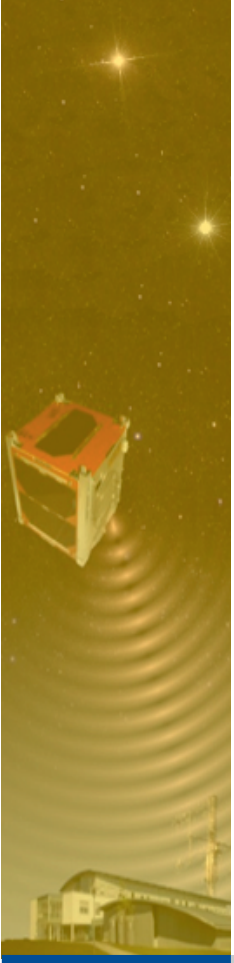


Institut für Informatik
Lehrstuhl für Robotik und Telematik
Prof. Dr. K. Schilling

Würzburger Forschungsberichte in Robotik und Telematik

Uni Wuerzburg Research Notes in Robotics and Telematics

Julius-Maximilians-
UNIVERSITÄT WÜRZBURG

Band 12

Hamidreza Houshiar

Documentation and mapping with 3D point cloud processing

Chair of Computer Science VII
Robotics and Telematics

Documentation and mapping with 3D point cloud processing

Dissertation

submitted to the Faculty of
Mathematics/Computer Science
of the University of Würzburg
in fulfilment of the requirements for the degree of
Doctor Rerum Naturalium (Dr. rer. nat.)

By
Hamidreza Houshiar

February 2017

Supervisor and first reviewer:	Prof. Dr. Andreas Nüchter
Second reviewer:	Prof. Dr. Claus Brenner

Abstract

3D point clouds are a de facto standard for 3D documentation and modelling. The advances in laser scanning technology broadens the usability and access to 3D measurement systems. 3D point clouds are used in many disciplines such as robotics, 3D modelling, archeology and surveying. Scanners are able to acquire up to a million of points per second to represent the environment with a dense point cloud. This represents the captured environment with a very high degree of detail. The combination of laser scanning technology with photography adds color information to the point clouds. Thus the environment is represented more realistically. Full 3D models of environments, without any occlusion, require multiple scans. Merging point clouds is a challenging process. This thesis presents methods for point cloud registration based on the panorama images generated from the scans. Image representation of point clouds introduces 2D image processing methods to 3D point clouds. Several projection methods for the generation of panorama maps of point clouds are presented in this thesis. Additionally, methods for point cloud reduction and compression based on the panorama maps are proposed. Due to the large amounts of data generated from the 3D measurement systems these methods are necessary to improve the point cloud processing, transmission and archiving. This thesis introduces point cloud processing methods as a novel framework for the digitisation of archeological excavations. The framework replaces the conventional documentation methods for excavation sites. It employs point clouds for the generation of the digital documentation of an excavation with the help of an archeologist on-site. The 3D point cloud is used not only for data representation but also for analysis and knowledge generation. Finally, this thesis presents an autonomous indoor mobile mapping system. The mapping system focuses on the sensor placement planning method. Capturing a complete environment requires several scans. The sensor placement planning method solves for the minimum required scans to digitise large environments. Combining this method with a navigation system on a mobile robot platform enables it to acquire data fully autonomously. This thesis introduces a novel hole detection method for point clouds to detect obscured parts of a captured environment. The sensor placement planning method selects the next scan position with the most coverage of the obscured environment. This reduces the required number of scans. The navigation system on the robot platform consist of path planning, path following and obstacle avoidance. This guarantees the safe navigation of the mobile robot platform between the scan positions. The sensor placement planning method is designed as a stand alone process that could be used with a mobile robot platform for autonomous mapping of an environment or as an assistant tool for the surveyor on scanning projects.

Acknowledgment

Firstly, I would like to express my deepest gratitude to my advisor Prof. Dr. Andreas Nüchter for the continuous support of my thesis and for his motivation and immense knowledge. His guidance assist me in all the research and projects during my study. It would have been quite impossible to carry on the research and make it to the final shape of the thesis without his encouragement. Additionally I would like to thank the other committee member of my thesis: Prof. Dr. Claus Brenner for his insightful comments and reviews.

I am also grateful to all my colleagues and friends in Bremen and Würzburg for making my study years such an enjoyable and interesting time of my life and for the hard days we were working together before deadlines. Especially Dorit Borrmann and Dr. Jan Elseberg, with their knowledgable feedbacks and their assist during the writing of my papers to make sure that not too many English mistakes get published.

Chapter 6 of this thesis was funded by the project "Fortlaufende semantische 3D-Kartierung von archäologischen Ausgrabungsstätten" by the German Federal Ministry of Economics and Technology by a Central Innovation Program (ZIM; No. KF24700098HM2) due to a resolution of the German Bundestag. I would like to thank all my colleagues from this project for all the insightful discussions and support during the project.

Chapter 7 of this thesis was made possible during my time at Zoller + Fröhlich GmbH. I would like to thank my colleagues there for all the long talks and collaboration on the project. Especially I would like to express my gratitude to Dr. Fröhlich for providing the initial idea for the development of the project and also giving me the opportunity to develop and finish the project.

I would like to acknowledge Zoller + Fröhlich GmbH for providing the Hannover dataset and ANGERMEIER INGENIEURE GmbH for letting my university group acquire the tunnel data during construction. Both datasets are used in Chapter 7. Additionally I would like to acknowledge denkmal3D GmbH & Co. KG and denkmalDaten Winkler KG for providing the archeology related datasets and images of Chapter 6. I am grateful to the "Bayerische Verwaltung der staatlichen Schlösser, Gärten und Seen" for access to the Würzburg Residence and make it possible to capture the point clouds from the White Hall and the Imperial hall.

Last but not least, I would like to thank my family: my parents, Abdolreza Houshiar and Shohreh Rejvani, my sister, Mahsa Houshiar, and my brother in law Houtan Shirzadi for supporting me spiritually through writing this thesis and my life in general. Without their love and unconditional support writing of this thesis would have been a lot more difficult.

Contents

Chapter 1

Introduction

Advances in technology provide easy access to sensors such as radar, ultrasonic, rangefinder and Light Detection And Ranging (LiDAR) sensors for multiple research fields. Range sensors are often used in robotics, surveying and archeology. Modern laser scanners are able to measure the surface of objects in the environment. They are capable of up to one million points per second measuring the reflectivity and the distance of a surface. With the help of photogrammetry they can also capture the color of the surface. The captured data is used for the generation of 2D and 3D maps of the environment. These maps are the essential part of digitisation process of archeological excavation sites. Additionally they are used in the robotics field for autonomous navigation and exploration. The generation of a complete 3D model from an environment without occlusion requires multiple scans from the environment. The overlapping areas from scans provide the necessary correspondences between captured point clouds for the registration process. 3D measurement systems capture point clouds in a local coordinate system defined by the scanner. The registration is the process of computing the relative transformation between overlapping scans to merge the data in one coordinate system. Several methods are proposed in the literature for scan matching. Two main categories are registration methods based on point correspondences and registration methods based on feature correspondences. In robotics the generated 3D models of an environment assist the robotic platform navigating that environment. However, in order to navigate in an unknown environment the robotic platform needs to generate the map and localise itself in it at the same time. In the robotics field this is a well know problem and is solved with the help of Simultaneous Localisation And Mapping (SLAM) solutions. The captured environment is digitalised in a 3D point cloud model. In archeology this data is commonly used only as the snapshot of the excavation site. This thesis introduces a framework for on-site semantic mapping of archeological sites. It utilises 3D point clouds from the environment in order to generate digital documentation for the excavation site. Additionally a novel method is introduced for minimising the number of required scan positions during the data acquisition. The aim of the method is to determine the next best scan position for autonomous indoor mapping of the environment with the help of the mobile robot platform. The combination of the next best view method and the archeological framework assists the archeological team on excavation sites to produce time efficient, objective and high quality documentation of the archeological sites.

1.1 Related Work

3D measurement systems produce large 3D point clouds. The data captured for the documentation and mapping consist of multiple scans. Many data formats for point cloud processing, transmission and archiving are proposed in the literature. The basis of point cloud processing is the ability to search for points in a short time. Bentley [19] developed the k-d tree for efficiently computing points closest to a query point. This tree structure is used to accelerate the search. In addition to the k-d tree other tree structures such as the octree are employed by [73, 91, 108, 199, 237] for storing and processing point clouds. This thesis presents the use of panorama images as a data structure for point cloud processing. The use of panorama images opens the paradigm of 2D image processing methods to 3D point cloud processing. The generation of panorama images from point clouds is based on the well-known cartography methods developed long ago for mapping the earth on a flat piece of paper. This thesis presents several projection methods for the generation of panorama images from 3D point clouds. These panorama images are the basis of most of the point cloud processing methods presented in this thesis. Point cloud registration is pivotal to the generation of occlusion free 3D maps of the environment. The Iterative Closest Point (ICP) [20] algorithm is a well-known algorithm for point cloud registration. This method relies on a good initial pose (position and orientation) estimate for the fast convergence of the registration process. This initial estimate is commonly available with the help of additional methods and tools. The Global Navigation Satellite System (GNSS) provides location and time information on the Earth provided an unobstructed line of sight to four or more satellites. Additionally local positioning systems are used to provide the required information in the absence of the GNSS signal. The local positioning systems rely on artificial landmarks installed in the environment. Roboticists proposed the use of a mobile robot platforms for initial pose estimates. Initially the robot odometry is used. In combination with Initial Measurement Unit (IMU) the estimates are more accurate. However, the small error generated during each step of the odometry calculation is accumulated over time and the overall pose estimation is uncertain for long robot paths. To overcome this problem roboticists utilise laser range finders to measure the environment. The robot localisation system provides the initial estimates for the registration process. Localisation and mapping solutions proposed in the literature estimate the relative robot position and orientation between scan positions. This information is used to correct the pose estimates. The use of the environmental information is the basis of the Simultaneous Localisation and Mapping (SLAM) methods. In the robotics field Smith and Cheeseman and Smith et al. [212, 213] introduce the use of the Extended Kalman Filter (EKF) to solve the SLAM problem. It is a probabilistic approach to calculate one posterior guess based on the a priori guess and the actions. Moutarlier and Chatila [163, 164] and Leonard and Durrant-Whyte [143] present the implementation of EKF SLAM. Several other SLAM solutions have been proposed in the literature such as [63, 67, 71, 87, 96, 97, 99, 100, 132, 159, 165, 176, 179, 225, 227]. In case of a lack of pose estimation during the acquisition process the only possible solution is to determine the pose estimates directly from the point cloud. Automatic algorithms favour the extraction of features in point clouds for the calculation of the pose estimates. Several methods are proposed in the literature for the automatic detection of features from the point cloud. These features provide the required correspondence between scans for the calculation of the pose estimates. Some approaches depend on the 3D structure for correspondence detection. Planer

patches are a common structure used with these methods [39, 183]. Other methods rely on key point detectors for the detection of the correspondences [24, 77, 119, 239, 243]. This thesis presents a survey of several 2D feature detector and descriptor methods for the calculation of the relative pose estimation between scans. It relies on the generation of panorama images from the point clouds. The effects of different projection methods on the registration process are evaluated.

An initial pose estimate is required for the final registration process of the point cloud using the ICP algorithm. This algorithm minimises the euclidian distance between the points in a scan pair. The number of points in the point clouds has a direct impact on the time efficiency of the registration method. Thus, data reduction becomes necessary for processing large point clouds. Efficient methods have been proposed in the literature for subsampling and reduction of point clouds. Several methods rely on special data structures for point reduction [141, 153, 223]. These methods uniformly subsample the point cloud to reduce the number of points. The uniform subsampling methods are problematic for the panorama generation methods. They introduce holes in the generated images. This produces unusable panorama images for further point cloud processing. Particularly this increases the likelihood of failure of the feature-based registration method. Thus, a point cloud reduction method based on panorama images is proposed in this thesis. It utilises the panorama map generated during the generation of panorama images for point cloud reduction. Additionally, it takes advantage of 2D image reduction methods to further reduce the points from the point clouds.

Besides data reduction for point cloud processing the large size of point clouds is problematic for data archiving and data transmission. Wireless networks are the common infrastructure used for transmitting the data from the scanning system to remote workstations. The limited bandwidth of these networks is the bottleneck of any data transmission process. Therefore, methods for point cloud compression are necessary for efficient data transmission and archiving. Standard binary formats are proposed in the literature for point cloud sharing. Additionally, compressed versions of the these binary formats are available [5, 114, 158]. Other compression methods are able to compress the point clouds based on tree structures such as k-d tree and octree [62, 73, 88, 108, 109, 185].

Captured 3D point clouds enable the generation of 3D models of environments. 3D measurement systems are utilised on archeological sites for digitisation of the excavation. These scans are often only used for the final representation of the data. Accuracy and time efficiency of modern laser scanners play a significant rule in the increasingly common practice of 3D laser scanning on excavation sites. However, the captured environment only represents a snapshot of the environment and it is not used during the scientific analysis. Many computer systems have been proposed in the literature to improve the data acquisition and analysis of archeological excavation [9, 51, 94, 120, 121, 137, 195, 247]. These methods lack the ability of the inclusion of the 3D data to the documentation and knowledge generation process. This thesis proposes a framework that utilises several components to achieve on-site semantic mapping of excavation sites based on captured 3D point clouds. The framework produces digital data and documentation from the excavation site. The aim of the framework is to replace the conventional methods of documentation with computer tools for generation, analysis and presentation of the data. At the core of the framework the Computer Aided SysTem for Labelling archeological Excavations in 3D (CASTLE3D) is presented to assist the archeologist in the generation of the

digital documentation from the captured point cloud on excavation site.

Scan position planning, point cloud acquisition and registration are challenging tasks. Mobile robot platforms improve the planning and the execution of the scanning process in an environment. The sensor placement planning reduces the number of required scans for an entire environment. The estimation of such view points is known as the Next Best View (NBV) problem. A manual determination of the view points is difficult. The automation of the next best scan position improves the quality and efficiency of the data acquisition process. 3D environment mapping and NBV methods are an active field of research in robotics. Many solutions have been proposed to solve this problem [11, 35, 93, 108, 173, 178, 210, 222, 246]. This thesis presents a novel approach for solving the NBV problem for 3D point clouds. It utilises the concept of holes in the captured point cloud. The holes are detected from the voxelised representation of the environment. A hole voxel is defined as an unseen voxel neighbouring an occupied voxel that has minimum one empty neighbouring voxel. The proposed NBV and hole detection algorithms are presented with more detail in this thesis. Additionally, to further improve the efficiency of the data acquisition process and to aim toward a fully autonomous indoor mobile mapping system, a navigation system is presented for the mobile robot platform. A combination of the NBV algorithm with the navigation system enables the robot to autonomously plan and execute the data acquisition process. The navigation system utilises the calculated NBV location to set a goal for the path planning method of the robot. The collision avoidance system of the robot guarantees a collision free travers between scan position for the robot.

1.2 List of Previous Publications

[1] **H. Houshiar** and A. Nüchter. 3D point cloud compression using conventional image compression for efficient data transmission. In *XXV-th International Conference on Information, Communication and Automation Technologies*, Sarajevo, Bosnia & Herzegovina, October 2015.

[2] **H. Houshiar**, D. Borrmann, J. Elseberg, A. Nüchter, F. Näth, and S. Winkler. CASTLE3D - A computer aided system for labelling archaeological excavations in 3D. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-5/W3:111 – 118, 2015.

[3] **H. Houshiar**, J. Elseberg, D. Borrmann, and A. Nüchter. A study of projections for key point based registration of panoramic terrestrial 3D laser scans. *Journal of Geo-spatial Information Science, Taylor & Francis*, 18(1):11 – 31, 2015.

[4] D. Borrmann, R. Hess, D. Eck, **H. Houshiar**, A. Nüchter, and K. Schilling. Evaluation of methods for robotic mapping of cultural heritage sites. In *2nd IFAC conference on Embedded Systems, Computer Intelligence and Telematics (CESCIT '15)*, Maribor, Slovenia, 2015.

[5] D. Borrmann, R. Hess, **H. Houshiar**, D. Eck, A. Nüchter, and K. Schiling. Robotic mapping of cultural heritage sites. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5/W4:9 – 16, 2015.

[6] D. Borrmann, **H. Houshiar**, J. Elseberg, A. Nüchter, F. Näth, and S. Winkler. Das Framework CASTLE3D zur fortlaufenden semantischen 3D-Kartierung von archäologischen Ausgrabungsstätten. *Allgemeine Vermessungs-Nachrichten (AVN), Wichmann Verlag*, 2015.

[7] D. Borrmann, **H. Houshiar**, J. Elseberg, A. Nüchter, F. Näth, and S. Winkler. Fortlaufende semantische 3D-Kartierung von archäologischen Ausgrabungsstätten. In *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage 2014*, pages 268 – 277. Jade Hochschule, Wichmann Verlag, February 2014.

[8] **H. Houshiar**, J. Elseberg, D. Borrmann, A. Nüchter, S. Winkler, and F. Näth. On-site semantic mapping of archaeological excavation areas. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-5/W1:163 – 168, 2013.

[9] **H. Houshiar**, J. Elseberg, D. Borrmann, and A. Nüchter. Panorama based point cloud reduction and registration. In *16th IEEE International Conference on Advanced Robotics (ICAR '13)*, Montevideo, Uruguay, 2013.

[10] D. Borrmann, **H. Houshiar**, J. Elseberg, and A. Nüchter. Vom Kombinieren von 3D-Modellen mit Farb- und Temperaturinformationen. In *Photogrammetrie Laserscanning Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage 2013*, pages 264 – 253. Jade Hochschule, Wichmann Verlag, February 2013.

[11] A. Nüchter, **H. Houshiar**, D. Borrmann, and J. Elseberg. Projektionen für die Scanregistrierung mit Hilfe von Bildmerkmalen. In *Photogrammetrie Laserscanning Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage 2012*, pages 12 – 21. Jade Hochschule, Wichmann Verlag, February 2012.

1.3 Outline

This work is structured as follows. Chapter 2 presents a short description of the point cloud acquisition methods and the differentiation of pulse based and phase based laser scanners. Additionally the common methods of data storage and data sharing for captured point clouds are overviewed. The point cloud representation methods are introduced especially the use of panorama images for storing 3D point clouds. To use panorama images as a representation medium this chapter introduces several projection method for mapping the captured 3D points onto a panorama image. Three different mapping methods between the 3D points and the 2D panorama pixels are presented. In Chapter 3 the scan matching problem is explained and an overview of the related literature is presented. The problem is divided into two main categories. Point-based registration and feature-based registration methods. The introduced feature-based registration method extends the idea of panorama images from 3D point clouds for feature-based

registration methods. This chapter presents a survey of several 2D feature detector and descriptor methods in combination with panorama projection methods. Finally, the globally consistent scan matching method is reviewed.

Chapter 4 and Chapter 5 further develop the use of panorama images from point clouds to 3D point cloud reduction and compression. Chapter 4 presents an analysis of point cloud reduction based on panorama images. Several reduction methods are proposed based on the panorama images. In addition to point cloud reduction the compression of point clouds are crucial. To address this issue Chapter 5 presents point cloud compression methods based on the panorama images. Utilising the panorama images enable the use of the image compression methods for 3D point clouds.

Chapter 6 focuses on the use of point cloud processing methods for documentation and mapping of the cultural heritage excavation site. A Computer Aided SysTem for Labelling archeological Excavations in 3D (CASTLE3D) is introduced to improve the quality of the archeological documentation. It consists of a set of tools for recording, geo referencing, labelling, semantic mapping and visualisation of the captured point clouds. Chapter 7 introduces a method towards fully autonomous indoor mobile mapping system. In addition to a short description of the mobile robotics, robot perception, localisation and mapping this chapter introduces a novel sensor placement planning. entire environment. Additionally, a path planning, path following and obstacle avoidance system is proposed for a mobile platform to autonomously map the entire environment.

A summary of the main results together with a concluding discussion is presented in Chapter 8.

Chapter 2

Point Cloud Representation

Active range sensors continue to be one of the most frequently used sensors in robotic, surveying, archeology and many other fields. These sensors provide direct measurements from the sensor to the objects in the environment and are commonly used for the localisation and mapping process on mobile robots. Active ranging sensors can be categorised into two groups. First category is time of flight systems that calculate the distance to objects based on the propagation speed of waves. In general the distance is given by:

$$d = c \times t, \tag{2.1}$$

where d is the traveled distance of the wave, c is the speed of the wave propagation and t is the time of flight. This equation calculates the distance traveled by the wave from the measuring unit to the object and back to the unit. Second category is triangulation based active ranging systems. These systems use geometric properties of structured light to calculate the distance. Structured light is projected to the environment and the scene is captured by an imaging device. The pattern is extracted from the image and based on the deformation and geometric values the triangulation system calculates the range.

Laser rangefinders use light as the electromagnetic signals to calculate the distance to objects. They consists of a transmitter which emits the laser and a receiver capable of detecting the reflected light. These sensors often are referred to as Light Detection And Ranging (LiDAR) sensors. LiDAR is an optical remote sensing technology that measures the properties of scattered light to find range and other information of a distant target. Figure 2.1 presents several LiDAR systems. 2D LiDAR systems use a mechanical system with a mirror to sweep the light to cover a plane of the environment. Additionally the 3D systems use an extra rotational mechanism to measure the environment in 3D.

2.1 Data Acquisition

3D laser scanners are standard instruments in the field of optical metrology. Laser scanners are a specific type of range sensors. They also measure the distance to a point on the surface of an object in the scanner environment by sending a focused laser beam in that direction.

(a) UTM-30LX a 2D scanner from HOKUYO Automatic Co., LTD.

(b) UXM-30LX-EW a 2D scanner from HOKUYO Automatic Co. LTD

(c) LMS 100 a 2D laser scanner from Sick AG

(d) IMAGER 5010C a 3D laser scanner from Zoller + Fröhlich GmbH

(e) VZ-400 a 3D laser scanner from RIEGL Laser Measurement Systems GmbH

(f) Faro Focus 3D a 3D X130 laser scanner from FARO Europe GmbH & Co. KG

Figure 2.1: Light detection and ranging systems. Pulse based laser scanners transmit short pulses of light and directly measures the amount of time for the pulses to travel from the scanner to the object and back to the scanner. Phase based laser scanners emit modulated laser light at a known frequency. They determine the distance to objects based on the changes of the phase between the emitted light and the received signal. Top row: Presents 2D laser range finders. Bottom row: Presents 3D laser scanners.

Similarly the distance is calculated by measuring the time difference of the emitted and reflected signals. Two different types of laser scanning technologies have established themselves in this field. Pulsed laser range finders send a short pulse of light and directly measure the amount of time it takes for the laser pulse to travel from scanner to the object and back to the scanner. The distance is then calculated via the speed of the light. Since the speed of light is almost 300000 km/h, tools with picoseconds (10^{-12} s) resolution need to be used in order to calculate the distance with an accuracy of 10 mm. Other than pulsed systems there are laser scanners that emit modulated laser light at a known frequency. The change of the phase between the emitted light and the received signal determines the light travel time. Since the phase shift in only intervals between 0 and 2π is non ambiguous, the maximum range of these devices is inherently

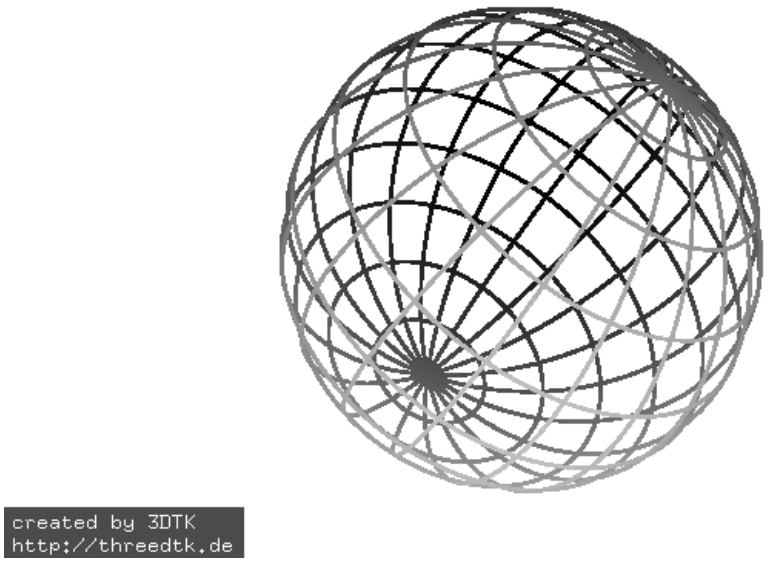


Figure 2.2: Scanned environment as an artificial sphere.

limited. Current phase-based laser scanners achieve a range of about 80-120 m whereas pulse-based laser scanners can measure distances up to several kilometres. However, since phase-based laser scanners emit light continuously they can operate at a higher measurement speed than pulse based systems. Besides the distance to objects, laser scanners determine the intensity of the reflected light. This yields information about the reflectivity of the scanned surface. Further combination of the LiDAR systems with color cameras incorporate the color information of the measured surface into the point cloud data.

2.2 Data Representation

Modern 3D laser measurement systems simplify the collection of large 3D point clouds. These 3D laser scanning systems are used to generate accurate and high resolution 3D point clouds. They measure the distance of the surface points of objects in the 3D scene. Advances in technology have improved the quality and quantity of the captured point clouds. 3D point clouds are captured for variety of applications in mobile robotics, computer vision, 3D environment modelling, surveying and archeology to name a few. Captured scans contain the range information from the measured surfaces in the environment. Additionally, modern scanners are capable of capturing the reflectance and color values of the measured surfaces. The captured environment can be imagined as a sphere with the measurement system positioned at the center of the sphere. Figure 2.2 presents the scanned environment as an artificial sphere. The raw data from a 3D laser measurement system is commonly stored as x, y and z. This represents the cartesian coordinates of the measured points in a coordinate frame originating at the scanner position. An alternative method is to represent the points in the spherical coordinate system. The θ, φ, r represent the horizontal and vertical angles of measurement with the measured distance r. Li-

DAR data is typically stored in vendor specific binary files and for exchange the simple ASCII representation is preferred. The ASCII format presents the measurements as a list of attributes on each line of the file. This method is flexible and easy to understand. However, it generates very large textual files with inefficient parsing capabilities. The American Society of Photogrammetry and Remote Sensing (ASPRS) created a simple binary exchange format namely LAS [5]. Additionally they introduced a compressed version of the LAS format. This lossless compressed format is called LAZ.

The large point clouds acquired by modern terrestrial 3D laser measurement systems in impressive rates pose large memory demands. To process them in the main memory of standard systems many methods have been proposed in the literature. These methods rely on special data structures, often used are tree structures such as the k-d tree and octree. These data structures are used to store the captured data from the 3D laser scanners. One of the basis of the point cloud processing is the ability to search for points in a short time. Since the development of ICP, k-d trees have been used to accelerate the search. Bentley [19] developed the multidimensional binary search tree or k-d tree. The k-d tree structure recursively splits and subdivides the point cloud and presents the data in a tree structure. For computing the closest points, k-d trees [85] are the standard search structures. Nüchter et al. [175] present a novel cached k-d tree search procedure by exploiting the iterative behaviour of the ICP algorithm.

The octree is a data structure for indexing 3D data. Similar to the k-d tree, octree represents the data in a tree structure. The tree divides the 3D environment into rectangular cuboids. These volumetric cuboids are represented as octree nodes. The root node of the tree represent the whole data. An octree node has up to eight children. Each child node corresponds to one octant of the cube. A node without a child implies that for the corresponding volume no further subdivision is necessary and the volume can be uniformly represented. Storing a point cloud in an octree has a stopping rule for the accepted volume. During the sorting the point cloud the stopping criteria is defined as both maximal depth and minimal number of points per node. This criteria is applied to volumes without points, such that child nodes are created only for volumes that contain points. Point cloud points are stored in the octree nodes. All the nodes without points are considered as empty space. Since laser scanners sample the surface of objects, the 3D point cloud is not fully volumetric. The point cloud only represents the occupied space in the environment and most of the environment is free. This produces octree nodes with only few occupied children. The octree data structure is therefore ideal for point cloud representation. Elseberg et al. [73] present an octree data structure with low memory consumption. Similarly Hornung et al. [108] present an open source framework for generating volumetric 3D environment models based on the octree structure. They use a probabilistic occupancy estimation method to explicitly present the occupied, free and unknown areas. The focus of this approach is to provide a volumetric representation of space for robotic applications including flying robots and robots that are equipped with manipulators. Rusu and Cousins [198, 199] also present another implementation of the octree data structure that is used for storing and processing 3D point clouds. Similarly [91] and [237] employ octree data structure for storing and processing 3D point clouds.

Some methods utilise the grid map method for representing point clouds. Depth images are generated from point clouds to represent the captured data. This is the process of mapping 3D data onto a 2D image. The 3D coordinates of points are mapped onto 2D image coordinates.

The measured range is used to represent the image pixel value. The additional information that laser scanners acquire, like reflectance or color, can be stored in additional images. In addition to data storage and representation of point clouds, the generated images are used for other applications for example feature detection methods. The detected features from point clouds provide correspondences between captured scans. This information is used for matching the scans. Additionally, these images are used for data compression, segmentation, normal calculation, point cloud reduction, data exchange and many other applications. The next section presents the use of 2D panorama images for storing and processing the point clouds in detail. The use of panorama images introduces the conventional 2D image processing methods to 3D point clouds.

2.3 Panorama Projection Methods

Image projection is the process of mapping a curved surface on to a flat image. Representation of the spherical view of the measured environment on a flat image requires the mapping of the 3D scene on a 2D medium. Mapping spherical coordinate onto 2D coordinates is a renowned practice in the cartographic field. Numerous methods have been developed over the years to map the earth. Similar methods are used in panoramic photography. The environment is considered as a surface of a sphere (for all viewing angles) that requires image projection to be mapped on an image. Captured point cloud of an environment can be considered as a sphere with the 3D measurement system positioned in the center of the sphere. Modern 3D laser measurement systems are capable of rapid full 360° field of view measurement. Captured spherical coordinates θ, φ, r of the measured points are used to project on to the panorama image coordinates x and y. The longitude or θ of the spherical coordinate system is the angular distance of east or west of the meridian, the constant longitude, at Greenwich, England. Therefore, meridian has the longitude $0° = 360°$. The latitude or φ of the spherical coordinate system is the elevation from the equator plane. The equator is a line around the sphere equally distant from both poles. Therefore, the equator has latitude 0° and the north and south poles have the latitude $\pm 90°$, respectively.

Each of the methods proposed in the literature for the image projection process has its own strengths and limitations. Distortion is inevitable in the process of mapping a 3D scene on a 2D image. In other words the original 3D surface cannot be converted perfectly to a flat 2D map. Distortion is the false presentation of angles, shapes, distances and areas. Disregarding the mapping method the introduced distortion for small angles is relatively minor due to the fact that the viewing arc is relatively flat. The viewing arc becomes a curve by increasing the viewing angle, thus introducing more distortion. Projection methods focus on different distortions and aim to minimize one or more types of distortion at the expense of others, i.e., projections have various attributes and limitations. There are many projection methods for a variety of purposes, however only a few of them are traditionally used for panoramic projection. In the following of this section a few of these projection methods are studied. Typically point clouds are represented using equirectangular projection method. As mentioned before every projection method has its own limitations therefore, this section extends the projection methods used for 3D point clouds. These methods have been modified to be suitable for 3D point clouds.

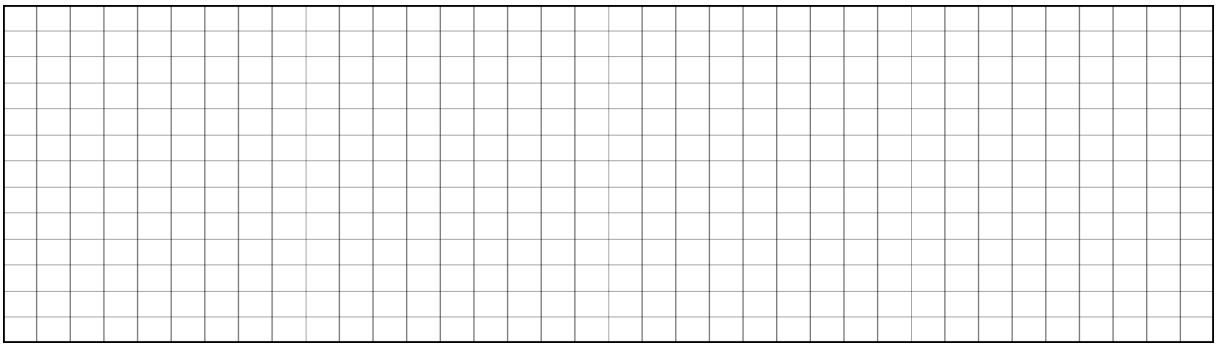


Figure 2.3: Latitude and longitude of mapped artificial sphere with equirectangular projection.

2.3.1 Equirectangular

This projection is among the oldest and simplest projections. This method is used for spherical panorama applications for mapping a portion of the surface of a sphere onto a flat image. This is the only common projection used for mapping point clouds and generating depth images. In this section several other methods are proposed and optimised for panorama image generation from point clouds.

This projection maps the longitude and latitude to the horizontal and vertical coordinates of a grid with no transformation and scaling applied. This projection is being used to create maps since it was invented by Marinus of Tyre around 100 A.D. Equal vertical and horizontal distances on the mapped grid are equivalent to equal angles on the original sphere. The straight vertical lines remain vertical and straight on the equirectangular projected images. However, this projection does not preserve the horizontal straight lines. All horizontal straight lines except the horizon become curvatures. Equirectangular projection covers full 360° horizontal and 180° vertical field of view. The θ and φ are mapped linearly to image coordinates. The spherical poles are stretched at the top and bottom of the image over the entire width of the panorama. This projection is a cylindrical equidistance projection, i.e., it preserves the distance. It is also known as rectangular projection [214]. transformation equations for this projection are:

$$\begin{aligned} x &= \theta, \\ y &= \varphi, \end{aligned} \tag{2.2}$$

where the θ and φ are longitude and latitude of the measured sphere of the environment. Figure 2.3 presents the equirectangular projection of an artificial sphere mapped onto a panorama image. Figures 2.4, 2.5 and 2.6 present a sample point cloud mapped with equirectangular projection onto a panorama image showing the color, reflectance and range information from the point cloud.

Figure 2.4: Color map of a 3D point cloud. Generated by mapping the 3D point cloud with equirectangular projection onto a panorama image.

Figure 2.5: Reflectance map of a 3D point cloud. Generated by mapping the 3D point cloud with equirectangular projection onto a panorama image.

Figure 2.6: Range map of a 3D point cloud. Generated by mapping the 3D point cloud with equirectangular projection onto a panorama image.

2.3.2 Cylindrical

Cylindrical projection is similar to equirectangular projection. It maps a portion of the surface of a sphere onto a flat image. It can be envisioned by wrapping a flat piece of paper around a

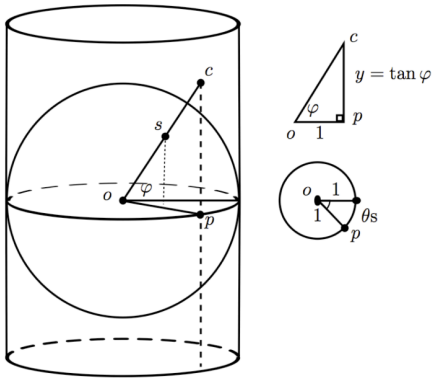


Figure 2.7: Cylindrical projection envisioned as a flat paper mapped around a sphere to form a cylinder. Emitting a light from the center of the sphere maps the surface of the sphere onto a flat paper.

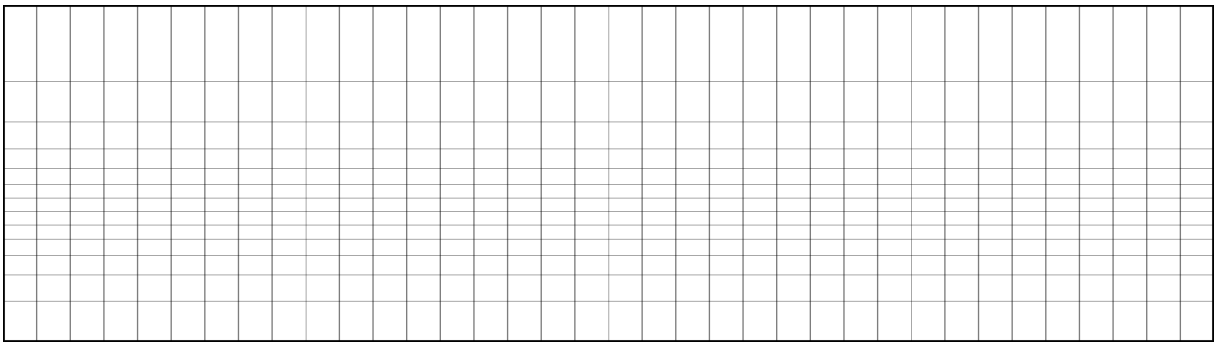


Figure 2.8: Latitude and longitude of mapped artificial sphere with cylindrical projection.

circumference of a sphere. The wrapped paper forms a cylinder and it is tangent to the sphere at its equator. Emitting light from the center of the sphere O will project the surface of the sphere onto the flat paper as shown in Figure 2.7. A cylindrical projection of a unit sphere centered at O consists of extending the line OS for each point S on the sphere until it intersects the cylinder at the corresponding point C [214]. Similar to equirectangular projection non vertical straight lines are mapped to curvatures except the horizon which will be mapped to the middle of the image as a straight horizontal line. This projection stretches the objects vertically, specially the closer they are to north and south poles of the sphere. This limits the projection to 120° or less vertical field of view. Therefore, it is not suitable for panorama images with large vertical angle of view. 360° horizontal field of view is the same as in the case of the equirectangular projection. This projection is more suitable for mapping point clouds of the outdoor environments due to the limitation of the vertical field of view. In outdoor environments the introduced distortion on top of the panorama image is mostly covered by sky. Therefore the distortions have less effect on the rest of the point cloud. The transformation equations for this projection are:

$$\begin{aligned} x &= \theta, \\ y &= \tan\varphi, \end{aligned} \tag{2.3}$$

Figure 2.9: Color map of a 3D point cloud. Generated by mapping the 3D point cloud with cylindrical projection onto a panorama image.

Figure 2.10: Reflectance map of a 3D point cloud. Generated by mapping the 3D point cloud with cylindrical projection onto a panorama image.

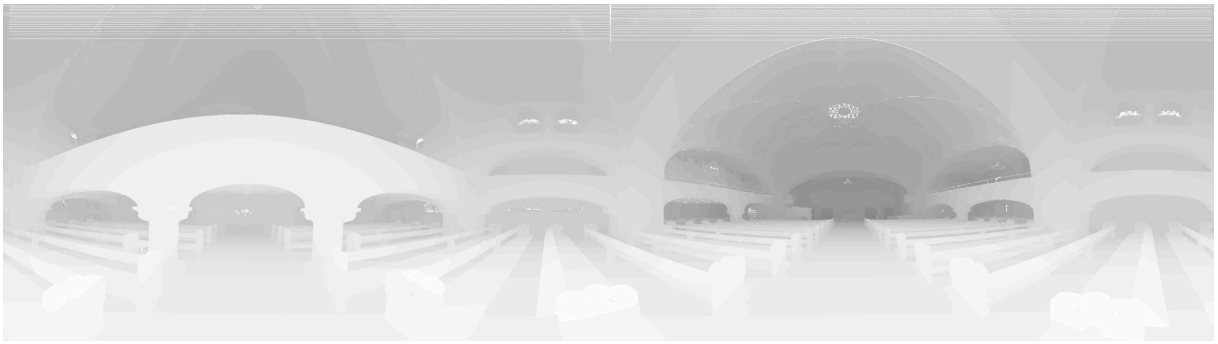


Figure 2.11: Range map of a 3D point cloud. Generated by mapping the 3D point cloud with cylindrical projection onto a panorama image.

where θ the longitude is mapped to the horizontal coordinates and the $\tan(\varphi)$ is mapped to vertical coordinates of the panorama image where φ denotes the latitude. Figure 2.8 presents the cylindrical projection of an artificial sphere mapped onto a panorama image. Figures 2.9, 2.10 and 2.11 present a sample point cloud mapped with cylindrical projection onto a panorama image.

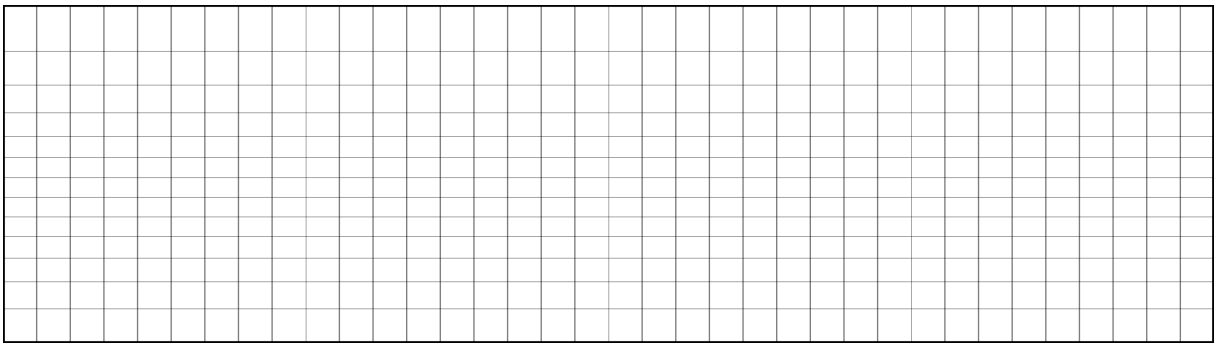


Figure 2.12: Latitude and longitude of mapped artificial sphere with the Mercator projection.

2.3.3 Mercator

The Mercator projection is one of the most renowned projections in the cartography field because of the use of this projection for flat maps of the earth. It was first presented by the geographer and cartographer Gerardus Mercator in 1569. This projection is a conformal projection, i.e., it preserves the angles. This ability of the Mercator projection made it the standard map projection for nautical purposes. The Mercator projection has equal linear scale in all directions for any point. This preserve the angle and shape of small objects. However, the shape of objects are distorted as the latitude increases from the equator. This projection is related to the equirectangular and cylindrical projections. However, It shows less pronounced distortion compared to the aforementioned projections. It has less vertical stretching, hence it supports larger vertical field of view. There is a variation of the Mercator projection called transverse Mercator which is used for very tall vertical panoramas [214]. However, for the purpose of panorama image generation from full 360° point clouds, the simple Mercator projection is preferred. The Mercator projection has a recommended vertical field of view of less than 150° and horizontal field view of 360°. The transformation equations for this projection are:

$$\begin{aligned} x &= \theta, \\ y &= \ln\left(\tan\varphi + \frac{1}{\cos\varphi}\right), \end{aligned} \tag{2.4}$$

where θ is the longitude mapped to the horizontal coordinate and a function of φ the latitude is mapped to the vertical coordinate of the panorama image. Figure 2.12 presents the Mercator projection of an artificial sphere mapped onto a panorama image. Figures 2.13, 2.14 and 2.15 present a sample point cloud mapped with the Mercator projection onto a panorama image.

2.3.4 Rectilinear

Rectilinear is a type of projection for mapping only a portion of the surface of the sphere onto a flat image. It is also known as “genomonic” or “tangent-plane” projection. This projection can be imagined by placing a flat piece of paper tangent to a sphere at a single point, and illuminate the surface of the sphere from the center. The image projected onto the flat paper

Figure 2.13: Color map of a 3D point cloud. Generated by mapping the 3D point cloud with the Mercator projection onto a panorama image.

Figure 2.14: Reflectance map of a 3D point cloud. Generated by mapping the 3D point cloud with the Mercator projection onto a panorama image.

Figure 2.15: Range map of a 3D point cloud. Generated by mapping the 3D point cloud with the Mercator projection onto a panorama image.

is the rectilinear projection of the sphere onto the flat image as shown in Figure 2.16. The rectilinear projection is obtained by projecting the point P_1 on the surface of the sphere onto a point P on a plane that is tangent to the point S on the sphere. Emitting light from O will map the surface point P_1 to an image point P. In Figure 2.16 tangent point S is the south pole however, this can be any point on the sphere. This projection is a fundamental image projection

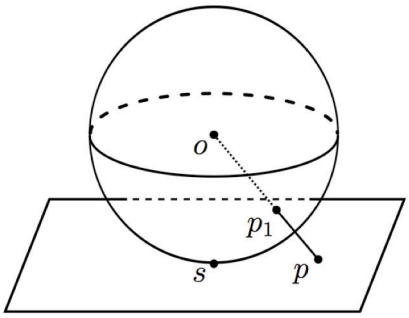


Figure 2.16: Rectilinear projection envisioned as a flat paper tangent to a point on a sphere. Emitting a light from the center of the sphere maps the surface of the sphere onto the flat paper.

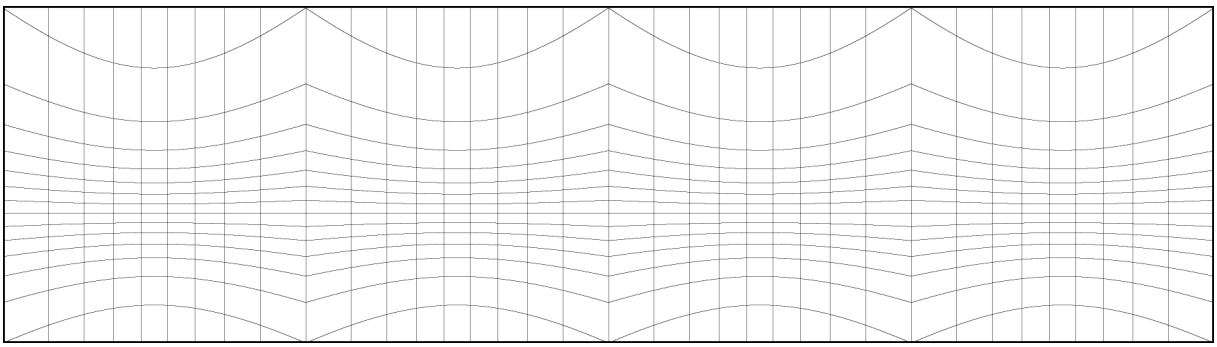


Figure 2.17: Latitude and longitude of mapped artificial sphere with rectilinear projection.

since most ordinary cameras produce images very close to rectilinear projection. The primary advantage of rectilinear projection is that it maps straight lines in a 3D space onto straight lines on the 2D image. Its main disadvantage however, is the smaller field of view as compared to the aforementioned projections. This makes this projection only applicable to images that do not cover extensively large range of longitude or latitude. Another disadvantage of rectilinear projection is the stretched corners of the mapped image. The distortion grows with larger fields of view [214].The transformation equations for this projection are:

$$\begin{aligned} x &= \frac{\cos\varphi \sin(\theta - \theta_0)}{\sin\varphi_1 \sin\varphi + \cos\varphi_1 \cos\varphi \cos(\theta - \theta_0)}, \\ y &= \frac{\cos\varphi_1 \sin\varphi - \sin\varphi_1 \cos\varphi \cos(\theta - \theta_0)}{\sin\varphi_1 \sin\varphi + \cos\varphi_1 \cos\varphi \cos(\theta - \theta_0)}, \end{aligned} \tag{2.5}$$

where θ is the longitude and φ is the latitude and θ_0, φ_1 are the longitude and latitude of the center of the projection. The projection center is the point on the sphere where the plane is tangent to the sphere. It is recommended to use this projection for horizontal and vertical fields of view of less than 120°. Therefore, the rectilinear projection for mapping a point cloud with a 360° horizontal field of view requires an additional step. The point cloud is divided into minimum three subsets with 120° or less horizontal field of view.

Figure 2.18: Color map of a 3D point cloud. Generated by mapping the 3D point cloud with rectilinear projection onto a panorama image.

Figure 2.19: Reflectance map of a 3D point cloud. Generated by mapping the 3D point cloud with rectilinear projection onto a panorama image.

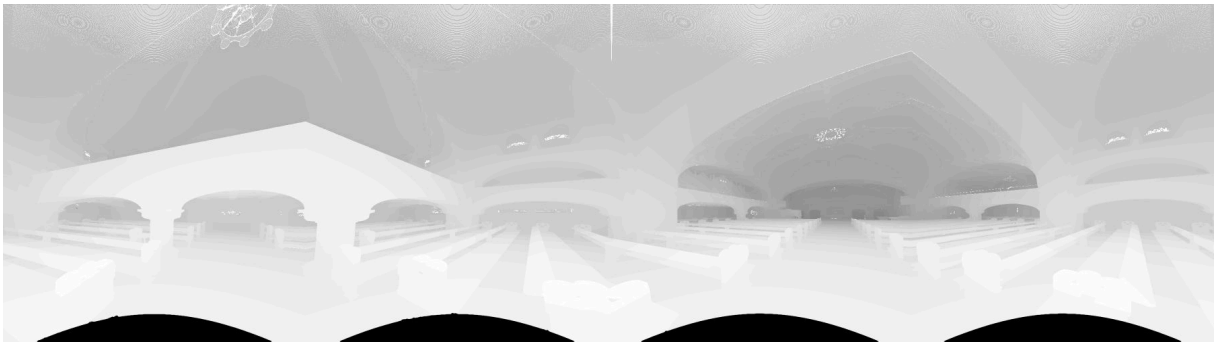


Figure 2.20: Range map of a 3D point cloud. Generated by mapping the 3D point cloud with rectilinear projection onto a panorama image.

Subsets are mapped individually onto an image using the rectilinear projection. The combination of the projected images produces a full panorama image of the point cloud. Figure 2.17 presents the rectilinear projection of four subsets of an artificial sphere mapped onto a panorama image. Figures 2.18, 2.19 and 2.20 present a sample point cloud mapped with rectilinear projection onto a panorama image. To generate these images the point cloud is divided into four subsets. Accordingly the projection of each subset is calculated and mapped onto one fourth of the image.

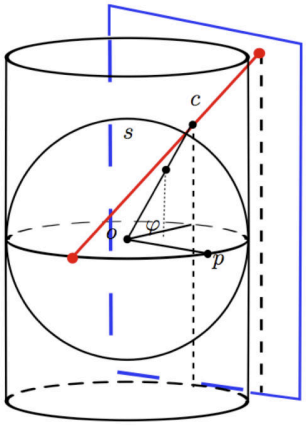


Figure 2.21: The Pannini projection envisioned as rectilinear projection of the sphere onto the tangent plane to the cylindrical projection of the sphere. This projection maps the cylindrical projected point c of the sphere point s onto the tangent plane to the cylinder.

2.3.5 Pannini

The Pannini projection is a mathematical rule for constructing perspective images with a very wide field of view. This projection is also known as "Recti-Perspective", "Panini" or "Vedutismo". It can be imagined as the rectilinear projection of a 3D cylindrical image. This image is itself a projection of the sphere onto a tangent cylinder. The center of the rectilinear projection can be any point and is on the view axis at the distance d from the cylinder axis (cf. Figure 2.21). The parameter d can be any non-negative number and determines the projection. With $d = 0$ the projection is rectilinear and with $d \to \infty$ it produces the cylindrical orthographic projection. Therefore, different projections are achieved with different d parameters. The projection with $d = 1$ has been defined as the Pannini projection. This projection is a cylindrical projection therefore, straight vertical lines on the sphere are mapped to straight vertical lines on the image. However, unlike other cylindrical projections it maps radial lines through the center of the projection onto straight lines. This implies a correct perspective for a wide angle of view with a central vanishing point. Similar to other cylindrical projections the Pannini projection renders the horizontal lines as curvatures. This projection produces images with a single central vanishing point. The images appear to have correct perspective even when using a wide field of view. The Pannini projection often resembles a normal rectilinear perspective. The horizontal field of view can be very wide, without the distortions near the edges which are noticeable when the rectilinear projection is pushed too far. This it the advantage of the Pannini projection over the rectilinear projection. The Pannini projection is suitable for views with single central vanishing point similar to a city street. However, it is also applicable for interior and wide viewing angles. The recommended field of view for the Pannini projection is less then 150° in both vertical and horizontal field of view. Therefore, the Pannini projection demands the division of a 360° point cloud into several subsets. They are mapped individually onto an image

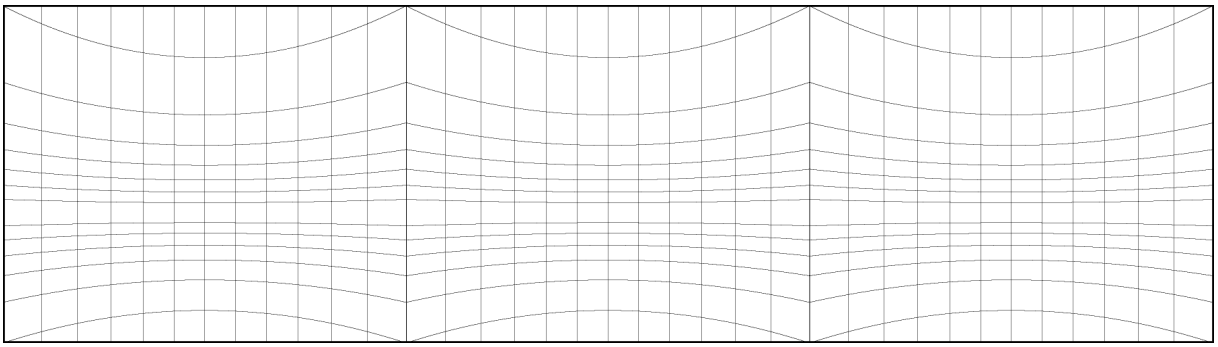


Figure 2.22: Latitude and longitude of mapped artificial sphere with the Pannini projection.

Figure 2.23: Color map of a 3D point cloud. Generated by mapping the 3D point cloud with the Pannini projection onto a panorama image.

using the Pnnini projection. The combination of the projected images produces a full panorama image of the point cloud. The transformation equations for this projection are:

$$
\begin{aligned}
x &= \frac{(d+1)\sin(\theta-\theta_0)}{d+\sin\varphi_1\tan\varphi+\cos\varphi_1\cos(\theta-\theta_0)},\\
y &= \frac{(d+1)\tan\varphi\left(\cos\varphi_1-\sin\varphi_1\left(\frac{1}{\tan\varphi}\right)\cos(\theta-\theta_0)\right)}{d+\sin\varphi_1\tan\varphi+\cos\varphi_1\cos(\theta-\theta_0)},
\end{aligned}
\tag{2.6}
$$

where θ is the longitude and φ is the latitude and θ_0, φ_1 are the longitude and latitude of the center of the projection. These are the enhanced equations of the Pannini projection that were modified to have projection center other than the south pole. This was an assumption for the original Pannini projection equations presented by Sharpless et al., 2010 [209]. Figure 2.22 presents the Pannini projection of three subsets of an artificial sphere mapped onto a panorama image. The projection center for each subset of the point cloud is the point on the sphere where the plane and the cylinder are tangent to the sphere. Figures 2.23, 2.24 and 2.25 present a sample point cloud mapped with the Pannini projection onto a panorama image. To generate these images the point cloud is divided into three subsets. Accordingly the projection of each subset is calculated and mapped onto one third of the image.

Figure 2.24: Reflectance map of a 3D point cloud. Generated by mapping the 3D point cloud with the Pannini projection onto a panorama image.

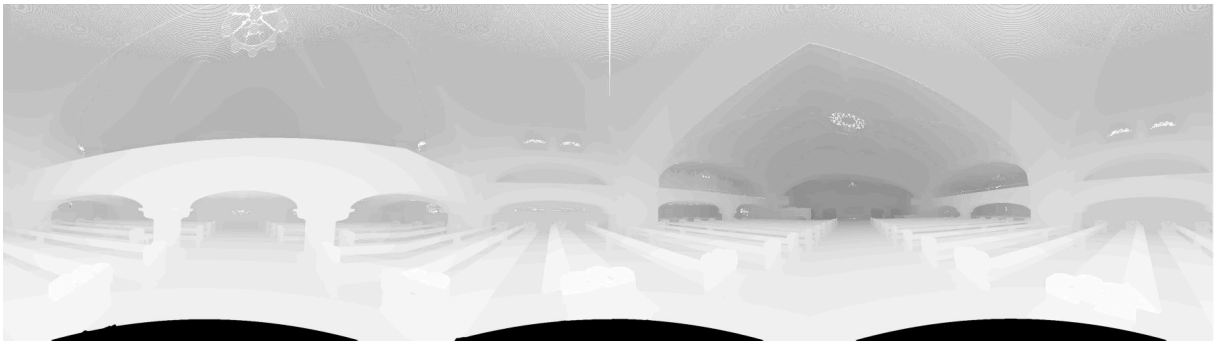


Figure 2.25: Range map of a 3D point cloud. Generated by mapping the 3D point cloud with the Pannini projection onto a panorama image.

2.3.6 Stereographic

This projection is an alternative method for mapping a portion of the surface of a sphere onto a flat image. Similar to the rectilinear projection it can be imagined by placing a flat paper tangent to a sphere and illuminate it from the antipode, i.e. the opposite surface point on the sphere diameter, rather than the center of the sphere. All points on the sphere cast a shadow on the tangent paper. The tangent point is the center of the projection. Selecting one of the sphere poles as the center of the sphere maps the latitudes lines as circles around the central point. Mapped points gain more distortion further from the pole up to the equator which is increased twice the size compared to the sphere. The other hemisphere is stretched more through the other pole where it is mapped to infinity. However, the center of the projection and the illuminating points are not bound to the sphere poles. The central point can be any point on the sphere and the illuminating point can have any distance R from the center of the projection [214]. Images over 330° are not very functional. It is advisable to take this further and utilise a 120° field of view and divide the image in to three subsets. Furthermore, the new parameter R which defines the distance of the illuminating point from the projection center is introduced. The

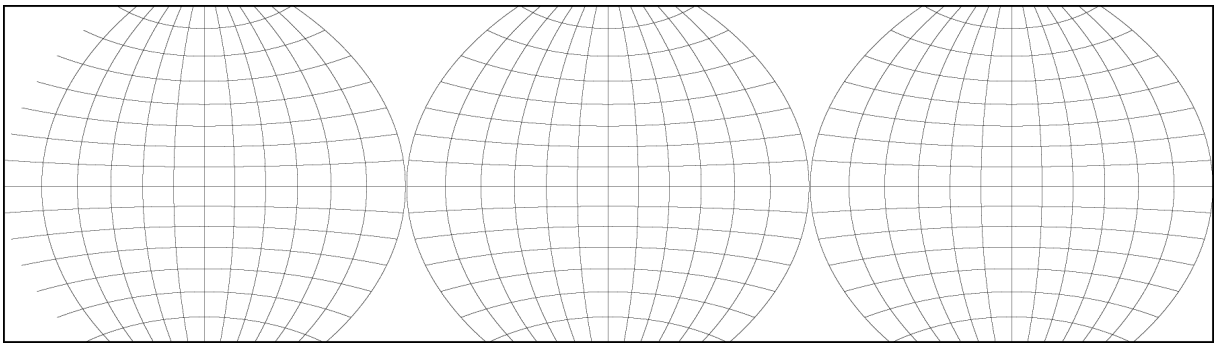


Figure 2.26: Latitude and longitude of mapped artificial sphere with stereographic projection.

Figure 2.27: Color map of a 3D point cloud. Generated by mapping the 3D point cloud with stereographic projection onto a panorama image.

transformation equations for this projection are:

$$
\begin{aligned}
x &= \frac{2R\cos\varphi\sin(\theta-\theta_0)}{1+\sin\varphi_1\sin\varphi+\cos\varphi_1\cos\varphi\cos(\theta-\theta_0)},\\
y &= \frac{2R(\cos\varphi_1\sin\varphi-\sin\varphi_1\cos\varphi\cos(\theta-\theta_0))}{1+\sin\varphi_1\sin\varphi+\cos\varphi_1\cos\varphi\cos(\theta-\theta_0)},
\end{aligned}
\tag{2.7}
$$

where θ_0, φ_1 are the longitude and latitude of the center of the projection in each subset of the data. The R parameter is any non negative value. $R = 1$ generates exactly the same equation as the Pannini projection and high values for R introduce more distortion. All the experiments with this projection utilise the value 2 for the parameter R. Figure 2.26 presents a stereographic projection of three subsets of an artificial sphere mapped onto a panorama image. The projection center for each subset of the point cloud is a tangent point on the equator. Figures 2.27, 2.28 and 2.29 present a sample point cloud mapped with stereographic projection onto a panorama image. To generate these images the point cloud is divided into three subsets. Accordingly the projection of each subset is calculated and mapped onto one third of the image.

Figure 2.28: Reflectance map of a 3D point cloud. Generated by mapping the 3D point cloud with stereographic projection onto a panorama image.

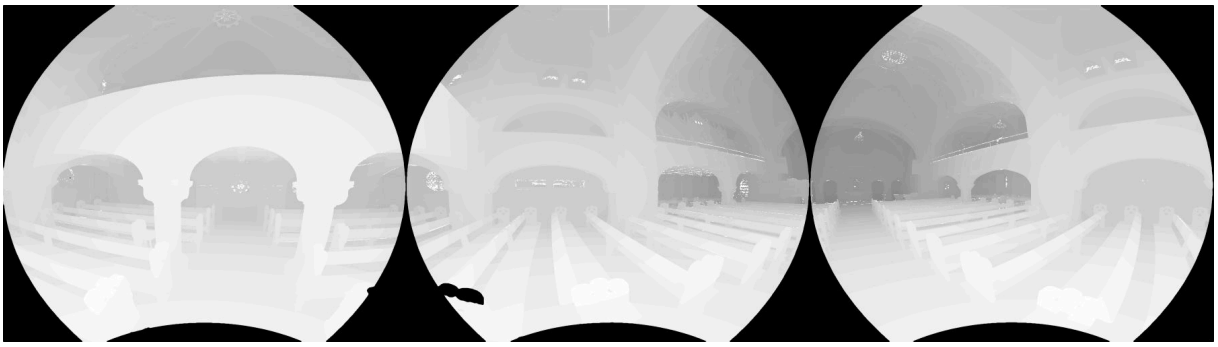


Figure 2.29: Range map of a 3D point cloud. Generated by mapping the 3D point cloud with stereographic projection onto a panorama image.

2.3.7 Z-axis

This projection is not a regular panoramic projection. The actual heights of objects in the real world are known through the sensing process and it is reasonable to use these values for the projection purpose. An approximation of the minimum and maximum values of points in z-axis direction, i.e., in height direction, from the scanned data and the horizontal angle of view of the scanner are used for the calculation of the panorama image. The transformation equations for this projection are:

$$\begin{aligned} x &= \theta, \\ y &= cz. \end{aligned} \tag{2.8}$$

This projection maps the longitude θ to the horizontal coordinates and the height of the points z to the vertical coordinates. Parameter c is a scene dependent constant to map the height of the object onto the panorama image. Consequently by utilising the aforementioned equations the 3D data will be mapped onto a 2D image. Figure 2.30 presents the z-axis projection of an artificial sphere mapped onto a panorama image.

Figures 2.31, 2.32 and 2.33 present a sample point cloud mapped with z-axis projection onto a panorama image.

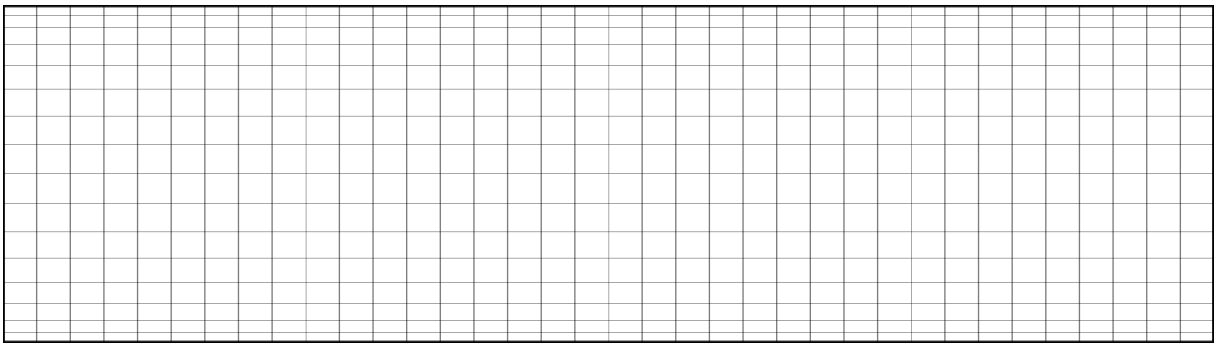


Figure 2.30: Height and longitude of mapped artificial sphere with z-axis projection.

Figure 2.31: Color map of a 3D point cloud generated with z-axis projection.

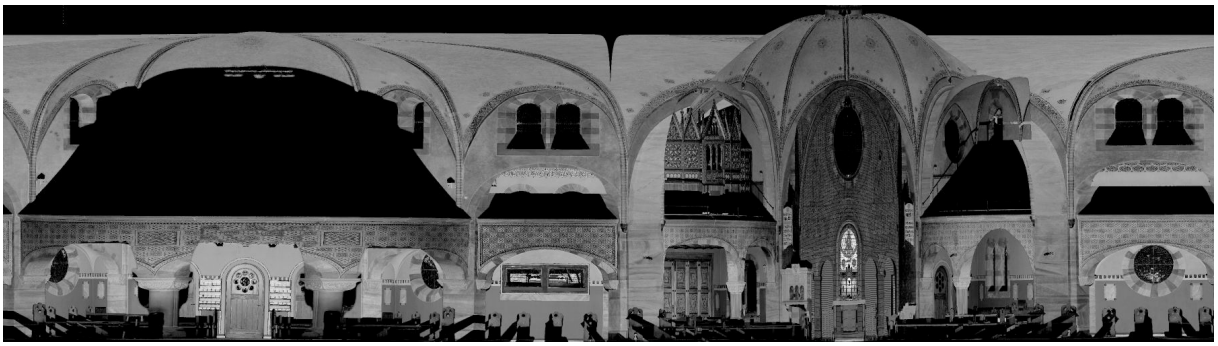


Figure 2.32: Reflectance map of a 3D point cloud generated with z-axis projection.

Figure 2.33: Range map of a 3D point cloud generated with z-axis projection.

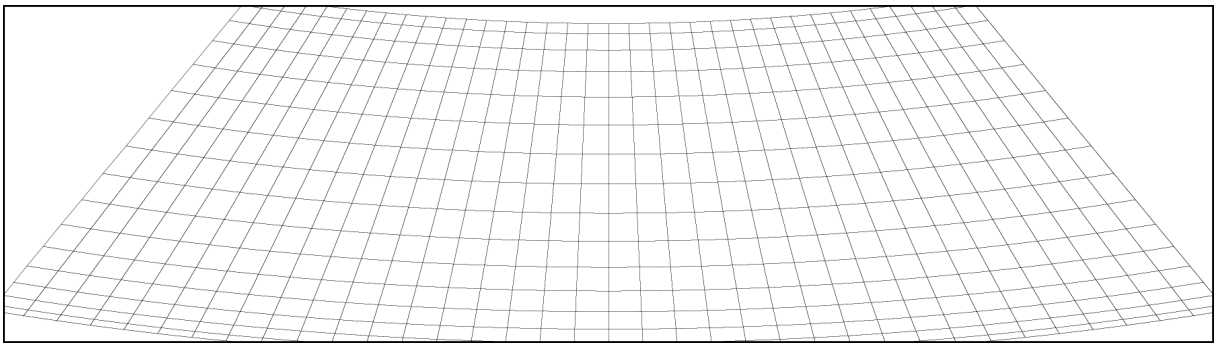


Figure 2.34: Latitude and longitude of mapped artificial sphere with the Albers equal-area conic projection. Standard lines at $\varphi_1 = -40°$ and $\varphi_2 = 60°$ are used for generation of this map.

2.3.8 Albers Equal-Area Conic

The Albers equal-area conic projection is similar to the cylindrical projection. A cone substitutes the tangent cylinder of the cylindrical projection. It can be imagined as a map wrapped onto a cone set over the sphere. The cone intersects the sphere at one or two latitude lines called standard lines. The term conic is referred to any projection in which longitudes are mapped to equally distanced lines, converging at a point which may or may not be a pole. Circles of latitude are mapped to arcs of circles, concentric in the point of convergence of the longitude lines. The resulting map has low distortion in scale, shape and area near the standard lines. This projection preserves the area, hence the equal-area in the name of this projection. The Albers equal-area conic projection has little distortion at the central latitude and none at the standard lines. This projection is suited for areas on the sphere that extend from east to west rather than from north to south [214]. In this projection the 90° angles between latitudes and longitudes are preserved. However, the scale along the lines of latitude does not match the scale along the lines of longitude, i.e., the projection is not conformal. This projection is recommended for a horizontal field of view of up to 360° and a vertical field of view of less than 60°. The transformation equations for this projection are:

$$\begin{aligned} x &= \rho \sin \lambda, \\ y &= \rho_0 - \rho \cos \lambda, \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} n &= \frac{1}{2}(\sin \varphi_1 + \sin \varphi_2), \\ \lambda &= n(\theta - \theta_0), \\ C &= \cos^2 \varphi_1 + 2n \sin \varphi_1, \\ \rho &= \frac{\sqrt{C - 2n \sin \varphi}}{n}, \\ \rho_0 &= \frac{\sqrt{C - 2n \sin \varphi_0}}{n}. \end{aligned} \tag{2.10}$$

The θ_0 and φ_0 are the longitude and latitude of the origin of the Cartesian coordinates. The φ_1 and φ_2 are the standard lines. All the experiments with this projection utilise $(\varphi_0, \theta_0) = (0°, 0°)$ and the standard lines at $\varphi_1 = -40°$ and $\varphi_2 = 60°$.

Figure 2.35: Color map of a 3D point cloud. Generated by mapping the 3D point cloud with the Albers equal-area conic projection onto a panorama image. Standard lines at $\varphi_1 = -40°$ and $\varphi_2 = 60°$ are used for generation of this map.

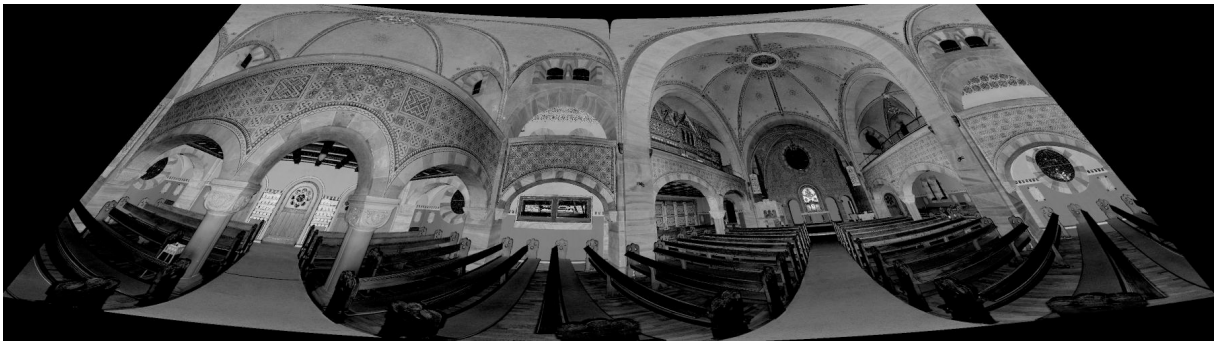


Figure 2.36: Reflectance map of a 3D point cloud. Generated by mapping the 3D point cloud with the Albers equal-area conic projection onto a panorama image. Standard lines at $\varphi_1 = -40°$ and $\varphi_2 = 60°$ are used for generation of this map.

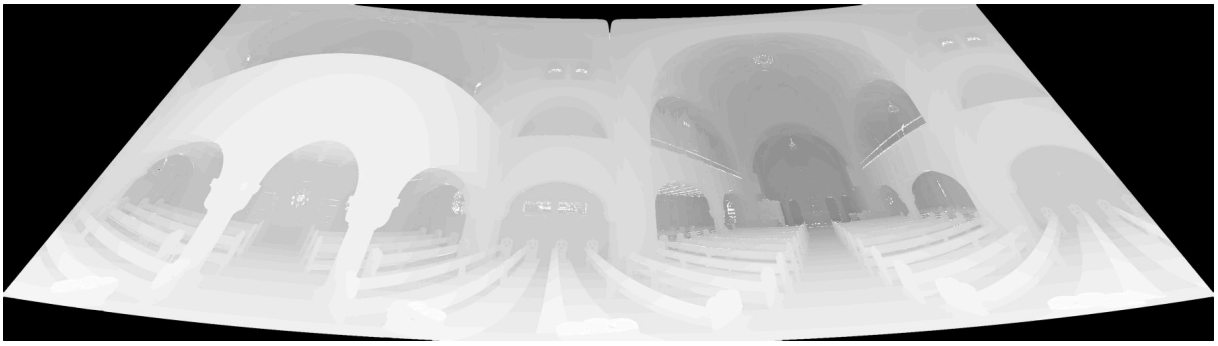


Figure 2.37: Range map of a 3D point cloud. Generated by mapping the 3D point cloud with the Albers equal-area conic projection onto a panorama image. Standard lines at $\varphi_1 = -40°$ and $\varphi_2 = 60°$ are used for generation of this map.

Figure 2.34 presents the Albers equal-area conic projection of an artificial sphere mapped onto a panorama image. Figures 2.35, 2.36 and 2.37 present a sample point cloud mapped with the Albers equal-area conic projection onto a panorama image.

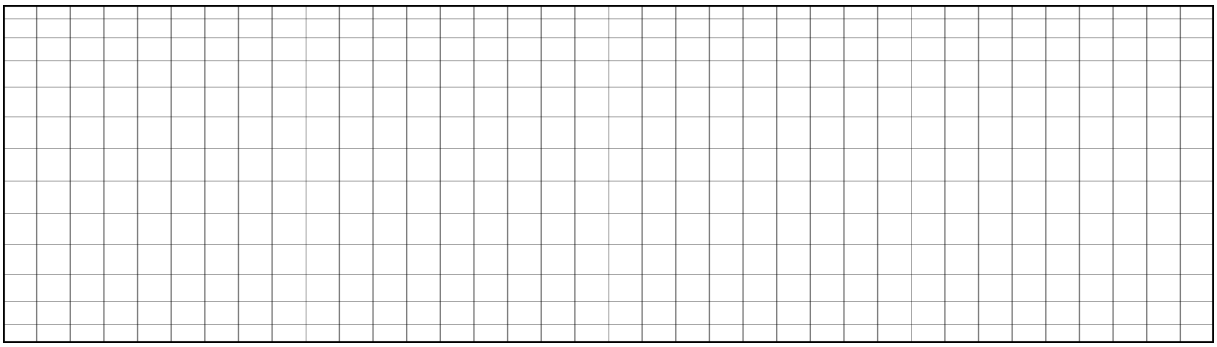


Figure 2.38: Latitude and longitude of mapped artificial sphere with the equal-area cylindrical projection. Standard lines at $\varphi = -30°$ and $\varphi = 30°$ are used for generation of this map.

Figure 2.39: Color map of a 3D point cloud. Generated by mapping the 3D point cloud with the equal-area cylindrical projection onto a panorama image. Standard lines at $\varphi = -30°$ and $\varphi = 30°$ are used for generation of this map.

2.3.9 Equal-Area Cylindrical

The equal-area cylindrical projection is a cylindrical projection that preserves the area. The longitude lines are mapped to equally spaced vertical lines and the circles of the latitude are mapped to horizontal lines. Similar to other cylindrical projections it can be envisioned by mapping a sphere onto a cylinder around the sphere. The projection is undistorted around the tangent line, but distortion increases rapidly towards the poles. Similar to other cylindrical projections the latitude lines away from the equator are stretched. Therefore, this projection is limited to a vertical field of view of less than 140°. The horizontal field of view for this projection, as for all cylindrical projections, is 360°. Various equal-area cylindrical projections are distinguished from each other solely by their north-south stretching. The ratio of the vertical and horizontal axis determines the standard latitude of the projection. The standard latitude has no distortion and distances along this line match the scale of the projection. Usually there are two sets of standard latitudes for equal-area cylindrical projection each at the same distance

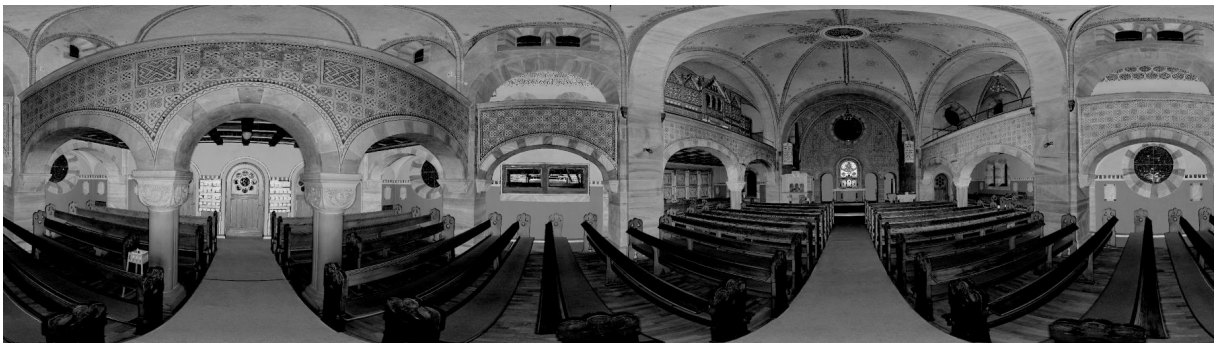


Figure 2.40: Reflectance map of a 3D point cloud. Generated by mapping the 3D point cloud with the equal-area cylindrical projection onto a panorama image. Standard lines at $\varphi = -30°$ and $\varphi = 30°$ are used for generation of this map.

Figure 2.41: Range map of a 3D point cloud. Generated by mapping the 3D point cloud with the equal-area cylindrical projection onto a panorama image. Standard lines at $\varphi = -30°$ and $\varphi = 30°$ are used for generation of this map.

north and south of the equator [214]. The transformation equations for this projection are:

$$\begin{aligned} x &= (\theta - \theta_0)\cos\varphi_s, \\ y &= \frac{\sin\varphi}{\cos\varphi_s}, \end{aligned} \tag{2.11}$$

where θ is the longitude, θ_0 is the horizontal center of the projection, φ is the latitude and φ_s is the standard latitude.Figure 2.38 presents the equal-area cylindrical projection of an artificial sphere mapped onto a panorama image.

Figures 2.39, 2.40 and 2.41 present a sample point cloud mapped with the equal-area cylindrical projection onto a panorama image.

2.3.10 Lambert Azimuthal Equal-Area

The Lambert azimuthal equal-area projection is an azimuthal projection that preserves the area. Unlike classical azimuthal projections this projection is derived mathematically and has no real perspective process. This projection is capable of mapping a full sphere onto a disk. However,

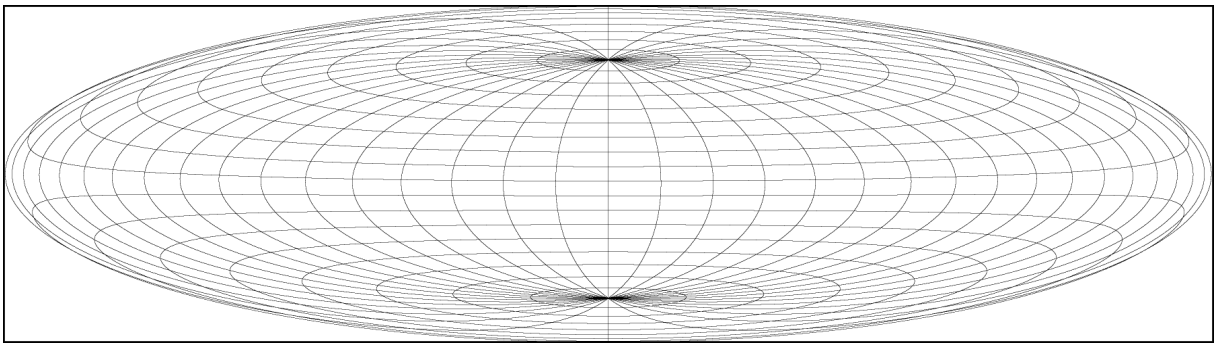


Figure 2.42: Latitude and longitude of mapped artificial sphere with the Lambert azimuthal equal-area projection.

for smaller distortions the full sphere is mapped onto two disk centered on antipodal points. This projection has a true scale and only at the center of the projection has no distortion. Distortion is moderate for one hemisphere but the map becomes strongly distorted for a full map of the sphere. The distortion increases for areas with longer radial distance to the center of the projection. This projection can be defined by placing a flat paper tangent to the sphere at point S. Any point P on the sphere other than the antipode of S is mapped to point P' on the flat image. Point P has a three dimensional distance d to point S. The mapped point P' on the flat paper is selected to have the same distance d from the center point S. This can be imagined as a circle perpendicular to the tangent plane and centered at S and passing through P. This circle intersects with the plane at two points and the mapped point P' is the point closer to the point P on the sphere. Using north or south pole as the center of the projection maps the latitude lines onto equally spaced straight lines intersecting at the central pole with true angles between them. It maps the circle of longitudes onto uniquely spaced circle centered at the pole [214]. The transformation equations for this projection are:

$$\begin{aligned} x &= k' \cos\varphi \sin(\theta - \theta_0), \\ y &= k'(\cos\varphi_1 \sin\varphi - \sin\varphi_1 \cos\varphi \cos(\theta - \theta_0)), \end{aligned} \tag{2.12}$$

where

$$k\prime = \sqrt{(\frac{2}{1 + \sin\varphi_1 \sin\varphi + \cos\varphi_1 \cos\varphi \cos(\theta - \theta_0)})}. \tag{2.13}$$

The θ and φ are the longitude and the latitude. The θ_0 and φ_1 are the central longitude and latitude of the projection. Figure 2.42 presents the Lambert azimuthal equal-area projection of an artificial sphere mapped onto a panorama image. The projection center of the projection is a tangent point on the sphere.

Figures 2.43, 2.44 and 2.45 present a sample point cloud mapped with the Lambert azimuthal equal-area projection onto a panorama image.

Figure 2.43: Color map of a 3D point cloud. Generated by mapping the 3D point cloud with the Lambert azimuthal equal-area projection onto a panorama image.

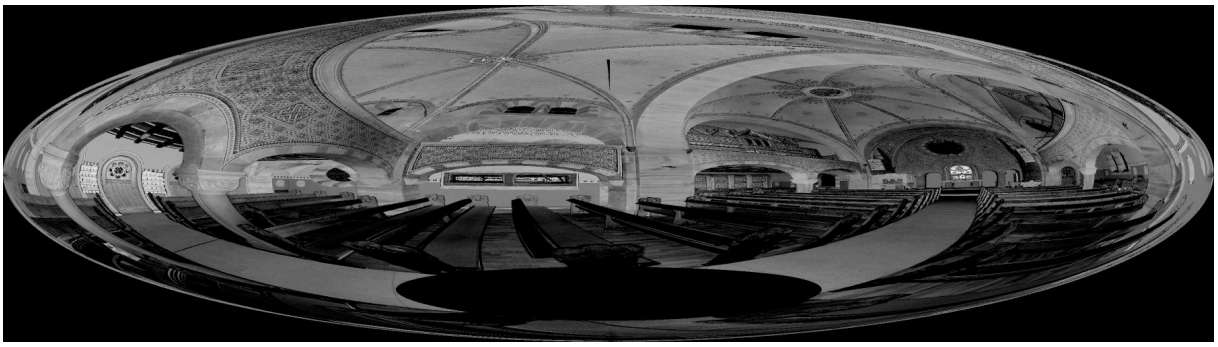


Figure 2.44: Reflectance map of a 3D point cloud. Generated by mapping the 3D point cloud with the Lambert azimuthal equal-area projection onto a panorama image.

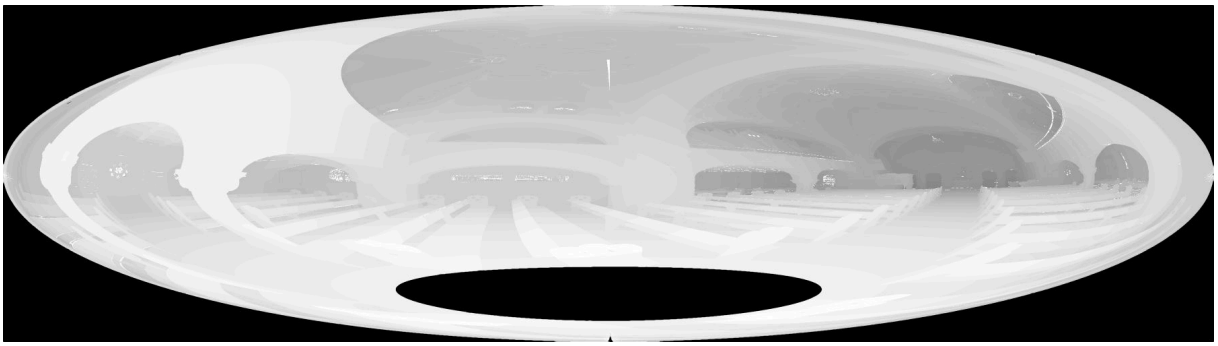


Figure 2.45: Range map of a 3D point cloud. Generated by mapping the 3D point cloud with the Lambert azimuthal equal-area projection onto a panorama image.

2.4 Panorama Map Methods

The panorama images are generated from the color, reflectance and range information provided by modern 3D laser measurement systems. The 2D representation of a 3D point cloud permits the use of conventional 2D algorithms for acquired 3D data. Section 2.3 presents the projection

methods for production of panorama images based on 3D point clouds. A predefined image size is required to generate such a panorama image. Since different panorama projections have different aspect ratios, using a fixed image size, i.e., fixed ratio, is not recommended due to the fact that fitting a panorama into a fixed image size requires different scaling for horizontal and vertical axes. Image size optimisation is carried out on the initial suggested image size. The optimised size has the aspect ratio of the used projection. This reduces the extra distortion introduced by fitting a panorama image to a fixed size image. All pixels of generated images contain color, intensity and range of the mapped 3D point cloud. Depending on the resolution of the panorama image, multiple 3D points are projected onto the same image pixel. Therefore, a panorama map is generated with a list of 3D points that fall into the array element. This list of 3D points are mapped to each panorama pixel. This method creates a connection between all the 3D points and panorama pixels. However, two additional methods can be used to select only one 3D point per panorama pixel. The panorama map method *FURTHEST* takes the point with the largest range from the list of 3D points of each panorama pixel. The panorama map method *NEAREST* takes the point with the smallest range from the list of 3D points of each panorama pixel. Depending on the point cloud and the required process on the data any of the panorama map methods can be used.

2.5 Summary

Advances in technology provide fast and reliable systems for remote sensing. Laser scanners are becoming more common among many research fields. Two categories of laser scanners namely the pulse based laser scanners and the phase based laser scanners are introduced in this chapter. Modern 3D laser measurement systems are capable of capturing a large number of points per second. The increased quality and quantity of the captured data with additional reflectance and color produces very dense point clouds. This chapter presented methods for point cloud representation.

This includes the typical representations such as k-d tree and octree. Additionally, it represents methods for mapping a point cloud onto a flat image. The 2D representation of the 3D data permits the use of conventional 2D algorithms for 3D point clouds. However, mostly only the equirectangular projection method is used. This chapter examines several panorama projection methods and modifies them for 3D point clouds. These methods are capable of mapping the captured range, reflectance and color of the point clouds. A panorama map is generated for each panorama image to contain the connection between each 3D point to the panorama pixels. Two additional methods are presented to filter the number of 3D points for each panorama pixel to one 3D point per pixel. Panorama images are generated for a variety of applications including feature-based registration introduced in Chapter 3, point cloud reduction and point cloud compression presented in Chapters 4 and 5.

Chapter 3

Point Cloud Registration

The aforementioned methods for point cloud storage represent the measured surface of captured objects in the local coordinate system. Digitisation of the entire environments without occlusions requires multiple scans. The generation of a rigid 3D model of structures, objects and environments requires the locally captured 3D point clouds to be merged into one coordinate system. This process is called scan matching or registration. The scan matching process is defined as the calculation of the relative transformation between a pair of overlapping point clouds. The basis of these algorithms is the establishment of correspondences between the two scans. The scans are examined for determination of the initial feature or point correspondences. An error measure is derived from the point correspondences. This error measure is minimised to compute the relative transformation between the local coordinate systems of point clouds. Manual registration methods require the help of the operator for selecting the scan pairs and the correspondences between scan points. Marker-based registration methods use defined artificial landmarks as corresponding points. This supervised data association ensures that the scans are registered at their correct location. Marker-based algorithms manually or automatically extract the 3D position of the markers for registration. However, they require the manual installation of markers on the scan site. This increases the time of the workflow and requires more interaction and alters the environment. Automatic scan-to-scan registration methods compute the transformation matrix between an overlapping scan pair based on the automatic detection of similarities in the point clouds. These similarities can be established based on either purely points or features in the point clouds.

In this chapter, first a technical background of registration methods is presented. Second, an automatic feature-based registration method is proposed. The proposed method relies on panorama generation methods (cf. Chapter2) to map the point clouds onto panorama images. 2D feature extraction methods are used to detected the required correspondences for scan registration. Section 3.2 provides an overview of the renown 2D features. After feature extraction process the proposed feature-based registration method introduces feature filtration methods to remove weak features. Next feature matching methods are presented and finally a random sample consensus (RANSAC) [76] based method is proposed for the calculation of the transformation matrix between a scan pair. This is a coarse registration process and can be optimised using Iterative Closest Point (ICP) algorithm. Algorithm 1 presents the simplified steps of the

Algorithm 1 The proposed feature-based registration method

Require: point clouds P_1 and P_2

1. $I_1, I_2 = generatePanoramaImagesFrom(P_1, P_2)$
2. $f_1, f_2 = extract2DFeaturesFrom(I_1, I_2)$
3. filter the extracted feature to remove the weak features
4. $m = detectMatchingFeatures(f_1, f_2, I_1, I_2)$
5. calculate transformation matrix between the scans P_1 and P_2 with the help of calculated correspondences from the feature matching step
6. optimise the calculated transformation using ICP

proposed feature-based registration. Finally, a few experiments with the proposed feature-based registration method is presented.

3.1 Technical Background

3.1.1 Featureless Registration

Featureless registration methods do not rely on the detection of features or artificial markers for the calculation of the relative transformation between an overlapping scan pair. Point-based registration methods register 3D point clouds based on the detected point correspondences. Many algorithms have been proposed in the literature. A popular algorithm for this task is the Iterative Closest Point (ICP) algorithm proposed by Besl and McKay [20]. Starting from an initial position estimate the algorithm calculates the correct transformation between two point clouds by iteratively searching for pairs of nearby points and minimising distances between corresponding points. Corresponding points are chosen based on Euclidian distance. The goal of the algorithm is to minimise the sum of all point to point distances. The algorithm converges to a local minimum. A good start estimate is essential to improve the matching result. This ensures the convergence to a correct local minimum. Given the reference point cloud M and a point cloud D with an initial pose estimate, the ICP algorithm detects for each point d_i in the point cloud D the point m_i in the point cloud M. The selection criteria are based on the distance and the point pair with the closest distance is selected. Afterwards the transformation that minimises the ICP error function is calculated (cf. Equation 3.1).

$$E_{ICP}(\mathbf{R},\mathbf{t}) = \frac{1}{N}\sum_{i-1}^{N} ||m_i - (\mathbf{R}d_i + \mathbf{t})||^2. \tag{3.1}$$

Nüchter et al. [174] present several minimisation methods for calculating the relative transformation matrix. The ICP algorithm iterates to find the best transformation between the two overlapping point clouds. Algorithm 2 presents the ICP algorithm.

Biber and Strasser [21] propose a method that is significantly different from the ICP algorithm. They introduce a new representation for a set of points, namely the Normal Distributions Transform (NDT). The NDT models the distribution of all points of one point cloud by a collection of local normal distributions. The method initially proposed by Biber and Strasser [21]

Algorithm 2 The Iterative Closest Point (ICP) algorithm

Require: point clouds M and D

1. $detectPointCorrespondences(M, D)$
2. minimise for rotation $\mathbf{R}$ and translation $\mathbf{t}$

$E_{ICP}(\mathbf{R},\mathbf{t}) = \frac{1}{N}\sum_{i-1}^{N} ||m_i - (\mathbf{R}d_i + \mathbf{t})||^2$

3. transform the source points with the returned pose ($\mathbf{R}$,$\mathbf{t}$)
4. iterate steps 1 and 2

works on 2D point clouds. The first step of the method subdivides the environment into regularly sized cells similar to an occupancy map of the environment. The occupancy grid maps represent a map of the environment on a grid. Each grid cell represents the presence of an obstacle in the location enclosed by the grid cell. Then for cells with a minimum of three points the mean vector and the covariance matrix are calculated. Finally the probability of measuring a point at position x in a cell is modelled by the normal distribution $N(q, \sum)$:

$$p(x) = C \cdot e^{(-\frac{(x-q)^t \sum^{-1} (x-q)}{2})}, \tag{3.2}$$

where q and $\sum$ are the calculated mean vector and covariance matrix for the cell containing the point x. C is the normalising constant that is set to one. Similar to an occupancy grid map the NDT method divides the environment into cells. However, an occupancy grid map represents the probability of a cell being occupied, the NDT represents the probability of measuring a sample for each position within the cell. NDT attempts to find a transformation between two point clouds by maximising the probability of the transformation. The NDT scan matching algorithm starts with building the NDT for the first scan and then initialisation of the transformation and rotation. The initial pose estimation is used to map the samples from the second scan to the first scan frame. Afterwards, the corresponding normal distribution of the mapped sample point is determined. A score is calculated for the used pose estimation by evaluating the distribution for each mapped sample and summing the results. A new transformation matrix is calculated to optimise the score. The algorithm iterates till a convergence criterion is met. Magnusson et al. [150] present a generalisation and improvement of the NDT method for 3D data.

3.1.2 Feature-Based Registration

State of the art registration methods including ICP mostly rely on initial position and orientation (pose) estimation of acquired data to determine the correct transformation between 3D point clouds. Sources for this estimation includes Global Positioning System (GPS) and local positioning using artificial landmarks [238]. Some methods rely on the ability of a robotic platform to roughly estimate its own pose. The pose can be estimated using the wheel encoder. This process is referred to as odometry. Additional Inertial Measurement Unit (IMU) devices are used to increase the accuracy of the estimated pose. The measured accelerations from the

IMU are integrated into the pose estimates. However, the error in the trajectory sums up and worsens the longer the path and the larger the amount of rotation on the path. To overcome this problem roboticists commonly use Simultaneous Localisation and Mapping (SLAM) solutions. The SLAM problem is the problem of localising the robot in a map while creating the map at the same time. The initial pose information is hard to acquire and in many scenarios prone to errors or not available at all. If no pose estimation were acquired during the data collection the remaining option is the determination of the pose estimation directly from the data. Automatic algorithms favour the automatic extraction of the natural features from the scans. These features are then exploited to find the point correspondences between scans. Thus, registration without initial pose estimation is a highly active field of research.

The information of the captured data with a laser scanner can be used for the detection of natural features. The 3D data is thus projected onto an image. The generation of panorama images enables us to use 2D feature matching methods. These features describe image patches with high intensity variations with distinguishable descriptors. These descriptors are used to detect the correspondences between scans. Hansen et al. consider an application of scale-invariant feature detection using scale-space analysis suitable for use with wide field of view cameras. They map the image to the sphere and obtain scale-space images as the solution to the heat (diffusion) equation on the sphere [102, 103]. Lee et al. present a matching method in order to find the correspondences of features in two omnidirectional images. Dominant corresponding feature pairs are found using a proximity matrix and a sum of squared differences (SSD) based similarity matrix, and then the remaining feature matching is accomplished by dynamic time warping (DTW) [142]. Parida et al. present a unified approach to detect, classify, and reconstruct junctions in images. Their approach is a combination of the two paradigms: edge detection followed by grouping of edges to form junctions (via Dynamic Programming) and treating a junction as a template matching process. They use a template deformation framework using the minimum description length (MDL) principle [181]. By extending this approach, Förstner et al. proposed a novel method for detecting scale invariant keypoints and a scale space mechanism for junction type features [80]. Common appearance based place recognition approaches often rely only on camera data and are not suitable for laser scans [25, 54, 55, 134, 234].

Captured intensity of the reflected light with laser scanners provide additional information for the registration process. Böhm and Becker suggest the use of scale invariant feature transform (SIFT) features for automatic registration and present an example of a successful registration of 3D scans with a small field of view [24]. Wang and Brenner extended this work by using additional geometry features to reduce the number of matching outliers in panoramic outdoor laser scans [239]. Kang et al. propose a similar technique for indoor and outdoor environments [119]. Weinmann et al. use a method that is based on both reflectance images and range information. After the extraction of characteristic 2D points based on SIFT features, these points are projected onto 3D space by using interpolated range information. Combining 3D points of a new scan with 2D observations on a virtual plane yields 3D-to-2D correspondences from which the coarse transformation parameters can be estimated via a RANSAC based registration scheme including a single step outlier removal for checking consistency [243]. They extend their method in [242] to calculate the order of the scans in unorganised terrestrial laser scan data by checking the similarity of the respective reflectance images via the total number of SIFT correspondences between them. Bendels et al. exploit intensity images and propose a fully automatic registration

technique using 2D-image features. The fine registration of two range images is performed by first aligning the feature points themselves, followed by a so-called constrained-domain alignment step. In the latter, rather than feature points, they consider feature surface elements. Instead of using a single 3D point as feature, they use the set of all points corresponding to the image area determined by the position and scale of the feature [17]. Other approaches rely only on the 3D structure. Brenner et al. use 3D planar patches and the normal distribution transform (NDT) on several 2D scan slices for a coarse registration [39]. Similarly, Pathak et al. evaluate the use of planar patches and found that it is mostly usable [183]. A solution using the NDT in 3D is given in a study by Magnusson et al. [149]. While this approach computes global features of the scan, several researchers use features that describe small regions of the scan for place recognition and registration [12, 110, 219]. Flint et al. [77] use a key point detector called THRIFT, to detect repeated 3D structures in range data of building facades. In addition to coarse registration, many authors use the well-known iterative closest point algorithm (ICP) for fine registration [20, 47, 251]. ICP requires no computation of features. Instead, it matches raw point clouds by selecting point correspondences on the basis of smallest distances and by minimising the resulting Euclidean error. This iterative algorithm converges to a local minimum. Good starting estimates improve the matching results drastically, i.e., they ensure that ICP converges to a correct minimum.

Another field of research is the co-registration of point clouds and camera data. Stamos and Allen present a semi-automatic method for image to model registration of urban scenes, where 3D lines are extracted from the point clouds of buildings and matched against edges extracted from the images [218]. Aguilera et al. use the Förstner operator [79] to exploit its high accuracy and reliability in the localisation of the interest points, and its robustness to noise [2]. Meierhold et al. use SIFT features for the task [154]. Böhm et al. also present a successful point cloud to camera matching algorithm [24]. Wu et al. use a novel viewpoint-independent patch (VIP) on the image sequence of urban scenes. They use structure from motion to compute the depth map and camera positions. VIPs are features that can be extracted from textured 3D models which combine images with corresponding depth maps. A single VIP correspondence uniquely defines the 3D similarity transformation between two scenes [245]. Köser and Koch extend the 2D feature concept to the third dimension by producing a descriptor for a constructed 3D surface region. They combine the concepts of stable texture interest points and 3D geometry. Their feature represents texture and surface information in a perspective invariant fashion by computing a normal view onto the surface. Their approach can be exploited for structure from motion, for stereo cameras or alignment of large scale reconstruction [135].

3.2 2D Features for 3D Point Clouds

Many algorithms have been proposed for the extraction of features, so-called keypoints, from grey scale images. Descriptors are used to encode the feature neighbourhood. They are used to present the keypoint with a highly distinctive structure. Robust and unique features are matchable from different viewing angles. The chapter proposes the generation of panorama images from 3D point clouds to introduce the use of 2D features for detecting 3D point correspondences. The matched features provide the 3D point correspondences for point cloud registration. The

proposed method in this chapter utilise these sets of points in a RANSAC-like algorithm to determine the transformation between scan pairs. This pose estimation is used as the initial pose estimation for high accuracy scan registration methods such as ICP. Next, many features and descriptors are described. Two types of features are presented. The first group are the features that are more robust and more invariant to noise, rotation and camera position. The second group is optimised for real time performance. These features sacrifice some accuracy to reduce resource consumption.

3.2.1 Scale invariant feature transform (SIFT)

The Scale Invariant Feature Transform or SIFT is proposed by Lowe [144, 145]. This is a popular algorithm in computer vision to describe and determine distinctive invariant local features from images. These features can be used to perform reliable matching between different views of an object or scene. To achieve reliable matching it is necessary that extracted features are invariant to scale and rotation. Furthermore, it is important that features provide robust matching across a substantial range of affine distortions, change in 3D viewpoint, addition of noise and change in illumination. SIFT features fulfil all these requirements. Additionally a high number of features are detected using an efficient algorithm. SIFT features are distinctive and single feature is likely to be correctly matched to a large database of features, thus providing a basis, e.g., for object recognition. The major steps of the SIFT feature extraction process are scale-space extrema detection, key point localisation, orientation assignment, and key point description.

The first step of feature detection is to identify interest points repeatedly under differing views and scales. Initially an image pyramid is built by applying Gaussian filters of different sizes to the image. An image pyramid with multiple images with various scales are the foundation of the scale invariance of this method. This allows feature detection over all scales and image locations. Subsequently, differences between consecutive levels of Gaussian convolved images are computed for efficient detection of stable feature location in scale space. This step is referred to as the Difference of Gaussians. Afterwards local maxima of the Difference of Gaussians are computed by comparing each sample pixel to its eight neighbours in the current image and nine neighbours in the scales above and below. All local maxima points are considered as potential key points. Thereafter, the potential key points are refined and filtered and all local maxima that do not correspond to edges in the image and that have high contrast are selected as features.

The feature orientation is determined on the basis of local image properties. Rotation invariance is achieved by representing the feature descriptor relative to the orientation of the feature. The feature orientation is determined by calculating the magnitude of the gradient $m(x, y)$ and the orientation $\theta(x, y)$ of the local neighbourhood around a feature on the image L of the scale where the feature was detected at pixel x and y:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}, \tag{3.3}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right). \tag{3.4}$$

Afterwards a 36 bin orientation histogram is generated. The magnitude of each neighbourhood point, weighted by a Gaussian-weighted circular window is added to the bin as determined by the angle $\theta(x, y)$. The largest peak is selected as the orientation of the feature. In case multiple substantial peaks are identified, multiple features with the same position and different orientations are generated.

After the calculation of the scale, position and orientation of the feature, the next step is the computation of the feature descriptor. The surrounding area of the feature in a window of size 16×16 is considered for the computation. This window of pixels is oriented according to the feature orientation. The orientation and magnitude for each pixel is computed. The window is grouped into a 4×4 array. In each of the 4 pixel wide cells an orientation histogram with 8 bins is computed in the same manner as in the previous step. A feature descriptor of $4 \times 4 \times 8 = 128$ dimensions is constructed by concatenating the orientation histograms.

The SIFT feature matching process utilises a modified version of the k-d tree search known as Best Bin First [16] due to inefficiency of the standard k-d tree for high dimensional spaces. The ratio of the Euclidean distance between the feature and the nearest neighbour to the feature and the second nearest neighbour is proposed to determine a positive match in the matching process. This method is utilised in favour of the method of thresholding the distance, considering that the determination of certain thresholds is a demanding task [144, 145].

3.2.2 Speed-up robust features (SURF)

The Speed-Up Robust Features or SURF algorithm is a scale- and rotation-invariant detector and descriptor[14, 15]. The SURF features are highly repeatable, distinctive and robust, yet their computation and matching process is efficient. SURF relies on integral images for image convolution. Its detector is based on existing methods of a Hessian matrix-based measure and a distribution-based descriptor. The search for distinctive image points is divided into two steps: First, interest points are selected in distinguishable and repeatedly detectable locations in the image. Next, the descriptor of the detected point describes the intensity of its neighbourhood. The descriptor of the interest point is distinctive and robust to noise.

The SURF interest point detection is based on Hessian detectors [155, 157]. SURF employs image filters that are simple approximations of the second order derivative of the Gaussian. The filters are so simple that the convolution can be computed extremely fast with the help of integral images. Pixels of integral images present the summation of pixels above and to the left of it. The filters vary in size, starting with 9×9 pixels at the finest resolution. Using differently sized filters results in a similar image pyramid as in SIFT. Local maximas are detected in the image pyramid by non-maximum suppression in a $3 \times 3 \times 3$ neighbourhood. Maximas below a threshold are eliminated. The remaining points are the detected features. In this process scale invariance is achieved by up-scaling the filter rather than reducing the image size.

The SURF descriptor represents the distribution of the intensity within the neighbourhood of the interest point. The descriptor is a vector of 64 dimensions and is based on the distribution of the first order Haar wavelet [98] responses in the x and y direction. Again, integral images are utilised for speed. The SURF descriptor generation has two steps: First, determine a reproducible orientation based on information from a circular region around the interest point to make the SURF descriptors invariant to rotation. Second, construct a square region aligned

to the assigned orientation and extract the descriptor.

The dominating orientation is estimated with a sliding orientation window. The Haar wavelet responses of the points surrounding the feature are 2D orientation vectors. For each orientation window the Haar wavelet responses falling inside the window are summed up. The window with the largest orientation vector is selected as the orientation of the interest point. Afterwards a square region around the interest point is constructed and oriented along the determined orientation. This region is divided into 4×4 square sub-regions. For each sub-region the Haar wavelet responses at 5×5 regularly arranged sample points are computed. The wavelet responses are summed up in each sub-region. In order to include the information about the polarity of the intensity changes, the sum of absolute values of the response are calculated. Therefore, each sub-region has a four-dimensional descriptor vector for its intensity structure. Concatenating these vectors for all 4×4 sub-regions obtains a 64 dimensional descriptor vector.

3.2.3 Features from accelerated segment test (FAST)

The majority of feature detection algorithms compute corner response functions across the image. An edge in an image defines a change in intensity that is the boundary between two regions of an image. The direction of a boundary changes rapidly at corners. The Feature from Accelerated Segment Test or FAST feature detector determines its features by examining a small circle with a circumference of sixteen pixels around corner candidates [192, 193]. A candidate point p is considered a feature if the intensity of a set of n contiguous pixels in the circle are all brighter or darker in comparison to the intensity of the candidate pixel by some threshold. The value of n was chosen to be twelve by Rosten and Drummond [192]. The testing process is optimised by examining the four pixels at the main compass directions first to reject candidate pixels more rapidly. A pixel is only a feature if at least three of the corner points are all darker or brighter than point p by the threshold. The full segment test is applied to the remaining candidates from the optimisation process by examining all sixteen pixels in the circle. This is a sufficiently fast detector with high performance. However, the optimisation does not generalise for $n < 12$ and the knowledge from the first 4 steps is discarded. Multiple features are also detected adjacent to one another. Rosten and Drummond present a machine learning algorithm to address the first two weaknesses of the original FAST feature detector, in which a non-maximal suppression has been utilised for the latter weakness. Furthermore, in [194] they present a new heuristic for feature detection in combination with machine learning that generates a feature detector that can fully process a live video using less than 5% of the processing power. They also optimised the detector for repeatability with little loss of efficiency.

3.2.4 Oriented FAST and rotated BRIEF (ORB)

The Binary Robust Independent Elementary Feature or BRIEF is an efficient feature descriptor that utilises binary strings [43]. Binary strings are highly distinguishable even with relatively few bits. Descriptors are directly computed from image patches. The individual bits are acquired by comparing intensities of pairs of pixels. Hamming distances are utilised for the matching process to evaluate the similarity between descriptors. This is more efficient in comparison to the more common $L2$ norm. BRIEF shows that only 256 bits, or even 128 bits, often suffice to obtain

superior matching results. Therefore, BRIEF is very fast in both the process of constructing and matching features.

BRIEF is based on the classification of image patches by the pairwise intensity comparison of a relatively small number of pixel pairs. Each bit τ in the descriptor corresponds to one pair of pixels x, y in the smoothed image. The bits are set as follows:

$$\tau(x, y) \coloneqq \begin{cases} 1 & \text{if } I(x) < I(y) \\ 0 & \text{otherwise} \end{cases}, \tag{3.5}$$

where $I(x)$ is the intensity of the pixel x. The descriptor is defined by n pairs of (x, y), and the bit string presentation is given by:

$$f_n \coloneqq \sum_{1 \leq i \leq n} 2^{i-1} \tau(x_i, y_i). \tag{3.6}$$

The pairs of pixels to compare are randomly selected. Different types of probability distributions can be used for the selection [43]. Drawing pixel pairs from a Gaussian distribution around the center presents the most promising results. The BRIEF descriptor is a good compromise between speed, storage, efficiency, and recognition rate. Since the process of generating the feature descriptor is based on the information from only a few pixels, the descriptor is noise sensitive. Thus, BRIEF utilises pre-smoothed patches to reduce the noise sensitivity.

Oriented FAST and Rotated BRIEF or ORB is a combination of an oriented FAST detector with a very fast binary descriptor based on BRIEF [196]. Given that these two techniques have shown promising performances they are both utilised for ORB features. FAST features are affected by the lack of orientation and BRIEF is limited as a consequence of the absence of rotational invariance. ORB introduces a fast and accurate orientation for FAST features and the efficient computation of rotated BRIEF.

The FAST detector utilises an intensity threshold between a pixel and pixels in a circular boundary around the center. FAST has a large response along edges. However, it lacks a measurement for the cornerness of detected corners. Thus, FAST features are filtered by using the Harris corner measure. First a low threshold is utilised in order to generate a high number of features. Afterwards, the N features with the highest Harris measure [155, 157] are selected. FAST features are scale dependent. ORB employs a scale pyramid of images to obtain scale invariance. The orientation measurement for FAST features is generated by calculating the angle of the offset between the center of the patch and the intensity centroid. This computation is simplified to:

$$\theta = \operatorname{atan2}\left(\sum_{x,y} yI(x, y), \sum_{x,y} xI(x, y)\right). \tag{3.7}$$

This generates orientation aware FAST features. ORB also introduces the rBRIEF, which is a rotation aware BRIEF descriptor. Rotation invariance is achieved by rotating the patches in accordance to the determined orientation. To improve the results of the descriptor the binary tests on pixel pairs are not randomly generated. Instead, a greedy search algorithm was employed on the PASCAL 2006 data set [75] to determine a good set of binary tests.

3.2.5 Center surround extremas (CenSurE)

The Center Surround Extremas or CenSurE feature detector is a scale invariant center surround detector [1]. This detector computes features at all scales using pixels in the original image to maintain accuracy. A Harris-Laplace [156] and Hessian-Laplace [156] detector computes the Hessian or Harris measure [155, 157] at all locations and scales. For every detected candidate the Laplacian is computed at all scales. This strategy is adapted by using extrema of a center surround response as an approximation. The CenSurE detector computes a simplified center surround filter at all locations and scales. Afterwards, the local extremas in a neighbourhood are found. Finally, these extremas are filtered by computing the Harris measure and eliminating those with weak corner response. The CenSurE uses bi-level center surround filters, which is a simple approximation of the Laplacian with the Difference of Gaussians. It multiples the image value by either 1 or -1. A filter that is both faithful to the Laplacian and is fast to compute with integral images needs to be designed. A circular filter is the most faithful to the Laplacian, however, hard to compute. Agrawal et al. proposed the use of an octagon filter for good performance and a box filter for good computation time [1]. In the experimental section of this chapter the STAR detector which is an OpenCV modified version of the CenSurE detector is used [37]. Agrawal et al. [1] proposed a non maximal suppression in a $3 \times 3 \times 3$ neighbourhood in the image pyramid. The magnitude of the responses is taken as an indication of the strength of the feature for being stable. The greater the strength, the more likely it is to be repeated. Weak responses are likely to be unstable. Furthermore, weak features are discarded using a threshold t_r for the response. To filter out the features that lie along an edge or line, the second moment matrix of the response function at the particular scale is used:

$$\boldsymbol{H} = \begin{pmatrix} \sum L_x^2 & \sum L_x L_y \\ \sum L_x L_y & \sum L_y^2 \end{pmatrix}, \tag{3.8}$$

where L_x and L_y are the derivatives of the response function L along x and y within a window that is linearly dependent on the scale of the particular feature point. This is the scale adapted Harris measure and is different from the Hessian matrix used by SIFT [145] to filter out line responses. Once the Harris measure is computed, its trace and determinant is used to compute the ratio of principal curvatures.

3.3 Feature Filtration Methods

Feature detection methods have a tendency of detecting many features on images. This behaviour is favourable most of the times to produce a broad set of features for matching. However, some of the detected features are not robust across several images. As an example in an outdoor environment feature detectors will detect many features on vegetation or mesh like surfaces such as fences. These features are very unstable and not suitable for matching. When using panorama images generated from 3D point clouds the range information of each image pixel is known. This range is the 3D distance of each 3D surface point that is represented by the image pixel. A feature filtration method can eliminate some of the unstable features to

improve the matching process. A simple filtration over the range information of features selects a 3×3 image pixel patch around the feature. Features that have range differences of more than a threshold with its neighbouring pixels are eliminated. A more sophisticated method is to calculate the standard deviation of the feature patch. Only features with a small standard deviation are selected for feature matching between panorama images. These feature filtration methods increase the efficiency of scan matching by reducing the required time for feature matching and by eliminating the unstable features.

3.4 2D Feature Matching Methods

To calculate the transformation matrix in a pairwise 3D point cloud scenario a list of point correspondences is required. This list contains a set of 3D points from one point cloud and the corresponding 3D point from the other point cloud. Matching the determined image features from 3D point clouds provides this correspondence list. The matching process depends on the type of utilised feature descriptors which must be the same for both input 3D point clouds. Matching means identifying a feature that is closest to a sampled feature-based on descriptor comparison. Algorithms such as k-nearest neighbour (KNN), Radius-KNN and brute-force are possible efficient solutions to this problem. The ratio nearest neighbour search as presented by Lowe [145] which is a modified version of nearest neighbour search has shown the most promising results. For the ratio nearest neighbour search the distances of the feature descriptors to the sampled feature are used. The ratio of the distance of the first closest descriptor to the distance of the second closest descriptor is calculated. A threshold on the ratio determines whether a sample feature is a valid match. Feature descriptor distances are defined as the Euclidean distance between descriptor vectors for the SIFT and the SURF and the Hamming distance in case of the ORB and the BRIEF feature descriptors. The Hamming distance calculates the minimum number of differences between two strings with the same length. This is used for ORB and BRIEF descriptors to determine the difference between feature descriptor strings.

3.5 Coarse Registration

3D point cloud registration is the last stage of feature-based registration. To generate a transformation matrix in a pairwise registration scenario a set of feature matches is used as correspondences. The registration process starts by taking as input a set of matched features from two panorama maps. As presented in Chapter 2 a panorama map stores for each panorama image pixel a set of corresponding 3D points from the point cloud. In the experiments of this chapter the point with the largest range is selected. This way the method favours points on buildings instead of vegetation [72].

The registration proceeds by testing combinations of 3 point pair matches and examining the two triangles that are generated by these point pairs. If the matched feature set has n elements the number of 3-combinations is equal to the binomial coefficient:

$$\binom{n}{3} = \frac{n!}{3!\,(n-3)!}. \tag{3.9}$$

To avoid the increase of the computation time based on the large number of matches a random sample consensus or RANSAC-like [76] algorithm is used. RANSAC is an iterative algorithm for mathematical model parameter estimation from a set of data with outliers. A randomly selected subset of combinations of 3 point pair matches is used. The probability that the algorithm finds the correct solution increases with the number of iterations. The algorithm calculates the centroid of each triangle and translates the triangles so that their centroids are at the center of the reference frame. The orientation that minimises the euclidean distance of the corresponding features in 3D is computed by the closed form solution proposed by Horn [107]. The rotation is presented in the form of a quaternion which is the eigenvector corresponding to the maximum eigenvalue of the following matrix:

$$\boldsymbol{N} = \begin{pmatrix} S_{xx}+S_{yy}+S_{zz} & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & -S_{xx}+S_{yy}-S_{zz} & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & -S_{xx}-S_{yy}+S_{zz} \end{pmatrix}, \tag{3.10}$$

with

$$S_{mn} = \sum_{i=1}^{3} p_{im}^{\alpha} . p_{in}^{\beta}, \tag{3.11}$$

where α and β correspond to the first and second triangles respectively, m and n to one of three coordinates x, y or z. Afterwards the rotation matrix R is given by:

$$\boldsymbol{R} = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{pmatrix}, \tag{3.12}$$

where a, b, c and d are the parameters of the eigenvector $\boldsymbol{v} = (a, b, c, d)^T$ which corresponds to the maximum eigenvalue of N. Consequently the transformation matrix for homogeneous coordinates is given by:

$$\boldsymbol{M} = \begin{pmatrix} & \boldsymbol{R} & & \boldsymbol{t} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.13}$$

where $\boldsymbol{t}$ is the translation vector and is computed by:

$$\boldsymbol{t} = \boldsymbol{R}(-\boldsymbol{c}_{\beta}) + \boldsymbol{c}_{\alpha}. \tag{3.14}$$

Here, c_α and c_β are the centroids of the triangles α and β. The quality of the transformation matrix M obtained for a pair of triangles is estimated by computing the distances between all pairs of matched points. The pairs of points with an error lower than a threshold λ_d are considered as inliers. In all the experiments in this chapter the threshold is set to $\lambda_d = 0.5$m. For transformations with more then 10 inliers the following quality metric is computed. It depends on both the number of inliers n as well as the sum of distances between matching points E :

$$Q = E - In^2. \tag{3.15}$$

Here, I is a parameter that scales the dimensionless number of inliers. The transformation matrix with the highest quality is selected as the output of this algorithm.

3.6 Globally Consistent Scan Matching

Registration of 3D point cloud pairs using the mentioned algorithms is prone to error. These errors are due to imprecise measurements and small registration errors. The small local errors for the registration of each scan pair add up and lead to a larger error for long sequences. Globally consistent scan matching algorithms aim to reduce these errors. The global matching algorithms take correspondences from all scans into account. Lu and Milios [146] presented a solution using a network of relations between laser scans poses. A single linear equation system yields optimal estimations for all poses. This approach is limited to 2D laser scans. Borrmann et al. [33] extended the linear estimation algorithm to work reliably with 3D scans and 6 degrees of freedom. Point pairs are determined for each scan pair on a graph of corresponding scans and the new error function solves for the transformation for all scans simultaneously:

$$E = \sum_{j \to k} \sum_{i} ||R_j m_i + t_j - (R_j d_i + t_k)||^2. \tag{3.16}$$

R and t are the rotation and translation and the m and d are the point correspondences and i and j are the list of corresponding points and scans respectively. Methods to minimise this error are presented in Nüchter et al. [174]. This method matches the laser scans with global consistency. Additionally, it avoids the common problems of sequential matching strategies by minimising a global error function in a bundle adjustment like manner.

3.7 Case Studies

The first three experiments were carried out on a machine with a Quad-Core processor AMD Athlon$^{\text{TM}}$II X4 640 CPU and 8GB RAM. For the implementation of feature detectors and descriptors and feature matching the OpenCV library [37] is utilised. The remainder of the process was implemented as part of 3DTK – The 3D Toolkit which provides methods to process 3D point clouds and algorithms for highly-accurate 6D SLAM [173, 177]. All scans in these experiments were acquired with a terrestrial laser scanner, the VZ-400 scanner by the RIEGL

Measurement GmbH. Four data sets were acquired, two small indoor data sets and two large outdoor data sets. The panorama projected reflectance images of each scan are generated with several of the proposed panorama projections in Chapter 2. Due to the restricted horizontal field of view of the Pannini, rectilinear and stereographic projections, scans were divided into three subsets for these projections. Afterwards, features are detected on each image and local descriptors are generated. Matched descriptors are detected in a pairwise approach. The ratio nearest neighbour based matching is utilised with a RANSAC-like filtering approach for these experiments. A threshold of 0.8 is used for the ratio based matching as a measure for eliminating false matches. Scan registration proceeds on sequential pairwise scans. The success of registration is determined via visual comparison of results with the ground truth. Ground truth was acquired by marker-based manual pre-registration and fine-registration using slam6D from the 3DTK. The reflective markers are used for the purpose of this manual pre-registration.

SIFT, SURF, ORB, STAR, and FAST feature detectors were used with combination of SIFT, SURF, and ORB descriptors which results in 15 possible combinations of these feature detectors and descriptors. Combining these 15 possibilities with 7 of the mentioned projections in Chapter 2 and with two resolutions produces 210 configurations for each scan pair of these three data sets.

The data for the last two experiments were acquired with the help of a mobile robot. The scanner was mounted on the mobile robot Irma3D (Intelligent Robot for Mapping Applications in 3D) for the acquisition of the scans in a stop-scan-go fashion. Irma3D is a small, battery-powered, light-weight three-wheeled vehicle. The robot is based on the Volksbot RT 3 chassis with two motorised front wheels. Each front wheel is actuated by an individual 150W DC Maxon motor. The motors are capable of moving the robot at maximum velocity of 2.2m/s. The robot chassis consist of a third wheel in the back of the robot. This wheel is swivel-mounted and thus completely passive as it follows the direction of the front wheels. The two wheel drive system of the robot is equipped with rotary encoders to measure wheel rotations. This information is used to provide pose estimates of the robot via odometry.

The pose estimates for the last experiment are improved with the help of the Xsens MTi IMU (Inertial Measurement Unit) device attached to the robot chassis. Additionally a Sick LMS 100 2D laser scanner is attached to the front of the robot to improve the localisation of the robot. Similar to other experiments in this chapter the central sensor on the mobile robot is the Riegl VZ-400 3D laser scanner. The Canon 1000D DSLR camera mounted on the scanner provides the color information for all the measured points in the last experiment. In this experiment an iSpace sensor frame is also mounted on top of the laser scanner. iSpace is a high precision position and tracking system from Nikon Metrology [169].

The optical laser based system consists of several transmitters. The transmitters are mounted on walls or tripods to cover the entire experimental environment. The transmitters consist of a rotational head that emits two perpendicular fan-shaped laser beams at a unique distinguishable frequency near 40Hz. The vertical opening angle of the laser beams is limited to 40 degrees and the detectable range lies between 2 to 55 meters. Several sensor frames can be located within the system. A sensor frame consists of at least one detector. The detector is a photo diode with a 360 degrees horizontal field of view and a 90 degrees vertical field of view. The captured sensor data is transmitted to the iSpace base station wirelessly with the help of a small radio frequency module. The iSpace base system is a PC running the iSpace control software. A sensor frame

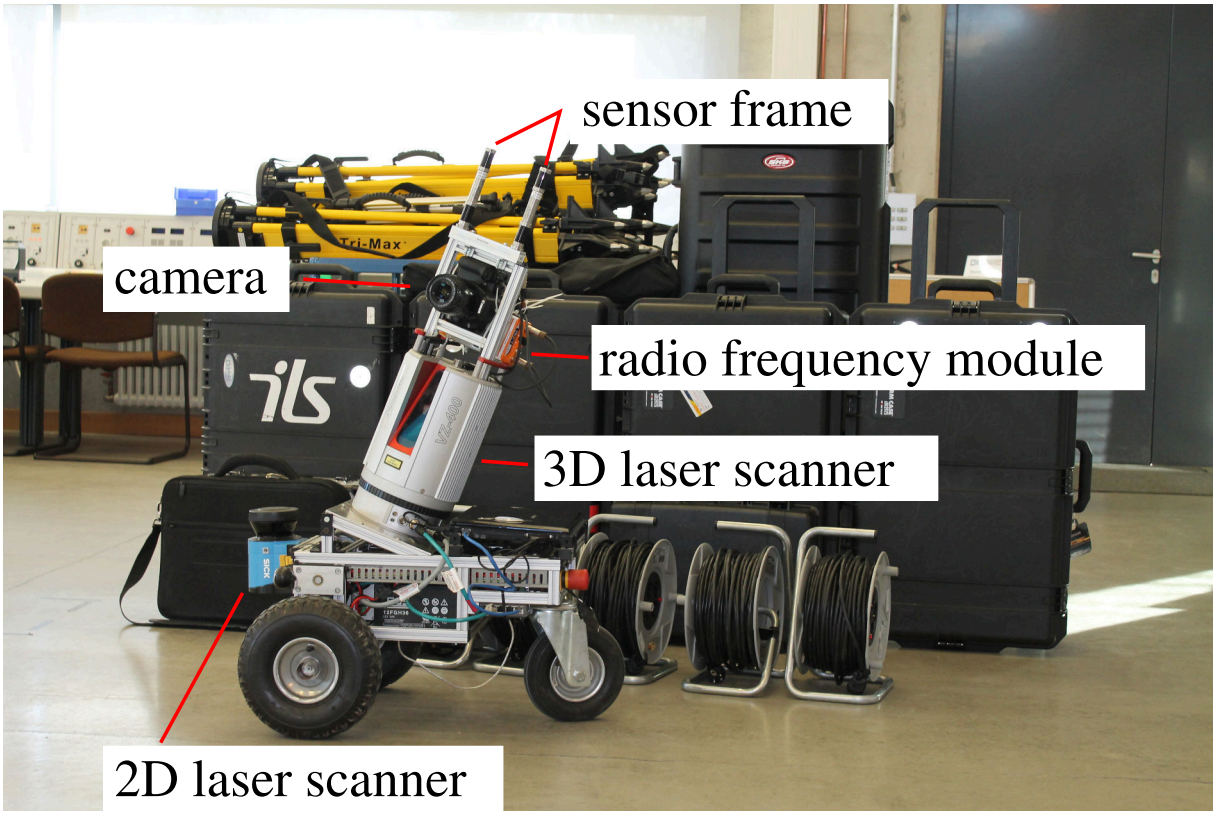


Figure 3.1: The Irma3D mobile robot with the setup used in the indoor experiment at the Residence Palace in Würzburg Germany.

with one detector is sufficient to detect the 3D position information. However, to measure the rotation and increase the accuracy of the position data a sensor frame with four detectors is used on the robot for these experiments. A sensor frame with two detectors, the handvector bar, is used to measure the location of points in the environment to evaluate the quality of the resulting model. In contrast to other position and tracking systems the iSpace transmitters do not actively observe the position of the sensor frames. The sensor frame transmits the received data from the transmitters to the iSpace control PC. The control system calculates the elevation and the azimuth angles between all detectors for a sensor frame and each visible transmitter based on the received data defining a straight line between the transmitter and the detector. Given the relative transformation between the transmitters the length of the line is calculated using triangulation. A calibration process determines the position of the transmitters. This procedure utilises a few hundred points from a special sensor frame. An optimisation process calculates the position of all transmitters in a self defined coordinate system. Three points, the origin, a point on the x axis and a point on the y axis allow the user to define its own coordinate system. In controlled environments the iSpace system is able to perform measurements at a sampling rate of 40Hz with a maximum error of $[\pm 0.25]$mm. However, in practice environmental factors such as size, reflection of the surface and occlusions of the transmitters have to be taken into consideration. A

Figure 3.2: Reflectance image generated from the scans captured with the Riegl VZ-400 scanner from the Interfaith House of the Jacobs University Campus. It shows the matched SURF features with the Mercator projection.

comparison of this method with the proposed feature-based registration method is presented in the last experiment of this chapter. Figure 3.1 presents the robotic platform used for capturing the data in the last two experiments. As mentioned before, the Irma3D is a battery powered, three-wheeled mobile robot baed on the Volksbot RT 3 chassis. The platform includes the Riegl VZ-400 3D laser scanner, the Canon 1000D DSLR camera, Xsens MTi IMU and an additional Sick LMS 100 2D laser scanner.

3.7.1 Interfaith house (indoor environment)

Two scans were acquired from the inside of the Interfaith House located on the campus of Jacobs University Bremen, Germany. Several panoramic projections are tested to generate panoramic reflectance images of the 3D point clouds received from the scanner. Two different resolutions were used for the panorama image projections. The lower resolution is 3600×1000 and the higher resolution is 5040×1400. The symmetrical aspects of the environment increase the difficulty of the registration process. Correct registrations are only obtained with the Mercator projection using the SURF feature detector and descriptor for both resolutions and with the SIFT feature detector and descriptor with low resolution images. A correct registration was also generated with the Pannini projection with the SURF feature detector and descriptor for both resolutions. This shows that both the type of projection and the type of feature detector and descriptor have a significant effect on the registration process.

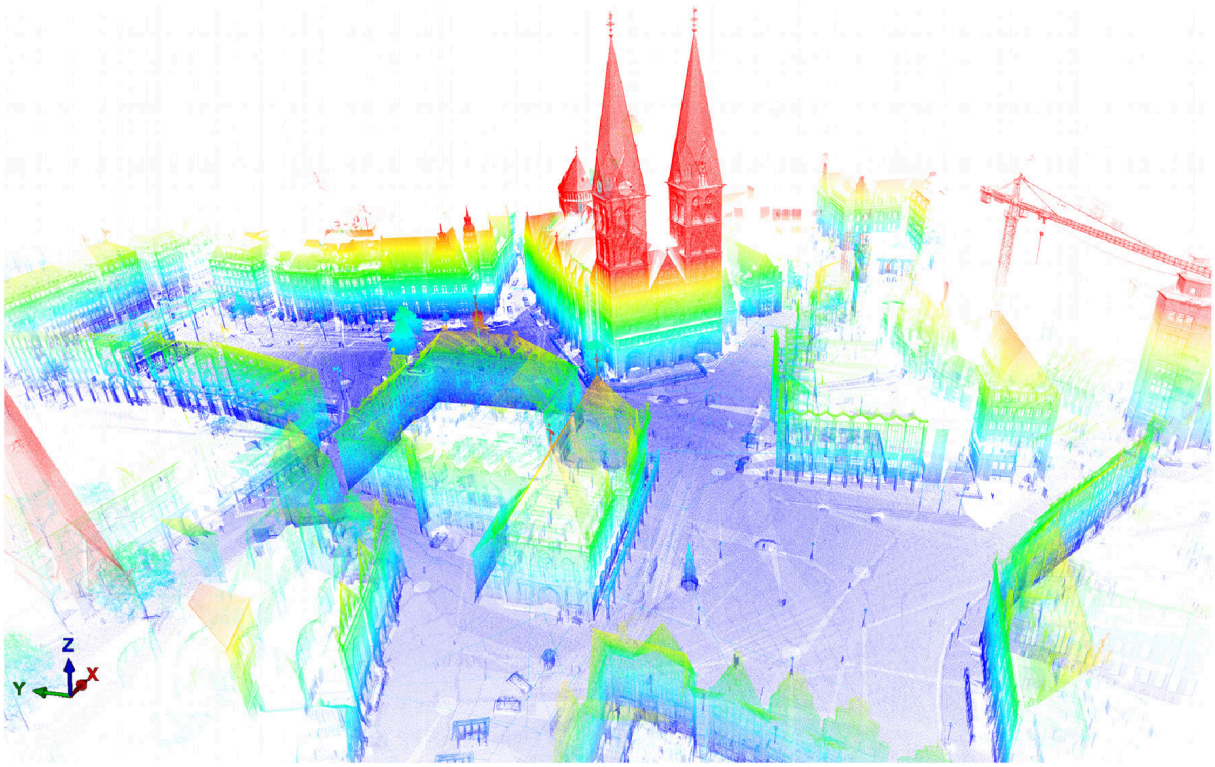


Figure 3.3: The City center of Bremen presented by registration of 13 scans captured with Riegl VZ-400 scanner.

3.7.2 Bremen city (outdoor environment)

This data set contains 13 scans from the city center of Bremen, Germany. The scan resolution was set to 0.04 degrees, which defines the amount of rotation between each measurement in both vertical and horizontal direction. The scans were acquired with several meters distance between each pair. Figure 3.3 presents a view of the scans after successful registration. Six combinations of feature detector and descriptor are used to generate registration for these scans. Table 3.1 displays the percentage of successfully registered subsequent pairs of scans for these configurations. It shows the results with seven possible projections with the image resolution of 5040×1400. Images are generated from the Bremen City data set. The FAST feature detector generates a high number of features on each panoramic image. Therefore the matching process, especially with SIFT and SURF descriptors, is extremely slow and ineffective. Due to the poor performance of some combinations of feature detector and descriptor, not all of the demonstrated combinations in Table 3.1 are practical. The combinations with SIFT as detector and descriptor and SURF as detector and descriptor are the most promising.

Table 3.1 presents 92% success rate for scan registration with SIFT feature detector and descriptor with the Pannini projection. Similar results are presented with SURF feature detector and descriptor with The Mercator projection. This result indicates that the entire Bremen City data set is successfully registered except for one scan pair, namely the loop closing pair.

Table 3.1: Comparison between the success rate of registration using six combination of detector and descriptor with seven panoramic projection methods. The image resolution used for the generation of the panorama images is 5040 × 1400. Please note, that 92% success indicates that the entire Bremen City data set is successfully registered except for one pair of scans, namely the loop closing pair.

Detector+Descriptor	**Projections**						
	EQ	**CY**	**ME**	**RE**	**PA**	**ST**	**ZA**
SIFT+SIFT	84%	69%	76%	30%	92%	76%	15%
SIFT+SURF	38%	38%	61%	30%	46%	46%	7%
SURF+SURF	76%	76%	92%	38%	76%	76%	23%
ORB+ORB	15%	15%	7%	0%	7%	0%	0%
STAR+ORB	15%	7%	7%	0%	7%	15%	0%
STAR+SURF	0%	7%	7%	0%	0%	0%	0%

The success of the registration is dependent on the feature detection process, descriptor method and the used panorama projection method. Additionally, it also relies strongly on the overlapping area between scan pairs. Detected features from the overlap area are the correct correspondences of the scan pair. These features are the inliers of the matching process. A certain amount of inliers is required for calculating the correct registration. Another crucial aspect for calculating of the correct registration is the number of detected features in the overlapping area and the uniqueness of detected features. In some circumstances the environment contains symmetric or repetitive objects. This reduces the uniqueness of the features and therefore increases the difficulty of the registration even with large overlapping areas. Due to the aforementioned facts, scan pairs can be divided into three categories. Scan pairs with a large overlap and a unique set of features, scan pairs with a reasonable overlap yet not enough features and scan pairs with tiny overlap and not enough unique features. The first category is the easiest and it is successfully registered with most of the configurations. The second category is difficult to register. The third category is the most challenging which generally generates incorrect results even with promising combination of feature detector, descriptor and panorama projection methods.

Tables 3.2 and 3.3 show a comparison between the number of matches, filtered matches and inlier matches. SIFT and SURF are utilised as the detector and descriptor for three scan pairs of the Bremen City data set. These pairs are representative examples for each of the three categories of scan pair registration difficulties. Scan pair 1-2 has a reasonable overlap yet not enough distinctive matchable features. Scan pair 5-6 is the most difficult out of the three selected scan pairs. The overlapping area is very small and calculation of the registration is nearly impossible. Scan pair 11-12 is the easy pair with a large overlapping area and many distinctive matchable features. In Tables 3.2 and 3.3 “matches” are the number of matched features based on the ratio nearest neighbour search,“filtered matches” are the number of matches after the RANSAC-like filtering process. The“inliers” are the number of matches that corresponds to the correct registration. The number of matches that the SURF detector generates is greater by a

Table 3.2: Comparison of three scan pair categories with different registration difficulties. The matched features are generated with SIFT feature detector and descriptor on several panorama projection methods. The 5040 × 1400 resolution is used for generating the reflectance panorama images.

	Scan Pair	**Projection**						
		EQ	**CY**	**ME**	**RE**	**PA**	**ST**	**ZA**
Matches	1 — 2	1413	1219	1388	1565	1566	1604	1802
	5 — 6	1677	1464	1649	2289	1948	1983	1148
	11 — 12	1600	1358	1624	1973	1973	2021	1838
Filtered Matches	1 — 2	41	47	53	22	39	33	30
	5 — 6	26	21	24	27	27	24	50
	11 — 12	91	117	97	77	101	58	33
Inliers	1 — 2	22	27	6	0	15	11	0
	5 — 6	0	0	0	0	7	0	0
	11 — 12	57	94	70	49	87	48	0

Table 3.3: Comparison of three scan pair categories with different registration difficulties. The matched features are generated with SURF feature detector and descriptor on several panorama projection methods. The 5040 × 1400 resolution is used for generating the reflectance panorama images.

	Scan Pair	**Projection**						
		EQ	**CY**	**ME**	**RE**	**PA**	**ST**	**ZA**
Matches	1 — 2	3730	3197	3430	4577	4276	4686	8218
	5 — 6	4426	4490	4332	6764	5160	5097	5026
	11 — 12	3856	3125	3600	5108	4924	5277	8666
Filtered Matches	1 — 2	82	78	94	45	90	69	63
	5 — 6	40	43	36	59	44	43	51
	11 — 12	162	130	139	135	153	127	68
Inliers	1 — 2	24	39	39	6	13	39	0
	5 — 6	0	0	6	0	14	0	0
	11 — 12	119	87	98	89	112	85	0

factor of roughly 3 in comparison to the SIFT detector. This is due to the fact that the SURF feature detector tends to detect more features from the same panorama image. Figures 3.4 and 3.5 present the features detected with the SIFT and the SURF detector from the panorama image generated from a scan in the Bremen City data set. The equirectangular projection method is used to generate these panorama images with a resolution of 3600 × 1000.

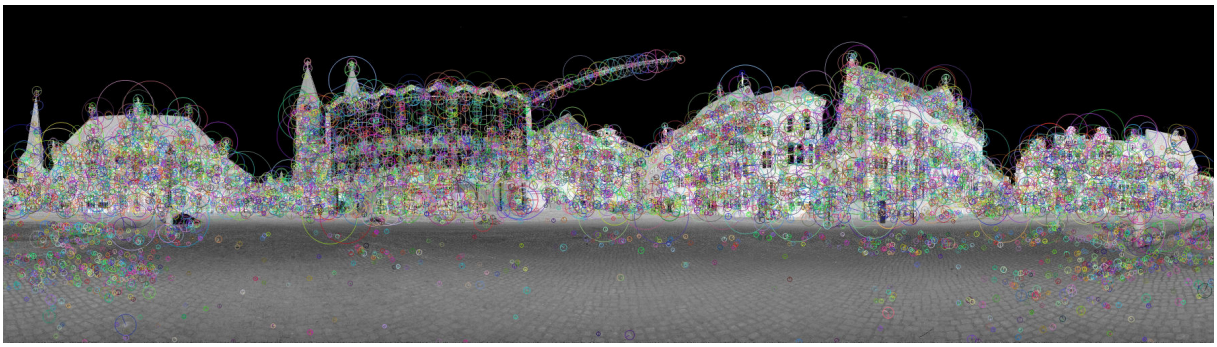


Figure 3.4: The SIFT features detected on a reflectance panorama image generated from scan 11 of the Bremen City data set. The equirectangular projection with 3600×1000 resolution is used for generating the panorama image.

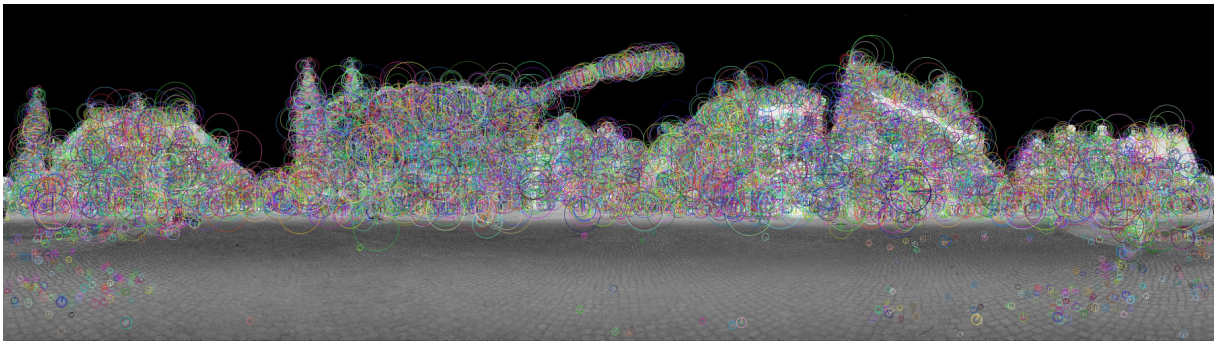


Figure 3.5: The SURF features detected on a reflectance panorama image generated from scan 11 of the Bremen City data set. The equirectangular projection with 3600×1000 resolution is used for generating the panorama image.

The quantity of the filtered matches and inliers are therefore higher for SURF. Figures 3.6, 3.7 and 3.8 present reflectance images of these three data sets with a line connecting the matched features. The presented matched features are the SURF features detected on the panorama images. The Mercator projection is used to project the scans onto panorama images with the resolution of 5040×1400. As evidenced by both Figure 3.7 and the corresponding rows of Tables 3.2 and 3.3 the overlap area in the scan pair 5-6 is small, containing a slightly tilted wall and lots of windows. Therefore, registration of this type of scan pairs due to the lack of sufficient unique features is a demanding process. This scan pair is the loop closing pair mentioned earlier in the Table 3.1. The small overlap between the scan pair makes it impossible to register with any of the proposed combinations of feature detector and descriptor and projection methods. In view of the presented facts in Tables 3.2 and 3.3 the registration of scan pair 5-6 is feasible only with the Pannini projection with both SIFT and SURF feature detector and descriptors. Additionally, the SURF feature detector and descriptor is capable of calculating the registration with the Mercator projection. In this example the SURF feature detector and descriptor outperforms the SIFT feature detector and descriptor.

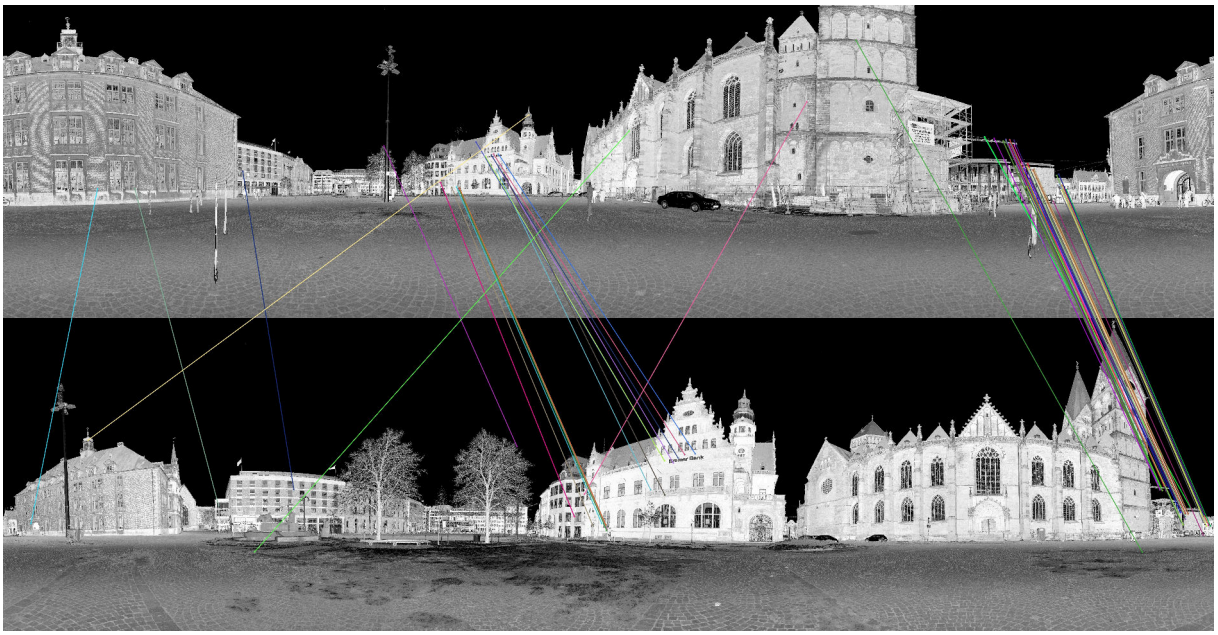


Figure 3.6: Reflectance panorama images presenting the scan pair 1-2. The Mercator projection is used to generate these images with the resolution of 5040×1400. The SURF feature detector is used to detect the correspondences. The filtered SURF feature matches are indicated by the lines.

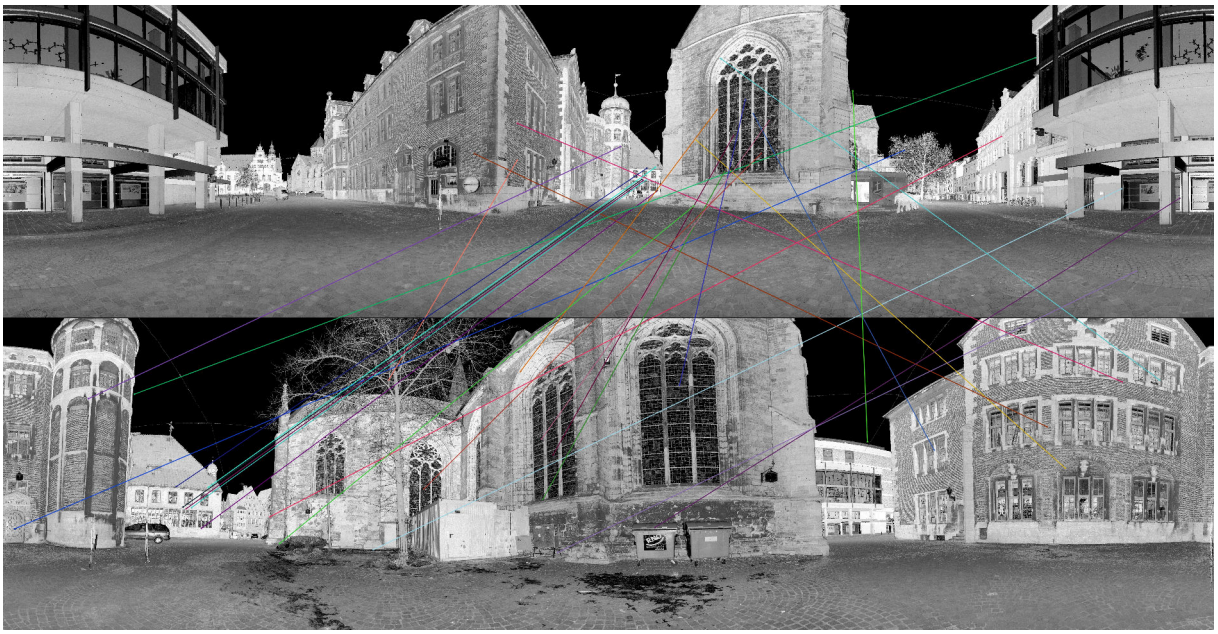


Figure 3.7: Reflectance panorama images presenting the scan pair 5-6. The Mercator projection is used to generate these images with the resolution of 5040×1400. The SURF feature detector is used to detect the correspondences. The filtered SURF feature matches are indicated by the lines.

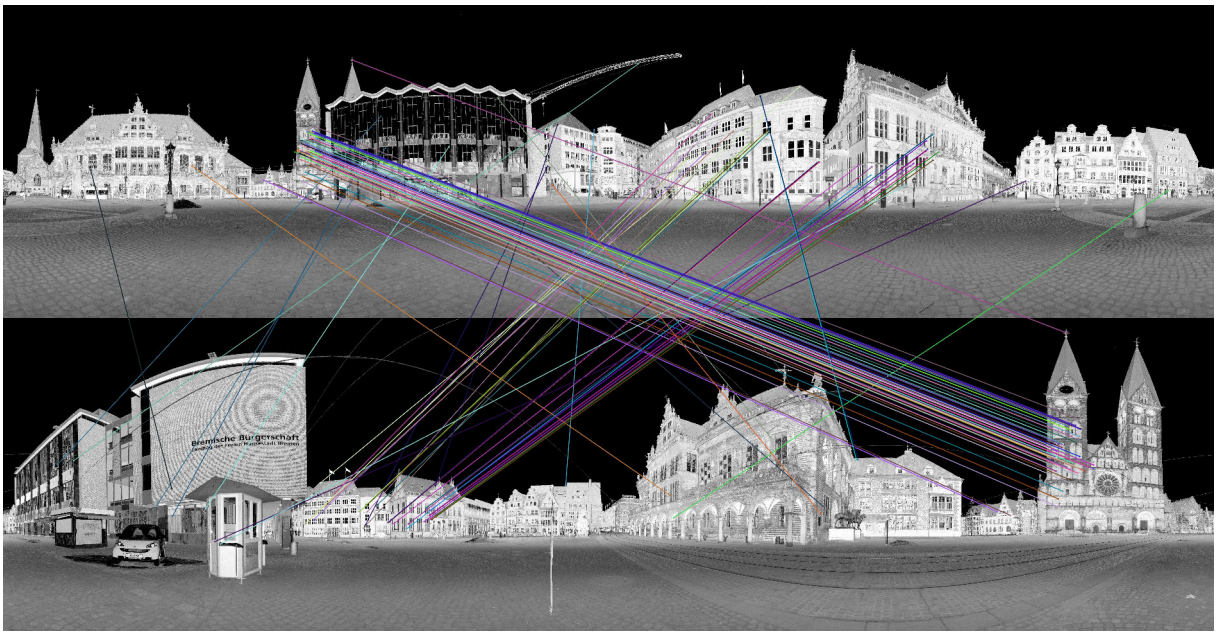


Figure 3.8: Reflectance panorama images presenting the scan pair 11-12. The Mercator projection is used to generate these images with the resolution of 5040×1400. The SURF feature detector is used to detect the correspondences. The filtered SURF feature matches are indicated by the lines.

The most time consuming part of the entire procedure is the registration step. There is a correlation between the amount of filtered matches and the required time for registration. Due to the fact that registration time is higher than the rest of the process by one order of magnitude, the required time depends strongly on the number of filtered matches. However, the number of filtered matches is in turn dependent on the amount of detected features; both of them rely on the projection and the feature detector methods. Furthermore, the resolution of the generated panoramic reflectance images from the data set is essential for the feature detection process. It is clear that by increasing the resolution the amount of detected features increases. A larger number of features yields a significantly higher number of filtered matches, and thus results in a more time consuming registration. Similarly lower resolution yields lower number of features and therefore faster registration. Since the amount of filtered matches is crucial to the registration process and a minimum number of filtered matches is essential for successful registration, two resolutions have been used in the experiments to determine a balance between success rate and speed of registration. Table 3.4 demonstrates the percentage of correct registrations of the Bremen City data set with both resolutions. The success rate by using a larger resolution is higher.

3.7.3 Campus of Jacobs University (outdoor environment)

This data set contains 122 terrestrial 3D scans captured from the campus of the Jacobs University Bremen. All the scans were acquired with the Riegl VZ-400 scanner. To acquire this data set the resolution of the scan was fixed to 0.04 degrees. This sets the increment of 0.04 degrees in both horizontal and vertical rotational directions between each measurement. These scans have been acquired with several meters distance between each scan acquisition. Figure 3.9 depicts an

Table 3.4: Comparison of the success rate of registration of the Bremen City data set with two different resolutions for panorama images. SIFT and SURF are used as the most promising combinations of detector and descriptor with seven projections.

Detector+Descriptor	Projection	3600x1000	5040x1400
SIFT+SIFT	Equirectangular	53%	84%
	Cylindrical	69%	69%
	Mercator	76%	76%
	Rectilinear	23%	30%
	Pannini	69%	92%
	Stereographic	76%	76%
	Z-axis	15%	15%
SURF+SURF	Equirectangular	53%	76%
	Cylindrical	30%	76%
	Mercator	46%	92%
	Rectilinear	23%	38%
	Pannini	61%	76%
	Stereographic	61%	76%
	Z-axis	15%	23%

orthoimage of the data set. The image contains part of the scans after registration. Figure 3.10 presents a typical scan pair of the data set. The images are generated by mapping the point cloud onto a panorama image. The Pannini projection method was used to generate the panorama images. The lines in the image present the detected feature matches between the scans.

With the results of the Bremen City data set in mind, two combinations of feature detectors and descriptors have been analysed for the Jacobs University Campus data set: the SIFT feature detector and descriptor and the SURF feature detector and descriptor in combination with seven of the presented projections in Chapter 2 and the two proposed resolutions. The scan pairs for the registration are selected sequentially based on the order of the captured scans. Table 3.5 presents the percentage of accurate registrations of the Campus data set for these configurations.

Similar to the results from the Bremen City data set a higher success rate is obtained for the higher resolution. The best projections are Pannini and Mercator which result in a higher success rate with SIFT and SURF respectively. Additionally, the result of equirectangular projection is promising. This is mostly due to the fact that the equirectangular projection keeps the original distribution of the points. Overall, the obtained success rate is lower on the Jacobs University Campus data set. This is due to the fact that the buildings on the campus are quite symmetrical. This puts the data set in the second category of matching. The scan pairs have a reasonable overlapping area but not enough distinctive features. Therefore the registration process is more challenging. In addition, the environment contains a high amount of vegetation. Trees produce a great quantity of features which results in a significant amount of wrong matches.

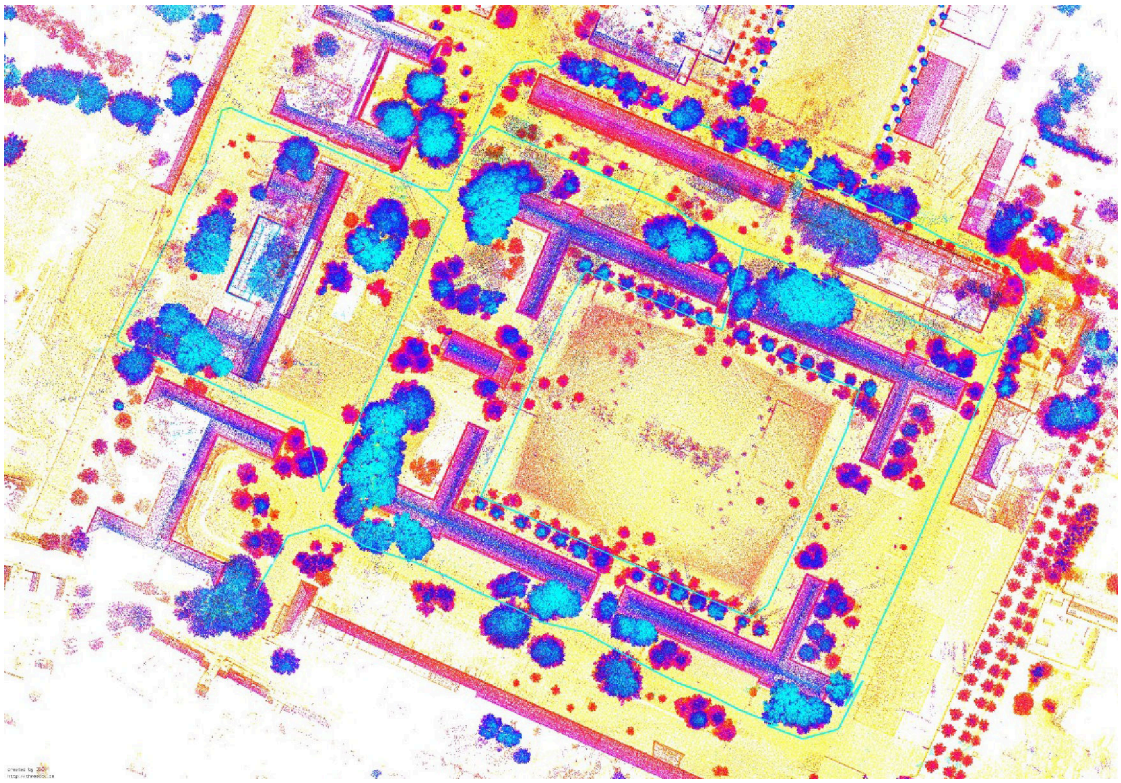


Figure 3.9: Part of the campus of the Jacobs University. The data set contains 122 terrestrial 3D scans captured by the Riegl VZ-400 scanner.

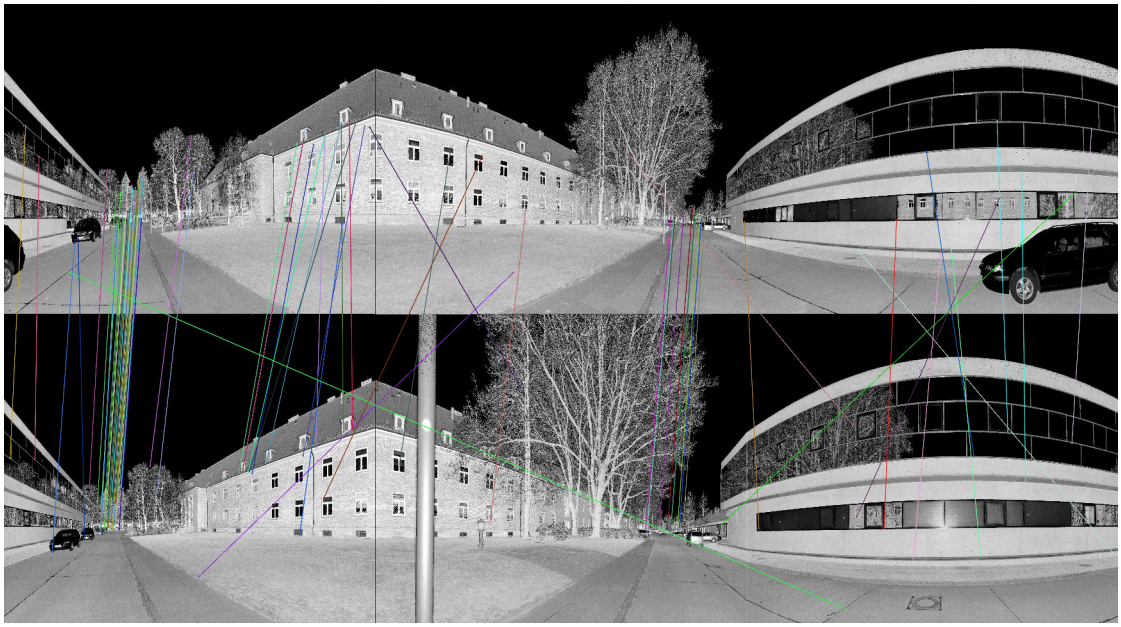


Figure 3.10: A typical scan pair from the Jacobs University Campus data set. The panorama images are generated with the Pannini projection. The Matched SURF features are depicted by lines.

Table 3.5: Comparison of the success rate of registration of the Jacobs University Campus data set. Two panorama image resolutions are used with SIFT and SURF as the most promising combinations of detector and descriptor methods. The results are shown for seven panorama projection methods.

Detector+Descriptor	Projection	3600x1000	5040x1400
SIFT+SIFT	Equirectangular	51%	55%
	Cylindrical	46%	53%
	Mercator	51%	55%
	Rectilinear	38%	29%
	Pannini	55%	59%
	Stereographic	55%	59%
	Z-axis	34%	27%
SURF+SURF	Equirectangular	45%	57%
	Cylindrical	38%	39%
	Mercator	44%	61%
	Rectilinear	28%	29%
	Pannini	41%	55%
	Stereographic	37%	52%
	Z-axis	34%	32%

3.7.4 The Residence Palace in Würzburg Germany (indoor environment)

The Residence Palace in Würzburg is a baroque palace in the city center of Würzburg, Germany. The palace has been a UNESCO World Cultural Heritage site since 1981. It is one of the most renowned baroque castles in Europe. The experiments were carried out in two halls in the palace namely the White hall and the Imperial hall, two large halls with impressive 3D structure. Together with the colorful paintings in the Imperial Hall the environment was captured with combination of two technologies, laser scanning and photography for generating colored point clouds. To capture the entire environment the scans were taken in a stop-scan-go fashion. The Riegl VZ-400 scanner is able to capture 125.000 points per second with a field of view of $360° \times 100°$. Thus, a laser scan with a resolution of $0.04°$ to $0.03°$ takes between 3 and 5 minutes. The scanner rotates around the vertical axis to achieve $360°$ data collection. After the scan is taken, the scanner is rotated in discrete steps to take 12 pictures with a resolution of 3888×2592 at each scanning location. The point cloud is colored based on the known transformation between the scanner and the camera. The calibration procedure used for determination of this transformation is presented by Borrmann et al.[35].

To capture the entire experimental environment, multiple scans were taken in both the Imperial Hall and the White Hall. To evaluate the feature-based registration process the iSpace system was used to measure the robot position with high precision. In the Imperial Hall the localisation system was set up using six transmitters to define a coordinate system (cf. Fig-

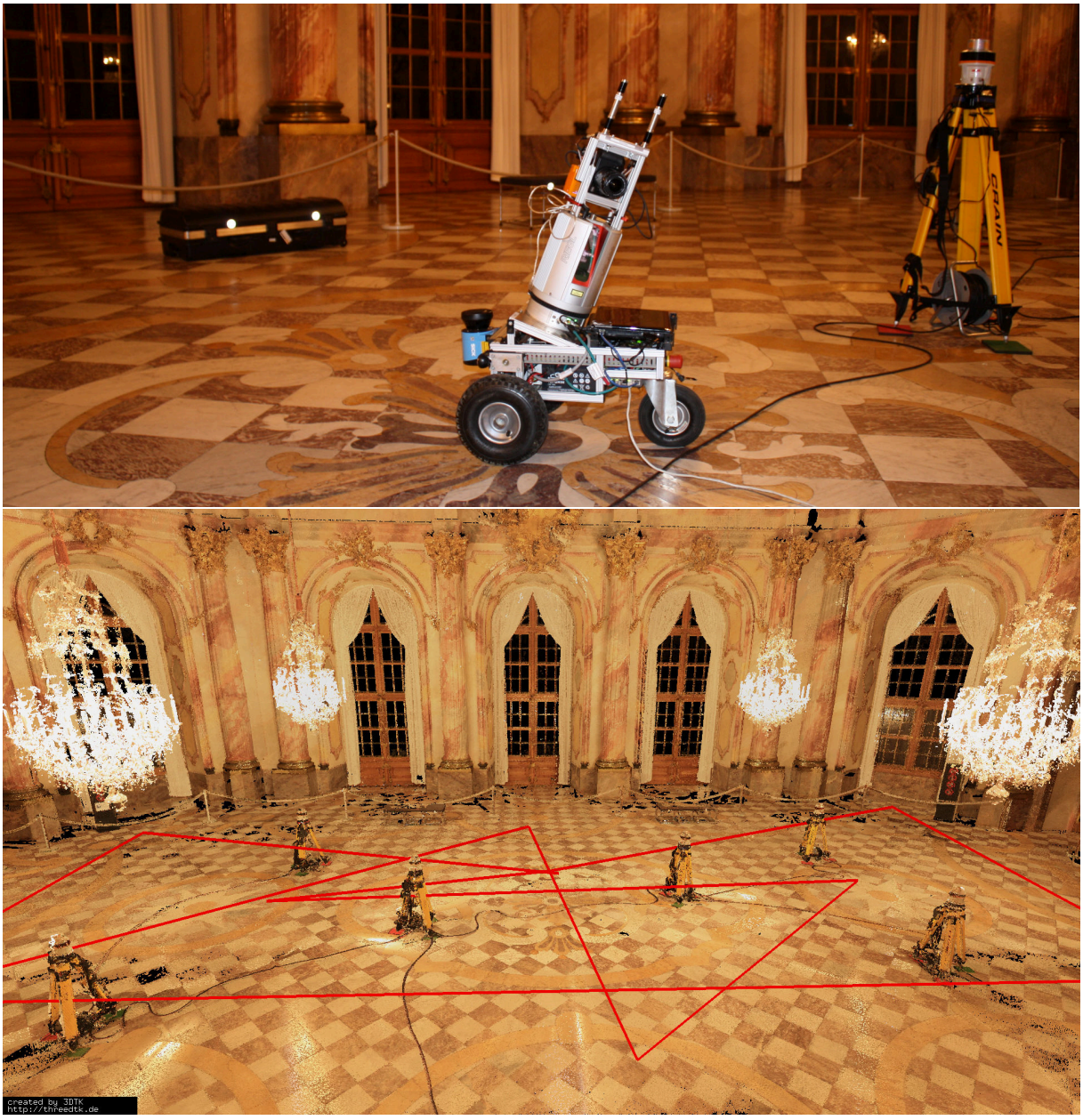
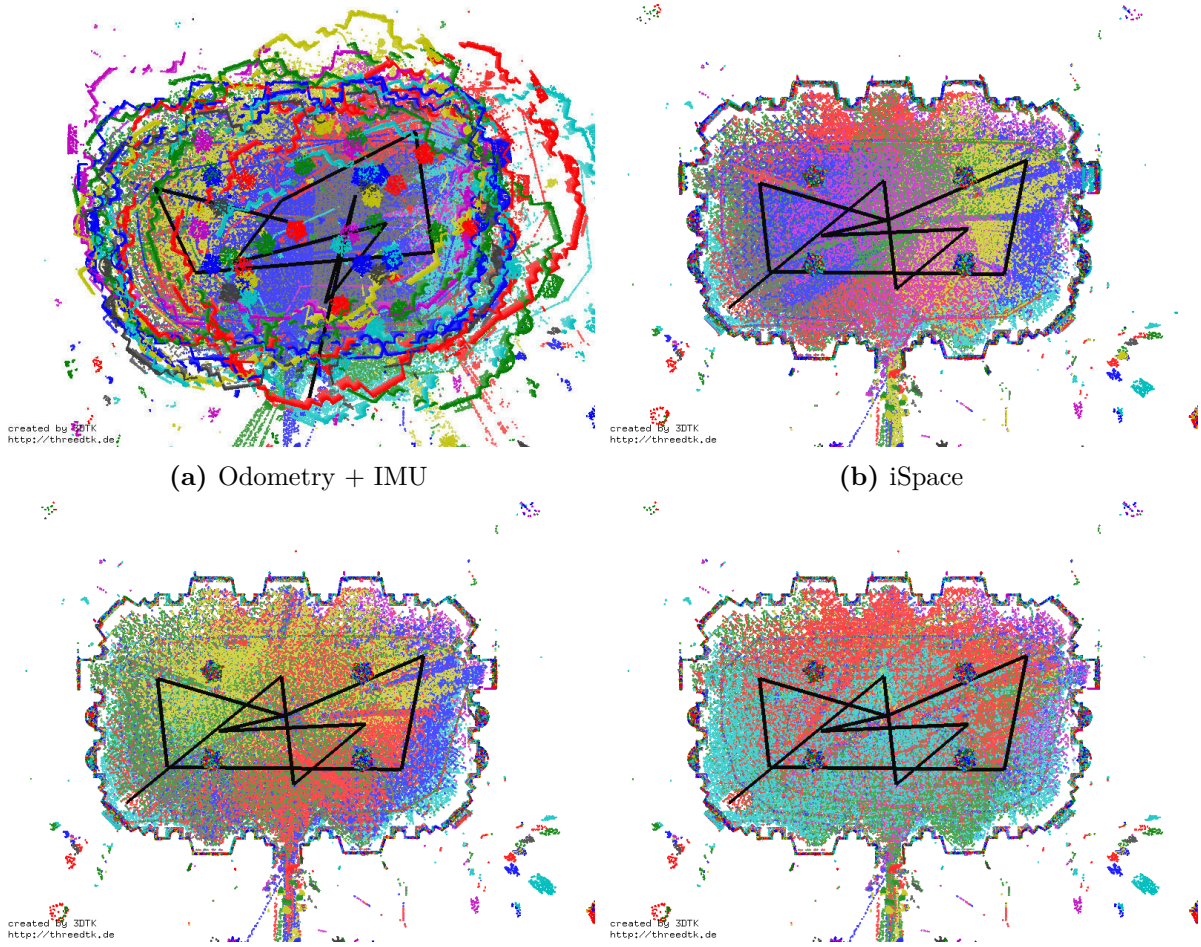


Figure 3.11: Top: Irma3D in the Imperial Hall of the Würzburg Residence. In the background is one of the six transmitters of the iSpace localisation system. Bottom: Final 3D model of the Imperial Hall presenting the six iSpace transmitters and the measured scanning locations.

ure 3.11). The system measures the robot position using the sensor frame attached to the top of the laser scanner. The robot was manually driven to 11 scanning locations. The first scan position is anchored in the iSpace coordinate system for the registration process. The position of the robot was collected at each scanning location using the iSpace localisation system.



(a) Odometry + IMU

(b) iSpace

(c) FBR

(d) ICP

Figure 3.12: Floor plan of the Imperial Hall at the Würzburg Residence. The black lines connect the scan locations. Top Left: Poses from Odometry + IMU. Top Right: Poses from the iSpace system. Bottom Left: Poses from the feature-based registration. Bottom Right: Poses from the registration with the global ICP using feature-based registration as initial pose estimates.

Figure 3.12 represents a top view of the resulting model. Additionally it represents a comparison between the localisation methods. The scanning positions are connected with black lines. The point clouds are color coded based on the scanning location. The first observation from the presented data yields that the position estimation from odometry contain very large errors. Second, the feature-based registration method presented in this chapter leads to very good results. The direct comparison of the feature-based registration map and the iSpace map shows thinner walls in the feature-based registration map, suggesting that the positions are more accurate. The ICP based registration further improves the map quality. This is visible based on the color distribution in the maps. Close to the individual scanning location the density of the points of that scan becomes higher. Thus, the color representing the scanning location is dominant, while in the other maps the distribution is more random.

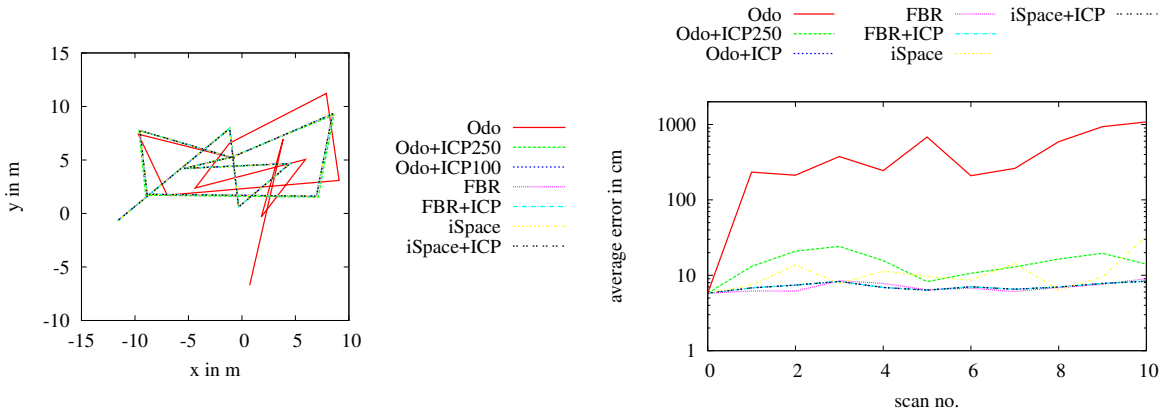


Figure 3.13: Evaluation of pose estimation methods. Odo = Odometry and IMU, ICP = Global optimisation with a maximum distance of 100cm between points, ICP250 = Global optimisation with a maximum distance of 250cm between points, FBR = feature-based registration using the Mercator projection with SIFT feature detector and descriptor for panorama image matching. Left: Represents the measured poses utilising different localisation methods. Right: Represents the average error of the measured control points for the generated map using the different mapping methods. The average error is presented in cm.

For a quantitative evaluation of the pose estimates 7 distinct points in the environment were measured using the iSpace handvector bar. These points were identified in each scan and transformed into the iSpace coordinate system using the pose estimates calculated with different methods. The average of the distance error between the transformed points and the globally measured points are calculated for each scan. Points that cannot clearly be identified in the scan are ignored in the averaging. The result is presented in Figure 3.13. The error of the odometry method is very large as expected. However, a registration is possible due to the fact that the scans were taken in a single large hall. This requires a two phase registration with ICP. First with a maximum distance of 250cm between point pairs and then with a maximum distance of 100cm to generate the correct model of the environment. The 2D scanner on the robot platform is also used to perform a 2D SLAM method for robot localisation.

The error for the model generated with the iSpace localisation system is surprisingly high. The interferences from the large windows and the glass chandeliers are the possible reasons for the inaccuracy in the measurement. The rotational accuracy promised of the system is not as high as the positional accuracy. Therefore, small rotational errors have a large impact on the final model given the large distance to the walls. To minimise the number of scanning positions and increase the captured environment for each scan the robot was moved to the corners of the hall, where the transmitter coverage was not optimal. A further disadvantage of the method is the transmitters that act as obstacles during the data acquisition. In combination with the time required for setup this makes the system infeasible for the purpose of scan matching.

Feature-based registration yields the best initial pose estimates. The results differ only very slightly from the final model. The remaining error is caused by three factors. First, not all points in the environment could be measured with high precision. Second, some of the reference points were difficult to capture in the point clouds due to the reflective properties of the material. In

Figure 3.14: The final 3D model of the White Hall in the Würzburg Residence.

combination with low resolution areas far away from the laser scanner the manually selected points lack precision. Third, the registration method used the first scan as reference. Therefore, the errors in the localisation of this scan are propagated to the remaining scans.

Similar to the data set captured from the Imperial Hall the robot was manually driven to 9 locations to collect data in the White hall. The scan locations are chosen to be in the corners of the hall to cover the most of the environment. Similar results have been achieved with the experiments with this data set. The feature-based registration pose estimations are very similar to the final model and odometry results in by far the largest amount of error. Figure 3.14 represents the captured data after applying the global optimisation on top of the pose estimates from the localisation methods.

3.8 Summary

This chapter gives an overview on point cloud registration methods, Including the well-known point-based ICP algorithm and feature-based registration methods. It proposes a feature-based registration method based on natural detected features from point clouds. It presents a survey and an evaluation of modern feature detector and descriptor methods for 3D scan registration. Several 2D feature detector and descriptors are presented. The proposed feature-based registration method utilises these 2D features on reflectance panorama images generated from 3D point clouds (cf. Chapter 2). The proposed method calculates a set of descriptors for each panorama image. This provides a set of points for correspondence matching between scan pairs. Matched features are used by the feature-based registration to generate the transformation matrix between the scan pair. Additionally, feature filtration methods are proposed to improve the efficiency of the feature matching process. Furthermore, several feature matching methods are implemented to select the correct correspondences between scans. A coarse RANSAC-like registration algorithm is developed to determine the transformation matrix. Finally, globally consistent scan matching is introduced to minimise the accumulated errors from the pairwise scan matching methods. The performance of different projection methods is analysed. The effects of utilising different projections with both real-time features and more robust and invariance features are studied. In the view of the results from the experiments the SURF and the SIFT feature detector and descriptors produce higher success rates for registration process due to their more robust and rotation invariant features than real-time methods. A realistic, real world evaluation shows that the projection of a 3D scan to a 360° panoramic image for proposed automatic feature-based registration is crucial. It is shown that the Mercator and the Pannini projections are the most suitable methods for the developed automatic registration method. The Mercator projection is an isogonic projection, i.e., angles are preserved; the Pannini projection is also designed to have as few distortion as possible.

Chapter 4

Point Cloud Reduction

Point cloud registration reconstructs environments from point clouds captured with 3D measurement systems. This is an important but time consuming part of any mapping system. As mentioned in Chapter 2 3D laser scanners use either two rotating mirrors to deflect the emitted laser beam or rotate a 2D laser scanner to measure the distance r to an object under a certain angles θ and φ. The resulting 3D point clouds of the scanned environment is in spherical coordinates (θ, φ, r). Data acquisition with 3D laser scanners is virtually unlimited with respect to resolution. This leads to point clouds with a large number of points (cf. Figure 4.1). Data reduction becomes necessary to efficiently perform operations, especially registration, on these point clouds. Reduced point clouds will notably increase the speed of the registration. There are approaches for reducing the complexity of scan registration by extracting planes from the data [40, 182]. However, by using geometrical features the precise point measurements are lost. Little work has been done on subsampling 3D point clouds, e.g., by utilising range images [151]. Other methods of point cloud reduction separate the space into voxels and sample points from each voxel. Efficient methods with uniform voxels use octrees [153]. Similarly Lee subdivides the point cloud into a non-uniform grid based on normals [141]. Suter et al. cluster the data with a k-nearest neighbour approach to remove points from the original data [223]. Methods that use multiple scans to reduce redundancies such as the method proposed by Sqadzba et al. [224] cannot be used for registration as they require knowledge about point correspondences beforehand. This chapter presents point cloud reduction methods based on panorama images. They exploit the panorama generation methods presented in Chapter 2 to reduce point clouds to speed up the registration process. The methods are contrasted with a state of the art reduction method based on octrees.

4.1 Octree Reduction

Given a large number of points from a 3D point cloud this section presents a method to uniformly subsample the entire point cloud to reduce the number of points. This is achieved by first binning the point cloud in a regular 3D grid and then by randomly selecting a number of points in each voxel. Both the number of points and the voxel size can vary to allow for many different point densities. An additional advantage of the uniformity of the subsampling is that surfaces closer

Figure 4.1: Illustration of a 3D scan acquired by a FARO Focus3D laser scanner. The scanner gages up to 976000 points/sec on a sphere around it by a rotating mirror and a second rotation around the vertical axis. The scanner has a built-in camera and is capable of acquiring color images. Hence, the colored point cloud.

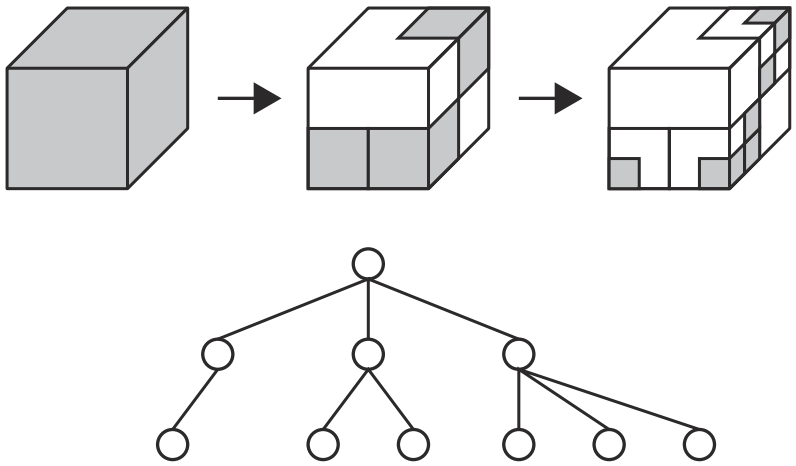


Figure 4.2: Top: Spatial subdivisions of an octree up to level 3. Occupied leaf nodes are shaded as grey. Bottom: The corresponding tree structure of the sparse data structure.

to the scanner do not unfairly influence the quality measure more than surfaces that are farther away. This influences the quality and success rate of the point-based registration methods. A reduction of the number of points in the 3D point cloud increases the speed of the registration process. Elseberg et al. [73] employ a spatial data structure called octree with a low memory consumption. As presented in Chapter 2 an octree is a data structure for indexing 3D data. It represents the data in a volumetric tree structure. Similarly Hornung et al. [108] present an open source framework for generating volumetric 3D environment models based on an octree structure. Rusu and Cousins [198, 199] also present another implementation of the octree data structure that is used for storing and processing 3D point clouds. Since laser scanners sample the surface of objects, the 3D point cloud is not fully volumetric. Therefore, most octree nodes will only have few children. The octree data structure is ideally suited for 3D point clouds. Figure 4.2 present the octree data structure with a depth of three.

To reduce the amount of data three methods can be used. The first method selects the center of the occupied voxel as the representative point of the volume of the voxel. This method creates uniformly distributed points in the reduced point cloud. However, it loses the original points measured by the scanner. The second method utilises the centroid of each voxel as the representative of the volumetric area. The reduced point cloud has a more realistic representation of the measured points. The third and the more accurate method is to reduce the data by selecting a fixed number of points from each voxel. This will remove more points in voxels closer to the scanner than in the voxels further from the scanner. After reduction the points will be uniformly distributed across the scanned environment. This is ideal for point-based registration. Point-based scan reduction is not suitable for panorama based registration. When creating a panorama image equally sized areas will be mapped onto differently sized areas in the panorama image depending on the projection type. The result of point-based reduction is an even distribution in 3D space. This will lead to an uneven distribution of points in the 2D projection, with too few points in areas near to the scanner. The result is a panorama image that contains many holes.

4.2 Panorama Reduction

For panorama based registration we require a reduction method that reduces points based on the 2D projection. Panorama generation implicitly reduces points by discretising the projection onto a 2D grid. The 2D grid is necessarily of lower resolution than the original point cloud. This is due to the distortions inherent to the projection onto a 2D plane. Some areas will be "compressed" and others "stretched". In the latter case, the original resolution of the 3D scan is often insufficient for complete coverage of the 2D grid. Consequently, for an image without holes the overall resolution of the 2D image has to be reduced. To generate a rectilinear panorama image based on the method presented in Chapter 2, for a medium sized scan with 22.5 million points, the resolution of the image should not exceed 3600×1000. This equates to 3.6 million points, i.e., a reduction by a factor of six. This is a sufficient resolution for feature-based registration. Further reduction leads to an even larger decrease of memory usage. Figure 4.3 shows an example of scan reduction. It is obvious that the point cloud reduced using the octree is more evenly distributed, while the panorama based reductions are a closer representation of the original data. On close examination the characteristics of the different projection methods can be seen in the reduced point cloud. Three methods are proposed for panorama based scan reduction using the correspondences between panorama image pixels and 3D points in the point cloud. The generation of the correspondences is explained in detail in Section 2.4.

For the first method all 3D points are projected onto the pixels of the panorama image. For each pixel all points except the one with the furthest distance from the scanner is discarded thus reducing the data to at most one point per pixel. This method is referred to as the *FURTHEST* panorama map method. Furthest point selection will remove some of the obstacles such as vegetation and tree leaves from both panorama images and point clouds. This subsampling method uses direct mapping between pixels and original point data. Therefore the reduced point cloud contains a subset of the original points from the input point cloud.

In the second method inverse panorama projection equations are used to recover the point data from the panorama images. The original point information is lost and replaced by an approximation based on the center of the pixel and the range information of the furthest point. In this method the panorama map is not used and the generated range image is the source of the reduced point cloud. Several panorama projection methods are presented in Section 2.3. To recover the reduced point cloud the inverse equations of these projections are used. Some projection methods such as the equirectangular and cylindrical projections have simple equations. Their reverse equations are presented in Equations 4.1 and 4.2, respectively.

$$\begin{aligned} \theta &= x, \\ \varphi &= y, \end{aligned} \tag{4.1}$$

and

$$\begin{aligned} \theta &= x, \\ \varphi &= \arctan y, \end{aligned} \tag{4.2}$$

where the x and y are panorama image coordinates and the θ and φ are the longitude and latitude of the 3D point. The range of the point is the value saved in the image pixel. However,

most projection methods presented in Chapter 2 have more complex reverse equations such as the reverse projection equations for the Mercator and the Albers Equal-Area Conic as presented in Equations 4.3 and 4.4, respectively.

$$\begin{aligned} \theta &= x, \\ \varphi &= 2\arctan(e^y) - \frac{\pi}{2}, \end{aligned} \tag{4.3}$$

and

$$\begin{aligned} \theta &= \theta_0 + \left(\frac{\arctan(\frac{x}{\rho_0 - y})}{n} \right), \\ \varphi &= \arcsin\left(\frac{C - (x^2 + (\rho_0 - y)^2)n^2}{2n} \right), \\ \rho_0 &= \frac{\sqrt{C - 2n\sin\varphi_0}}{n}, \\ n &= \frac{\sin\varphi_1 + \varphi_2}{2}, \\ C &= \cos^2\varphi_1 + 2n\sin\varphi_1, \end{aligned} \tag{4.4}$$

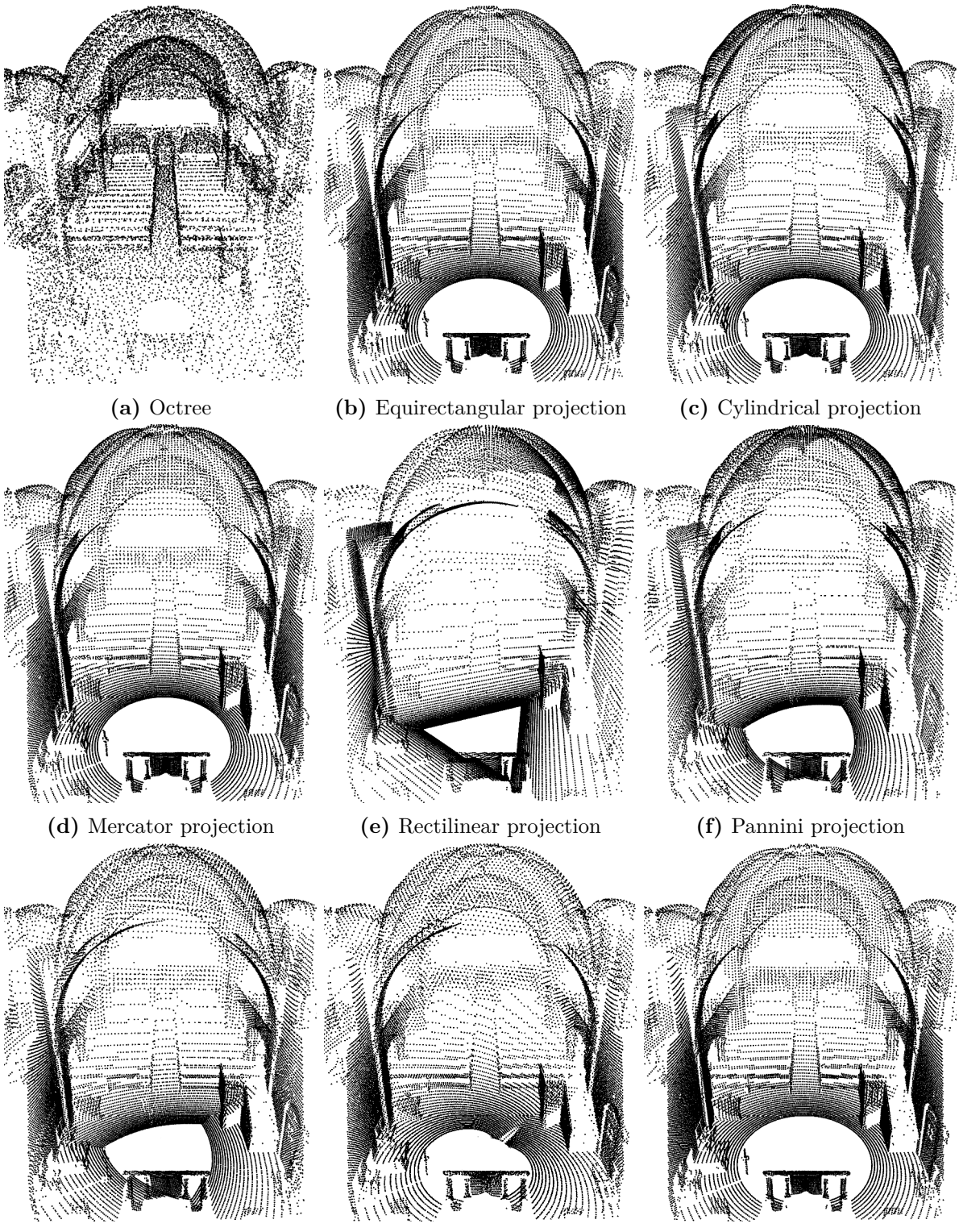
where the x and y are panorama image coordinates and the θ and φ are the longitude and latitude of the 3D point. The range of the point is the value saved in the image pixel. The θ_0 and φ_0 are the longitude and latitude of the origin of the Cartesian coordinates. The φ_1 and φ_2 are the standard lines of the Albers Equal-Area Conic projection. The reflectance and color information of the original point data are retained in the separate panorama image upon availability.

The third method for point cloud reduction proposed in this chapter is used on top of the first two methods. After creating the panorama image conventional image resizing algorithms are applied. Due to the resizing of the image reflectance and color values have to be reassigned to the points as well. Different approaches utilised in this chapter are explained in more detail in Section 4.3.

4.3 Case Study

This chapter presents the experimental results of the three aforementioned panorama based scan reduction methods. The data used for the experiments was acquired in a church in Bremen-Grohn, Germamy with a FARO Focus3D laser scanner. A total number of 12 scans were captured with color images to cover the complete indoor environment of the church. Each scan contains around 28 million points. Figure 4.4 presents a sample colored panorama image generated from the captured data. The reduction methods are compared with regards to their performance in panorama based scan registration. The required time for registration is recorded as well as the success of the registration process. As there was no ground truth available the success of the registration method was decided upon visual inspection of the resulting transformation.

Based on experiments in Chapter 3 the SIFT feature detector and descriptor outperform other feature detection and description methods in panorama based registration. Accordingly



(a) Octree **(b)** Equirectangular projection **(c)** Cylindrical projection

(d) Mercator projection **(e)** Rectilinear projection **(f)** Pannini projection

(g) Stereographic projection **(h)** Conic projection **(i)** Equal-area cylindrical projection

Figure 4.3: Point cloud reductions

Figure 4.4: Illustration of a colored panorama image generated with the Mercator projection from a sample 3D scan captured from a church in Bremen-Grohn, Germany. The data was captured with a FARO Focus3D laser scanner. The scanner has a built-in camera that captures the color of the point cloud.

Table 4.1: Performance of the first two panorama based reduction methods, namely subsampling and point recovery methods. The projection method used for the experiment is the Mercator projection. Feature detector and descriptor is SIFT. Time is given in seconds. The success of the registration method is decided upon visual inspection of the resulting transformation due to the lack of ground truth.

Resolution	**No. Points in 1,000**	**Subsampling**			**Point Recovery**		
		Matches	**Time**	**Reg.**	**Matches**	**Time**	**Reg.**
8640×2880	20,305	112	707.39	yes	119	623.64	yes
7200×2400	15,882	139	679.18	yes	137	608.63	yes
3600×1200	4,309	27	31.78	yes	23	25.04	no
1800×600	1,079	7	0.98	no	4	0.03	no
900×300	269	2	0.08	no	2	0.07	no

SIFT is used in all experiments in this chapter. The Mercator projection was shown to be preferable for panorama based registration (cf. Chapter 3). The initial experiments show the applicability of panorama based point cloud reduction using the Mercator projection. In the last part of the experiments an evaluation of all the presented projections is performed. Features are detected on each scan pair. Matched features are detected by the ratio nearest search algorithm as explained in Chapter 3. Afterwards the transformation matrix is computed by using RANSAC and a least squares method. Figure 4.3 shows an example of each panorama reduction method used in this experiment as well as the octree based reduction method presented in Section 4.1.

First a comparison of the first two panorama reduction methods is presented, namely subsampling and the point recovery methods. The Mercator panorama projection is used for generating the panorama images for this experiment. The results for different resolutions of the Mercator panorama images are given in Table 4.1. The table presents the number of detected matches in one scan pair for different resolutions. Additionally it presents the required time in seconds for calculation of the relative transformation matrix for the scan pair and whether the registration

process was successful for both reduction methods with several resolutions.

The subsampling method provides a successful registration for resolutions down to 3600×1200. Both methods fail on images with a resolution lower than 1800×600. In these failure cases the runtime is very low. The subsampling method, i.e., selecting the furthest point per pixel from the original data, performs more reliably but also slower. The experiments show that the modification of the point coordinates produced by using the inverse projection equations has a negative influence on the registration quality, especially with often decreasing number of features.

The results for the third panorama based reduction method on top of the other two aforementioned methods are shown in Table 4.2 and Table 4.3. Panorama images with a resolution of 7200×2400 are scaled by the factor indicated in the first column of the tables. This is done using the scaling algorithms that are implemented in OpenCV [38]. The bilinear interpolation and the nearest neighbour interpolation methods are used. The former is a true interpolation in the sense that new pixel values are computed, whereas the latter chooses one of the pixels in a small neighbourhood in the original image as a representative. The latter may cause problems when the scaling factor does not divide the original image in whole numbers. In such a case the subsampling may choose a point that, once reprojected in to the reduced panorama, falls outside of the pixel it represents. As only a single point per pixel is permissible, this results in a smaller point cloud which eventually produces empty pixels in the panorama generated from the reduced point cloud. The bilinear interpolation is slower but more reliable than the nearest neighbour approach. Both the bilinear interpolation and the reverse equation of panorama projection will produce new sets of points. Therefore, the subsampling method without scaling is the preferable method for panorama based scan reduction.

A comparison of different projection methods used for point cloud reduction is presented in Table 4.4. Two scan pairs are selected from the Bremen-Grohn data set. Panorama images with a resolution of 3600×1200 are generated with different projections. The subsampling reduction with no scaling is used to generate the reduced scans. SIFT is the feature detector and descriptor method. The transformation matrix from feature-based registration is used as the initial guess for the ICP algorithm. Columns 7 and 8 of the Table 4.4 show the time and number of iterations required for the ICP algorithm to finalise the registration. Point pairs are accepted for ICP with a maximum distance of 10 cm. A very strict convergence criteria is used.

For scan pair (0-1) the reduced point clouds with the cylindrical and Pannini projection could not be registered with ICP using a maximum point distance of 10 cm but required a distance larger than 1 m and a more lenient criteria in order to converge. The rectilinear projection produced point clouds that could not be registered. This means that for these projections the initial guesses were inferior to the others. Scan pair (6-7) was matchable after applying all the reduction methods. The Pannini reduced point cloud gave by far the worst initial guess. The Albers equal-area conic and the equal-area cylindrical projections outperform the other projections for both scan pairs as they require the least number of ICP iterations to converge.

In contrast to the previous results (cf. presented results in Chapter 3) the Pannini projection performance is among the worst. The Pannini projection has large distortions with growing distance from the horizon. While the feature-based registration was previously applied to outdoor data and indoor data with a small vertical field of view. The new experiment deals with indoor data that were captured with a laser scanner with a larger vertical field of view thus

Table 4.2: Performance of the point cloud reduction method based on panorama scaling reduction. The nearest neighbour method is used for interpolating the pixel values. The first column presents the scaling factor used on panorama images with a resolution of 7200×2400. The panorama projection method is the Mercator projection. Time is given in seconds. The number of detected matches from SIFT feature detector and descriptor is presented. The success of the registration method is decided upon visual inspection of the resulting transformation due to the lack of ground truth.

Scale	**Nearest-Neighbor Interpolation**						
	# points	**Subsampling**			**Recovery**		
	in 1,000	**Matches**	**Time**	**Reg.**	**Matches**	**Time**	**Reg.**
100%	15,882	139	679.18	yes	137	608.63	yes
75%	8,966	17	7.08	no	90	170.62	yes
50%	4,014	49	49.85	yes	54	57.32	no
40%	2,556	7	0.76	no	31	9.91	no
30%	1,438	1	0.22	no	22	6.05	no
25%	1,003	26	1.48	no	25	1.42	no

Table 4.3: Performance of the point cloud reduction method based on panorama scaling reduction. The bilinear method is used for interpolating the pixel values. The first column presents the scaling factor used on panorama images with a resolution of 7200×2400. The panorama projection method is the Mercator projection. Time is given in seconds. The number of detected matches from SIFT feature detector and descriptor is presented. The success of the registration method is decided upon visual inspection of the resulting transformation due to the lack of ground truth.

Scale	**Bilinear Interpolation**						
	# points	**Subsampling**			**Recovery**		
	in 1,000	**Matches**	**Time**	**Reg.**	**Matches**	**Time**	**Reg.**
100%	15,882	143	709.25	yes	136	723.14	yes
75%	9,695	93	244.54	yes	93	280.09	yes
50%	4,309	47	46.73	yes	49	47.65	yes
40%	2,757	31	31.23	yes	32	25.88	yes
30%	1,551	18	6.15	no	17	5.92	no
25%	1,077	8	1.59	no	5	0.87	no

decreasing the performance of the Pannini projection. In support of the previous results (cf. presented results in Chapter 3) the Mercator projection achieves promising results. After the Albers equal-area conic and the equal-area cylindrical projections the Mercator projection has the fewest required ICP iterations for convergence.

Table 4.4: Performance of the panorama based point cloud reduction method based on 8 different projection methods on church data set in Bremen-Grohn, Germany. The panorama resolution used for the experiment is 3600×1200. The subsampling reduction method without scaling factor was used for the reduction of the point clouds. The number of detected matches from the SIFT feature detector ans descriptor is presented. Time is given in seconds. The success of the registration method is decided upon visual inspection of the resulting transformation due to the lack of ground truth of the captured data. The required time and the number of iteration for the ICP fine registration are presented.

Scan Pair	Projection	# points in 1,000	Matches	Time	Reg.	ICP	
						Time	Iteration
0–1	Equirectangular	4,305	32	43.47	yes	11.71	448
	Cylindrical	4,026	28	31.38	yes	-	-
	Mercator	4,309	27	31,78	yes	7.82	404
	Rectilinear	3,906	12	9.64	no	-	-
	Pannini	4,047	13	11.76	yes	-	-
	Stereographic	3,720	43	64.72	yes	17.23	765
	Albers equal-area conic	3,475	34	33.43	yes	1.40	39
	Equal-area cylindrical	4,301	32	45.71	yes	1.03	29
6–7	Equirectangular	4,310	69	61.98	yes	1.11	39
	Cylindrical	4,038	44	27.04	yes	1.04	36
	Mercator	4,313	76	53,81	yes	0.79	32
	Rectilinear	3,907	40	26.71	yes	1.63	54
	Pannini	4,052	41	34.99	yes	16.70	705
	Stereographic	3,730	62	67.94	yes	1.32	46
	Albers equal-area conic	3,479	54	30.52	yes	0.80	26
	Equal-area cylindrical	4,308	71	47.60	yes	0.90	30

4.4 Summary

This chapter presents point cloud reduction methods. The octree based reduction and panorama based reduction methods are presented. Three point cloud reduction methods based on panorama images are introduced. The first method utilises the correspondences between 2D panorama pixels with 3D points of the point clouds that are generated during the panorama projection process. This method subsamples the point cloud by selecting only one 3D point per 2D image pixel. The second method utilises the reverse equations of panorama projections to estimate the 3D position of the image pixels. This method uses the x and y coordinate of the image pixel to calculate the longitude and the latitude of the 3D point with the help of reverse equations of the panorama projection method. The third method utilises the conventional 2D image scaling methods on top of the two aforementioned methods for further reduction of the point cloud. The presented results have shown that some of the reduction methods are very effective at drastically

reducing point clouds while retaining all necessary information for panorama based registration. Reduced input sizes strongly decrease the runtime of the registration algorithm.

Chapter 5

Point Cloud Compression

Light Detection And Ranging (LiDAR) systems are used to generate accurate and high resolution 3D point clouds. Advances in modern 3D laser scanners provide easy to use methods and tools to collect large 3D point clouds. Sensors such as the Microsoft Kinect, stereo cameras, and 2D/3D laser scanners produce large amounts of data. New achievements in technology have widened the areas of use for 3D data acquisition techniques. 3D point clouds are used in many applications such as 3D environment modelling, archeology, surveying and especially for mobile robotics. Mobile robots are in turn used for various applications. One key application for mobile robots is their use in dangerous and hazardous environments. This includes the exploration and digitisation of hazardous environments in search and rescue mission. Depending on the application these systems are either controlled remotely or function semi-autonomously or fully-autonomously. Semi-autonomous and remote controlled systems require communication with a remote system. The robot transmits the data to an operator. Operators need to see the robot's environment adequately to manipulate the platform. In many applications the fully autonomous system transmits the data to a more powerful remote work station for processing of the new captured data. Therefore, the data acquired by sensors on the robot have to be transmitted fully or partially over a network to the off-board system. Wireless networks are a common type of infrastructure used for transmitting the data. Common wireless networks such as WiFi and HSPDA are relatively slow networks with limited bandwidth. Additionally, the maximum bandwidth of wireless network usually cannot be achieved, due to the distance between sender and receiver and environmental factors such as walls, objects and interference with other wireless networks. Vision sensors such as stereo cameras and 2D/3D laser measurement systems generate a very large amounts of data. Therefore, the bottleneck in tele-operating these systems is often the limited network capabilities. In addition to user interaction with the robot, the use of an off-board system to increase the process power is advantageous for fully autonomous robots. In real world applications the transmission limitation prevents a full stream of data over the network. In addition to the complexity of large data transmission, 3D point cloud archiving requires a very large amount of storage due to the large amount of captured data with laser scanners. Modern laser measurement systems are capable of very fast point measurements. For example the rotational speed of the Z+F IMAGER 5010C scanner is 3000 rpm and it is capable of measuring more than 1 million points per second [254]. Furthermore, high-end laser

scanners have a large dynamic range, e.g., the Riegl VZ-400 has a range of 1.5 m to ∼600 m with an accuracy of a few millimetre [191]. The scans contain the range and the reflectance values of the measured surface. Many 3D laser measurement systems are capable of capturing the color of the environment in addition to the range and reflectance measurements. E.g., the Z+F IMAGER 5010C has an integrated High Dynamic Range (HDR) color camera [86]. The HDR camera captures color information for each measured surface. The raw data from a 3D laser measurement system or a vision system is commonly stored in an ASCII format. This method generates large files that are used for the transmission and archiving of the point clouds. This increases the problem of data storage and minimisation of the amount of data with no or limited loss of information.

5.1 LiDAR data Compression Methods

Data captured with LiDAR scanners are usually stored in vendor specific binary files. For exchange between users the simple ASCII representation is preferred. The ASCII representation represents the data in lines containing the attributes of each measurement. This is flexible and very easy to understand, especially for humans. However, storing millions of LiDAR measurements in a textual format generates very large files for which only inefficient parsing capabilities exist. The American Society of Photogrammetry and Remote Sensing (ASPRS) created a simple binary exchange format called the LAS format [5]. Furthermore, LAZ, a lossless compressor for LiDAR data stored in the LAS format is presented in [114]. The LASzip compressor is a lossless, non-progressive, order-preserving compressor that provides random-access to the data. Mongus et al. [158] present a three-step lossless method for compression of LiDAR LAS files. They apply the predictive coding, a variable length coding and an arithmetic coding consequently. The key to their method is the use of three predictors for x, y and z coordinates and a separate predictor for scalar values, associated with each point. Several compression methods have been proposed in the literature that rely on data structures such as the octree and k-d tree. A k-d tree approach presented by Gandoin and Devillers [62, 88] uses a tree structure for recursively splitting and subdividing the point cloud. Elseberg et al. [73] present a method to encode one billion points in the cloud in an octree data structure on conventional hardware. Similarly Hornung et al. [108] explore the compression of point cloud based on octrees. Peng et al. [185] and Huang et al. [109] use the octree structure for progressive encoding of point clouds. The octree structure provides a multi resolution spectral decomposition of the point cloud. This allows for encoding of point clouds of higher resolution from subsampled low resolution versions of the point clouds. Progressive point cloud coding techniques are particularly important in scenarios with low bandwidth networks or with unstable networks with a high possibility of connection loss. This provides a receiving system for low resolution point clouds at first and based on that generates the higher resolution point cloud. A prediction scheme is used by Schnabel et al. [203] and Peng et al. [186] to further improve the octree based compression approach. Octree approaches are specifically effective on volumes with a large amount of free space. This is due to the structural design of octrees that are used to present the occupied surface points measured by a range measuring system. Similar to these approaches, Kammerl et al. [118] present a novel compression approach which exploits spatial and temporal redundancy within the point cloud

data stream. Their algorithm compresses the octree by expressing the differences of the current point cloud with respect to previous one.

LiDAR data captured with 3D laser measurement systems can be encoded as panorama images. Panorama projection is the process of mapping the spherical coordinates θ, φ, r of the measured points onto the image coordinates x and y. Chapter 2 presents several methods for encoding point clouds in panorama images. Image based compression methods can be applied to these images. Savakis [202] presents a comparative study of many lossless compression algorithms (UNIX compress, GZIP, LZW, old lossless JPEG, JPEG-LS based on LOCO, CALIC, FELICS, S+P Transform, PNG and other non grey scale methods). Kaess et al. [117] focus on real time compression of laser data on board of mobile robot platforms. For compression purposes they present laser data as a grey scale depth image. They have considered UNIX compress, GZIP, BZIP2, PNG, JPEG-LS and wavelet transformations for compression. Krishnamurthy et al. [136] investigated the use of image compression algorithm on depth based images. They introduce two improvements. First region of interest coding where regions with the most crucial depth are identified and second reshaping the dynamic range of the depth map. They achieve improvement over the standard JPEG2000. Prediction models for image transformation are used by Ghamisi et al. [89] to introduce a less complex and efficient lossless compression method for LiDAR rasterised data. They presented a novel real time algorithm for lossless compression for remote sensing data. They compare their method with other compression methods such as lossless JPEG and lossless version of JPEG 2000. Nenci et al. [167] have extended the compression of range data to video compression methods. Mobile robots require the ability to transmit a stream of data to a base station or an operator. Transmission of raw sensor data over low bandwidth network is problematic. In [167] they present a method that exploits H.264 compression to reduce the size of the data stream from sensors such as the Kinect. Next sections in this chapter propose the use of conventional image compression method to compress 3D point clouds. This method is based on the generation of panorama images from point clouds. The proposed compression method introduces novel range encoding methods for lossy and lossless image compression methods. Finally, a filtration method is proposed to improve the results from the lossy image compression method.

5.2 Conventional Image Compression Methods for Point Cloud Compression

Point clouds captured with terrestrial 3D laser measurement systems are projected onto panorama images with several projection methods. The generation of these images allows the use of conventional image compression methods to obtain compressed point clouds. Compression methods can be categorised into lossless and lossy methods. Lossless methods provide the compression without loss of information. With these methods all the original data can be reconstructed from the compressed data. On the other hand lossy methods sacrifice some information in the compressed data for a significantly higher compression ratios. Lossless compression methods are preferred for point cloud compression. They preserve the original data and can therefore be used for data archiving. The higher compression ratio of lossy methods generates point clouds with smaller memory size. They can be used for visualisation and fast data transmission over

low band-width networks.

The Joint Photographic Experts Group (JPEG) standards are well-known image compression methods. The JPEG standard contains many methods for image compression. The most widely implemented method in JPEG is a lossy form of compression based on discrete cosine transform (DCT) [236]. The compression process starts by grouping the source image samples in to 8×8 blocks. Their values are shifted to center around zero. The DCT is applied to the shifted blocks. The DCT coefficients are uniformly quantised in conjunction with a quantisation table. Quantisation is the process of removing the high frequency data. This is the lossy part of DCT-based JPEG. The final step of compression is the entropy coding. This step achieves additional lossless compression by encoding the quantised DCT coefficients with Huffman coding [111].

The JPEG compression method is usually lossy. This means some of the original image information is lost and cannot be restored after the decompression of the data. Therefore, it is not the most suitable method for point cloud compression. This data loss affects the image quality and also the quality of the measured 3D points of the recovered point cloud. However, many image applications can sacrifice small deformation of the data for a higher compression ratio. This chapter investigates the effects of the deformations produced by lossy compression methods on 3D point clouds. In addition to lossy JPEG compression the JPEG standard presents less common lossless compression methods. JPEG 2000 is a compression method based on the discrete wavelet transform (DWT), scalar quantisation, context modelling, arithmetic coding and post compression rate allocation [200]. JPEG 2000 provides both lossless compression and lossy compression with higher rates of compression.

The Portable Network Graphics (PNG) [235] is a patent free, W3C recommendation for coding still images. PNG is an extensible file format for the lossless, portable, well-compressed storage of raster images. Due to its lossless nature PNG compression is a suitable compression method of rasterised 3D point clouds. A predictive scheme and entropy coding are the two steps of this compression method. The prediction is done on the nearest neighbours and there are five predictors in the specification. The entropy coding is done by non patented lossless data compression method known as Deflate algorithm [61]. This method compresses the data using a combination of the LZ77 algorithm [252] and the Huffman coding [111]. PNG supports grey scale and color with alpha channel images.

The Tagged Image File Format (TIFF) [166] is a container format used for storing images. It incorporates many options. A TIFF file can be considered to hold the raw data of the image, i.e., the image data without compression. It also supports many compression methods. The most common and general purpose compression algorithm used with TIFF is the lossless Lampel-Ziv-Welch (LZW) [244] compression method. Similar to LZ77 and LZ78 [253], LZW is a dictionary based lossless compression method that works on generic data coding. Another common compression method supported by TIFF is the PackBit compression scheme. This is a simple and fast lossless compression that relies on run length encoding (RLE). This algorithm encodes the redundant information by sorting the series of identical pixels with a control bit and the value of the compressed pixel.

This chapter presents the use of the aforementioned conventional image compression methods for compressing the rasterised images generated from point clouds captured by a 3D laser measurement system. Separate panorama images are generated (cf. Chapter 2) for range, color and reflectance data from the point cloud and subsequently compressed to minimise the required

memory for each panorama image.

5.3 Range Encoding and Point Cloud Compression

The approach proposed in this chapter starts with encoding range, color and reflectance values from a 3D point cloud in separate panorama images. Then these images are compressed with conventional image compression methods. This effectively compresses the 3D point clouds. It is a common practice for 3D laser measurement systems to represent the reflectance data with 8 bit grey value. Color information is often presented with three 8 bit red, green and blue values. Range information however is a floating point value to present the distance between the measured environment and the 3D laser measurement system in meters. A grey scale image and an RGB image are generated to represent the reflectance and color data from the captured point cloud. A problem of conventional image compression methods for range data is the number of bits per pixel that compression method supports. Most compression methods support only 8 bits per pixel channel. A single precision floating point has a width of 32 bits. In order to encode the range information in an image for compression, the range value is multiplied by a scale factor of 10^4 and the float is converted to integer. This generates integer range values with 0.1 millimetre accuracy.

5.3.1 Range to Color Map Encoding

There is a common misconception that the floating point representation provides more precision than integer. Storing a value as floating point means that the precision of the value varies depending on the number that the value represents. Numbers closer to zero have more precision. On the other hand 0.1 millimetre range accuracy is higher than the range accuracy of most 3D measurement systems. The highest 24 bit integer value is $2^{24} = 16777216$ and with 0.1 millimetre accuracy ranges up to 1676 meters can be encoded in 24 bits of data. The approach presented in this chapter uses this method to map the range information with 0.1 millimetre accuracy on an RGB image. The 24 bit range is divided bitwise in to three 8 bit values to fill the red, green and blue values of the image, cf. Figure 5.1.

The RGB color model is an abstract mathematical model describing a way to represent a color with three numbers. A color space is the interpretation of the components of a color model. Adding a mapping function between color model and a reference color space creates a footprint or gamut which defines the color space for a specific color model. The proposed range to color encoding method has no connection to any globally known system of color interpretation and it is capable of mapping 2^{24} range values to 2^{24} colors. Several other color mapping methods exist, such as jet, hot and rainbow spectrum colors. However, these methods contain a subset of the RGB color space. For example using jet color map with only 32000 different colors maps range values from a point cloud with maximum 400 meter range with only 1.25 centimetre accuracy. The maximum range and accuracy supported with this range to color map encoding is lower than the range and accuracy of many professional terrestrial 3D laser measurement units. Higher accuracy is achieved by utilising all 2^{24} colors of the RGB color map to map the range values to an RGB color. Using color maps that only contain part of the RGB color model results in loss

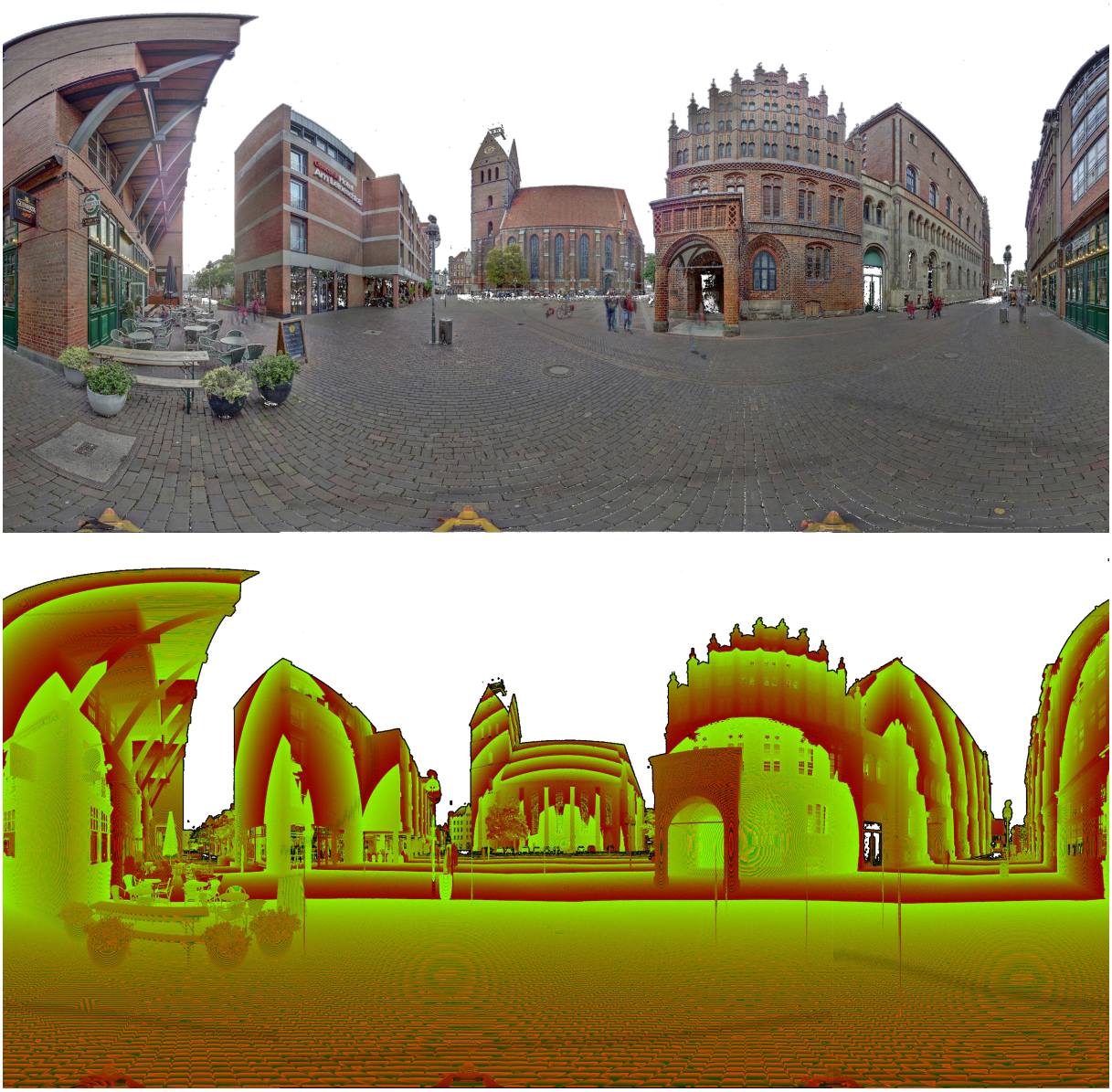


Figure 5.1: Panorama images with the equirectangular projection built from a point cloud captured with Z+F IMAGER 5010C from the city center of Hannover, Germany. Top: Color panorama image. Bottom: Panorama image presenting the range encoding in three channels.

of data for point clouds with long range and high accuracy. Therefore, these methods are not suitable for range encoding of point clouds generated by terrestrial laser scanners.

In the proposed range encoding method the range value is stored in a three byte integer as red, green and blue. The value of red corresponds to the least significant byte, i.e., the lowest range, and the value of blue to the most significant byte, the highest range. According to the

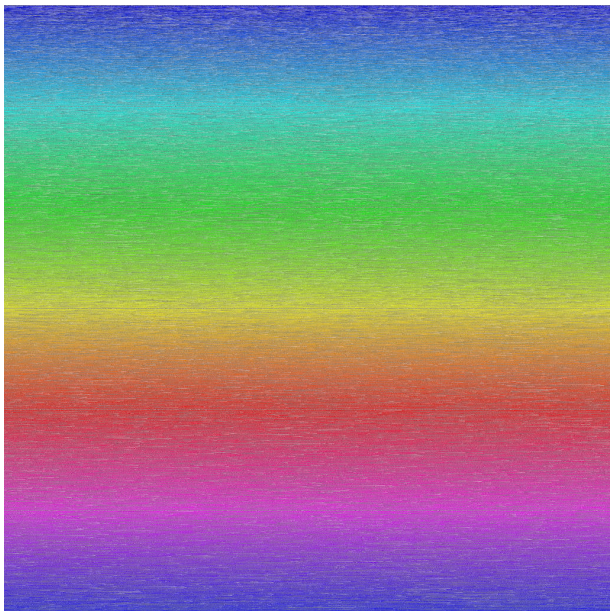


Figure 5.2: All 2^{24} RGB colors are mapped to the HSL color model. It maps the range 0 to top left corner and 16777216 to the bottom right corner. The values are sorted based on hue, saturation and lightness. The image represents this color map with 24 bit of depth.

resolution smallest change in the blue channel corresponds to a change on range of approximately 6.5 meters. A scan with a maximum range of 200 meters has the blue channel with maximum value of 30. Therefore, the blue color is not the prominent in the image and shades of green and red are mapped to every 6.5 meters of range (cf. Figure 5.1). A more intuitive representation of range data is presented as a smooth transition between colors to represent the increase in range. This color map is generated by mapping the sorted RGB colors based on blue, green and red to the Hue-Saturation-Lightness (HSL) color model. Furthermore, to produce the smooth transition of colors the values are sorted based on hue, saturation and lightness. Figure 5.2 represents the 2^{24} RGB colors mapped and sorted based on the HSL color model. This maps the range 0 to top left corner and 16777216 to the bottom right corner of the Figure 5.2. This color map is used to generate a three channel image with the encoded range value of a scan. Figure 5.3 presents the image generated with this color map from a scan. The same scan as for Figure 5.1 is used to generate the image for Figure 5.3. The color transitions from darker blue to lighter blue. The variation and transition of colors are dependent on the maximum range and the variation of the range in the scanned environment. Therefore, scans with longer range include more colors from Figure 5.2.

5.3.2 Range to Grey Scale Encoding

Lossy compression methods sacrifice the quality of the compressed image to the higher compression ratio. The JPEG compression method for color images is tailored to the human eye in order to reduce the detectable artefacts in compressed image. The human eye is more sensitive to changes in intensity than the hue of a color. Therefore, the JPEG compression method

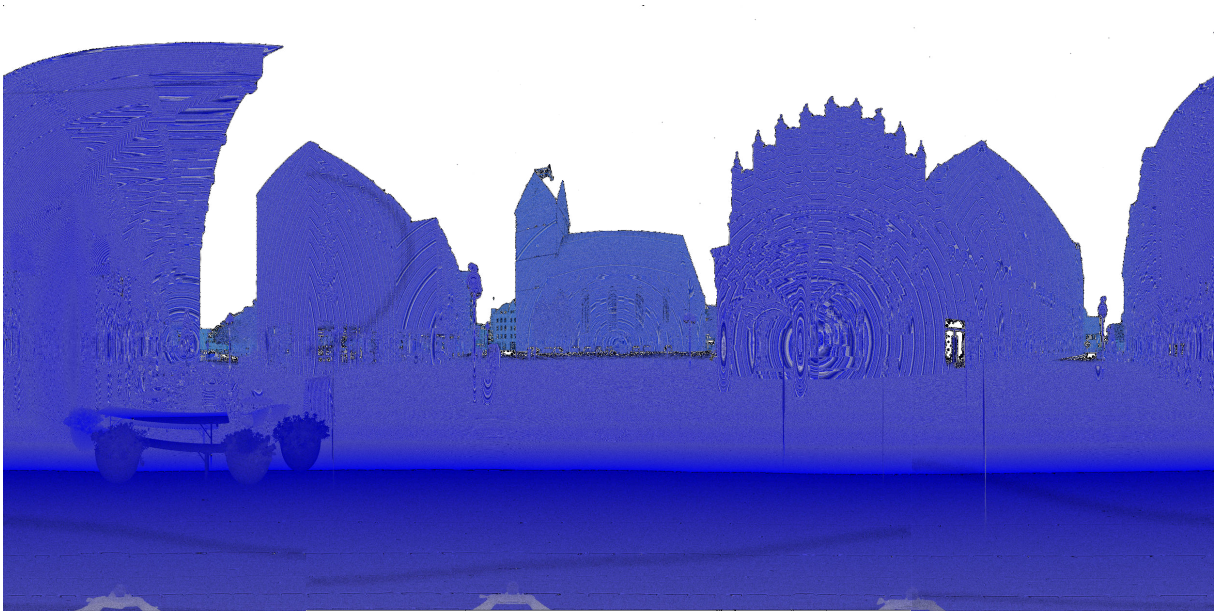


Figure 5.3: Range encoding onto three channel panorama image using HSL color. The range value is encoded onto blue, green and red values. These values are mapped onto the HSL color model and sorted out based on hue, saturation and lightness.

is optimised to have a higher compression on the hue of the color than the intensity. JPEG compression methods mostly convert the RGB color scheme to YCbCr before the compression. Additionally, they use higher compression ratio on the chrominance than the luminance of the image. This provides a higher compression ratio of the overall image with less detectable artefacts. For the purpose of the experiments in this chapter and to avoid different compression ratios for the chrominance and the luminance with the JPEG compression method a new method of range encoding is proposed. In addition to the use of the three channel image for compression a separate grey scale image is generated for each channel. This guarantees the use of identical compression ratios for all parts of the encoded range value. The JPEG method on grey scale images utilises the same compression ratio used for luminance on color images. Due to the lossy nature of JPEG compression, the compressed image is not a complete match of the original data. Similarly the recovered 3D point cloud from the decompressed panorama image does not match the original measurements. The proposed three grey scale image for encoding of the range is used for JPEG compression to reduce this difference.

5.3.3 Panorama Compression

A 3D point cloud is mapped onto a panorama image. A 24 bit color image or three separate grey scale images are used for range values. A grey scale image and a color image are used respectively for reflectance and color information of the 3D point cloud. These images are compressed using conventional image compression methods. They can be archived or transferred over the network efficiently. To recover the 3D point cloud, images containing range, reflectance and color values are decompressed and the range information is decoded from the image based on the method

Algorithm 3 Filtration algorithm used to remove the outliers of recovered point cloud from panorama image. These images are compressed by JPEG compression method.

Require: *image* // *The JPEG compressed panorama image of a 3D point cloud*
Require: $T \geq 0$ // *The scale factor to determine the tolerance of the filtration process*
  for $x = 0$ **to** *image.width* **do**
    for $y = 0$ **to** *image.height* **do**
      $p \leftarrow image[x][y]$
      $n \leftarrow$getNeighboursOfPixel(p)
      sortByRange(n)
      $q_1 \leftarrow$ getLowerQuartile(n)
      $q_2 \leftarrow$ getUpperQuartile(n)
      $interquartile \leftarrow (q_2 - q_1) * T$
      $minBoundary \leftarrow q_1 - interquartile$
      $maxBoundary \leftarrow q_2 + interquartile$
      if $p.range \geq minBoundary$ **and** $p.range \leq maxBoundary$ **then**
        Use this pixel for recovered point cloud
      else
        This pixel is considered an outlier and removed from recovered point cloud
      end if
    end for
  end for

used for encoding the information onto the image. The 3D points are recovered from the mapped panorama images based on the reverse equations of the projection method used for generation of the panorama image.

5.4 Filtration Method For Recovered Range Values

The Lossy JPEG compression method introduces artefacts in the compressed image. Therefore the recovered point cloud from the compressed JPEG panorama image is imprecise. The artefacts produced by the JPEG compression are inevitable and occasionally map to points on the recovered point cloud that do not exist in the original data set. These points are defined as outliers. This section presents a filtration method for the decompressed point cloud from the JPEG compression. This method is used to remove the outliers of the recovered point cloud that are produced by artefacts from the JPEG compression method. To determine the outliers for each image pixel the eight surrounding pixels are checked. The lower and upper quartile of the respective ranges are calculated. The interquartile range is calculated to determine the boundaries for outlier selection. A scale factor is multiplied by the interquartile range to define the tolerance of the filtration method for outlier selection. The boundaries are calculated by adding the scaled interquartile to the upper quartile and subtracting from the lower quartile. Range values outside these boundaries are considered to be outliers and are therefore removed from the recovered point cloud. This algorithm is described in Algorithm 3.

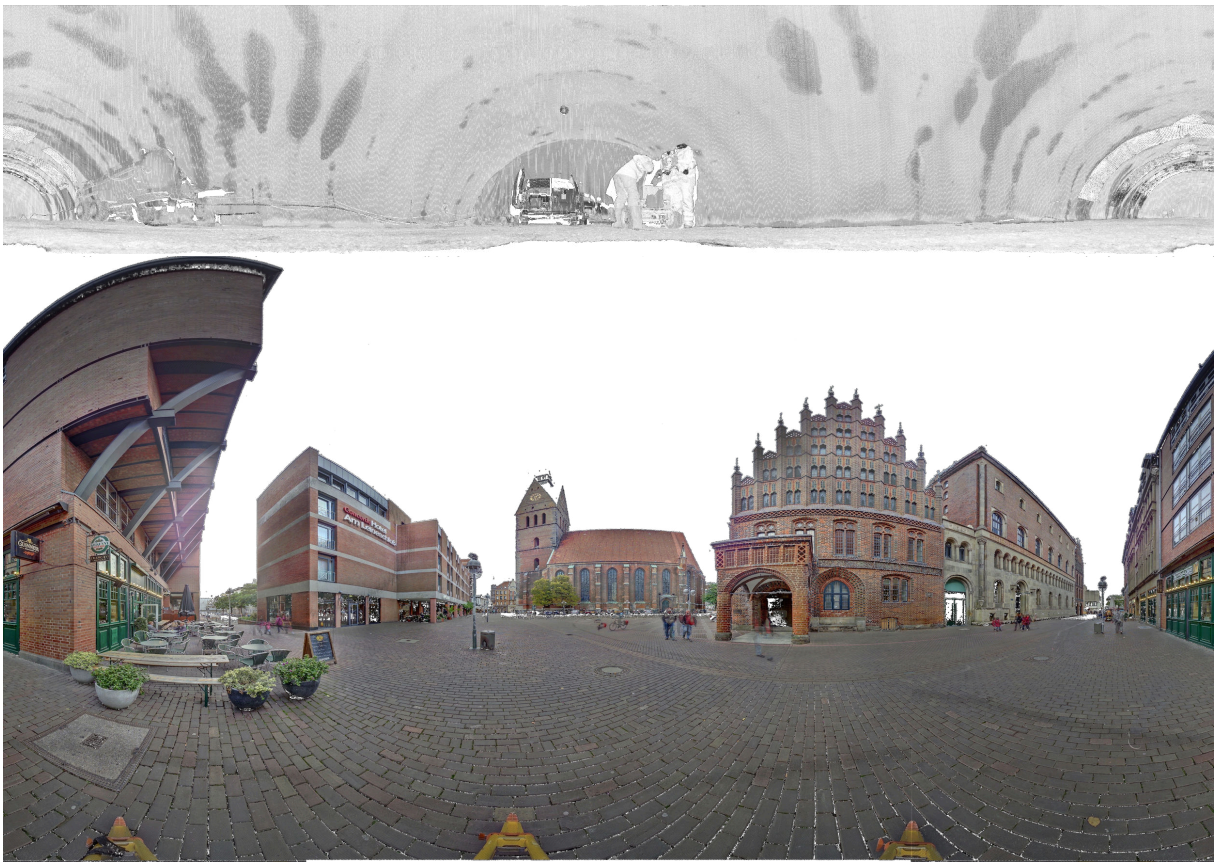


Figure 5.4: Panorama presentation of captured data with the Riegl VZ-400 and Z+F IMAGER 5010C. The Riegl VZ-400 has a 360-degree horizontal field of view and 100-degree vertical field of view. The Z+F IMAGER 5010C has a 360-degree horizontal field of view and 160-degree vertical field of view. The panorama image size is dictated by the field of view of the measurement system. This causes vertically smaller panorama images for the Riegl VZ-400. Both images are generated using the Mercator projection method. Top: Reflectance panorama image presenting the data captured by the Riegl VZ-400 from inside a tunnel. Bottom: Color panorama image presenting the data captured by the Z+F IMAGER 5010C from the city center of Hannover, Germany.

5.5 Case Study

The experiments in this section are designed to evaluate the potential of image compression methods for point cloud compression. Furthermore, they evaluate the effects of lossy compression methods on point clouds. First, a comparison of lossless compression methods are presented. Second the impacts of lossy JPEG compression method with several quality settings are evaluated. Third a comparison of two presented range encoding methods based on RGB and HSL color maps are presented. Finally the effects of different projection methods on the compression is presented. To compare all the aforementioned compression methods, two data sets have been used. The first data set is a single scan captured with the Riegl VZ-400 3D laser measurement system from inside a tunnel (See Figure 5.4 and Figure 5.5). This 3D laser measurement system

Figure 5.5: 3D views of the point clouds used in the experiments. Left: Rendering of the point cloud captured by the Riegl VZ-400 (viewed from inside the tunnel). The Riegl VZ-400 provides range and reflectance measurements for each 3D point. Right: A rendering of the point cloud captured by the Z+F IMAGER 5010C from the city center of Hannover, Germany. The Z+F IMAGER 5010C provides range, reflectance and color for each measured point.

Table 5.1: Comparison of different compression methods for the two data sets. Tunnel data set was captured from inside a tunnel with the Riegl VZ-400 3D laser measurement system. This data set contains range and reflectance values. City data set was captured from the city center of Hannover, Germany with the Z+F IMAGER 5010C 3D laser measurement system. This data set contains range, reflectance and color values. The file size presented for image based compression methods is the combination of all generated panorama images to encode range, reflectance and color information.

Compression Method	Tunnel Dataset	City Dataset
ASCII	681.5 MB	1.44 GB
LAS	440.5 MB	1.16 GB
LAZ	62.3 MB	170.9 MB
OCT	352.5 MB	889.5 MB
PNG	35.8 MB	142.2 MB
JPEG 2000	51.3 MB	149.2 MB
TIFF with no compression	90 MB	350 MB
TIFF with LZW	47 MB	144.8 MB
TIFF with PackBits	62.4 MB	247 MB

only provides the range and reflectance information of the environment. The second data set is a single scan captured by Z+F IMAGER 5010C from the Hannover city center in Germany (See Figure 5.4 and Figure 5.5). In addition to range and reflectance this 3D laser measurement system provides color information for measured points.

The experiments in this chapter utilised libpng, libjpeg, libtiff and libjasper for compressing images. A comparison of lossless image compression methods are presented in Table 5.1. Additionally this table includes the standard ASCII and the LAS format for comparison. Two other compression methods are the lossless compression of the LAS format namely LAZ and the representation of point cloud in octree structure based on the method presented by [73]. The

Table 5.2: Detailed representation of the file sizes of compressed point clouds. This table presents the file size for three channel image representing range, grey scale image representing reflectance and RGB image representing the color information of a scan from the Hannover data set.

Compression Method	Three Channel Range	Grey Scale Reflectance	RGB Color
PNG	29.2 MB	24.4 MB	88.6 MB
JPEG 2000	55.3 MB	37.7 MB	61.2 MB
TIFF with no compression	150 MB	50 MB	150 MB
TIFF with LZW	27.5 MB	29 MB	88.3 MB
TIFF with PackBits	103.5 MB	36 MB	107.5 MB

results in Table 5.1 show that the PNG compression is the method with the highest compression ratio. Furthermore the uncompressed TIFF file format encodes the point clouds in images with smaller file sizes than the LAS and octree method. This reduction of size is achieved based on the two features used for generating these images. The first feature is the use of pixel coordinates as longitude and latitude of the 3D points. Secondly the encoding method used for converting the range value from floating point to 24 bit three channel pixel. The presented file sizes for the image based compression methods are the combination of the three channel image representing the range value, the grey scale image representing the reflectance value and the RGB image representing the color information of the point cloud. Table 5.2 provides a more detailed comparison of the image based compression methods for the Hannover data set.

The main advantage of lossless compression methods is to provide a compression method without loss of data. The introduced error in lossy methods is inevitable. The JPEG compression method is designed to perform higher compression on chrominance of the image than the luminance to reduce the artefacts of the compression method that are detectable by the human eye. The JPEG quality setting of the JPEG compression is a measurement unit to produce images with different compression ratio. Higher quality settings generate less compressed images. Additionally this chapter presents the encoding method for range information to be encoded into three grey scale images to reduce the loss of information. This method is used to ensure the same level of compression on all encoded range channels. Figure 5.6 shows a recovered point cloud from panorama images compressed with the JPEG compression method using a quality setting of 100. The recovered point cloud for both three channel and three grey scale images range encoding methods are presented.

The results of the compressed point clouds with the lossy JPEG compression method is presented in Table 5.3. The results contain several quality settings of the JPEG compression method. Additionally, it includes the comparison of these two range encoding methods for lossless compression methods. As expected the lossy JPEG compression method with the lowest quality setting has the highest compression ratio. Due to the higher compression ratio of chrominance of the three channel image than the luminance, the three channel range encoding method have a higher compression for JPEG method. The introduced error based on the lossy method results in a very imprecise recovered point cloud. The use of three grey scale images

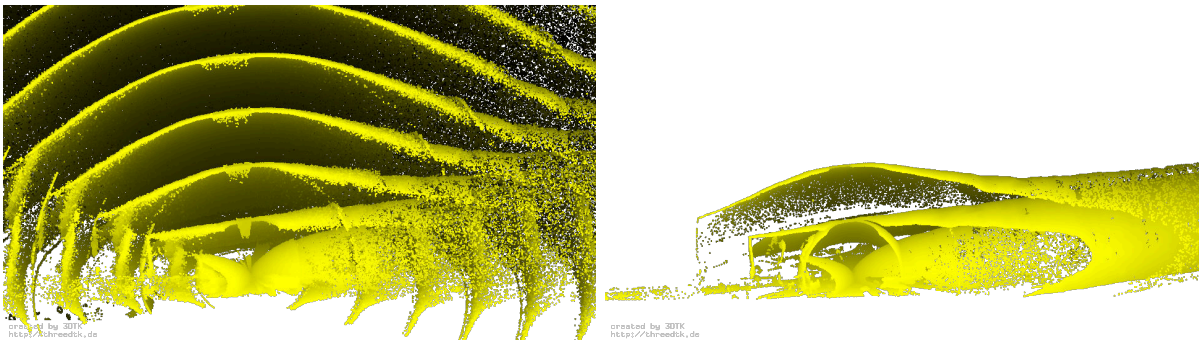


Figure 5.6: Recovered point cloud from encoded range in JPEG panorama image with compression quality setting of 100. Left: The recovered point cloud from the image using the three channel image range encoding method. Right: The recovered point cloud from the image using the three grey scale images range encoding method.

Table 5.3: Comparison of three channel and three grey scale range encoding method. The table presents file sizes generated from panorama images compressed by lossy JPEG compression method with a few different quality settings. Furthermore, file sizes of generated panorama images with lossless compression methods is presented. The tunnel data set was used for the creation of this table.

Compression Method	Three Channel Range	Three Grey Scale Range
JPEG 100	26.2 MB	38.3 MB
JPEG 95	15.1 MB	23.8 MB
JPEG 50	4.4 MB	9.1 MB
PNG	24.4 MB	20.1 MB
JPEG 2000	37.6 MB	25.4 MB
TIFF with no compression	67.5 MB	67.5 MB
TIFF with LZW	31.4 MB	27.9 MB
TIFF with PackBits	46.6 MB	35.5 MB

for encoding the range value improves the recovered point cloud. However the recovered point cloud still contains many outliers. Using three grey scale images for range encoding improves the compression ratio of lossless methods. In contrast the lossy JPEG compression ratio decreases as a result of same compression on all three channels of range as a substitute for higher compression on chrominance than luminance.

Figure 5.7 shows a visual comparison of the difference of recovered range values from images compressed with the JPEG compression method and the original range values of the point cloud. The result is the comparison of a subset of the point cloud. A section of the panorama image is used to recover a subset of the point cloud. The results reveal less outliers for the three grey scale images encoding method. Outliers are the artefacts generated by the lossy JPEG compression method. A lower quality JPEG used for higher compression ratio results in a higher amount of artefacts in comparison to a quality of 100 (cf. right side of Figure 5.7). Additionally, Figure 5.7

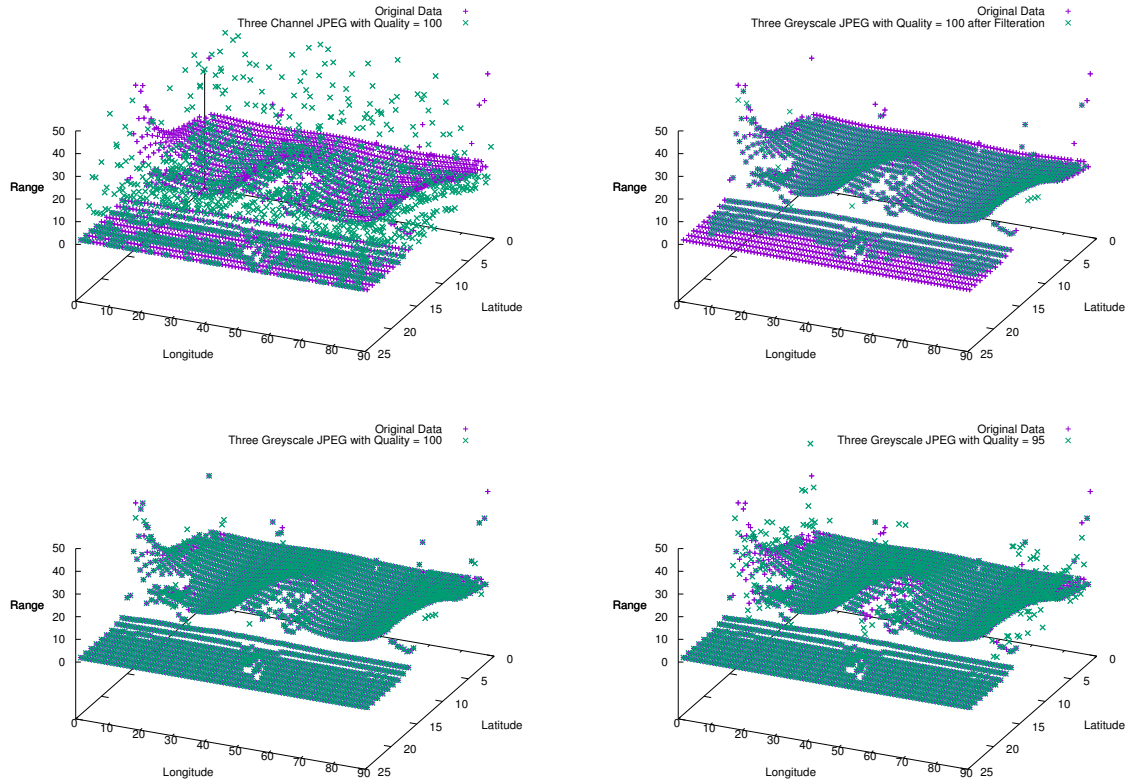


Figure 5.7: Comparison of the original range values with the recovered range values from the JPEG compression based on three channel image range encoding and three grey scale images range encoding. Image coordinates x and y are mapped back to longitude and latitude and the range value is presented as elevation for each plotted pixel. This shows improvement of the three grey scale images range encoding method over three channel range encoding method for recovered point clouds from the JPEG compression. Additionally, it shows the comparison of different quality setting for the JPEG compression method. Finally, the improvement of the filtration method designed to reduce the artefacts introduced by the lossy JPEG compression method is presented. Top Left: Original data and three channel image range encoding method. Top Right: Original data and three grey scale ranges encoding method. Bottom Left: Original data and three grey scale ranges encoding method with a quality of 100 after filtration. Bottom Right: Original data and three grey scale range images encoding method with a quality of 95.

shows the comparison of filtered recovered point cloud from a panorama image compressed using the JPEG compression method with the original data. The filtration method tends to remove the outliers to minimise the effect of the introduced artefacts of JPEG compression method. The recovered point cloud from JPEG compression method after the filtration process is presented in Figure 5.8. The drawback of the filtration method is that it also removes some of the points.

Range values from point clouds are encoded to 24 bit integer and divided into three 8 bit channels to be presented as color. These color values are used to map the range onto a three channel RGB image. Additionally these values are used to map the range values onto three grey scale images. Two range to color encoding methods are presented in this chapter. First method

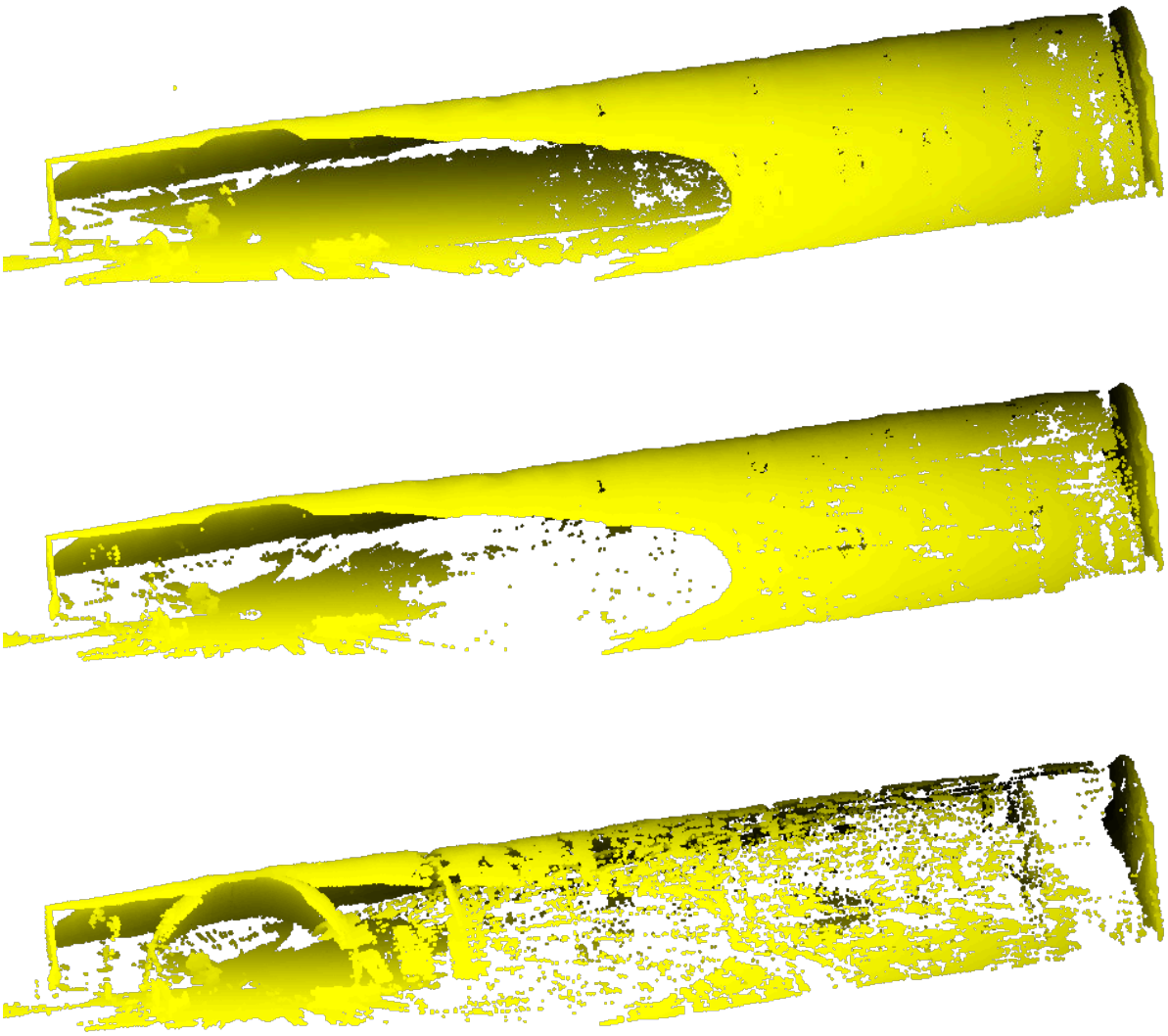


Figure 5.8: Filtration performance on recovered point clouds from the JPEG compression. The three grey scale method is used for range encoding with quality settings of 100 and 95. Despite the filtration process the main structure is visible and understandable by humans. Top: Original point cloud. Middle: Recovered point cloud from three grey scale images range encoding method for JPEG images with a quality setting of 100 and after filtration of range outliers based on our filtration method. Bottom: Recovered point cloud from three grey scale images range encoding for JPEG images with a quality setting of 95 and after filtration of range outliers based on our filtration method. More aggressive filtration boundaries were used for this quality setting.

encodes the range values to RGB colors sorted based on blue, green and red color values. Second method encodes the range values to RGB colors sorted based on HSL color model. Table 5.4 presents a comparison between these range encoding methods. Utilising HSL to sort the RGB

Table 5.4: Comparison of color mapping methods for range encoding. Color maps represent all the colors in the RGB color model with 8 bit depth for each color channel. The RGB map sorts the colors based on blue, green and red colors. The HSL map sorts the colors based on HSL color model. The comparison presents the difference between three channel range encoding method and three grey scale image range encoding method generated from the tunnel data set.

Compression Method	Three Channel Encoding		Three Grey Scale Encoding	
	RGB Map	HSL Map	RGB Map	HSL Map
JPEG 100	26.2 MB	27.9 MB	38.3 MB	66.8 MB
PNG	24.4 MB	42.2 MB	20.1 MB	45.3 MB
JPEG 2000	37.6 MB	39.9 MB	25.4 MB	47.5 MB
TIFF with no compression	67.5 MB	67.5 MB	67.5 MB	67.5 MB
TIFF with LZW	31.4 MB	48.9 MB	27.9 MB	62.3 MB
TIFF with PackBits	46.6 MB	46.8 MB	35.5 MB	54.3 MB

colors generates panorama images with smooth transition between colors. This transition shows the increase of range in the measured environment. However, a smooth transition in color results in the RGB values to be close to each other and not necessarily a small value. On the other hand the encoding method with the RGB color map has small values mostly for the blue channel based on the maximum range of the point cloud. Furthermore, the green and red channels have repetitive behaviour. This results in a higher compression of data, specifically on the blue channel.

Finally Table 5.5 shows the evaluation of the effects of the panorama projection methods on the compression of the point clouds. For this experiment the PNG compression method is used in combination with the panorama projection methods presented in Chapter 2. The Rectilinear projection outperforms other projection methods. However, the limited vertical field of view of the rectilinear projection methods results in the loss of data throughout the projection mapping process. Similarly the cylindrical projection method is affected by the limitation of the vertical field of view. At the same time in comparison to the equirectangular projection method, the Pannini, Mercator and stereographic projection methods improve the compression ratio without loss of data. On the other hand the Albers equal-area conic, equal-area cylindrical and the Lambert azimuthal equal-area projection methods produced lower compression ratio in comparison to the equirectangular projection method.

5.6 Summary

This chapter compared several image compression methods in terms of compression of generated point clouds from 3D laser measurement systems. 3D point clouds are mapped onto panorama images. A range, reflectance and color data from the point cloud are mapped onto separate panorama images. Range encoding method is presented to encode the measured range values from floating points onto 24 bit integer values. These values are mapped onto a three channel image. This process is required for the use of image compression methods. Most image compres-

Table 5.5: Evaluation of the effects of the panorama projection methods on the point cloud compression. The PNG method is utilised as the compression method for panorama images generated with several projection methods. The Hannover city data set was used for this experiment.

Projection Method	Three Channel Range	Three Grey Scale Range	Grey Scale Reflectance	RGB Color
Equirectangular	29.2 MB	21.9 MB	24.4 MB	88.6 MB
Cylindrical	14.1 MB	9.1 MB	13.8 MB	48.9 MB
Mercator	20.1 MB	12.8 MB	19.1 MB	69.9 MB
Rectilinear	9.8 MB	6.7 MB	9.4 MB	33.9 MB
Pannini	12.9 MB	8.7 MB	11.9 MB	43.1 MB
Stereographic	25.8 MB	19.6 MB	20.5 MB	74.1 MB
Conic	30.7 MB	24.5 MB	21.8 MB	76.6 MB
Equal-Area Cylindrical	46.3 MB	33.5 MB	27.8 MB	93.6 MB
Azimuthal	32.8 MB	25.8 MB	23 MB	79.3 MB

sion methods work only with 8 bit grey scale or 24 bit three channel color images. Furthermore, the three grey scale images range encoding method is introduced. This method is presented for the JPEG compression method to avoid the difference compression ratio on chrominance and luminance of color images. Comparison of several lossless image compression methods and lossy JPEG is presented in the experiments of this chapter. The experimental evaluation shows a high compression ratio for point clouds with use of conventional image compression methods. These methods outperform the industry standard LAZ compression for point clouds. The usage of PNG compression is recommended for 3D point cloud compression, resulting in a lossless compression with the highest compression ratio. If errors can be accepted, lossy JPEG can be used. This chapter presents a filtration method to reduce the outliers in the recovered point cloud. These outliers are unavoidable due to the artefacts introduced during the lossy JPEG compression. Additionally the effects of different panorama projection methods on point cloud compression is evaluated.

Chapter 6

Application: Point Cloud Processing in Archeology

Modern archeology utilises the state of the art 3D laser scanning methods for modelling archeological excavation sites, historical sites and even entire cities or landscapes. Accuracy and time efficiency are important aspects of any excavation site. Modern laser scanners are capable of fast data acquisition. They provide a very detailed 3D point cloud of the environment. Point clouds are used to generate the 3D model of the excavation. 3D point clouds and 3D models of an excavation site provide a better representation of the environment for the archeologist and for the documentation generation process. The documentation of findings on an excavation site is an essential archeological task. Documentation of archeological excavation sites with conventional methods and tools such as hand drawings, measuring tape and archeological notes is time consuming. Furthermore, this process is prone to human errors and the quality of the documentation depends on the qualification of the archeological team on-site. Automated systems are considered advantageous for achieving a higher accuracy and time efficiency. The use of modern technology and methods in 3D surveying and 3D robotics facilitate and improve this process. Computer-aided systems and databases improve the documentation quality and increase the speed of data acquisition. These systems tend to minimise the human inaccuracies by replacing the repetitive and difficult process of data gathering with automated systems. This assists the human operative to focus on data analysis and generating knowledge for publication.

This chapter provides an overview on the state of the art in archeology. It describes conventional methods as well as computer-aided methods. Afterwards, this chapter proposes a general framework for using software for the main steps of excavation. The goal of the framework is to assist the excavation process by organising the process into few steps and utilising software solutions for them. Additionally, this chapter introduces a computer aided system for labelling archeological excavations in 3D (CASTLE3D). Consisting of a set of tools for recording and georeferencing the 3D data from an excavation site, CASTLE3D is a novel documentation approach in industrial archeology. The focus of this software is the documentation component of the framework. It provides a 2D and 3D visualisation of the data and an easy-to-use interface that enables the archeologist to select regions of interest and to interact with the data in both representations. The 2D visualisation and a 3D orthogonal view of the data provide cuts of

the environment that resemble the traditional hand drawings. The 3D perspective view gives a realistic view of the environment. CASTLE3D is designed as an easy-to-use on-site semantic mapping tool for archeologists. Each project contains a predefined set of semantic information that can be used to label findings in the data. Multiple regions of interest can be joined under one label. Further information such as color, orientation and archeological notes are added to the label to improve the documentation. The available 3D information allows for easy measurements in the data. The full 3D information of a region of interest can be segmented from the entire data. By joining this data from different georeferenced views the full 3D shape of findings is stored. All the generated documentation in CASTLE3D is exported to an XML format and serves as input for other systems and databases. Apart from presenting the functionalities of CASTLE3D this chapter evaluates its documentation process in a sample project. For this purpose the data is exported to the Adiuvabit database [59] where more information is added for further analysis. The Adiuvabit is used as the database component of the proposed framework of this chapter. Finally the documentation process is compared to traditional documentation methods and it is shown how the automated system helps in accelerating the documentation process and decreases errors to a minimum.

6.1 Conventional Archeology

Archeology is the study of history and human activity in the past, through the analysis and recovery of their legacies and environmental data that they have left behind such as artefacts, architecture, cultural landscapes, art, etc.

In many countries large scale construction projects are prefaced by a series of inspections. These are performed to determine the possible existence of artefacts or other remains of cultural or historical significance. Archeologists accompany the construction projects to prevent the destruction of archeological sites. Efficient excavation is desired to minimise the time and cost for the client. Therefore, development of a framework such as the one presented in this chapter comes to effect: an easy-to-use hardware and software system allows the digitisation of the excavation site with a 3D laser scanner and generation of digital documentation. This will increase the efficiency of archeological activities leading to cost saving results. Succeeding the excavation, collected data and documentation of the potential findings will be evaluated. After analysis and post processing a report will be made available to the authorities.

Conventional methods of documentation for archeological features are based on manual drawings on graph paper and archeological notes. These sketches resemble the excavation by providing level, colors and numbers for findings with a connection to a detailed catalog. Findings are photographically recorded within these catalogs. This process is designed to document the excavation site meticulously during the excavation process (cf. Fig 6.1). These documents are generated to record the findings as precisely as possible and to describe them chronologically in three dimensions. Documents are evaluated after compilation of the excavation. The analysis of the findings provides insights on early culture, religion, handcraft, daily life and social process on both local and national levels. However, lack of access to the original findings on the excavation and dependency of the documentation on the knowledge and understanding of the on-site archeologist make the documents biased.

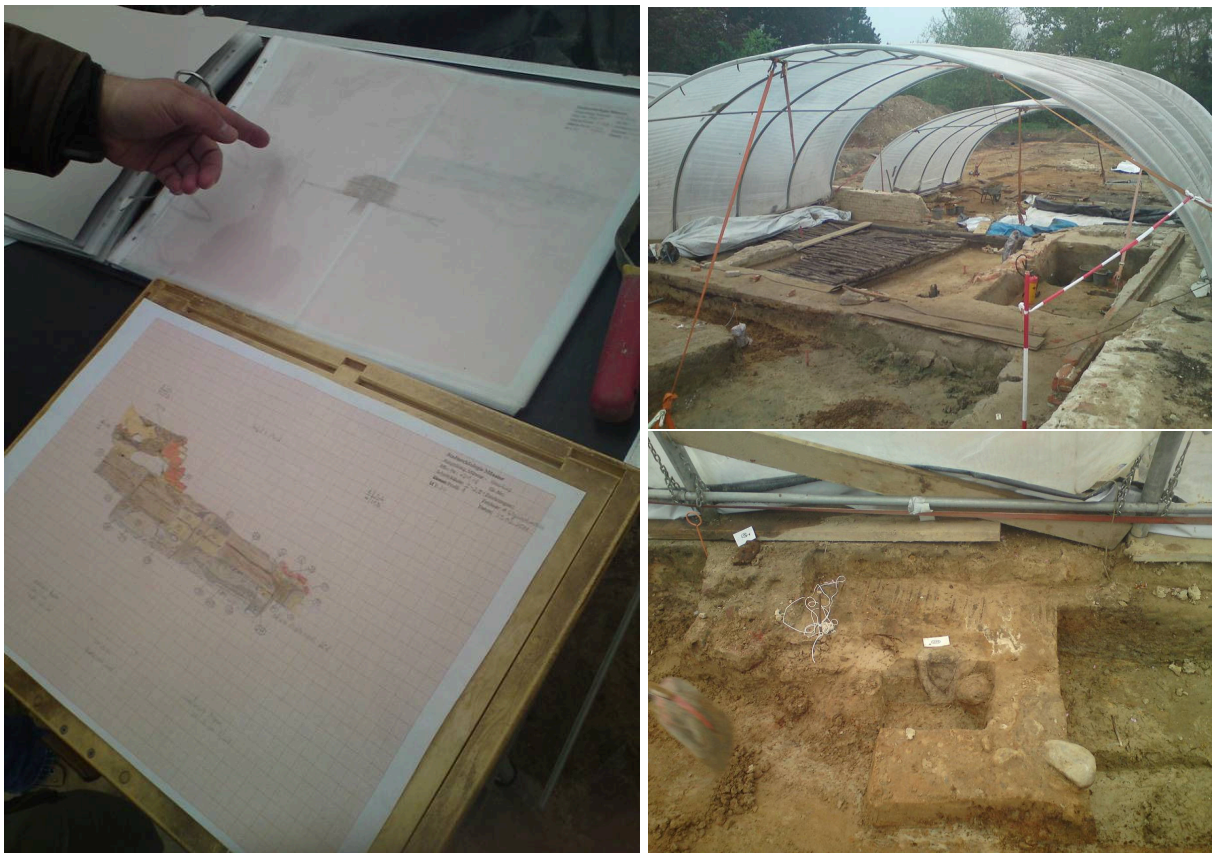


Figure 6.1: Photos demonstrating the conventional archeological documentation methods at "Haus Kump", a building from the 16th century, which is the oldest storage building in Münster, Germany. Left: Findings on graph paper that were meticulously sketched by an archeologist on-site. Top right: Overview of the excavation site. Bottom right: Numbered excavated artefact.

6.2 Modern Archeology

Archeological artefacts and historical sites have an inestimable value. However, often they have only been recognised after they are damaged or destroyed. An archeological excavation usually leads to the discovery and understanding of archeological remains. In most cases it also leads to the destruction of the context and integrity of those remains. Technological achievements in the last 10 years enable higher accuracies in the documentation and reconstruction process of cultural heritage objects. Moreover, they accelerate this process with regard to economical constraints. These new achievements create at the same time a large amounts of data that is no longer subject to the evaluation of the archeologist. Automatic data acquisition systems completely substitute the archeologist during the acquisition process. Photogrammetry, 3D laser scanning and electronic measuring devices are used in modern archeology. However, the data captured with cameras and 3D laser scanners from the excavation sites often only serve as snapshots or as as-built documentation and reference. Campana and Remondino [44] remark in their publication that "The balance of the load decades of archeological research in the use of 3D

digital documentation/representation in terms of scientific investigation is quite disappointing if we think at the possibility offered nowadays by newly developed software and hardware. The use of 3D images was typically oriented to suggest final reconstructions and not to contribute to scientific interpretation".

3D representations of environments, especially excavation sites are becoming the standard for archeologist. 3D laser scanners are the standard instrument in the field of optical metrology. With the use of 3D modelling systems it is possible to reconstruct cultural heritage sites. Katsianis et al. discuss a formal data model and a complete digital workflow for the documentation of excavation sites in 3D by using Geographic Information System (GIS) [121]. Appolonio et al. propose a framework that uses 3D digital models as a restitution of the real object and as a metaphor for navigation through the data [9]. They describe the procedures, techniques and pipeline used for creating the framework. Their ultimate goal is to use the framework to generate the base elements needed to develop a 3D GIS of a large archeological site. Wüst et al. propose applying the 3D GIS project, namely the Digital Landscape Server (DILAS) to archeology and cultural heritage projects [247]. The original aim of DILAS was the efficient generation, management and visualisation of large 3D landscapes and city models. Rua and Alvito propose the creation of a series of methods and tools for testing and analysing theories and hypotheses for historical scenarios through the use of 3D modelling tools and virtual reality engines [195].

Karmacharya et al. suggest the use of a web platform based on knowledge management and semantic web technologies [120]. It is used to store the data during the excavation process and to manage the knowledge acquired during the identification process of findings. Since different technologies are being used during the excavation, different patterns of data are generated. One of the sources of data is the 3D point clouds obtained through the terrestrial laser scanning process. Point clouds, floor plans, images and archeological notes are collected during the project for the creation of 3D object models. Non-academic archeology generates a huge amount of data in a very short period of time. Thus, the collected data is stored in a repository. Once the data is stored, the process of identification of findings is carried out. Cosmas et al. introduce the "3D Measurement and Virtual Reconstruction of Ancient Lost World of Europe" system (3D MURALE) [51]. This system consists of a set of tools for recording, reconstructing, encoding, visualising and database searching or querying. These tools are linked together by a common database.

The database serves several purposes. It contains information about the pieces. It also serves as a repository for archeologist to help them to classify findings, to prepare restoration and to keep track of statistics. The database is a major gateway to the wider public and to other archeologists. Grabczewski et al. describe the 3D MURALE multimedia database system architecture [94]. The goal of this system is to digitally record, store, restore and visualise archeological findings. Reu et al. suggest the use of image-based 3D modelling methods for the recording, documentation and visualisation of the excavated archeological heritage [190]. This method provides great possibilities to increase the quality of the acquired archeological records from the excavation sites. Gilboa et al. report the development of a computerised automatic system for the illustration of complex archeological objects based on 3D scans of the artefacts [90]. They present methods for the automatic translation of 3D models into 2D or 3D line drawings, to colored images to enhance the visualisation of the artefacts. The primary goal of these methods is to replace the traditional hand drawings. The traditional methods are

very expensive to produce and not accurate enough. The presented system uses the 3D scans of archeological artefacts to produce images for better visualisation of the artefacts. Kuzminsky and Gardiner propose the use of 3D laser scanners to offer a sophisticated method for documentation and study of prehistoric human skeletons [137]. The method is based on the major benefit that laser scanners offer a cost-effective method of creating a digital record of skeletal collections for museum archives. The 3D imaging methods are enabling scientists to expand and improve their research. They present the use of advanced but affordable 3D scanners for new research in the documentation and preservation of fragile skeletal material.

A lot of work has been done to use computer-aided systems to facilitate the archeological excavation processes. 3D laser scanners are used in several projects for data representation as a snapshot of excavations. 3D models are generated to present findings for museums. However, few systems are presented that allow archeologists to use 3D laser scanners as the main tool for documentation and semantic mapping. Semantic perception, mapping and exploration for intelligent systems has seen a lot of progress recently, especially in computer vision and the robotics community [172]. Segmentation and annotation of natural scenes, e.g., from images or point clouds is one of the topics in semantic perception and mapping. Semantics is the study of meaning and hence semantic perception focuses on interpreting and organising sensor information in a symbolic form. Semantic mapping combines the extraction of semantics with mapping. Actually there is no map without meaning. While a map can be two dimensional or three dimensional, it always stores either raw sensor data, like 3D point clouds, or syntactic elements, like lines or 3D planes, or semantic features, like interpreted objects. Semantic perception, mapping and exploration is also related to scene understanding, cognitive vision, object detection, object recognition and sensor data processing in general. Utilising these methods assist archeologist to use 3D laser scanners to achieve more precise documentation and to produce more precise results in the same form as conventional methods in shorter time.

In the next section an easy-to-use framework is presented. This framework combines several tools and steps to assist the excavation process. The main goal of this framework is the use of 3D scans for documentation and semantic mapping of the excavation. To generate objective data for archeological teams for further studies and analysis. This framework presents a workflow that combines different systems such as a 3D laser scanner, a tool for marking and labelling the findings and a database. Additionally it provides an infrastructure for other tools to be included in the framework.

6.3 Framework

New technologies provide tools and methods to assist archeologist during the excavation process and to generate an objective documentation of the findings. The goal of this section is to present a framework that uses several components to achieve online semantic mapping of the excavation. This framework is used to produce digital data and documentation from excavation sites. Figure 6.2 presents an overview of this framework.

The data generated based on this framework is used for further studies and analysis of the archeological sites. This framework is designed to provide computer aided systems to substitute the conventional methods of the archeological documentation and data analysis process.

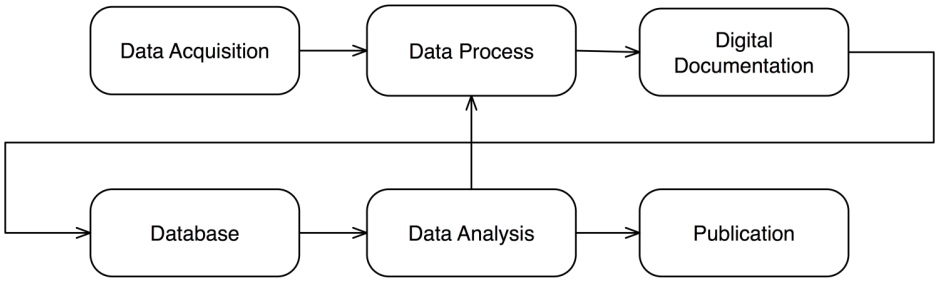


Figure 6.2: Overview of the presented framework. The goal of the framework is to create a standard for computer-aided archeological documentation. The framework provides computer aided systems and tools for assisting archeologist during documentation of excavation sites. Each tool in the chain is exchangeable to comply with the archeologist's need and the available infrastructure.

Conventional methods of documentation include hand drawing of findings and manual measurements of objects. This information is accompanied by comprehensive archeological notes. The quality of gathered data depends to a great degree on the skills and expertise of the archeologist on the site. Therefore, these documents are prone to human error. Furthermore, managing and studying this inadequate data is a challenging process. The conventional method present the excavation site from the perspective of the archeologist on-site. Due to the essence of the excavation, providing new information and extending or modifying the documentation based on conventional methods are demanding tasks. New technologies such as photogrammetry, 3D imaging and 3D laser scanning are vastly used in modern excavations to provide a snapshot of the excavation for publication purposes. However, the captured data has hardly been used for the documentation or analysis process. Modern archeology utilises computer systems mostly for saving the captured data and for managing the generated documentation with databases. This framework presents new ways for semantic mapping of the excavation site. Additionally, this framework creates objective data by capturing the environment in 3D. This provides means for archeologist to study and analyse an excavation based on their own knowledge. This reduces the human inaccuracies and increases the efficiency and speed of the excavation. The framework presents the combination of several tools and incorporates the idea of using computer aided systems in archeology through a standard workflow. Figure 6.2 presents the entire processing pipeline that is undertaken for an excavation. It starts with the data acquisition at the excavation site, continues with the processing and analysis of the captured data and ends with the publication of the results. The individual components are explained in more detail in the following. The idea of this framework is to create a standard for computer-aided archeological documentation. Each tool in the chain is exchangeable to comply with the archeologist's need and the available infrastructure.

Figure 6.3: Illustration of a 3D scan reconstruction of the Imperial Hall of the Würzburg Residence, Würburg, Germany. Data is captured by the mobile robot platform (Irma3D) with a VZ-400 laser scanner by RIEGL Measurement GmbH. Attached to the top of the scanner is a Canon 1000D DSLR camera. After a 3D scan has been acquired the camera is used to acquire color information for the point cloud.

6.3.1 Data Acquisition

One of the main limitations for industrial excavations is the limited time for the excavation. Data acquisition in conventional archeology includes hand drawings of the findings and sometime the whole excavation area. This requires an expert to manually measure the findings and to draw a scaled version of the finding on a graph paper. Advances in 3D laser scanning systems reduce the required time for capturing the data from an excavation site. At the same time these systems improve the quality of the captured data. 3D scans withdraw the human inaccuracies in drawing and provide realistic representation of the findings and the excavation site. Modern laser scanners are capable of capturing up to 1 million points per second. This provides a very detailed 3D point cloud of the environment. 3D point clouds and 3D models provide a better representation of the environment for analysis and the documentation. 3D point clouds present the environment in a spherical coordinate system. Measured points are represented by the polar coordinates and the measured distance of the points. Modern scanners provide the reflected energy of the measured surface. The combination of 3D point clouds with color images will produce colorful scans. Figure 6.3 presents a reconstruction of a sample colored point cloud. Using 3D scanners and cameras provide more detailed information than the conventional hand drawings. Furthermore, these technologies reduce the time and cost by creating objective data from the excavation site.

6.3.2 Data Process

All the captured data in the form of the images and 3D point clouds are processed and categorised in different projects. Many scan formats from different laser scanners and different images can be combined into one project to provide more detailed information of the excavation. Captured data can be modified, e.g., by removing unnecessary parts of the 3D point cloud. This makes processing the data of a project easier and faster for adding documentation and further studies. As shown in Figure 6.2 after data analysis, archeologist are able to access the raw data to check their analysis or to modify the project for further study based on the new entry of data. The organisation of an excavation in such a project improves the data management and collaboration between archeological teams.

6.3.3 Digital Documentation

The primary motivation of computer aided systems for archeology is to produce new ways of recording, cataloging, encoding and visualising archeological artefacts and monuments. 3D technologies have produced portable and fast systems to record large amounts of precise 3D data efficiently. Captured data from the excavation sites are used for digital documentation of the excavation. Computer aided systems are used to substitute the conventional methods of documentation. These systems provide tools for archeologist to visualise the captured data in 3D on-site. They provide simple tools for scan registration, measurement, region of interest selection, object removal, semantic mapping, georeferencing and adding archeological notes. This chapter proposes the use of a Computer Aided SysTem for Labelling archeological Excavations in 3D (CASTLE3D) software (cf. Section 6.4) for digital documentation component of this framework. This computer aided system is designed for on-site semantic mapping and generation of digital documentation of excavation sites. This system improves the documentation process with possibility of accessing the original data at all times. Archeologist are able to add, modify or remove the documentation and generate the necessary information based on their knowledge and understanding from the original captured data from an excavation site. Digital documentation increases the speed of the documentation process of an excavation and leads to cost savings for the project. Additionally digital documents provide the possibility for sharing and collaboration between archeologists.

6.3.4 Database

Management of the generated data from the archeological groups on-site or analysis teams off-site is problematic for conventional methods. The conventional methods of data storage and cataloging simply cannot cope with the amount and speed of data generation. Data management is widely used in archeological projects to store and retrieve the generated information during and after an excavation. In the framework presented in this chapter the database Adiuvabit [59] is used as the back-end of the framework for data management. The aim of this framework is to provide seamless connection between the digital documentation process and the database. The use of standardised XML schema to transfer the data from the documentation process to the database makes the framework modular. The aforementioned computer aided system for semantic mapping (CASTLE3D) implements the export functionality to prepare the generated

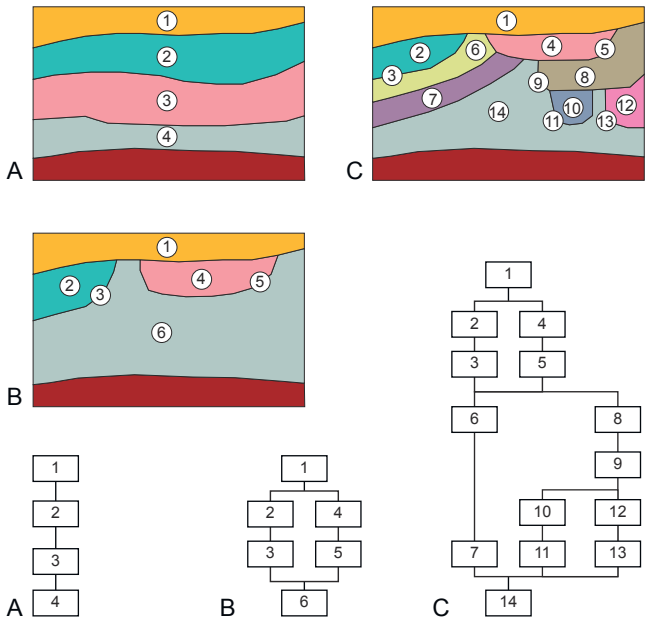


Figure 6.4: The Harris matrix reflects the relative positions and stratigraphic contacts between stratigraphical units [69].

documents for transfer to the database. Adiuvabit as the back-end of the framework implements the capability of importing the XML files and automatic generation of the database fields and tables. The use of a standardised XML schema in the framework provides the possibility to substitute the database with the existing database of any archeological project.

6.3.5 Data Analysis

It is essential to analyse and study the captured data after the excavation. This process produces knowledge about the environment and the era of the excavation site. During the excavation process images, floor plans and other data such as archeological notes are collected. Additionally the environment is captures in 3D and digital documentation is generated based on the 3D point cloud. These data are of a great value in the analysis of archeological findings in any archeological project. Computer Aided Models are used to study the stratigraphy of the excavation. CAD models are used to generate the 3D model of the excavation site. These models represent the excavated findings in their original structure. They reveal the purpose and functionality of the findings. Methods are available to verify the interpretation of the findings on-site. The Harris matrix is a tool to represent the stratigraphic units and the relations between different layers and findings [104]. This matrix reflects the relative position and stratigraphic contacts of observable stratigraphic units. In general, one can assume that the recent layers lie above the rest of the layers and the earliest at the bottom. The matrix is constructed by representing the connection of these layers. The lines represent the direct stratigraphic contact (cf. Figure 6.4). This graph representation of layers assists in informing the archeologist on the physical process

of site formation and to highlight areas where loops on recorded sequence occur. These loops in the matrix produce temporal anomalies. These anomalies present that in a sequence of contacts, the earliest in the sequence appear to be later than the latest contact. This happens due to the error during excavation or recording. During urban excavations, thousands of results will be produced, so the on-site consistency check is vital. Regular, e.g., daily scans of the excavation site enable the compilation of the Harris matrix. This matrix is generated automatically or semi-automatically with the help of an archeologist. The framework presented in this chapter provides a tool for compilation of the matrix as it uses the database Adiuvabit [59] as the back-end for data management. This matrix can be regenerated after the excavation from the original data. This improves the objectivity of the analysis of the findings. Utilising computer aided system and generating the digital data and analysis provide the basis for data sharing between archeological teams. Excavation sites can be studied by the archeological community and they can compile a new Harris Matrix based on their understanding from the original scans and captured data.

6.3.6 Publication

The last step of any study, especially of the archeological excavation and analysis is the publication of the findings. Computer aided systems and all the captured data in the form of 3D point clouds, images, archeological notes, semantic mappings and all generated digital documentation in the database provide a huge amount of data for archeologist to publish their findings. Furthermore, availability of the original data, XML documents generated during the documentation process with CASTLE3D, CAD and 3D models generated for finding on the excavation site and the analysis of data create a possibility for other archeologists for further study and comparison of their analysis. Computer systems make the publication, analysis, comparison and modification of the data very simple. Additionally, they improve the quality and the quantity of the publication in archeology. The digital basis of the generated data provides the infrastructure for instantaneous collaboration. Furthermore, the data can be shared among the archeological teams.

6.4 The Computer Aided System for Labelling Archeological Excavations in 3D

Computer aided systems introduce new ways of recording, cataloging and visualising archeological data. Data management systems are used in archeology to store and retrieve the generated data during and after the excavation process. Digital documentation is the most important part within the proposed framework in this chapter. Despite the advance of technology in modern archeology no satisfactory solution has been presented for digital documentation of the excavation process. For this purpose this section proposes the CASTLE3D (Computer Aided SysTem for Labelling archeological Excavations in 3D) software to create digital documentation of the excavation site. This software is developed as a possible component for the digital documentation component of the framework proposed by this chapter. Figure 6.5 presents the overview of this software. CASTLE3D provides an easy to use set of tools for archeologist to improve the

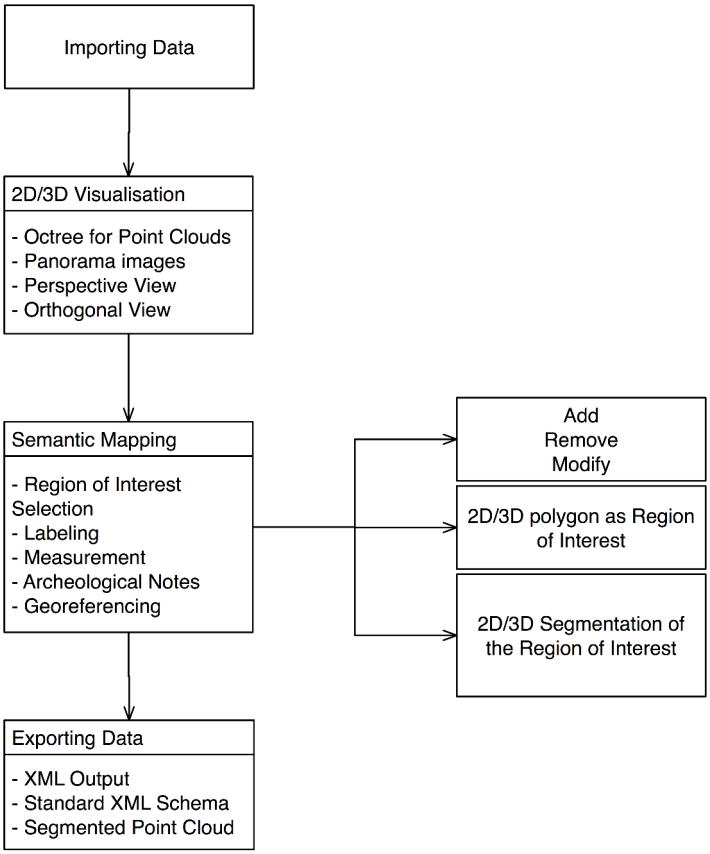


Figure 6.5: Overview of the presented Computer Aided SysTem for Labelling archeological Excavations in 3D (CASTLE3D). The software provides an easy to use set of tools for archeologist to improve the documentation process of the excavation sites. The aim of the software is to assist the archeological teams on-site to include semantics to the captured point clouds on-site. The software provides tools for importing the point clouds by supporting several point cloud types. It provides 2D/3D visualisation tools for archeologist. The core functionality of the software is to provide semantic mapping methods for on-site documentation. Finally it provides the means for exporting the generated data in to other systems and for publication.

documentation process of excavation sites. It provides tools for importing the captured data from an excavation directly from the scanning unit to the software. The visualisation tool provides different views of the captured point cloud. This assist the archeologist to have a better understanding of the excavation area. Additionally it provides access to the original data for remote archeologist. The on-site semantic mapping is available to the archeological team on-site to generate documentation of the excavation during the excavation process. All the generated data in the CASTLE3D can be exported for further study and analysis to a database system.

6.4.1 Importing Data

CASTLE3D provides a tool for importing the captured point clouds from the excavation site. The system uses the internal data structure for handling the data and for visualisation purposes. Therefore, different formats of point clouds are converted in to the internal data structure during the import process. Scanners have different methods to represent the captured data such as reflected value and color from the measured points.

CASTLE3D provides different reflectance normalisation methods for generic visualisation of different scan formats. Additionally, there are several standards and file formats for encoding the point clouds. CASTLE3D supports many different scan formats, amongst them the file formats from known laser scanner producers such as Riegl (rxp), Z+F (zfs) and FARO (fls), or standard ASCII file formats (xyz). This tool supports scans with reflectance and color information with different scales. Normalisation, scan scale and point values are selected during the import process and the scan information is saved in the project. Imported scans are available for visualisation and semantic mapping process.

6.4.2 Visualisation

Recently [73] introduced a data structure to improve the visualisation of point clouds with millions of points. A fast and easy to use visualisation tool is integrated into the CASTLE3D to visualise the point clouds in a matter of seconds (cf. Figure 6.6). This is achieved as the result of the octree data structure that allows for real time point density reduction. Depending on the hardware the system can be configured to visualise only a subset of points to retain 30 frame per second for fast and smooth visualisation. This improves the interaction of archeologist with the recorded data of an excavation site. Nonacademic archeological sites are often only available for a very short period of time. Therefore, It is essential for the excavation process the be very efficient. Recording data with 3D laser scanners will provide detailed data of the excavation site. Fast and efficient visualisation tools assist archeologist on-site to inspect the completeness of the captured data. Reconstruction and visualisation of the data after the actual excavation assists archeologist in studying the site more precisely. It also produces an infrastructure to capture the original data from the excavation for other archeologist to study the site based on their own knowledge.

Modern terrestrial 3D scanning systems acquire data at an impressive rate. Apart from range information, many laser scanners capture reflectance and color information. To handle the large amounts of data from laser scanners efficiently, we need special data structures for 3D point cloud. The CASTLE3D uses two efficient data structures for 3D point clouds: an octree and a panorama image array. Elseberg et al. [73] describe a spatial data structure called octree with a low memory consumption. The octree and panorama data structure store the range information as well as reflectance or color information from the laser scanners as explained in Chapter 2. The 2D representation of a 3D point cloud permits the use of conventional 2D algorithms for acquired 3D data. CASTLE3D uses the range information to produce panorama images of scans (cf. Figure 6.7).

All pixels of the image contain color, intensity and a list of 3D points, that fall into the array element. A panorama image offers a full view of the excavation site in one image and

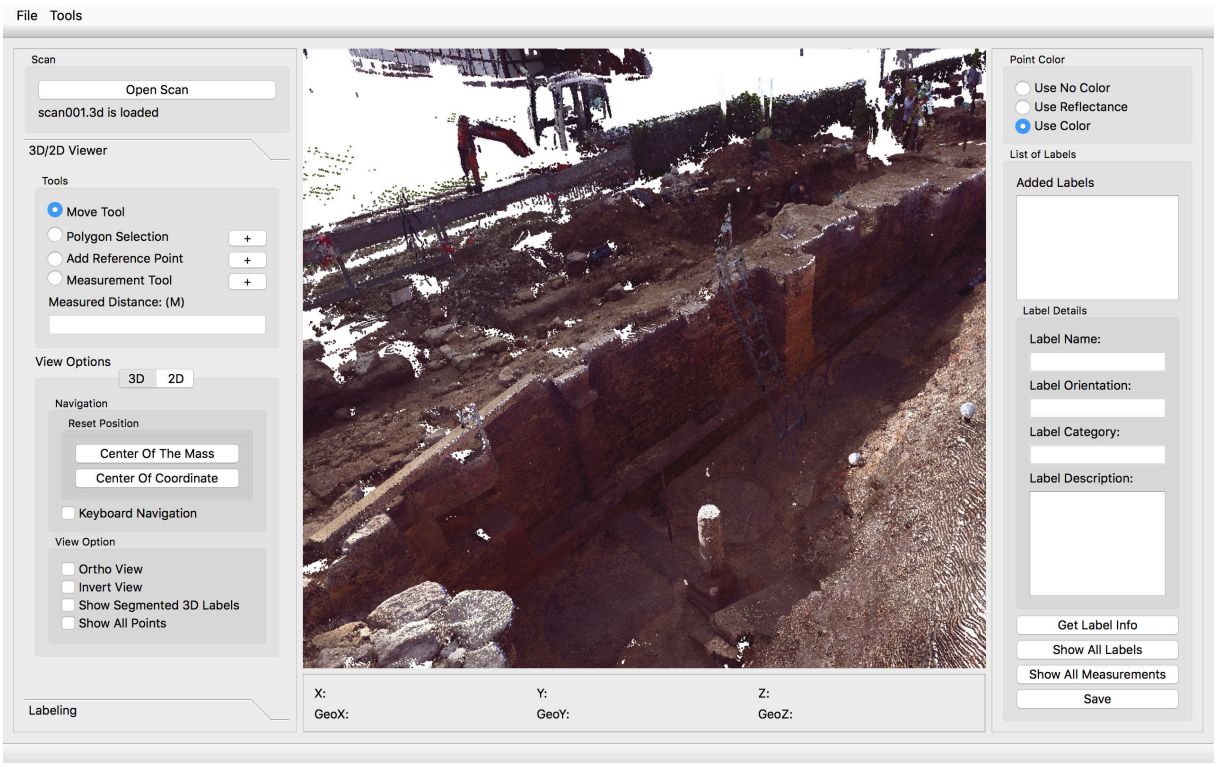


Figure 6.6: Visualisation of a 3D point cloud. Data captured from an excavation site in Germany

Figure 6.7: Panorama image of an excavation site.

provides a better understanding of the environment. The visualisation of point clouds provides a detailed representation of the excavation site. This is a realistic presentation of the excavation. In comparison to conventional methods, where hand drawings of the findings are used for further study, the 3D point cloud is an extremely accurate and precise model of the actual excavation.

6.4.3 Semantic Mapping

Archeological observations are commonly recorded on-site using conventional methods (cf. Figure 6.8). The outlines of the findings are drawn on graph paper with detailed information based on subjective understanding of the findings. Images, floor plans and other data such as archeological notes and measurements are collected during the project. These data are of great value in analysis of archeological findings in any project. However, acquisition of these

Figure 6.8: Conventional methods to record excavation sites include, hand drawings, manual measurement and archaeological notes.

data with conventional methods is a very time consuming and demanding task. Due to lack of availability of the excavation site after the data acquisition process any error or insufficiency in the captured data is not retrievable. 3D data acquisition and computer aided systems will improve this process. They are less prone to human error and the captured data is objective. In regard to 3D data acquisition the entire environment is captured, therefore the original data is always accessible. In CASTLE3D a series of labelling tools assists archeologists in documenting their findings on-site for further inspections. These tools facilitate the creation of several labels on visualised data by selecting regions of interest. Afterwards, selected regions are available as polygons in both 2D and 3D representations of the scan (cf. Figure 6.9).

Labels are connected with one or more regions of interest. Additionally, labels contain the measurements of the findings that are generated from the point cloud, georeferencing information for the label, direction and archeological notes. Multiple coloring tools are available to group similar findings and labels. A series of predefined categories are available for semantic mapping of the label. These categories are included in each project during the creation process as an XML file. The categories are modified for each project to contain the required level of information for the excavation. Additionally, these regions of interest can be segmented from the rest of the point cloud. The segmented region represents findings, such as monuments, buildings, stones and other structures.

Figure 6.9: Left: 2D presentation of a selected region of interest in a point cloud. Right: 3D presentation of a selected region of interest in a point cloud.

Georeferencing methods are available to transform the point cloud in to the global coordinate system. Point cloud registration methods are used to merge scans of the findings from different viewing angles. Utilising the registration information the segmented findings are merged to produce the full 3D model of the findings. Figure 6.9 (left) illustrates a panorama image with the marked region of interest presented in the 2D viewer of the CASTLE3D software. Additionally, Figure 6.9 (right) presents the 3D segmented region of interest encoded with green color in the 3D viewer of the CASTLE3D. A measurement tool is used to add more data for each region of interest and label. Due to the high accuracy of modern laser scanners the calculated distances based on point clouds are more precise than the conventional methods of manual measurements on-site. A detailed description of the finding is added to each label. The orientation of the findings provides more information regarding the state of the finding with respect to other labels and the excavation site. This computer system enables the identification of the findings on-site. It also provides feasible tools to study and observe the excavation site and findings both for archeologist on the site and for public.

6.4.4 Exporting Data

Documentation is essential to archaeological analysis. Therefore, it is crucial to have a proper amount of documentation for each finding. large amounts of data increase the necessity of a proper data management systems. The generated data is based on the archaeologists understanding and experience. Since the data has been acquired by terrestrial laser scanners and represented by computer aided systems, they can be studied in the archaeology community. The data can be used for further studies, catalogues, archaeological databases, and archaeological data management systems. CASTLE3D offers an Extensible Markup Language (XML) file format to export the generated data and documentation of findings. This is essential for multi-platform communication. The XML file can be easily imported into other software. For this purpose the data is transferred to an Adiuvabit database. Further documentation is included in the database for each finding. Analysis and publication of data is carried out based

Figure 6.10: Aerial view of the large excavation site in Ganderkesse, Germany.

on the database. In addition to the XML file for transferring the documentation, georeferencing information, selected region of interest and measurement information, segmentation of findings are available. This produces small point clouds containing only the findings with complete information prepared for publication.

6.5 Case Study

This section presents a workflow of the presented framework on an excavation site in Ganderkesee, Germany. Figure 6.10 shows an aerial view of the excavation site. After the deduction of the level of the topsoil to the relevant layer by an excavator potential records are flagged by an archaeologist. The flagged regions are cleaned by an excavation team consisting of a minimum of two people. This reduces the level to the first planum presenting the potential findings clean and free of crumbs (cf. Figure 6.11). The excavation specialist documents the archaeological relevant findings. The potential findings are sequentially numbered.

The numbered findings are photographed immediately after surface cleaning with the accompanying photo panel, scale and north arrow (cf. Figure 6.11). 3D scans are captured for visualisation purposes. Afterwards the cutting direction of the profile is determined with at least two nails and the limits of the findings are marked. A database record with an initial descrip-

Figure 6.11: Conventional methods to record excavation sites. Top left: Potential records are flagged by an archeologist for further process. Bottom left: The flagged regions are cleaned to reduce the level to the first planum presenting the potential finding clean and free of crumbs. Top right: Findings are numbered and photographed with photo panel, scale and north arrow. Bottom right: The marked limits of the finding is measured with a tachymeter or a GPS.

Figure 6.12: Profile view of a potential findings on Ganderkesse excavation site in Germany.

tion is generated. The marked limits and cutting directions are measured with a tachymeter or a GPS (cf.Figure 6.11). The profile is created on the measured records by removing a verti-

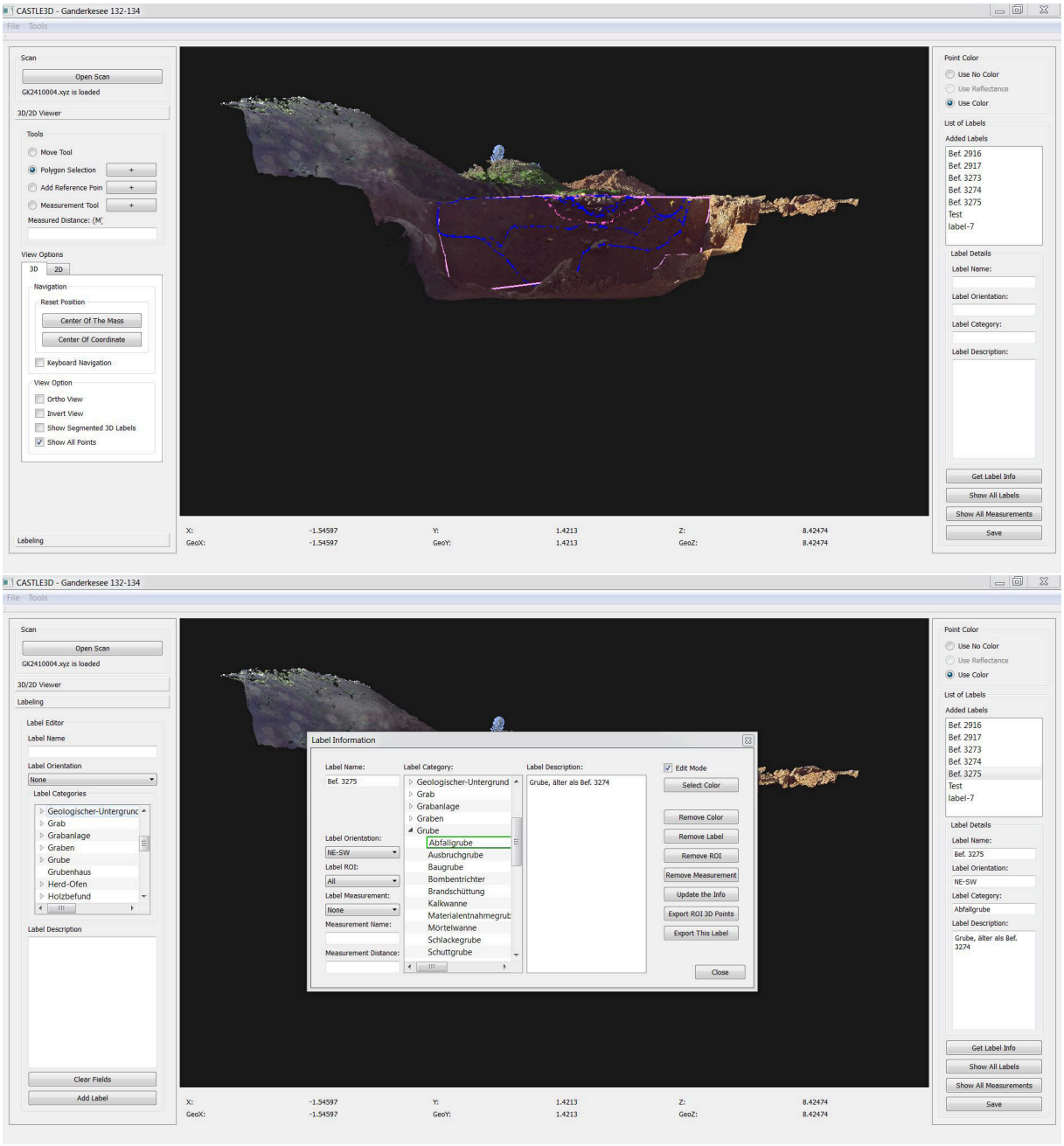


Figure 6.13: Presentation of CASTLE3D with added labels and selected regions of interests.

cal box of soil in longitudinal direction of the marked cutting nails. Afterwards the front half of the profile is removed to salvage the findings. Figure 6.12 presents samples of profile view of findings. Profiles with archaeological artefacts are prepared for further documentation by an archaeologist. The conventional documentation of the profile consist of image acquisition

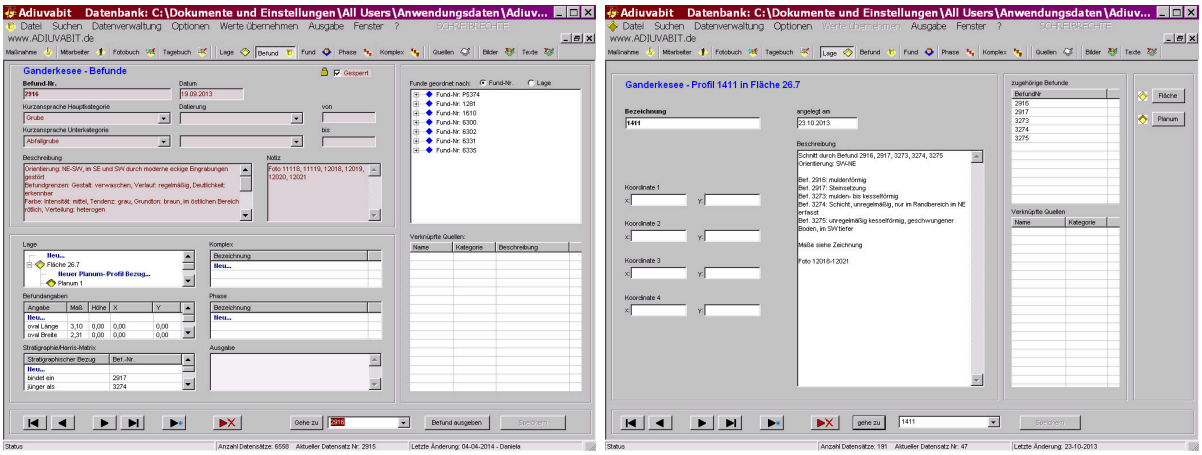


Figure 6.14: Generated record for a finding in Adiuvabit database.

from the numbered profiles, scaled hand drawings, manual measurements and archaeological notes. Some archaeological projects contain a database and the excavation team transfers all the documentation from the site to the database. However, this process is insufficient and time consuming.

It takes days for archaeologist to combine all the gathered data and to transfer it to the database. With the use of the framework presented in this chapter this process could be reduced to hours of work. This framework recommends tools and computer aided systems to improve and accelerate the documentation process. In this framework the excavation process continues to retrieve the rest of the profile and prepare findings for 3D scanning. After the data acquisition process scans are imported in to suitable projects. Visualised scans form the basis for interactive tools for archaeologist to mark the findings simply by selecting a polygon containing the finding. The selected regions of interest are labeled and additional documentation, orientation and accurate measurements are included in the CASTLE3D project (cf. Figure 6.13). The archaeologist exports the generated data from the CASTLE3D into the existing Adiuvabit database for the project. Database records are generated or merged automatically for the marked findings (cf. Figure 6.14). Profile views of the selected regions of interest are generated from the database.

This framework provides a comparable view to the conventional hand drawings of the findings. Figure 6.15 presents a comparison of the conventional hand drawings with the generated data from CASTLE3D. Additionally, 3D point clouds, 2D images, segmented findings and all the extra documentation provide higher quality of the representation of the excavation. This improves the documentation and reduces the required time for data generation and organisation. The available multimodal representation of the data in this framework facilitates the analysis of the findings. Further studies are carried out on the gathered documentation and results are published. The digital nature of this framework makes the analysis and the original data available instantly to other archaeologist for further study.

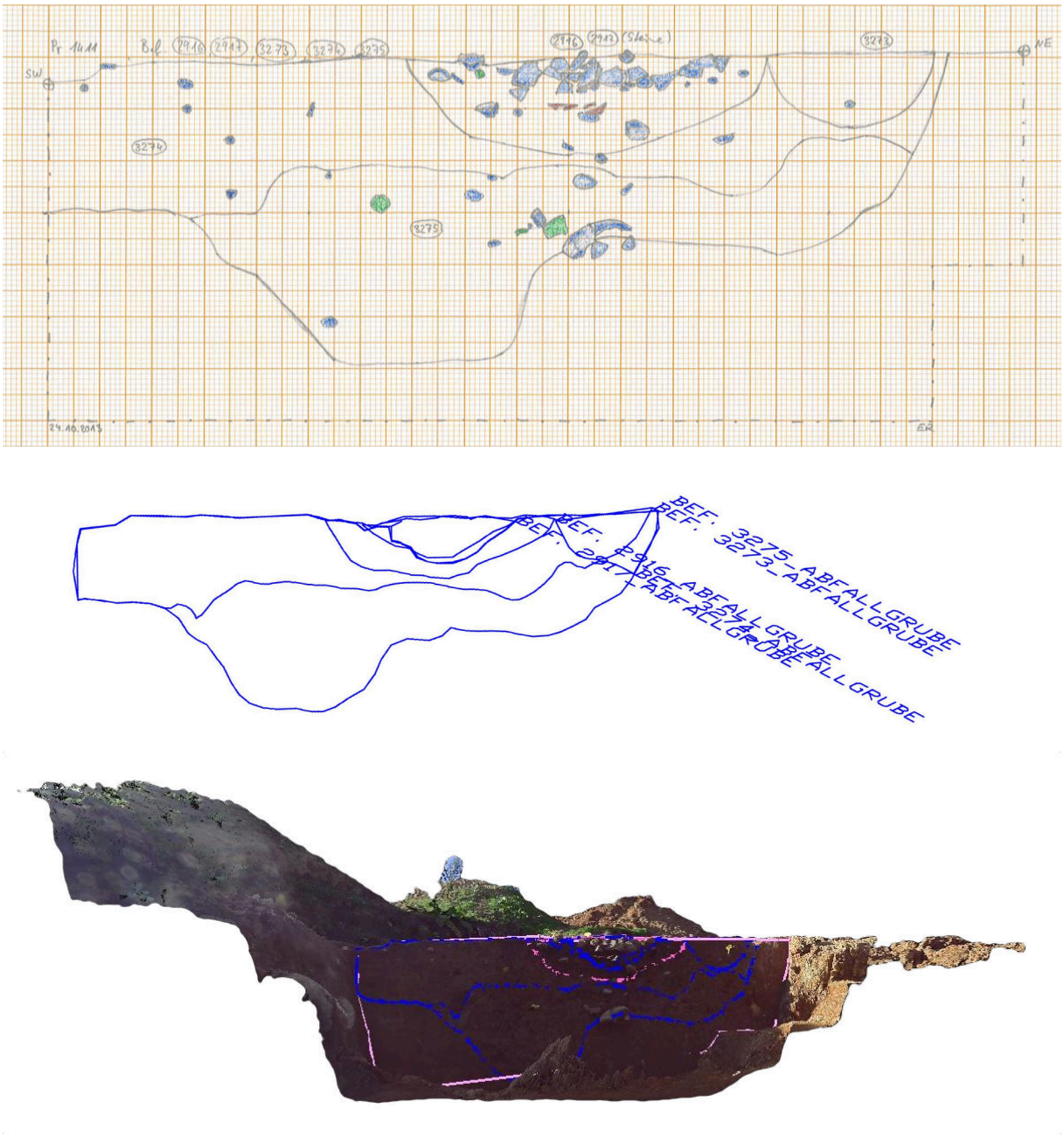


Figure 6.15: Top: Conventional hand drawing presentation of the excavation. Middle: 2D CAD model of the excavation based on the digital documentation. Bottom: Marked region of interest on a 3D point cloud of an excavation.

6.6 Summary

This chapter presents a framework to improve the process of archaeological excavation and documentation. The aim of this framework is to assist the archaeologist during excavation and to substitute the conventional methods of documentation. The generated data in this framework combines 2D images, 3D point clouds, archaeological notes and CAD models to present the excavation. 3D scanners are capable of data acquisition with a very high rate. 3D technologies have produced portable and fast systems to record large amounts of precise 3D data efficiently. This chapter presents a set of hardware and software to facilitate the data acquisition of the archaeological excavation. CASTLE3D (A Computer Aided SysTem for Labelling archaeological Excavations in 3D) is a novel tool for digital documentation. In combination with Adiuvabit as the back-end database of the framework. They are used to execute the workflow of the framework. The generated data in CASTLE3D is exported as standard XML format to the database. This chapter shows the combination of CASTLE3D and Adiuvabit in a case study at a sample excavation site. The framework is able to create similar documentation results as with conventional methods but more efficiently and with higher quality. The availability of the original data helps to achieve objective integrity even with destructive methods as new studies and analysis can be carried out over the original data. The achievement in technology provides tools from several disciplines such as, surveying, computer science, laser scanning and 3D modelling to assist archaeologists during an excavation.

Chapter 7

Application: Towards A Fully Autonomous Indoor Mobile Mapping System

The industrial manufacturing world is one of the largest industries utilising robotics. Robot arms or manipulators are bolted to a specific position in the assembly line. They are capable of fast and high accuracy movements to perform repetitive tasks with more precision and faster than a human operator. The main drawback of the manipulators is their lack of mobility. These systems have a limited range of motion. In contrast mobile robotics provide a wide range of motion for the robot. This is achieved by the ability of the robot to travel through the manufacturing plant and to accomplish its task. Mobile robots are used in rough environments, such as surface of other planets. Additionally they are used in environments that are dangerous for a human operator, such as nuclear facilities. Another use case for mobile robots is in collaboration with humans in an environment such as museums. All these platforms are controlled by a human operator or the system provides semi or full autonomy. Even teleoperated systems that are controlled by humans, the operator performs the localisation and cognition activities and the robot provides the motion control mechanism. Autonomous Guided Vehicles (AGV) are used to autonomously deliver parts, food and medication in different scenarios. Additionally they are used to perform tasks such as cleaning of large buildings. The combination of these mobile robots with vision sensors produces a mobile mapping system that captures the environment in 2D or 3D. A fully autonomous indoor mobile mapping system requires to be able to percept the environment, localise itself in it and generate the map of the environment, plan the next goal and a path to the goal from the current position and finally execute path following with obstacle avoidance to reach the goal. This chapter presents an application towards a fully autonomous indoor mobile mapping system. The main focus of the chapter is to present a novel sensor placement planning algorithm with a mobile robot capable of path following with obstacle avoidance system.

7.1 Mobile Robotics

Mobile robots require a locomotion mechanism that enables the system to move in the environment. In research there are a large variety of locomotion mechanism used for mobile robots. There are platforms that can walk, jump, slide, swim, fly and roll. Each of these locomotion mechanisms have limitations.

The wheel based robots achieve extremely high efficiency on flat ground. They have a simple mechanical design and are easy to control, as each wheel can only go forward and backwards. However, they are mostly limited to smooth surfaces. On the other hand legged robots have complex kinematics, their locomotion is limited by their leg and body mass and the characteristics of the used motors. They are able to work in fairly rough environments, especially they can overcome stairs and low obstacles. The efficiency of the wheeled locomotion highly depends on environmental qualities. The human environments are usually engineered and consist of flat surfaces especially in indoor environments. Therefore, wheel based mobile robots are preferred due to their simple design. Focus of this chapter is on wheel based mobile robots for indoor mapping [211]. Other than the locomotion mechanism mobile robots can be categorised based on the environment in which the robot moves. This classification includes land robots that are referred to as Unmanned Ground Vehicles (UGV). Wheel based mobile robots are included in this category. Aerial robots that are referred to as Unmanned Aerial Vehicles (UAV) are the second category in this classification. The third category is the underwater robots that are referred to as Autonomous Underwater Vehicles (AUV). Finally the fourth category is the watercraft that is referred to as Unmanned Surface Vehicles (USV). The remainder of this chapter presents the application of wheel based UGVs for mapping indoor environments.

7.2 Wheel Based Mobile Robots

Wheels are the most commonly used locomotion mechanism for mobile robots. These platforms achieve a very high efficiency with relatively simple mechanical implementation. Balance of these systems is guaranteed by minimum three wheels. However, there are mobile robots that provide the sufficient balance with just two wheels. Since the robot balance is not difficult to achieve on these platforms the research focus of wheel based robots are traction, stability and control over the robot. Four basic types of wheels are available for wheel based mobile robots. The first category is the standard wheel that has two degrees of freedom namely the rotation around the wheel axle and the rotation around the contact point of the wheel with the ground plane. Rotation of the wheel can be provided with a motorised system. Second category of wheels is similar to the standard wheel. However, their rotation is around an offset steering joint. This group of wheels is called castor wheels. Third category is referred to as Swedish wheels. They have three degrees of freedom namely the rotation around the wheel axle, rotation around the rollers on the wheel and the rotation around the contact point of the wheel on the surface. Similar to standard wheels the rotation around the wheel axle can be motorised. The fourth category is ball or spherical wheels. The sphere has a total 360° freedom and similar to castor wheels these wheels are mostly used to provide balance for the robot. These wheel categories differ widely based on their kinematics. Therefore, the choice of wheel type for the robot has a large effect on the kinematics of the mobile robot. The standard and castor wheels are similar

by having primary axis of rotation and are thus highly directional. To move in any direction the wheel must rotate around the vertical axis to face the direction of the movement. The difference between these two wheels is the center of the rotation. It passes through the contact point for the standard wheel whereas the castor wheel rotates around an offset axis. The Swedish and spherical wheels are designed with omnidirectional movement in mind. Swedish wheels have their primary axis to serve as only actively powered joint. However, the passive small rollers attached around the circumference of the wheel provide low resistance for movements in other directions than forward and backward movement. The spherical wheels are the real omnidirectional design. They can be powered actively to spin along any direction. Regardless of wheel type, mobile robots with more than three wheels require a suspension system to maintain wheel contact with the ground. The simplest approach to suspension is to provide flexibility into the wheel itself. For instance four wheeled indoor robots similar to the robot used in the experiments of this chapter have two active wheels in front and two castor wheels in the back to provide the balance. A deformable tire of soft rubber is used for castor wheels to provide a primitive suspension. This is a limited solution and mostly used for indoor robots that move through mostly flat environments.

Stability, controllability and manoeuvrability of a mobile robot platform is strongly linked to its wheel configuration. Several wheel configurations exist for mobile robots to maximises the aforementioned characteristics of the robot in a variety of environments. A minimum number of two wheels are required for static stability. A two wheeled robot can achieve static stability only if the center of the mass is below the wheel axle. However, a more common wheel configuration for indoor mobile robots utilises three or four wheel for more stability. In this case the center of gravity must be contained in triangle or rectangle formed by the contact points of the wheels with the ground. One of the common configuration for indoor mobile robots consist of three wheels. Two independently driven wheels in front or back and one unpowered omnidirectional wheel to provide the third contact point for stability (cf. Figure 7.1). To even further improve the stability and to increase the payload a configuration of four wheels are used. Two independently driven wheels in front or back and two omnidirectional wheels (cf. Figure 7.1). Many other configurations are used for different environments such as use of steered wheels, motorised steered wheels, four motorised castor wheels, three motorised Swedish wheels arranged in a triangle, three synchronously motorised and steerable wheels and many more.

Omnidirectional robots have the highest manoeuvrability. This indicates that the robot can drive in any direction on the ground plane (x, y) regardless of the robot orientation. To achieve omnidirectional manoeuvrability the mobile robots use wheels with the capability of movements in any direction such as powered Swedish or spherical wheels. Another class of robots are designed to achieve high manoeuvrability by an initial rotational motion of the robot. This makes motion in a particular direction possible for the robot. The popular configuration of such a robot requires differential drive wheels where the robot can rotate around the center point of the chassis. The ackermann steering configuration that is common in automobiles has a turning diameter larger than the size of the chassis. Sideway movement for this configuration requires repetitive steering and forward and backward motions. This directional and steering geometry provides a very good lateral stability in high speed turns, hence the use of this system for cars.

The controllability complexity of a robot system increases with higher manoeuvrability. Omnidirectional designs usually have greater degrees of freedom at the wheel. Swedish wheels

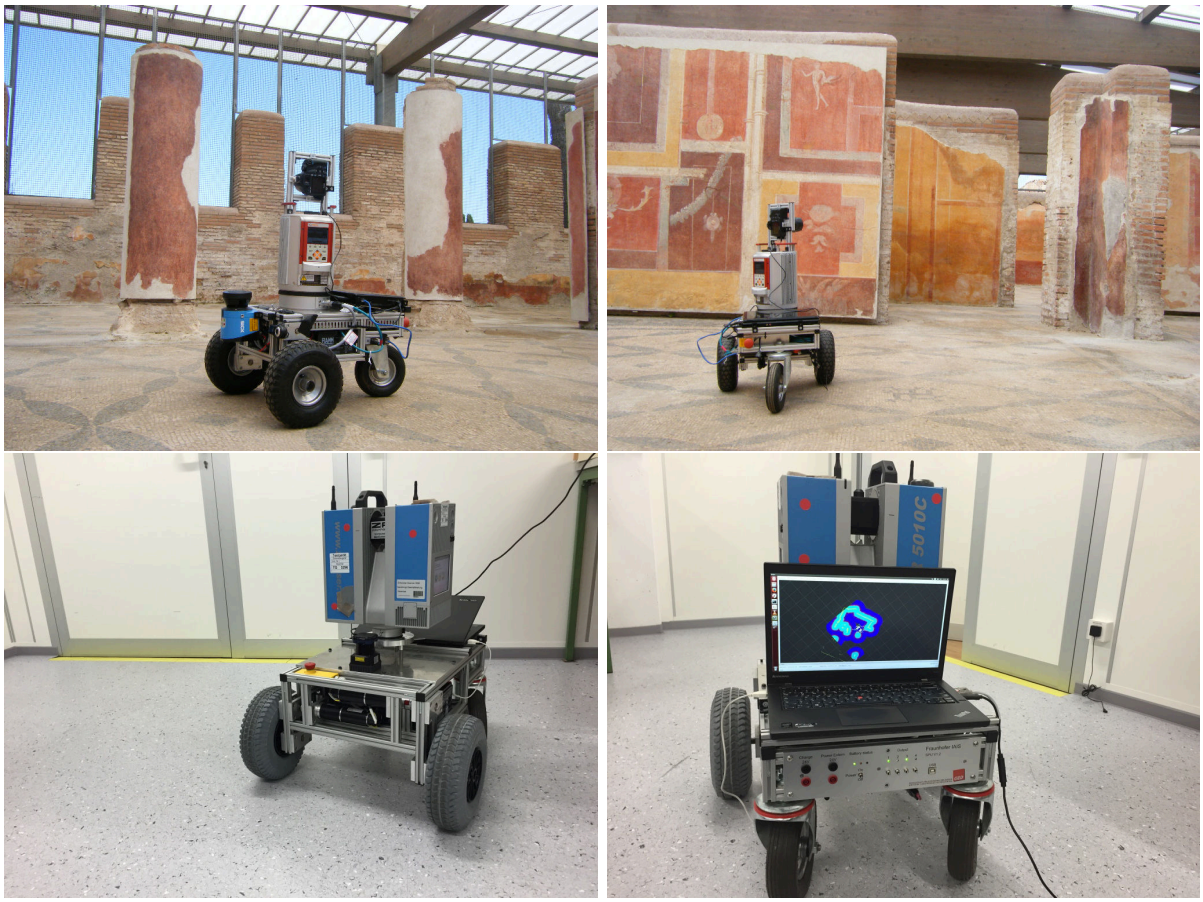


Figure 7.1: These robots are equipped with a differential drive system. Two motorised wheels in the front and one or two castor wheels in the back for stability. Both robots are equipped with 2D/3D laser scanners. Top: Irma3D the three wheeled mobile robot. An additional camera is mounted on the scanner to provide color information for the measured points. Bottom: ZF-Volksbot the four wheeled mobile robot. This configuration allows for higher payloads than the three wheeled configuration.

have a set of free rollers along wheel surface to provide its omnidirectional manoeuvrability. In addition to the increase in the complexity of the system, this causes the accumulation of slippage and reduces the accuracy of the positioning of the robot using the previously determined positions. This process is know as dead reckoning. Controlling an omnidirectional robot to follow a straight line is considerably harder than with the Ackermann design that can easily be achieved by locking the steering wheels and driving the robot. On the other hand differential drive robots have to control the motors attached to wheels to achieve exactly same velocity profile to drive on a straight line. This is challenging considering the differences between motors, wheels and environmental factors. Therefore, there is no general solution that simultaneously maximises stability, manoeuvrability and controllability of the mobile robot. This chapter focuses on three/four wheel differential drive robots with two motorised wheels in front and one or two castor wheels in the back of the robot chassis [211].

7.2.1 Differential Drive Robot Kinematics

Kinematics is the study of the mechanical system of a robot that is concerned with the motion of the robot without reference to the forces which cause these motions. The kinematics of a mobile robot are essential to the design of the control system of the robot. The robot kinematics are also essential for robot manipulators. These robot systems often have more moving joints than simple mobile robots. Hence the increase of complexity of the kinematics of such manipulators. In the field of manipulator robotics there has been extensive research for complete understanding of the kinematics of these robots [52, 205]. The control system and the kinematics of a mobile robot define the possible trajectory of the robot in its work space. In contrast with manipulators that are attached to a fix position, mobile robots have the ability to move as a self contained system in the environment. This increases the complexity of mobile robot pose estimation. The system must incrementally integrate the motion of the robot for pose estimation. Similarly the inaccuracies in motion estimation accumulate over time. Therefore the exact pose estimation of mobile robots is an extremely difficult task. Many mobile robots use a drive system known as differential drive. This system usually consists of two drive wheels mounted on a common axis and additional one or two omnidirectional wheels to provide more stability for the system. The control system of each drive wheel is independent and they can be driven either forwards or backwards. The contribution of each wheel defines the motion of the robot. Various velocities for each wheel enable the robot to perform different rolling motions. Wheels impose constraints on the robot motion and manoeuvrability of the robot. The wheels are attached to the robot chassis therefore the combined constraints of the wheels define the overall motion constraints of the robot.

In this chapter the notation and terminology presented in [45] is used to define the kinematics of a differential drive mobile robot. The robot is modelled as a rigid chassis on wheels. The operational environment of the robot is simplified to a horizontal plane. The robot has three degrees of freedom namely the (x, y) position on the horizontal plane and the orientation along the orthogonal axis to the plane. Additional degrees of freedom and flexibility of the robot due to the wheel axles, steering wheels, castor wheels are ignored for simplification. The robot position is defined based on the relationship between the local robot coordinate system and the global coordinate system of the plane. Figure 7.2 represents this relationship.

The position of the robot is defined by the point P on robot chassis. Additionally this point presents the local coordinate system of the robot. The point P is defined with (x, y) in the global coordinate system and the rotation is defined by the angular difference between the local and the global coordinate systems. Therefore, a robot pose is defined as a vector with three elements that describes the pose in the global coordinate system.

$$Pose_G = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}. \tag{7.1}$$

To describe the robot movement given its geometry and speed of wheel, the robot motion along the axis of the global coordinates is mapped to motion along the axis of the robots

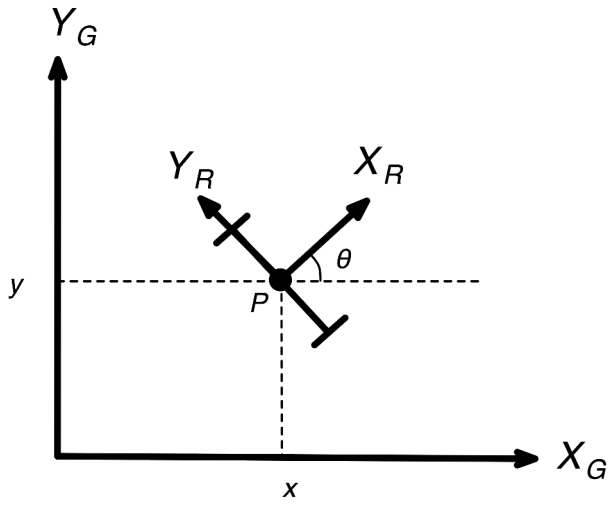


Figure 7.2: The robot modelled as a rigid chassis on wheels. The operation environment is simplified to a horizontal plane. X_G and Y_G define the global coordinate system and X_r and Y_r the robot coordinate system at point P, respectively. The point P represents the position of the robot in the global coordinate system. The robot is driving in the direction of the x-axis. The robot orientation θ is defined by the angular difference between the x-axis of the global and the robot coordinate system defined at point P.

coordinate frame. Orthogonal rotation matrix is used for this mapping based on current pose of the robot.

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7.2}$$

$$Pose_R = R(\theta) Pose_G. \tag{7.3}$$

The orthogonal rotation matrix is used to map the robot motion from the global coordinate frame to the local coordinate frame. Given the velocity $(\dot{x}, \dot{y}, \dot{\theta})$ in the global coordinate frame, the robot motion along the local coordinate frame is calculated by equation 7.3. This equation is used for the calculation of the forward kinematics of a differential drive robots. These robots have two powered wheels with nominal radius r. The distance between the center of the two wheels is L. The forward kinematics of the robot predicts the robot motion in the global coordinate frame based on wheel radius r, wheel axle length L, robot orientation θ and the angular velocity of the wheels as u_l and u_r. Variation of each wheel velocity enables the robot to perform rolling

motion around a point that lies along the wheel axis. This point is called Instantaneous Center of Rotation (ICR).

The local coordinate frame of the robot is aligned such that the forward motion of the robot is along the positive X_R direction. The wheel constraints of a differential drive robot prevent the robot from moving in Y_R direction. Therefore the linear velocity of the robot, along Y_R axis of the local coordinate frame is always zero. Robot only has linear velocity along X_R and angular velocity. The contribution of each wheel is considered separately to calculate the linear and angular velocity of the robot. Consider point P as the center point of the robot centered on the wheel axle. Consider a scenario that one wheel of the robot is rotating and the other wheel is static. The robot is rotating around with P as the center of mass and the static wheel as the contact point. The velocity of the top of the rolling wheel is the double of the center of the mass and the contact point has zero velocity. Therefore, with this analogy the contribution of each wheel to the linear velocity of the robot is:

$$V_l = ru_l \tag{7.4}$$

$$V_r = ru_r \tag{7.5}$$

$$V = \frac{V_l + V_r}{2}, \tag{7.6}$$

where r is the wheel nominal radius, u_l, u_r are the spinning speed of the wheels and V_l, V_r are the linear velocities of each wheel. V is the linear velocity of the robot at point P. A stationary spinning robot has zero linear velocity while wheels are rotating with exactly the same speed in opposite directions. The last step is the computation of the angular velocity of the robot. This is denoted by W. In a similar scenario when the right wheel is rotating and the left wheel is static the robot has an anticlockwise rotational motion around the left wheel. The angular velocity of point P is calculated based on the wheel movement along an arc of a circle with radius L. The same calculation applies to the left wheel with the exception of clockwise rotation. Therefore the angular velocity of the robot is the sum of the angular rotation of left and right wheels:

$$W_l = \frac{-ru_l}{L}, \tag{7.7}$$

$$W_r = \frac{ru_l}{L}, \tag{7.8}$$

$$W = W_l + W_r = \frac{V_r - V_l}{L}. \tag{7.9}$$

The combination of the linear and angular velocity of the robot and Equation 7.3 produce the kinematic model of the differential drive robot:

$$Pose_G = R(\theta)^{-1} \begin{bmatrix} V_x \\ V_y \\ W \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{V_r+V_l}{2} \\ 0 \\ \frac{V_r-V_l}{L} \end{bmatrix}, \tag{7.10}$$

where $Pose_G$ is the predicted global position, $R(\theta)^{-1}$ is the inverse of the orthogonal matrix of the robot, V_r, V_l are velocity of each wheel along the ground and L is the distance between the center of the wheels [68, 211].

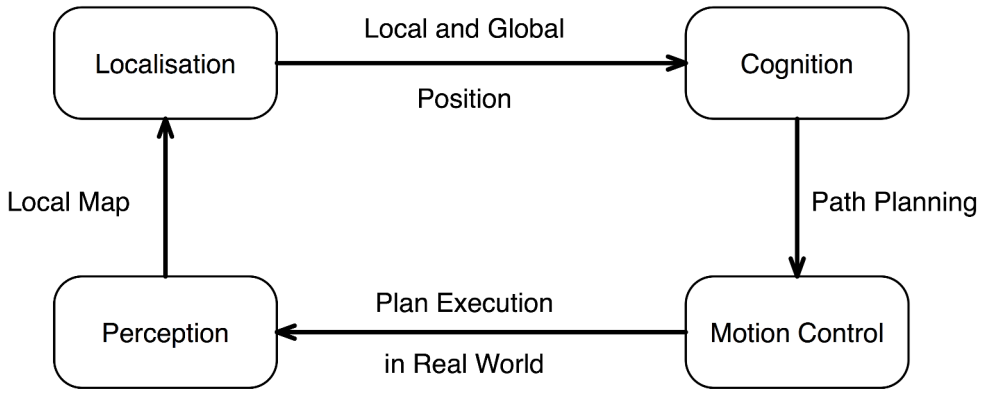


Figure 7.3: Autonomous mobile robot control cycle. This represents the main four stages of perception, localisation, cognition and motion control of an autonomous mobile robot.

7.3 Mobile Robot Perception

Advances in mobile robotics in recent years produce robust and efficient approaches to autonomous systems. The most important task of an autonomous system is the ability to sense its environment. Mobile robots use the attached sensors to their platforms to acquire knowledge about their environment. An autonomous system requires to extract meaningful information from the captured measurements to localise itself in the environment. Afterwards it plans and executes its next task (cf. Figure 7.3).

In robotics field sensors can be classified as proprioceptive sensors that are used to measure the internal state of the robot such as motor speed, battery voltage and internal temperature and as exteroceptive sensors that capture information from the robot environment such as distance measurements, light intensity sensors, cameras and laser range finders. Another classification for sensors categorises them into two categories. Passive sensors measure the environmental energy entering the sensor such as microphones and cameras. Active sensors emit energy in to the environment and measure the environment reaction such as ultrasonic sensors and laser range finders [105, 211]. This section focuses only on the sensors that are available on the mobile robots used for this application. Figure 7.4 presents the robots used for the application presented in this chapter.

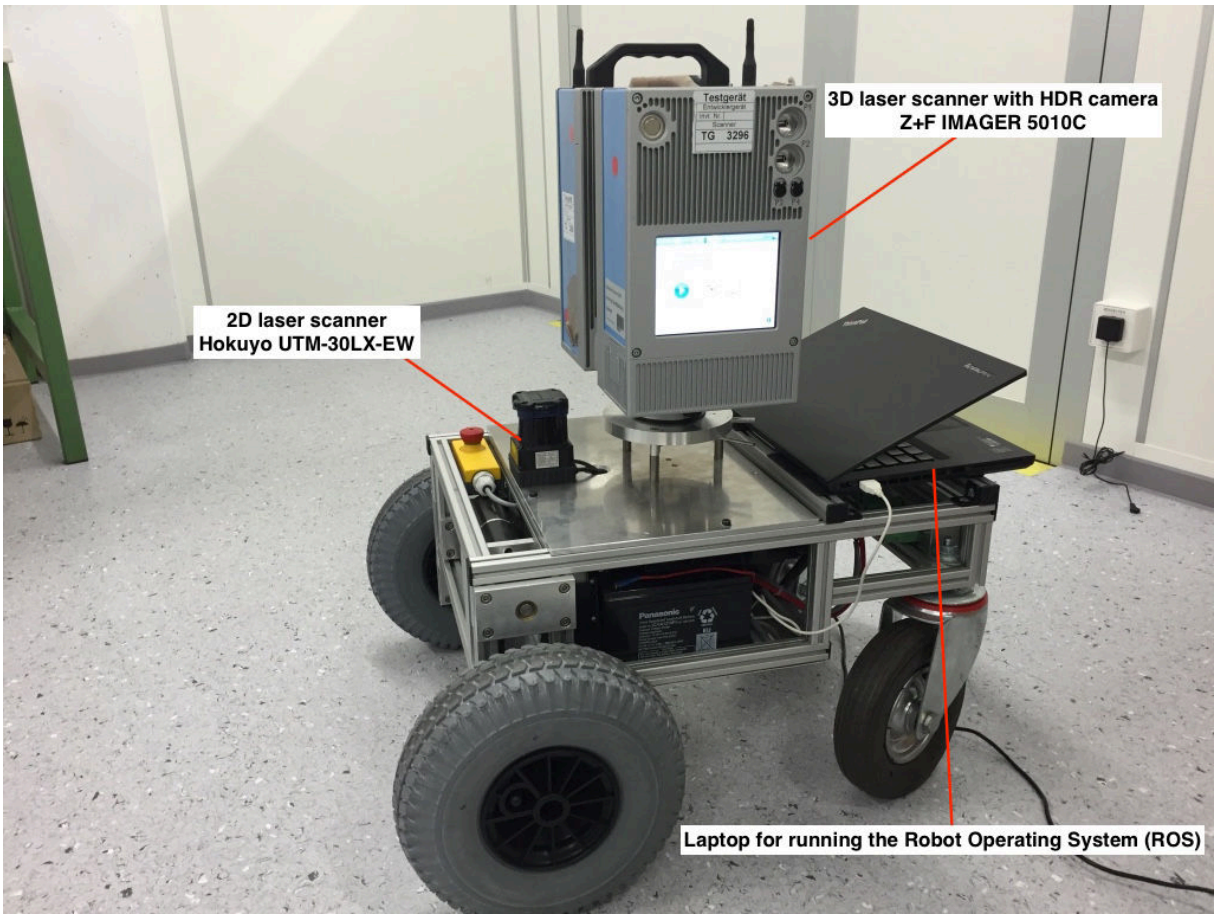


Figure 7.4: ZF-Volksbot the four wheeled mobile robot. This robot is equipped with a differential drive system. Two motorised wheels in the front and two castor wheels in the back. The wheel encoders are included in the configuration for odometry calculation. The robot is equipped with a 2D range finder namely the Hokuyo UTM-30LX-EW [106]. Additionally the Z+F IMAGER 5010C [254] is used for capturing the 3D point cloud of the robot environment. This scanner is equipped with and HDR camera for capturing the color values for measured points.

7.3.1 Wheel/Motor Encoder Sensor

Wheel/Motor sensors are used to measure the internal state of the mobile robot. The most popular sensors to measure the wheel/motor status are optical incremental encoders. These encoders are used to control the position and speed of motor powered wheels on mobile robots. The incremental encoders are used for measuring the angular velocity of the motorised wheel. The angular velocity of wheels is used to estimate the position of the mobile robot. An optical encoder is a light source that produces a certain number of sine or square wave pulses for each rotation. The system consists of three main parts. First source of light. Second a rotating disc mounted on the rotating shaft with optical grid pattern. Third a fixed optical decoder sensor. The rotation of the shaft with the optical pattern interrupts the light emitted onto the optical decoder sensor. In robotics the received sine wave from the optical decoder sensor is transformed

into a discrete square wave. This defines two states of light or dark for the system. The number of Cycles Per Rotation (CPR) defines the resolution of the encoder. The optical encoder can be manufactured with 10000CPR. The encoders used in mobile robotics often have a second set of light and detectors. They are shifted 90 degrees with respect to the original pair. These encoders are called quadrate encoders. They provide additional information such as direction of the rotation. Additionally they improve the resolution by a factor of four. This is based on the resulting twin square waves from both light sources [211].

7.3.2 2D/3D Range Sensor

Active range sensors provide direct measurements from the robot to objects in environment. Most mobile robot platforms rely on these sensors for object detection and avoidance. Additionally these sensors are used in the localisation and mapping process of the robot. As introduced in Chapter 2 active range sensors can be categorised into time of flight and triangulation based categories. Laser range finders belong to the first category. This chapter focuses only on the sensors available on the mobile robot platform used for the experiments. Figure 7.4 presents the mobile robot platform with its 2D/3D range sensors.

A mobile robot utilises the active range sensor to determine its relation to the objects in the environment. A mobile robot platform extracts information from one or more sensors to update the robot knowledge of the environment. Afterwards, robot actions are executed in this environment. The system relies on the extracted features to interact with the environment. Features are the recognisable structures in the environment. The system is capable of extracting geometrical features such as lines, polygons and circles from the raw sensor data. Additionally, LiDAR sensors present the environment in the form of 2D or 3D point clouds. This is used by the mobile robot platform to generate the 2D/3D model of the environment. Furthermore, the system is able to localise itself in its environment [105, 211].

7.4 Localisation and Mapping

Navigation is a challenging task for a mobile robot. The navigation system of a robot is divided into four basic steps. Perception: the mobile robot uses its sensors to capture data from its environment. It interprets this data to produce meaningful information. The next step is the robot localisation: the mobile robot determines its position based on the data captured from the environment. The third step is the robot cognition: the mobile robot in this step plans its movement based on the environment to achieve its goals. The last step is the motion control: the mobile robot utilises its locomotion mechanism to follow the plan generated from the cognition step to reach its goal from a desired trajectory (cf. Figure 7.3).

Localisation is the process of answering the question “Where is the robot?”. Global Navigation Satellite System (GNSS) are used to detect the absolute position of the robot in Earth reference frame. However, the existing GNSS only provides accuracy within a several meters. This is acceptable for navigation of large platforms however, it is not suitable for small mobile robots. Additionally, GNSS systems only provide the position estimation in open space outdoor environments. Therefore, it is not capable of providing the position information required for navigation of indoor mobile robots. The localisation process is more than the determination of the absolute position of the robot. It also includes the determination of the relation of the robot

with objects in its environment. The mobile robot requires to detect static and dynamic objects in the environment and calculate its relative position in this environment. This is required for the cognition process on a mobile robot. The robot generates a map of the environment from the captured data from its perception sensors. The localisation and mapping provides information for robot to select a strategy to follow for achieving its goals. It consists of positioning of the mobile robot with respect to its local environment and a map. Sensor noise increases the complexity of the robot localisation problem. Captured noisy data from the perception sensors provides limited information for the robot. The robot movement is measured based on the wheel rotations. The readings from robot wheels introduce uncertainty about the state of the robot. These errors are part of the robot position estimation based on its kinematics and dynamics.

Monitoring the wheel revolutions to calculate a relative position and orientation (pose) is the most widely used method for determining the robot pose. This method is known as odometry. Odometry provides relative pose estimation between two absolute pose measurements. This method is employed in most mobile robot applications [27, 29, 48, 49]. Absolute pose estimation methods such as, active or passive landmarks and scan matching systems are available for mobile robots. However, these methods introduce limitation for the positioning system. Scan matching methods are accurate but slower in comparison with odometry. On the other hand odometry and in general the proprioceptive sensors have the well-known disadvantage of inaccurate measurements with an unbounded accumulation of errors. Odometry calculates the momentary pose of the robot relative to a known starting pose. This calculation is a simple geometric equation based on incremental horizontal displacement of the drive wheels. This displacement is calculated from the incremental wheel encoders. Odometry errors can be categorised in to two groups, namely non-systematic and systematic odometry errors. Non-systematic errors are caused by the interaction of the robot with the environment, such as uneven floors, slippage and human interaction with the robot. These errors are particularly problematic due to their unpredictability. Borenstein [26, 32] presents a method for reducing these non-systematic errors by using redundant encoders. Systematic errors are related to the robot parts. These errors usually stay unchanged during the usage of the robot. Therefore, they can be measured and removed from the odometry calculation. These errors are caused by imperfections in robot design and implementation. The main source of systematic errors are unequal wheel diameters and uncertainty about the effective wheel base [28, 29, 53, 74]

Borenstein and Feg [28] present a method for measurement and correction of systematic errors for differential drive robots. They define a benchmark for odometry accuracy that assure the different odometry errors do not compensate for each other. Their method produces a single numerical value to represent the quantitative measure of the systematic odometry error. Doh et al. [65] present a comparison of this method with an algorithm based on the boundness property of the error for generalised Voronoi graph based paths. Von der Hardt et al. [60] present an interactive algorithm for determining the odometry parameters. Komoriya and Oyama [133] propose a sensor fusion algorithm to increase the accuracy of the robot pose estimation. Similarly Goel et al. [92] present a Kalman filter approach to estimate the robot pose based on gyroscope, wheel encoders and GPS system. Antonelli et al. present a systematic method for odometry calibration for differential drive robots. Their proposed calibration method is based on the least square method. This provides a measure of the numerical conditioning of the data to generate guidelines for test trajectories and to evaluate the accuracy of the solution [6–8].

7.4.1 Mapping

The odometry information provides the mobile robot with a relative pose estimation from a well-known pose. It is through localisation on a map that the robot can recover its pose and detect its arrival at the goal location. A map is the representation of an environment with a list of objects and their locations. The precision of the map must match the precision of the robot goals. Type of presented features and their precision match the robot sensors. The complexity of the representation of the environment with maps has direct impact on the complexity of the mapping, localisation and navigation processes of the mobile robot. Maps are categorised into feature-based and location based categories. Feature-based maps are a list of features in the environment. Feature indices represent the feature location and properties. Location based maps represent locations in an index list. Planar maps are often presented in specific (x, y) world coordinate. The indexed position contains the property of the world coordinate. These maps are volumetric and they offer an index for any location in the world. Volumetric maps present the presence of an object. Additionally they contain the information about the absence of object in the environment. These maps present both empty and occupied space of the environment on the map. However, feature-based maps only contain the shape and position of the objects in the environment [228].

Continuous map representation is a method to present the exact position of the features in the environment. Combination of this method with the closed world assumption is a common approach in robotics. This assumes all objects in the environment are presented on the map and the remaining area on the map corresponds to the empty space. This optimises the required memory for representation of a map.

An example of this method is the presentation of the environment and objects with lines and polygons in a continuous valued 2D coordinate space. Figure 7.5 represents a sample continuous map of an indoor environment. On the other hand the classical map representation uses a fixed decomposition method. This tessellates the real world and transforms the continuous environment onto a discrete approximation of the map. The main disadvantage of this approach is the discretisation of the environment. This effects small objects and narrow passageways. However, this concept is the most common map representation in mobile robotics. The occupancy grid presented by Moravec and Elfes [161] is the well-known fixed decomposition map representation utilised in mobile robotics. The occupancy grid is a location based map. The environment is represented in a discrete grids, where x, y coordinates represent a binary occupancy value that specifies the occupancy of the presented environment. The cell is either filled that specifies that the space is occupied with an object or the cell is empty that represents free space of the environment (cf. Figure 7.5). The most common occupancy grid maps are 2D floor plan maps. They describe a 2D slice of the 3D environment. This map representation is used in mobile robots navigating on a flat surface. This method is of particular value for robots equipped with range based sensors. The range value from the sensor combined with the absolute robot pose is used to calculate the occupancy of each grid cell. The main disadvantage of this method is the relation of map size with the environment size. Increase in the environment increases the size required for representing the map. Additionally, the occupancy grid approach is not compatible with the closed world assumption. In contrast, this method maps every region in the environment to empty, unknown or occupied cell [105, 211, 228].

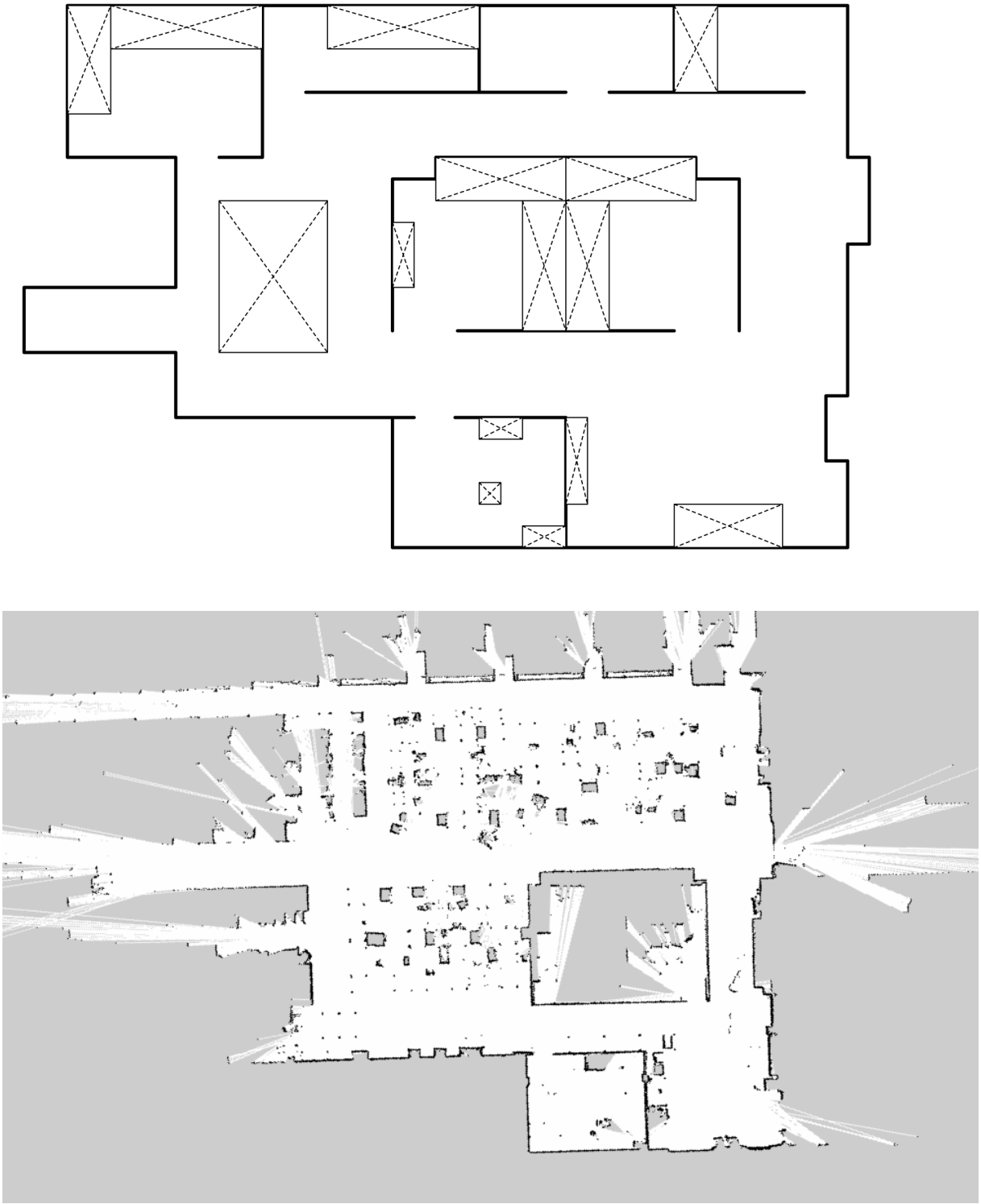


Figure 7.5: Top: Representation of the continuous map of a environment. The obstacles are represented as polygons. Bottom: Representation of an occupancy grid map. The white cells are the empty environment. The black cells are occupied and the grey cells are unseen area of the environment.

7.4.2 Localisation

Mobile robot localisation is the process of determining the robot pose relative to the map of the robot environment. This is also known as the position estimation of the mobile robot. Localisation is essential to robot movement and interaction with the environment. Localisation is considered as a coordinate system transformation problem. It is the process of establishing the correspondence between the map coordinate system and the local robot coordinate system. The robot utilises odometry to track its motion from a precise location on the map. However, most robots do not have a noise free sensor for measuring the pose. After some movement odometry errors accumulate and the robot pose estimation will be uncertain. Therefore, the robot must localise itself based on the captured data from the environment to avoid the increase of uncertainty. The information provided by the robot sensors from the environment and the robot odometry is combined to enable the robot to localise itself with respect to a map. Robot localisation is categorised into local and global methods. Local localisation or position tracking solves the localisation problem with a known initial pose. This method accommodate the noise in robot motion to achieve localisation. The global localisation methods are more general. In these methods the initial pose is unknown to the robot.

Markov Localisation

Markov localisation is the application of a Bayes filter to the localisation of mobile robots. This method allows for localisation starting from any unknown position. Markov localisation tracks the state of the robot using an arbitrary probabilistic density function to represent the robot pose. It transforms the probabilistic belief at time $t-1$ into a belief at time t. This method addresses both local and global localisation [42, 105, 131, 171, 211, 228]. The localisation method is capable of recovering from ambiguous situations due to the fact that the robot tracks multiple possible poses. To update the probability of all the poses at any time the method requires a map of the environment. Burgard et al. [41, 42] present a precise navigation method using a grid based map utilising Markov localisation.

Kalman Filter Localisation

The Markov localisation method is very general, therefore it is inefficient. The key to a robust localisation method is sensor fusion. Mobile robots include multiple sensors that provide estimation for the robot pose. However, these estimations lack the accuracy and they suffer from failure modes. To optimise the localisation, the Kalman filter method utilises the information from all sensors. It is an optimal recursive data processing algorithm that combines all information regardless of their precision to estimate the robot pose [105, 211]. The Kalman filter is a mathematical method for estimating the system state based on the knowledge of the system and measuring devices and the measurement noise [152]. This fuses the sensor data in an optimal way.

Monte Carlo Localisation

The Monte Carlo localisation algorithm represents the robot state with particles. Similar to Markov localisation this method is used for both local and global localisation. The key idea of the Monte Carlo localisation is to represent the initial global uncertainty with a set of N weighted

random pose particles. The method proceeds in two steps. First the robot generates N new particles during the robot motion. This approximates the robot pose after the movement. The new set of particles are generated by randomly selecting a particle from the previously computed particles. The second step incorporates the sensor readings by assigning importance factors to particles. This approach is computationally efficient with the ability of representing arbitrary distributions. An adaptive extension of the Monte Carlo localisation is capable of updating the number of particles online. This only invokes large particle sets when necessary. Additionally a small number of uniformly distributed random particles is added after each estimation step. This is essential for relocalisation when the robot loses track of its pose [81, 83, 105, 228].

7.4.3 Simultaneous Localisation And Mapping

All the localisation methods mentioned in this chapter require a map of the environment. The assumption of availability of a map is legitimate in a few real world applications. However, some applications do not provide the map of the environment and the robot ought to build its own map. The ability to learn a map from scratch reduces the efforts for installing a mobile robot in the environment. Additionally, it enables the robot to adapt to the environmental changes. Learning a map is a challenging problem without the accurate localisation of the robot. This is a "chicken and egg" problem. The localisation problem requires the map to improve the robot pose estimation. Robot movement in the environment accumulates the odometry errors and gradually reduces the robot certainty about its pose. Localisation methods rely on the existence of the environment map to determine the robot pose. On the other hand constructing a map with an uncertain robot pose is a difficult process. In the absence of both accurate localisation and a map the robot has to create the map and localise itself at the same time. In the robotics community this process is called Simultaneous Localisation And Mapping (SLAM). The SLAM difficulty is based on the relation between the robot pose update as it localises itself and the process of learning the map. The robot updates the pose from the observation of the map features. Therefore, the pose estimation correlates to feature position estimation. On the other hand the map correlates to the pose estimation. The sensor measurements from the estimated pose is used to update the map. The robot only has the sensor measurements from the environment and the robot controls. SLAM methods utilise the current robot pose estimation, the map, the sensor measurements and the robot control to update the map and the robot pose. The SLAM process consists of multiple parts. Feature extraction, data association, state estimation, state and feature updates are the steps in any SLAM solution. The goal of this process is to use the environment for updating the robot pose and the map.

There are two main forms of the SLAM problem. The first category is known as online SLAM. This estimates the pose over the momentary pose along the map. The second SLAM category is known as global SLAM. This calculates the pose for the entire estimated poses along the map [228]. In robotics the EKF method for SLAM is introduced by [212, 213]. Moutarlier and Chatila [163, 164] and Leonard and Durrant-Whyte [143] present the original implementation of EKF SLAM. Dissanayake et al. [64] presents another extended Kalman filter SLAM solution. Thrun [226] presents an extensive review of existing methods, such as maximum likelihood presented by [78, 84], and expectation maximisation presented by [229]. Similarly several other methods have been proposed in the literature [63, 71, 99, 159, 225]. Methods

from 2D mapping are extended for mapping the 3D environment. These methods estimate 6D poses [241]. Some methods rely on 2D laser range finders to build 3D volumetric maps of the environment [87, 100, 227]. Few other methods rely on the 3D data. This data is acquired either by a 3D scanner or a rotating 2D scanner [4, 176, 179, 207, 208]. In the experiments of this chapter two SLAM methods are used that are both available as open source projects.

GMapping SLAM

Doucet et al. [67] and Murphy [165] introduce Rao-Blackwellized particle filters to solve the SLAM problem. Additionally, Grisetti et al. [97] present two approaches to improve the performance of the Rao-Blackwellized particle filters for solving the SLAM problem. They present an adaptive method to reduce the number of particles in a Rao-Blackwellized particle filter method. This is used for creating the grid map of the environment. Additionally they propose a method to reduce the uncertainty about the robot pose during the prediction step of the filter. One of the issues of the Rao-Blakwellized approach is the elimination of the correct particles during the resampling step of the method. This effect is known as particle depletion. Grisetti et al. [96, 97] present an approach to selectively carry out the resampling to reduce the problem of particle depletion.

Hector SLAM

Many SLAM solutions work best in a planar environment. They often rely on accurate odometry and do not leverage the high update rate of the modern LiDAR systems. Kohlbrecher et al. [132] present a system for fast online generation of occupancy grid maps. Their focus is to reduce the required computational resources. Their method combines a 2D scan matching approach using a LiDAR system in a planar map with an integrated 3D navigation system based on an Inertial Measurement Unit (IMU). The 2D information from the scanner is one possible source of information for pose estimation. Using a fast approximation of map gradients and a multi resolution grid provides reliable localisation and mapping capabilities [132].

7.5 Sensor Placement Planning

3D laser scanning methods are well stablished in the surveying and robotics fields. They are widely used for high precision, high density 3D reconstruction of the surface of objects or an entire environment. Dense and detailed 3D models are used in many fields, such as industrial modelling, medical imaging, entertainment, archeology and architecture. These models are used for documentation of sites. Scanning from different poses enables the digitisation of an environment and resolves the occlusion. An autonomous mobile robot requires to observe the environment and capture information of its environment. The generated knowledge from the environment is used to assist the robot to plan its next move toward achieving its final goal to create a complete model of the environment. The process includes planning a set of views, taking scans in planed poses, registering the acquired data to a common coordinate system and finally generating the full model of the environment. Registration algorithms from robotics and geodesy are used to automatically align scans from different poses.

View planning remains an open problem. This process requires choosing an efficient set of sensor poses to cover the entire environment without occlusion. The estimation of such view points is known as the Next Best View (NBV) problem. Manual determination of the views is a time consuming process. Automation of this manual process improves the quality and the efficiency of the reconstruction process. View planning methods are categorised into model-based and non model-based view planning methods. Model-based methods plan a set of scan poses based on a priori model of the environment. On the other hand non model-based methods which dominate the existing view planning algorithms, plan a set of scan poses in an unknown environment. These methods are also known as exploration methods in robotics. The main goal of these algorithms is "given what you know about the world, where should you move to gain as much new information as possible" [248]. The same goal is valid for non robotic applications. The aim of scanning an environment is to capture as much data as possible from the environment with the minimum number of scanning poses.

3D environment mapping with mobile robots is an active field of research in robotics. Surmann et al. [222] and Nüchter [173] present a 3D environment mapping system using mobile robots that are equipped with 3D scanners. Traditional NBV algorithms compute view poses until the entire scene has been observed and a complete model of an environment is generated from the captured data [11, 93]. These methods determine the next best location based on the data collected from previous scans. Most methods select locations on the border between explored and unexplored regions. This reveals the largest amount of unknown environment [70, 230]. Selection of the next best view on the boundary between open space and uncharted territory is the key idea behind the well-known frontier based exploration methods. Dornhege and Kleiner [66] extend this idea by a novel approach that combines frontiers in 3D with the concept of voids. Voids are unexplored volumes in 3D that are automatically generated from observed data from 3D sensors. The voids are generated after extracting all occluded or enclosed regions by obstacles. They utilise octomaps [108, 246] for efficiency in 3D computations.

The calculation of view points is similar to the art gallery problem. Shermer [210] presents the art gallery problem as a task to find an optimal placement of guards on a polygonal representation of an environment in 2D that the entire environment is observed by the guards. Nüchter et al. [178] propose a next best view method for digitising a 3D environment with mobile robots. They represent the 3D environment as polygons. The obstacles are presented as detected lines and the free space is presented as lines connecting the detected obstacles. From this representation the next best view is selected according to the 2D information gain from each potential location. However, the 3D data are actually acquired at each location. Joho et al. [116] extend this method from a 2D representation to a 2.5D elevation map. Algorithms based on the 2D horizontal plane in the 3D environment can be used efficiently for mobile robot exploration. However, this is not sufficient for the generation of a complete 3D model of an environment. To achieve a complete 3D model a method is required to search for unexplored area in 3D. Sensor placement planning in 3D for large environments requires significant memory space for storing the occupancy information of the whole environment. Borrmann et al. [35] propose a method that combines 2D and 3D planning. Their main idea relies on a typical indoor environment, composed of enclosed spaces. Their algorithm explores the environment based on only 2D measurements and follows the NBV poses obtained from 2D planning. After detection of an enclosed space the NBV search switches from 2D to 3D planning. This method utilises the whole 3D

environment captured by the 3D scanner. The 3D method only explores the detected enclosed space thus it requires to store the 3D information of this small part of the environment. After the full 3D exploration of the enclosed space the 3D algorithm terminates and the 2D algorithm continues. Their 2D NBV planning algorithm is based on their previous work presented in [250].

Blaer and Allen [23] present a method that addresses the joint problem of data acquisition and view planning for large scale indoor and outdoor environments. Their method proceeds in two stages. First the system uses a 2D map to plan a minimal set of locations. A 3D scanner is used to acquire data at each location. Their system is combined with a mobile robot that automatically computes and follows the required path to navigate to scan locations. The second step of the system proceeds by using the initial scans as an approximate 3D model of the environment. The planning method relies on a voxel based occupancy approach to select the next best view. The new scan is acquired and the system computes the further locations until a complete 3D model is obtained.

Potthast and Sukhatme [188] describe an automatic data acquisition and view planning method by iteratively estimating the knowledge gain of potential scanning poses. Their approach estimates the knowledge gain in cluttered environments based on the unobserved space using probabilistic methods. A major survey of the NBV methods specifically for 3D sensors is presented by Scott et al. [206]. Kawashima et al. [123] propose a new computer aided method to find a sequence of next best views for efficient and exhaustive measurements of plant piping systems. Their method is specialised for as built modelling of piping systems without any a priori information of the piping objects. The non model-based method seeks to generate models with no a priori information. These methods include volumetric methods. This chapter presents a volumetric method for estimation of the next best scanning pose for modelling a 3D environment. This is a generalised method that can be used for the mobile robot exploration process or as a semi-automatic assistance system for surveyors during the 3D data acquisition of large environments. The system relies on detection of holes in the captured environment and estimating the next best view to maximise the information gain.

7.5.1 Hole Detection

Dense and complete 3D models of large environments are beneficial for many fields. Data from the 3D data acquisition process often contain holes due to occlusion, reflectance or transparency. However, to capture the complete environment multiple scan poses are required. Captured data from one scan pose lacks the data from the shadow of the occluding objects. Therefore the environment obscured by objects requires additional measurements. The captured environment is presented in 3D point clouds. Points are an unstructured surface representation without connectivity information. Defining and detecting holes in these datasets is a challenging task. However, the knowledge of holes in the data is crucial in many applications. It is used to direct the next scan pose for capturing the missing information from the environment. Bendels et al. [18] present a method to investigate properties of points to derive criteria for automatic hole detection. They combine several criteria into an integrated boundary probability. Finally a boundary loop extraction method is used for automatic hole detection. Chalmovianský and Jüttler [46] describe a method for detecting and filling the holes in point clouds. Their method aims to detect the boundary of holes. Additionally, they fit an algebraic surface patch to the hole and sample points to fill the hole.

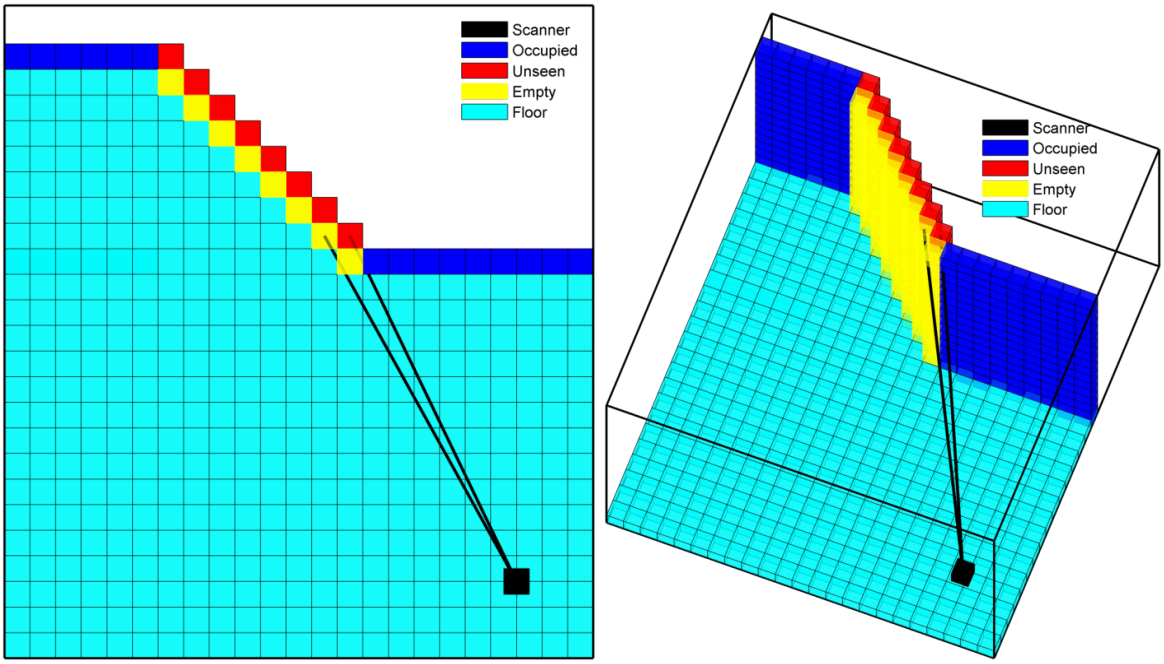


Figure 7.6: Hole voxel representation as the intersection between the unseen and empty voxels. Empty voxels are the unoccupied voxels that are visible from scan pose. Unseen voxels are the unoccupied space that are obscured by objects. Left: Top view of a sample environment and the representation of hole as the intersection between yellow voxels as empty with red voxels as unseen space. Right: 3D view of a sample environment and the representation of hole as the intersection between yellow voxels as empty with red voxels as unseen space.

The estimation of the next best view method presented in this chapter relies on the detection of the obscured space in the scanned environment. This presents the hole in the point cloud. This method defines the hole as the border between the obscured and the visible environment. To detect these areas the point cloud is presented in an octree data structure. This is a volumetric representation of the environment. The space is presented in cuboid nodes of the octree. As presented in Chapter 2 each node in the octree represents the volume formed by a cube that is often axis aligned. Experiments in this chapter utilise an octree implementation that is derived from the PCL octree [198, 199]. An octree contains a bounding box that defines the volumetric area that the entire octree represents. Additionally it contains a resolution that defines the smallest node size. The cuboid nodes are also known as voxels. This volumetric representation discretises the entire 3D environment into voxels. Each voxel of the 3D environment can be marked as occupied, empty or unseen. The occupied voxels are presented as leaf nodes in the octree. Empty voxels are empty space that is visible from the scanner pose. The laser beam is able to reach these voxels by traversing through the environment without intersecting with an object. Unseen voxels are the obscures space. The laser beam is not able to reach these voxels without intersecting with objects in the environment. The hole detection method in this chapter for next best view planning defines the holes as intersection of empty and unseen voxels. The hole voxels are selected from the unseen voxels that have neighbouring empty voxels. Figure 7.6 demonstrates the representation of a hole in the volumetric 3D environment. The yellow voxels

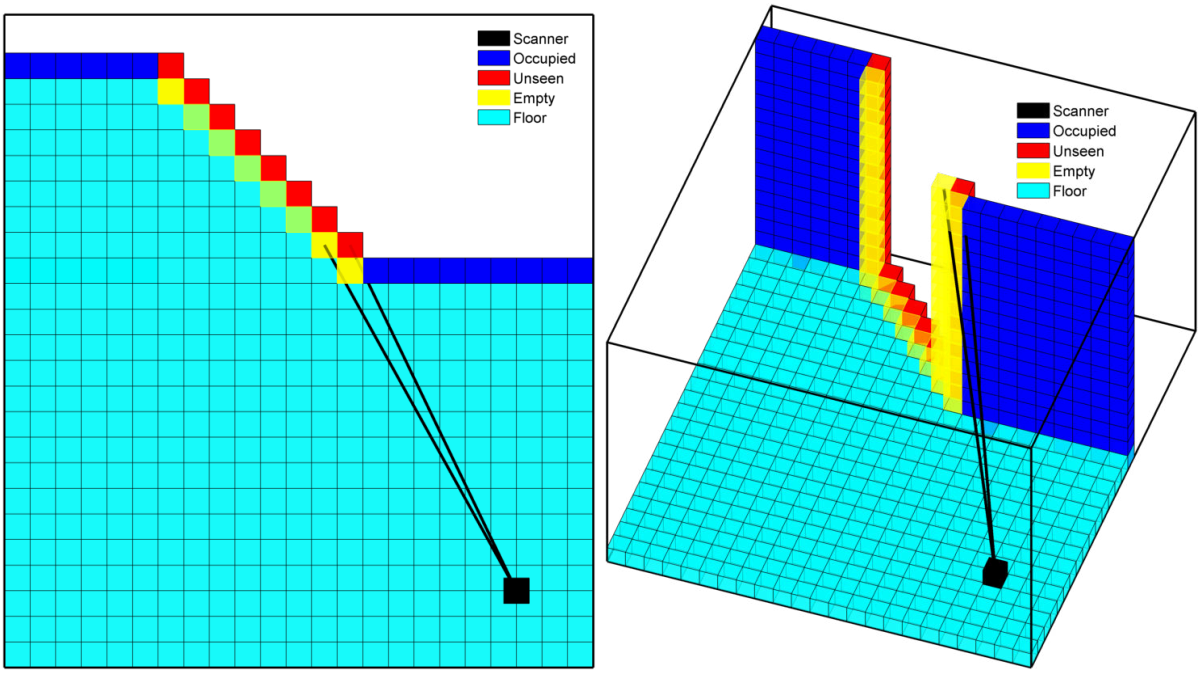


Figure 7.7: Hole border voxel representation as the unseen neighbouring voxels of occupied voxels with and empty neighbour voxel. Empty voxels are the unoccupied voxels that are visible from scan pose. Unseen voxels are the unoccupied space that are obscured by objects. Left: Top view of a sample environment and the representation of hole border. Right: 3D view of a sample environment and the representation of hole border.

represent some of the empty voxels. These are unoccupied voxels that are visible from scanning pose. The red voxels represent some of the unseen voxels. These are unoccupied voxels that represent obscured space. The hole voxels are selected from the unseen voxels that represent the intersection between the unseen and empty space.

The representation of a hole is further optimised by detecting and presenting only the border of the holes. This is achieved by only selecting hole voxels that are next to an occupied voxel. To detect these voxels the method traverses over all the octree leaf nodes as representation of occupied voxels. The hole voxels are selected as unseen voxels that are a neighbouring voxel to an occupied voxel and have a minimum of one empty voxel as neighbour. Figure 7.7 presents the detected border of a hole in a 3D environment.

The six face neighbour voxels of each occupied voxel are calculated. Occupied neighbouring voxels are detected by checking their existence in the octree and removed from the search process. The status of the remaining neighbouring voxels are determined with the help of a ray tracing algorithm. Similar to the measurement system of the scanner that emits a laser beam and receives the the reflection of the measured surface, the ray tracing method emits a virtual ray from the scanning pose to the query voxel. The algorithm is based on the volumetric discretisation of the octree voxels. It counts the occupied voxels of the octree that have intersections with the ray. In general ray tracing methods have an origin and a direction for intersection search. These methods usually check all the voxels in the entire environment that intersect with

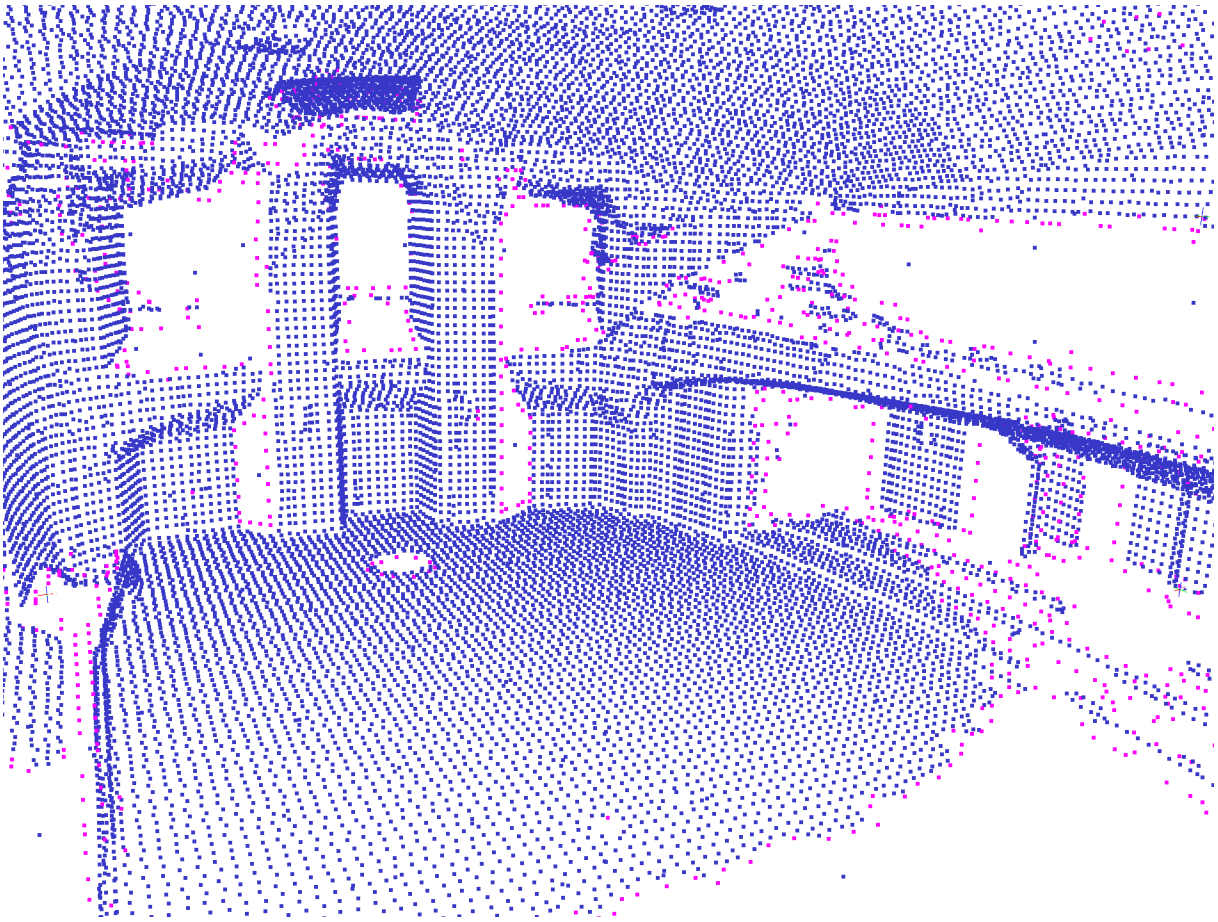


Figure 7.8: Representation of the detected holes in a single point cloud. The blue points represent the center of occupied voxels of the octree. This octree is used for detection of the hole voxels. The pink points represent the center of the hole border voxels.

the ray for calculation of the occupied intersections. To improve the required calculation time the ray tracing method is terminated on detection of the first occupied voxel or by reaching the query voxel. The status of a voxel is determined as empty without detection of any occupied voxel intersection. This can be observed as an area that the laser from the scanner traveled through before measuring a surface of an object. On the other hand the status of a query voxel is determined as unseen with detection of an intersection with an occupied voxel. This can be observed as an obscured space that the laser of the scanner did not reach. This determines the status of the unoccupied neighbouring voxels of an occupied voxel. To select a voxel as a voxel representing the border of a hole, a neighbouring pair of empty and unseen voxel is required. Therefore, a further neighbour status search is required for all empty and unseen voxels detected in the previous step. Similarly the six neighbours of each voxel are calculated and their status is determined by the ray tracing method. Finally the selection of all unseen voxels with empty neighbouring voxel presents the hole border voxels in the environment.

Figure 7.8 presents the holes detected in one scan. The blue points represent the centroid

Algorithm 4 Hole detection algorithm used for detecting the boundary of obscured space in a 3D point cloud. The hole voxels are defined as the unseen neighbouring voxels of occupied voxels. The hole voxels require minimum one empty neighbouring voxel.

```
Require: scan        // The captured 3D point cloud from an environment.
  octree ← createOctreeFrom(scan)
  for l = 0 to octree.leafCount do
    vNeighbours ← getVoxelFaceNeighbours(octree.leaf(l))
    for n = 0 to vNeighbours.count do
      vNeighbourStatus ← getVoxelStatus(vNeighbours[n])
      vNeighbourNeighbours ← getVoxelFaceNeighbours(vNeighbours[n])
      for nn = 0 to vNeighbourNeighbours.count do
        vNeighbourNeighbourStatus ← getVoxelStatus(vNeighbourNeighbours[nn])
        if vNeighbourStatus == Unseen and vNeighbourNeighbourStatus == Empty
        then
          addToHoleVoxels(vNeighbour[n])
        end if
        if vNeighbourStatus == Empty and vNeighbourNeighbourStatus == Unseen
        then
          addToHoleVoxels(vNeighbourNeighbour[nn])
        end if
      end for
    end for
  end for
```

of occupied voxels of the octree that represents the environment. The red points represent the unseen voxels of the environment that are adjacent to the occupied voxel with an empty neighbouring voxel. This process requires multiple ray tracings to the same voxels for status determination due to the fact that adjacent occupied voxels have overlapping neighbouring voxels. Similarly this redundant status determination is required for neighbouring voxels of empty and unseen voxels. To improve the computation time a temporary list of empty voxels is generated. These voxels are presented in a separate octree. Therefore, their status determination is reduced to determination of their existence in an octree. This new octree accumulates the empty voxels with a lazy accumulation method. An unoccupied voxel status is determined first by checking the empty octree and second in case of undermined status with ray tracing. Additionally an empty detected voxel with ray tracing is added to the empty octree. A similar method is used for temporary accumulation of unseen voxels to further improve the efficiency of the algorithm. Algorithm 4 presents the simplified steps of the hole detection algorithm presented in this chapter.

7.5.2 Floor Detection

For the estimation of the next best scanning pose of the presented next best view method in this chapter, it is crucial that the estimated poses are accessible in the environment. The scanning

unit or the mobile robot require the ability to travel through the environment and access this pose. In most applications the scanning unit is installed on the floor. Therefore, detection of the floor in the captured point cloud is an important step for estimation of the potential scan poses. To improve automatic processing of point clouds segmentation of the points is desired. This is required to reduce the data size and decrease the noise of the data. Segmentation is the process of grouping the measurements in the point cloud. Groups are generated to contain points from similar surfaces or regions. Different segmentation methods often vary in the similarity criterion used for grouping the points. Groups contain points with the similarity measure within a given threshold. Additionally, the points can be spatially connected.

Various segmentation methods are proposed in the literature. Edge based methods detect the border of different regions. This step is followed by grouping the points inside these boundaries. Sappa and Devy [201] and Wani and Arabnia [240] present typical edge based segmentation methods. Point-based segmentation methods use local point properties as similarity measurement for grouping the points. Rabbani et al. [189] present a segmentation method for point clouds using smoothness constraints as the similarity measurement. Their method uses the local surface normals and point connectivity to generate smoothly connected segments on the point cloud. Plane detection is often required in segmentation process. The detected plane is prerequisite to a wide variety of tasks, such as, modelling, object detection and classification, SLAM and many more. Region growing and RANSAC [76] based methods are the two main point-based approaches for 3D point cloud segmentation. Region growing algorithms are fast and easy to implement [127, 216]. Bauer et al. [13] and Boulaassal et al. [36] present RANSAC based algorithms for the detection of facades of buildings in the 3D point cloud. Deschaud and Goulette [115] present a fast and accurate algorithm to detect planes in unorganised point clouds. Their method utilises the filtered normals and voxel growing. It is based on region growing with improvements in normal estimation and the growing process. Biosca and Lerma [22] present a plane segmentation method for point clouds based on fuzzy clustering methods. Popinga et al. [187] propose a fast plane detection in noisy 3D range data. Yang and Förstner [249] propose a new approach in plane detection in point cloud data by integrating RANSAC and the minimum description length principle. Moosmann et al. [160] present a graph based approach for segmentation of floor and objects from 3D scans. They introduced a novel criterion dubbed local convexity that uses local geometric features for segmentation. Borrmann et al. [34] present a novel approach using the Hough Transform for detecting planes in 3D point clouds reliably. Additionally they evaluate different variants of Hough Transform for plane detection.

This section proposes a combination of RANSAC based plane fitting and normal based region growing method to detect the floor in the point cloud. The process starts with subsampling the octree generated from the scan into an octree only containing the voxels below the height of the scanning unit. Afterwards, the RANSAC based plane fitting is used to fit multiple horizontal planes into the remaining voxels. Multiple planes are used to detect floor with varying height in one environment such as an environment that the floor is divided with a step into two levels. A set of best fitted planes are selected. A threshold is used to introduce tilted planes for fitting. This is required for the detection of the floors with slope. Detected planes are represented by occupied voxel from the environment. This presents the horizontal surface of the environment with clusters of voxels. The region growing method combines the neighbouring clusters from the previous step. The largest cluster is selected as the representation of the floor

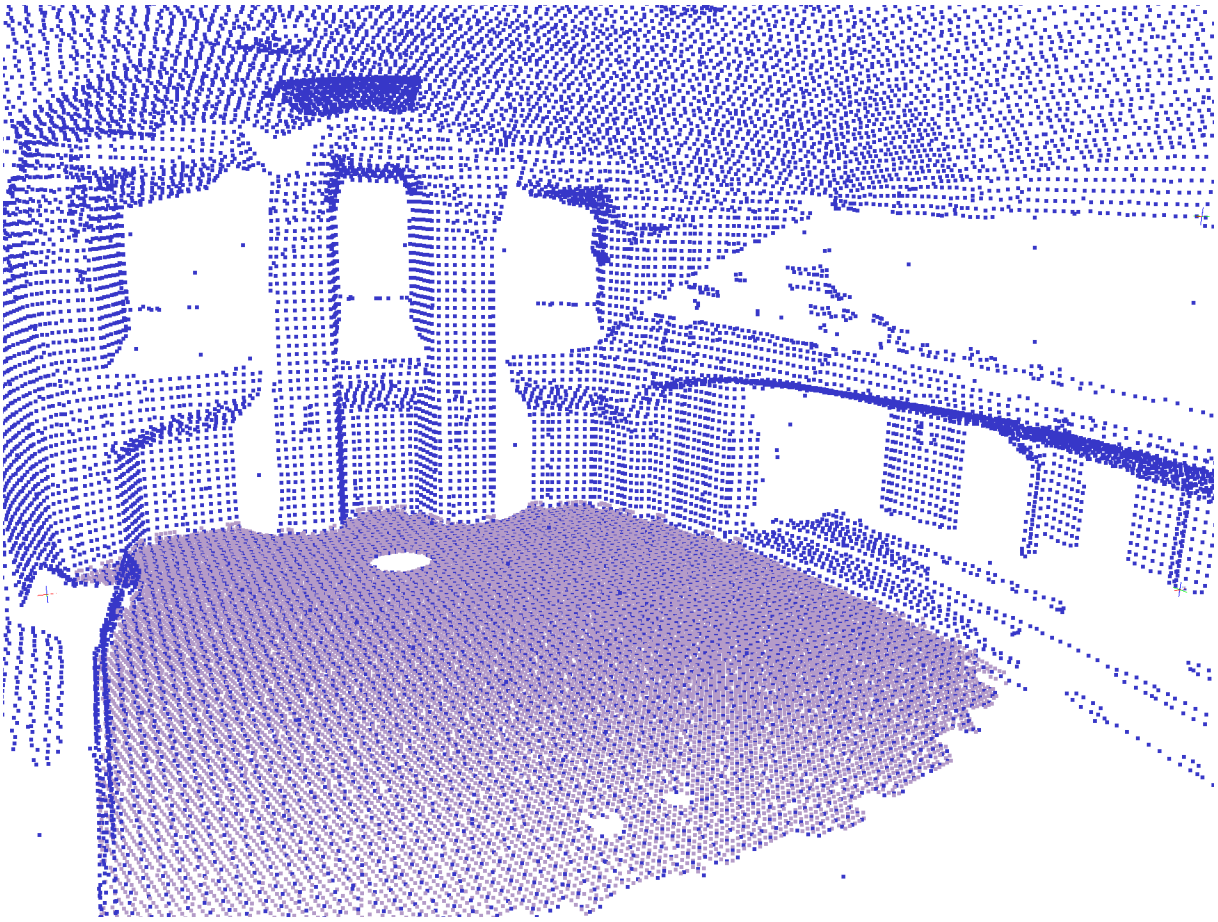


Figure 7.9: Representation of the detected floor in a single point cloud. The blue points represent the center of occupied voxels of the octree. This octree is used for detection of the floor voxels. The grey points represent the center of the detected floor voxels.

of the environment. Additional voxel normal examination removes all the vertical clusters of voxels. Figure 7.9 presents the detected floor from a data captured from one scan pose. The occupied voxels of the octree are presented by blue points and the grey points present the floor voxels.

7.5.3 Next Best View

The floor detection algorithm provides the available ground floor area for selection of the new scan pose. Detection of the floor requires minimum of one 3D scan of the environment. This is used as initial model for the 3D NBV planning algorithm. The main idea of the 3D NBV planning algorithm is to select a sequence of scan poses to optimise the scan acquisition process and to generate the complete 3D model of the environment. The candidate next scan poses are all voxels at the height of the scanning unit in the entire voxelised representation of the 3D empty environment. Potential poses are the candidate poses with scanning unit height

Algorithm 5 The next best view planning algorithm detects the hole and floor voxel in the octree representing the 3D environment. The octree, floor and hole voxels of the entire environment are accumulated from multiple scans. The potential next scan poses are selected based on the floor voxels and the height of the scanning unit. The potential next scan pose with the maximum number of visible hole voxels is selected as the next best scan pose.

Require: *scan* // *The captured 3D point cloud from an environment.*
  while Environment is not complete **do**
    $scan \leftarrow captureScanAtBestScanPose()$
    $octree \leftarrow createOctreeFrom(scan)$
    $holes \leftarrow detectHoles(octree)$
    $floor \leftarrow detectFloor(octree)$
    $accumulate(octree, accumulatedOctree)$
    $accumulate(holes, accumulatedHoles)$
    $accumulate(floor, accumulatedFloor)$
    $potentialNBV \leftarrow accumulatedOctree.calculatePotentialNBV(accumulatedFloor)$
    $NBV \leftarrow selectTheNBV(accumulatedOctree, accumulatedHoles, potentialNBV)$
    if *ROBOT* **then**
      Drive the robot to next best scan pose
    else
      Set up the scanner at the next best scan pose
    end if
  end while

from the floor voxels. The aim is to choose the potential pose with the maximum information gain for capturing the obscured space of the environment. The number of visible hole voxels are counted for each potential scan pose. The ray tracing algorithm is used to determine the visibility status of the hole voxel from the potential scan pose. The ray is traced from the potential scan pose to all detected hole voxels. The counter is increased for each hole voxel that no intersection with an occupied voxel is detected by the ray tracing algorithm. This indicates that all the traversed voxels by the ray are empty and therefore the hole is visible from the potential scan pose. The hole voxels that the algorithm takes into account are actually the edges of the explored and unseen area. The idea is that a scan pose that sees the border of the obscured space will be able to see most of the unseen space represented by the hole border. Additionally, the iterative nature of the next best view problem guarantees that this method fills all the holes in the environment. Many algorithms utilise the unseen voxels for calculation of the next best scanning pose. However, consideration of every unseen voxel is unnecessarily time consuming. The method proposed in this chapter extends the idea to only the unseen border voxels with minimum one empty neighbouring voxel. This reduces the number of voxels that require the visibility determination from the potential next scan poses. This approach is similar to the frontier edges in 2D exploration methods in the way that hole voxels that are used present the 3D frontier edges. They present the border between explored and unexplored space. After the detection of a location with maximum number of visible hole voxels, i.e., the next best view, a new scan is acquired in this location. The procedure continues in three stages. First

the hole and floor voxels of the new scan are detected. This process is carried out on one scan in its local coordinate system to increase the efficiency by eliminating the effect of the growth of the environment size in the calculation time. However, many newly detected hole voxels are detected or filled in the previous scans. The second step of the algorithm utilises an accumulation method to combine the newly detected hole voxels with all previously detected hole voxels. The algorithm eliminates the holes that are occupied or visible from any scan pose. The remaining hole voxels are the voxels that are obscured from all scan poses. Similarly an accumulation method is utilised to accumulate the detected floor and the octree of the entire environment. Required memory for presenting the octree increases with the size of the environment. To improve the memory efficiency the accumulated octree of the entire environment only represents the occupancy structure of the three. The actual points of the point cloud are not stored in the octree. Finally the method calculates the potential next best positions from the accumulated floor and weights them from best pose with the maximum number of visible hole voxels to worst with the minimum number of hole voxels. The potential next pose with the highest weight value is selected as the next best view for scanning. Algorithm 5 presents the steps of this method.

This is a generalised 3D next best view planning method. It can be used for the cognition process during the exploration on a mobile robot platform. Additionally, it is a semiautomatic decision support system for digitisation of a large 3D environment in the surveying field.

7.6 Robot Motion Planning

Wheeled mobile robots are used in industrial and service robotics. Mobile robots solve the problems of mission planning, navigation and motion control to reach their goal positions. Cognition generally represents the purposeful decision making and execution that a system performs to achieve its goals. Mobile robot navigation systems require the partial knowledge about their environment. The onboard sensors of the mobile robot platform are used to capture this knowledge. Additionally, a goal position or series of goals is calculated in the environment, such as detection of next best view. The navigation system of a mobile robot implements the ability to process its knowledge of the environment and sensor information to reach the goal position as efficiently as possible. Exploration is the problem of controlling a mobile robot to maximise the knowledge about its environment. In many applications the main goal of the mobile robot is to capture information from the environment. Previous sections of this chapter present the ability of the robot to use its on board sensors to capture information about its environment. This information is used to generate a 2D/3D map of the environment. The robot is capable of localising itself in the environment. The next best view method defines the sequence of goal poses for the robot to capture the environment efficiently. The last step of the exploration problem is the ability of the robot to navigate to the goal positions. Many approaches have been proposed in the literature for solving the navigation problem of mobile robotic platforms. The focus of this chapter is the robust and efficient approaches for autonomous navigation of mobile robots in indoor environments. Many approaches to indoor navigation use laser range finders to acquire a map. They present the map in a 2D occupancy probability grid. Additionally, they utilise probabilistic particle filter based approaches for pose estimation. Some approaches for autonomous navigation in indoor environments with differential drive robots utilise the onboard sensors for

localisation. These methods are based on the integration of information obtained from odometry, ultrasonic, camera and other onboard sensors. The majority of control systems assume the environment to be completely static. Movable parts of the environment such as doors are not considered. Some approaches detect the moving objects, such as people, and remove them from the process. The robot is able to utilise information about the movable objects to interact with them and achieve its goal. This behaviour is presented as for example opening the door to reach a goal position in the enclosed room. Nieuwenhuisen et al. [168] present a method to address the problem of the door by the explicitly representing the doors as door leaves and joints. They extend a standard approach to indoor navigation. They present the static environment with a 2D occupancy grid map. Additionally, they represent the movability of doors adequately in the map. Opening and navigation through doors is a challenging task. Gray et al. [95] present a planning framework that handles non-spring and spring loaded doors. Their approach uses a low dimensional graph based representation of the problem in order to plan a door opening procedure quickly and reliably. Stachniss and Burgard [217] propose an improvement to localisation in dynamic environments by using local grid maps of typical configurations of the environment. Schulz and Burgard [204] present a dual approach to localisation and object state estimation. Hähnel et al. [101] propose a classification method based on expectation maximisation. They group the measurements in static or dynamic objects. The static environment is used for SLAM.

The perception, localisation, mapping and cognition processes of the mobile robot determine the robot pose in the environment and select the goal pose that the robot has to achieve. The navigation process for the mobile robot answers the question "How do I get there?" [211]. The problem of motion planning of the robot is defined as follows: find a path that moves the robot from the start location to the goal location while avoiding all the obstacles. The control system of an autonomous robot consists of, path planning, navigation and obstacle avoidance. Given a map and a goal location the path planning process identifies a trajectory for the robot to reach the goal location. The navigation is the process of trajectory tracking and path following for the robot to gradually reach the goal location. This represents the classic artificial intelligence "sense-plan-act" approach. Finally the obstacle avoidance uses the real time sensor readings to modify the trajectory in order to avoid collisions. This is the reacting approach to the changes in the environment. Planning and reacting have a strong complementarity. Without obstacle avoidance the planning approach will not be successful and the robot will not reach its goal location. Without planning the reacting approach cannot guide the robot toward the goal location and essentially the robot will not reach the goal.

7.6.1 Path Planning

The field of path planning studies the required trajectory that guides the robot from the start location to the goal location. Most mobile robot applications operate the robot at low speeds. Therefore, dynamics are rarely considered during path planning. The path planning for mobile robots is simplified as a problem that only considers the kinematics of the robot. Although the robot motion is defined in the real world, the path planning is often done in a representation of the environment called configuration space. A robot configuration q, describes the pose of the robot relative to a fixed coordinate system. Consider a robot with k degrees of freedom. The robot configuration is a k-dimensional point in the configuration space C of the robot. This

simplifies the complex shape of the robot into a single k-dimensional point. The configuration space is the list of all possible configurations of the robot in the physical world.

The aim of the path planning method is the determination of a path from the start location to the goal location while avoiding all the known objects in the environment. This problem is solved in the configuration space. The objects are defined in the configuration space as a list of configurations O which the robot collides with them. The free space configuration is defined as $F = C - O$. The collision free path is selected from this subset of possible configurations. The pose of the nonholonomic differential drive robot operating on a flat surface is defined as position and orientation (x, y, θ) and linear and angular velocity $(\dot{x}, \dot{y}, \dot{\theta})$. The nonholonomic constraints limit the robot velocity. Differential drive robots are able to rotate in place. Therefore, a holonomic path can be mimicked [138]. The path planning methods only consider (x, y, θ) as the holonomic robot configuration for simplifying the process. The mobile robot is assumed to be a simple point for further simplification. This reduces the configuration for a mobile robot to a 2D representation (x, y). It represents the physical world with its identical 2D map. The reduction of the mobile robot to a point is compensated by sliding the robot along the edges of the obstacles in the configuration space and inflating them with to match the size of the radius of the robot [211].

The path planning systems represent the robot environment into a discrete map suitable for the path planning method. Many approaches have been proposed in the literature for discretisation of the configuration space. Some approaches present the connectivity of the free space in the robot configuration space with lines. The graph constructed from these lines is called road map. The path planning is the process of connecting the start and goal location with the road map and searching for a series of roads to connect the start location to the goal location. Visibility graphs [140] and Voronoi diagrams [138] are the most common road map approaches. In case of visibility graphs, roads are as close as possible to obstacles and the resulting path is a minimum length solution. On the other hand Voronoi diagram paths are as far as possible from the obstacles [211]. Many approaches decompose the environment in non overlapping geometric areas or cells. These cells represent the free space and areas that are occupied by objects. The main idea is to determine adjacent cells and to construct the connectivity graph. Afterwards, a search for a path in the connectivity graph that connects the start location to the goal location [197]. If the cell boundaries are presented as a function of the structure of the environment, the method is called exact cell decomposition. The decomposition with approximation of the actual map is called approximate cell decomposition. These methods are used for the generation of the environment map. Due to the popular use of occupancy grid maps for the representation of the robot environment, the approximate cell decomposition method is the most popular technique for path planning. A series of methods abandon the concept of explicit representation of free and occupied space. They rely on a collision detection algorithm to determine the status of some configurations in the configuration space of the robot. Kavraki et al. [122] propose probabilistic road maps. Their main idea is to take random samples from the configuration space and determine their status. The free configurations are used as the vertices of the road map and the local planner connects the nearby vertices. Lavalle and Kuffner [139] present rapidly exploring random trees. It aggressively probes and explores the configuration space by expanding incrementally from an initial configuration. The explored environment is presented as a tree with the initial configuration as its root. Potential field

path planning is another group of methods that aims at capturing the connectivity of free configuration space. It imposes mathematical functions over the space. It creates an artificial potential field across the robot map that directs the robot, represented as a point in configuration space, to the goal location from multiple prior locations [138]. The goal act as an attraction force on the robot and obstacles are repulsive forces. Superimpose of all forces is applied to the robot and it smoothly guides the robot toward the goal while avoiding known obstacles. Khatib and Chatila [124] proposed an extension to this method. They introduced the rotation potential field and task potential field in addition to attractive and repulsive forces. The rotation potential field incorporates the distance from the obstacle and relative orientation of the robot to an obstacle for the generation of repulsive forces. The task potential field considers the current robot velocity. It filters out the obstacle that has no effect on the near term potential based on robot velocity.

In mobile robotics the most common approach to configuration space representation is the 2D occupancy grid map. This chapter focuses on the use these maps for robot localisation. Additionally, this is used for path planning and obstacle avoidance. The path planning method is characterised as the search problem on this 2D occupancy grid map. This is defined as finding a sequence of adjacent cells as path that connects the current location of the robot to the goal location. Mobile robots often operate in an environment that is partially known to the system. Dynamic environments introduce new occupancies in the environment. Therefore, the recalculation of a new path with updated knowledge of the environment is required. Incremental heuristic search methods are used to improve the search by focusing the search and reusing the information from previous steps. On the other hand uninformed search methods such as, breadth first and depth first methods only rely on the current problem definition and have no further information about the environment. A* is one of the most widely known informed search approaches. It uses heuristic knowledge in form of approximations of the goal distances to improve the search. The distances are the combination of the path cost to the current cell and the estimation of the goal distance from the current cell. This method calculates the cheapest path to the goal through the current cell [170]. The shortest path generated with the A* algorithm on a grid map is often very close to obstacles. This increases the risk of collision by cutting occupied corners. Map smoothing methods are used to solve this problem. Convolution methods are used to smooth the map. This is presented as an inflated area around the obstacles. The cells in an occupancy grid map represent the presence of an obstacle at the location of the cell in the environment. An estimate of the posterior probability is calculated for the cells. Above zero probability is assigned to cells around the obstacles. This method presents the obstacle larger on the map than their physical size. The cell occupancy probability is used as the cell traversal cost for calculation of the shortest path. The newly generated path is larger than the previous method. However, this reduces the risk of collision.

Mobile robots often operate in unknown or partially known environments. Additionally, the occupancy of grid cells of the map changes in a dynamic environment. This change in the occupancy can block the previously planned path. A new path is required to be planned. The complete replanning of the path from scratch from the current location to the goal location in large environments is very costly and often adds up to substantial idle times. Several approaches based on the A* algorithm are proposed to improve this process [129, 184, 231]. Stentz [220, 221] present a heuristic search method that improves the required time for replan-

ning over repetitions of the A* search algorithm. The method avoids the replanning of the entire path. It incrementally modifies the previously calculated path. The modifications occur in a local area around the current robot location. This method has been extensively used in the robotics field, due to its combined efficiency of incremental and heuristic methods. Koenig and Likhachev [128, 130] introduce the D* lite algorithm as an alternative to the D* algorithm proposed by Stentz [220, 221]. It implements the same navigation strategy but it is algorithmically different. These methods are based on the assumption that the robot navigates in an unknown or partially known environment. The robot starts from the start cell and moves with cost one to the next cell until it reaches the goal cell. It always calculates the shortest path by assuming that the unknown cells are free. Afterwards, the robot follows this path either to reach the goal or a cell occupancy on its path changes. In the latter case the robot recalculates the shortest path from the current cell to the goal cell. However, these methods utilise the initial knowledge about the cell occupancy to only update few cells path near the current cell. They efficiently recalculate a shortest path from the current cell to the goal location. They recalculate only the goal distances that have changed or have not been calculated and are relevant to the shortest path. These incremental heuristic search algorithms are able to focus and plan a new path based upon previous solution [128, 130]. The D* algorithm is very complex and difficult to analyse. Despite the extreme usage of this method in robotics, it has not been extended and often used as a black box method. On the other hand the D* lite method is based on the previously proposed Lifelong Planning A* algorithm [129] by the same authors. Lifelong Planning A* is an incremental version of A* and very similar to the A* algorithm. The D* lite method is simpler, easier to understand and easier to extend. In the experiments of this chapter the D* lite algorithm is used for the calculation of the shortest path from the robot location to the goal location. Additionally, this method is used for the calculation of short local paths for collision avoidance.

7.6.2 Path Following

The navigation system is divided into path planning and path following. This approach of first planning a path and then following it is the most popular method of navigation in robotics. In mobile robotics the problem of how the robot gets to the final configuration includes following the shortest calculated path. This is the ability of the robot to follow any path in its configuration space. Only mobile robots with omnidirectional wheel configuration with unconstrained wheels are capable of following any path in the environment. Therefore, kinematic constraints must be expressed for nonholonomic robots for the path following problem. Kinematics describes the effect of the motion control action on the robot pose. In the case of a mobile robot operating in a planer environment the pose is denoted by its x, y location relative to an external reference coordinate frame and its bearing θ as an angular orientation. The robot with $\theta = 0$ is heading to the x axis of the coordinate system. The positive rotation produces the left turn rotation. The experiments of this chapter are carried out on a differential drive robot with four wheels configuration and a detailed explanation of the kinematics of differential drive robots is presented in Section 7.2.1.

Various studies have been carried out on motion control for nonholonomic mobile robots [3, 10, 45, 50, 56, 58, 162, 215]. The main idea of motion control is to follow a path that is

described by its position or velocity profiles as a function of time. The path is divided into line and circular segments. The control system follows these segments to drive the robot from its pose to the goal pose. The calculation of a precomputed trajectory with an open loop solution that disregards the measured robot pose is difficult. The robot is not able to correct this trajectory in a dynamic environment. Additionally, the precomputed trajectory is usually not smooth. A more applicable solution utilises the robot pose measurement information as the feedback information for the controller. The system divides the planned path into smaller segments and the robot performs motion control to reach these intermediate goals [45, 56, 58, 215]. Consider the situation with an arbitrary robot pose and a predefined goal pose. The task of the control system is to find the rotational and translational velocities to control the robot toward the goal pose. The translational velocity at time t is denoted as V_t and the rotational velocity at time t is denoted as W_t. Indiveri [112] presents a closed loop, time invariant and globally stable control law for a general 2D nonholonomic kinematic model moving in only one forward direction. This method incorporates the steering constraints that are usually neglected with a simpler unicycle kinematics model. Physical constraints of the robot such as its maximum allowed curvature are crucial to the path planning and control system of the robot. Canudas de Wit et al. [57] proposes a nonlinear controller for path following. This method assumes that the path curvature is uniformly bounded and is differentiable. It solves the unicycle path following problem. The drawback of this method is that the convergence for a non-linear path is guaranteed only for a certain initial conditions. This is due to the fact that the controller attempts to drive the robot to the closest point on the path. Some methods utilise a virtual target moving on the reference path for convergence to solve this problem [3]. Indiveri and Corradini [113] propose a non-linear path following controller for the kinematics model of a car-like robot with bounded curvature. There method is based on the work of Canudas de Wit et al. [57] and guarantees global asymptotical convergence to the path. The improved convergence speed is represented by periodical recomputation of control gains. This method is implemented and used on the mobile robot platform to perform the experiments in this chapter. Additionally, this method is used to predict the near future pose of the robot during the path following for collision detection and avoidance.

7.6.3 Obstacle Avoidance

The aim of mobile robots is to carry out missions in the environment. They aim to navigate in unknown, partially known or dynamic environments. They capture periodic measurements from the environment to update their knowledge of the surrounding area. This process includes or modifies the occupancy knowledge of the environment. The goal of the obstacle avoidance algorithm is to avoid collisions with objects. This focuses on changing the robot trajectory based on the current occupancy information. Obstacle avoidance methods are usually based on the local map that represents the local environment around the robot and they are implemented with respect to the current sensor readings. These methods utilise the goal pose and the relative pose of the robot to the goal to steer the robot around the obstacles. The simplest obstacle avoidance method is to follow the contour of each obstacle on the robot path to avoid it. The Bug algorithm utilises this method for obstacle avoidance [147, 148]. The Bug1 method circles each encountered obstacle once. Afterwards the robot departs from the obstacle at the closest

point to the goal. The Bug2 method follows the obstacle always on the left or right side. The robot departs from the obstacle when the direct line from start to goal is crossed. Khatib [126] proposed a real time obstacle avoidance approach for manipulators and mobile robots based on artificial potential field concept (cf. Section 7.6.1). This concept additionally has been used for solving the path planning problem. Later on Koren and Borenstein [30] proposed another method based on the potential field concept. To improve the ability of the robot to drive through the narrow spaces, Borenstein and Koren [31] developed the vector field histogram (VFH) method. They represent the local environment on a small local 2D occupancy grid. For obstacle avoidance they generate a histogram that represents the angle and probability of an obstacle in each direction. The steering direction is computed from this histogram. All open paths that the robot is able to drive through are detected and the path with the lowest cost is selected. Later, Borenstein and Ulrich [232, 233] extend the VFH method to yield VHF$^+$ and VFH*. The VHF$^+$ method includes the dynamics and the robot trajectory in the selection of the path. The idea is that the obstacle blocking a given direction also blocks all trajectories going through this direction. Khatib et al. [125] introduce the method that utilises the maximum free space around robot that can be reached without any risk of collision. This is used as band of bubbles that connects the start pose to the goal pose. Fox et al. [82] propose the dynamic window approach for reactive collision avoidance for mobile robots equipped with synchro drive. The synchro drive system is a two motor configuration where one motor rotates all the wheels to produce motion and the other motor turns all wheels to change the direction of the robot. Their method is designed to deal with constraints imposed by limited velocity and acceleration. This method approximates trajectories in short time intervals as circular curvatures. This results in a two dimensional search space of translational and rotational velocities.

The experiments of this chapter utilise an obstacle avoidance strategy that combines the concept of few aforementioned obstacle avoidance methods. It represents the robot local environment with an occupancy grid map. This map is generated from the 2D range finder measurements. The obstacles are inflated with the robot size for the usage of one point representation of the robot on the grid map. Similar to the bubble band method a safety bubble is used. The obstacles are inflated more to achieve the extra cushion area. This guarantees safe movement around the obstacles. Additionally the probability of the obstacles with inflated areas are calculated. Obstacles have the highest probability and this value decreases based on the distance from the obstacle. This probability is used to define the cost of traversal for each grid cell on the occupancy grid map. This local map is updated with every new measurement from the onboard sensors. The pre-planned path and the current robot pose are used to predict the near future poses of the robot. The proposed method by Indiveri and Corradini [113] for path following is used for the prediction of the future poses of the robot. Additionally this method is used for motion control system of the robot. The predicted poses are examined on the occupancy map for detection of collision with an occupied cell. The cell occupancy probability is used to calculate the risk of the collision. After the detection of collision the original planned path is traced on the local grid map to locate the first free cell on the path after the detected collision. This pose is used as a new intermediate goal. D* lite [128, 130] is used on the local map to calculate the shortest path to the intermediate goal. The robot follows this temporary path to avoid the obstacle. During the traversal of the temporary avoidance path the original path is observed periodically. The system returns to following the original path in case of a change in

Algorithm 6 The obstacle avoidance strategy utilises the planned path for detection and avoidance of obstacles. The occupancy information is stored in a local grid map. In case of collision on the planned path a temporary path is calculated for avoiding the obstacle.

Require: *path* // *The planned shortest path to the goal pose.*
  while Goal pose has not been reached **do**
    if *collisionAvoidancePath* **then**
      $plannedPathStatus \leftarrow checkTheCollisionStatusOfPlannedPath()$
      if $plannedPathStatus == COLLISION$ **then**
        Continue with the collision avoidance path.
      else
        Remove the collision avoidance path and continue with the original planned path.
      end if
    end if
    $captureSensorReadings()$
    $updateTheLocalMap()$
    $inflateTheObstaclesOnLocalMap()$
    $localiseRobot()$
    $predictFutureRobotPoses()$ // *Use the current robot pose and the path to calculate the near future robot poses based on the path following method.*
    $collisionStatus \leftarrow checkForCollision()$
    if $collisionStatus == NOCOLLISION$ **then**
      $followThePath()$ // *Use the path following method to follow the path.*
    else if $collisionStatus == COLLISION$ **then**
      $getTheFirstFreeCellOnThePlannedPathAfterCollision()$
      $collisionAvoidancePath \leftarrow calculateShortestPathToTheFreeCell()$
    end if
  end while

the collision status of the original path. This assists the robot to continue following the original path in the dynamic environment. Algorithm 6 presents the strategy used as collision avoidance method.

7.7 Case Studies

Experiments were carried out with the robot ZF-Volksbot (cf. Figure 7.4). The aim of the experiments is to evaluate the application for autonomous indoor mobile mapping. The generated 3D map of the environment is based on the 3D point clouds captured by the Z+F IMAGER 5010C. This scanner is capable of capturing more than 1 million points per second. The captured point clouds contain the range and the reflectance value of the measured surface. This scanner is capable of capturing the color of the environment. The color is captured by the integrated High Dynamic Range (HDR) color camera. This camera captures color for each measured surface point [86, 254]. The robot is a differential drive robot equipped with two powered standard wheels in the front and two castor wheels in the back. The wheel encoder sensors on the robot

Figure 7.10: Representation of the occupancy grid map generated with the GMapping SLAM algorithm. This map represents the occupied and free cells of the environment. The estimated robot poses are presented as two paths. The green path represents the robot pose estimation from the SLAM solution. The red path represent the robot pose estimation from wheel odometry. The accumulation of errors in the odometry calculation is the source of the divergence of the two paths.

are used to provide the odometry information. The Hokuyo UTM-30LX-EW [106] captures a 2D slice of the environment. This 2D range finder is constrained by a horizontal field of view of 270°. The guaranteed measuring range of this sensor is 30 meters. The captured 2D scans from the environment are used to generate the 2D occupancy grid map of the local environment of the mobile robot.

All the experiments and implementation of the robot localisation, mapping and navigation systems are carried out as part of the Robot Operating System (ROS) [180]. The GMapping [96, 97] method is used to solve the simultaneous localisation and mapping problem. This SLAM method uses the odometry information in addition to scan matching methods to estimate the robot pose. Figure 7.10 presents a sample captured 2D map of an indoor environment with the GMapping SLAM method. The green path presents the traversed path by the robot generated from the SLAM pose estimations. The red path represents the odometry pose estimations. To capture the complete 3D model of the environment 3D scans are captured from several view points. 3D scan matching algorithms such as ICP based methods are used in combination with the 2D SLAM pose estimation for precise 3D scan matching.

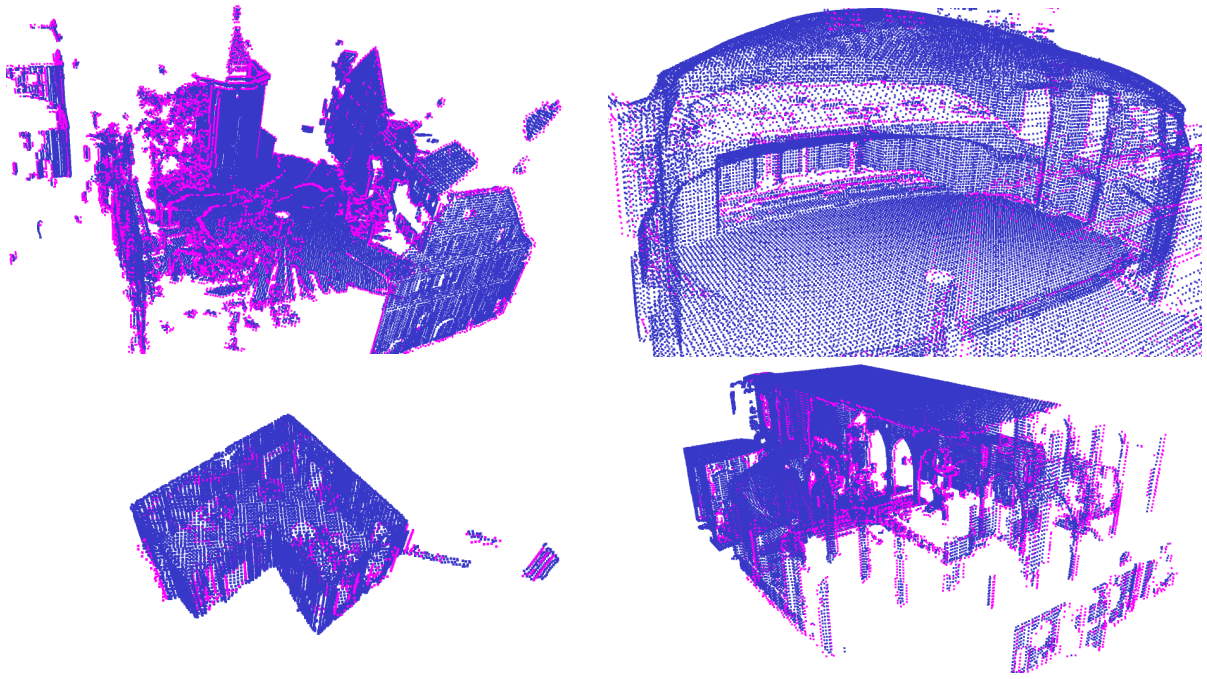


Figure 7.11: Results from the hole detection method. The detected hole voxels are presented as pink points. The blue points represent the octree voxels of the queried scan. Scans are captured from the city Wangen im Allgäu, Germany. Top Left: Scan from the city center. Top Right: Scan from a school theatre. Bottom Left: Scan from the city hall. Bottom Right: Scan from the Saint Martin church.

The main contribution of this chapter is the proposed next best view algorithm. The method detects hole voxels in a voxelised representation of 3D point clouds and generates a sequence of scanning positions to maximise the information gain from the obscured data. Figure 7.11 presents the detected holes from single 3D scans captured from a few scanning projects in Wangen im Allgäu, Germany. The pink points present the 3D border of seen and unseen environments. An octree data structure is used to represent the environment as a voxelised environment. The blue points present the occupied voxels of the octree. The hole voxels are detected as unseen voxels with an empty neighbouring voxel. To improve the efficiency of the method the search for hole voxels is limited to the neighbouring voxels of occupied voxels of the octree.

Figure 7.12 shows the detected floor of these scans. The floor voxels are required for the calculation of the potential scan positions. These positions are selected based on the detected floor points and the scanning unit height. The selected potential scanning position are ordered. Figure 7.13 shows the calculated potential positions with the color coding to represent the next best scanning positions. The potential position points from the scan position with the maximum information gain are represented as red and points from the scan position with the minimum information gain are represented as green. The information gain is calculated from the number of visible hole voxels from each potential scan position.

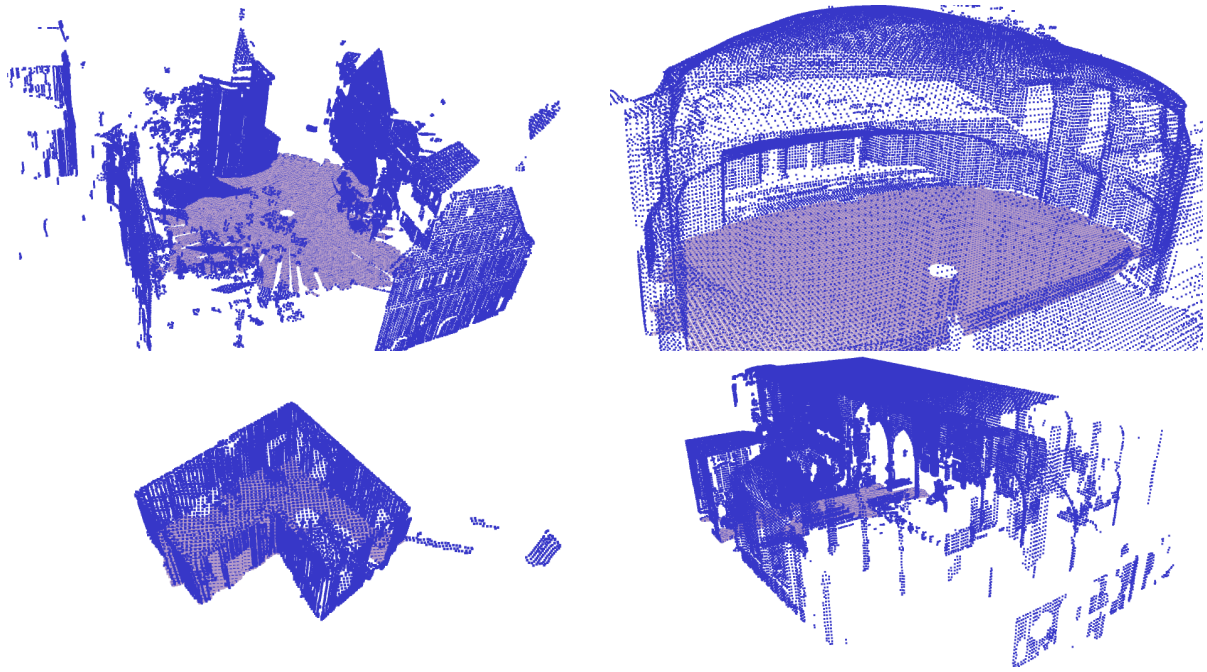


Figure 7.12: Results from the floor detection method. The detected floor voxels are presented as grey points. The blue points represent the octree voxels of the queried scan. Scans are captured from the city Wangen im Allgäu, Germany. Top Left: Scan from the city center. Top Right: Scan from a school theatre. Bottom Left: Scan from the city hall. Bottom Right: Scan from the Saint Martin church.

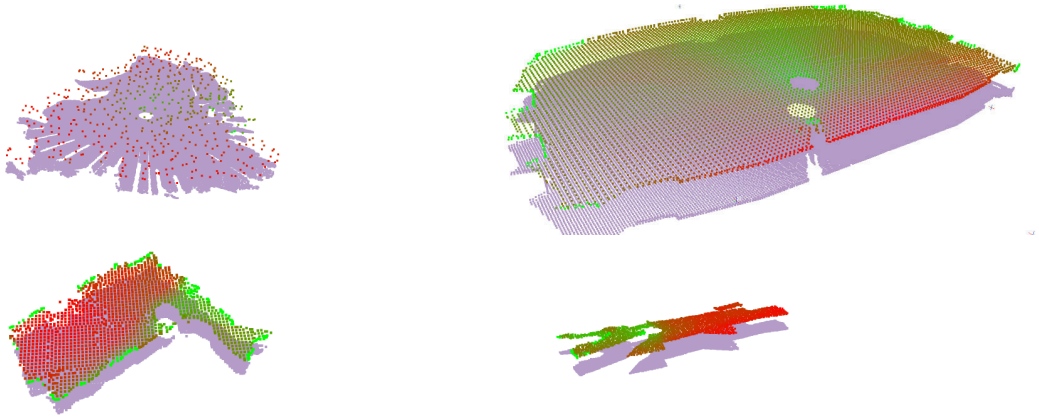
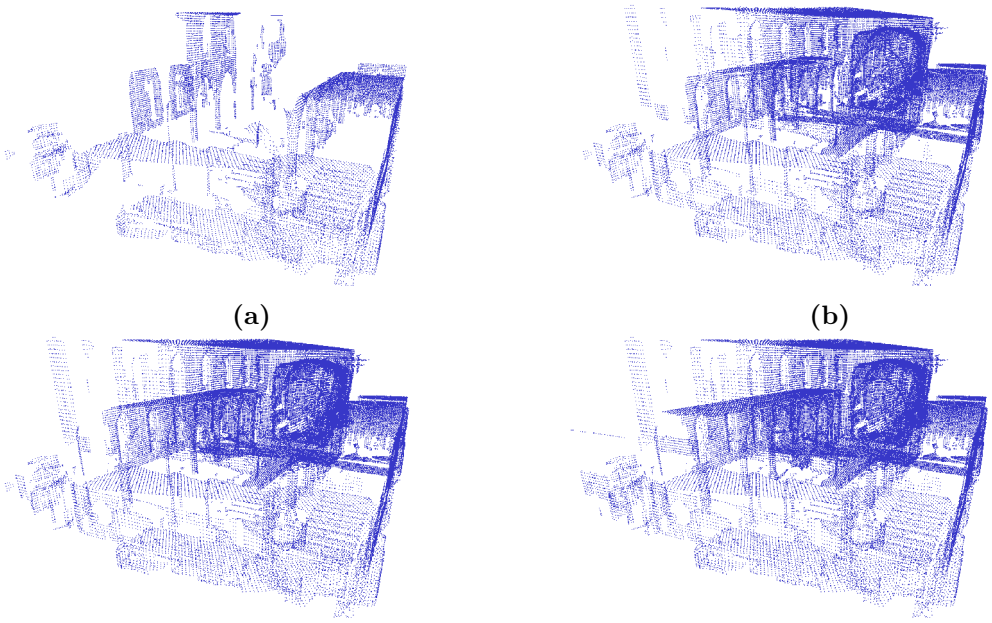
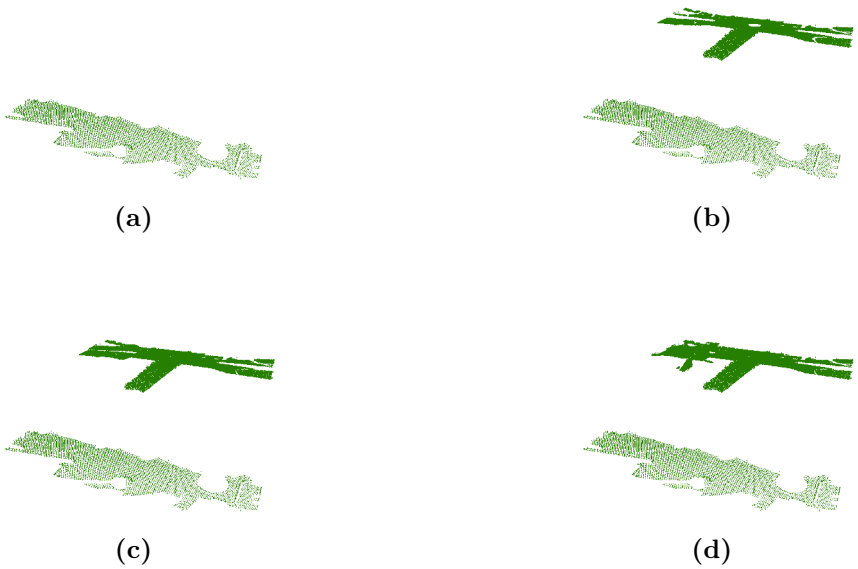


Figure 7.13: Representation of the selected potential next scan positions. The potential positions are ordered based on the maximum information gain of each position. The positions with a higher number of visible hole voxels are presented as red and the ones with a lower number of visible holes are presented as green. The grey points represent the detected floor voxels of the queried scan. All the scans are captured in the city of Wangen im Allgäu, Germany. Top Left: City center. Top Right: School theatre. Bottom Left: City hall. Bottom Right: Saint Martin church.



(a) **(b)** **(c)** **(d)**

Figure 7.14: Representation of the octree voxel accumulation. The centroid of the occupied voxels of the accumulated octree are presented as blue points. It represents the octree accumulation of four scans.



(a) **(b)** **(c)** **(d)**

Figure 7.15: Representation of the floor voxel accumulation. The centroid of the floor voxels of the accumulated floors are presented as green points. It represents the accumulation of the floor voxels from four scans. The detected floor from the first scan is completely separated from the other scans. However, the method is capable of correct detection and accumulation of the floor voxels.

Figure 7.16: Representation of the hole voxel accumulation. During the accumulation the previously detected hole voxels are filtered based on the visibility status of them from the new scan position. The hole voxels detected from the new scans are additionally inspected to remove the voxels that are visible from one of the previous scans. This process accumulates hole voxels with unseen status after the accumulation of the scans. Left: Representation of the detected hole voxels from one scan. The shadow of the person in the scan is detected as a hole in the captured scan. Right: Representation of the accumulated holes after the addition of the second scan. This scan filled the obscured environment behind the person. Therefore, the hole voxels representing this environment are removed from the accumulated hole voxels.

To capture an occlusion free and complete model of a 3D environment several scans are required. The NBV algorithm calculates the best next scanning position for the maximum information gain. To calculate the next scan position the captured data from previous scans are accumulated. The proposed NBV method accumulates an octree to represent the entire environment. Voxels of newly acquired scans are added to the previously accumulated octree. This represents the entire captured environment in one octree. This octree is used for the estimation of completeness of the data acquisition and to guide the NBV algorithm for calculation of the next scan position. To improve the memory efficiency of the algorithm this octree only contains the occupancy information of the voxels. The point information of the voxels is stored in separate local octrees for each scan. Figure 7.14 presents the accumulation process of a few sequential scans acquired from the Saint Martin church data set. In addition to the occupancy information the NBV algorithm accumulates the detected hole and the floor voxels. The floor voxel accumulation process adds the newly detected floor voxels from the last captured point cloud to the accumulated floor voxels. Figure 7.15 shows the progress of the floor voxel accumulation on the same sequence of scans from the Saint Martin church data set. The centroid of the detected floor voxels are presented as green points.

The hole voxel accumulation process requires an additional step. First the newly detected hole voxels are examined. A filtration method discards the voxels that are seen or occupied from previous scans. Additionally, the new scan changes the status of previously detected hole voxels. The visible and occupied hole voxels based on the new scan are removed from the list of

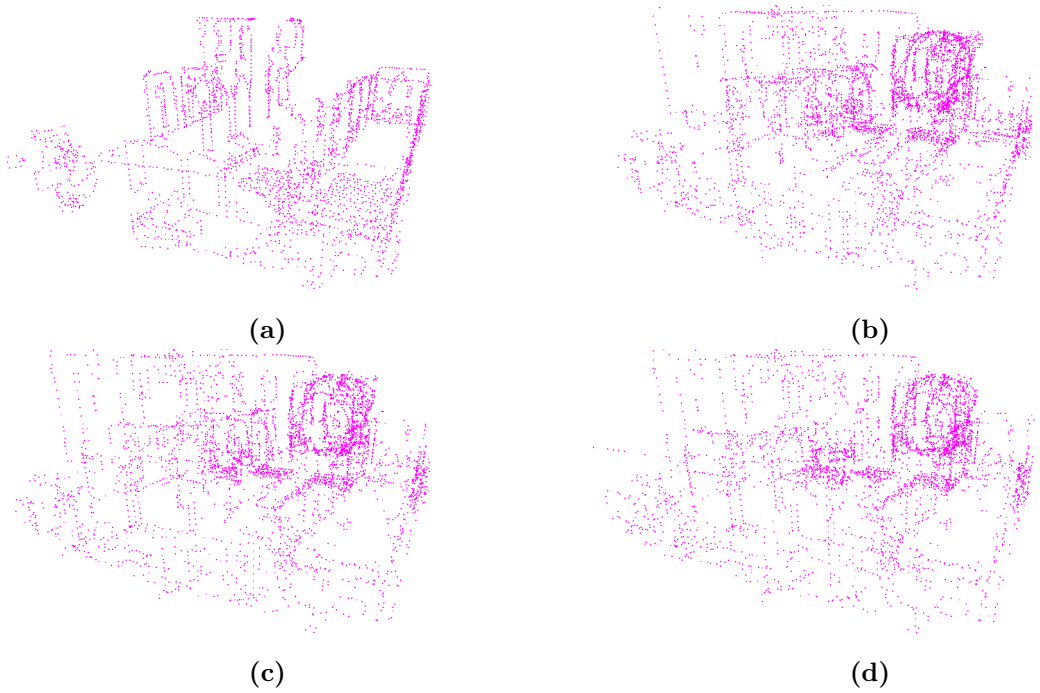


Figure 7.17: Representation of the hole voxel accumulation. The center of the voxels from the accumulated hole voxels are presented as pink points. This represents the accumulation of the hols from four scans.

accumulated hole voxels. This process guarantees that the seen hole voxels are removed from the process. Figure 7.16 shows two subsequent scans from an indoor environment. The detected hole borders from one scan are removed during the accumulation of the second scan. The detected hole voxels from the first scan are presented as pink points. These points represent the center of the hole voxels. The accumulated hole voxels after the addition of the second scan and the removal of the filled hole voxels are presented as blue points. These points represent the center of the hole voxels. Figure 7.17 presents the hole voxel accumulation of the sequence of scans captured from the Saint Martin church. Similar to the octree and floor accumulation process the same sequence of scans are used for generation of these figures.

After the accumulation process the NBV algorithm calculates a set of potential next scan positions. These potential scan positions are color coded to present the areas with the maximum information gain for scanning. Figure 7.18 presents the color coded potential next best view positions from the same sequence of scans. The colored points represent the potential positions with the highest information gain as red and the positions with the lowest information gain as green. The Saint Martin church scans were captured with the help of an operator by setting the scanner on tripod at scan positions. All scan positions were chosen based on the expertise of the operator and the scanning process was stopped based on the decision made by the operator. The algorithm is able to process the data even when the operator does not follow the instruction from the system, for example the second scan was taken completely separated from the first scan and the selected position was not the calculated position of the NBV method.

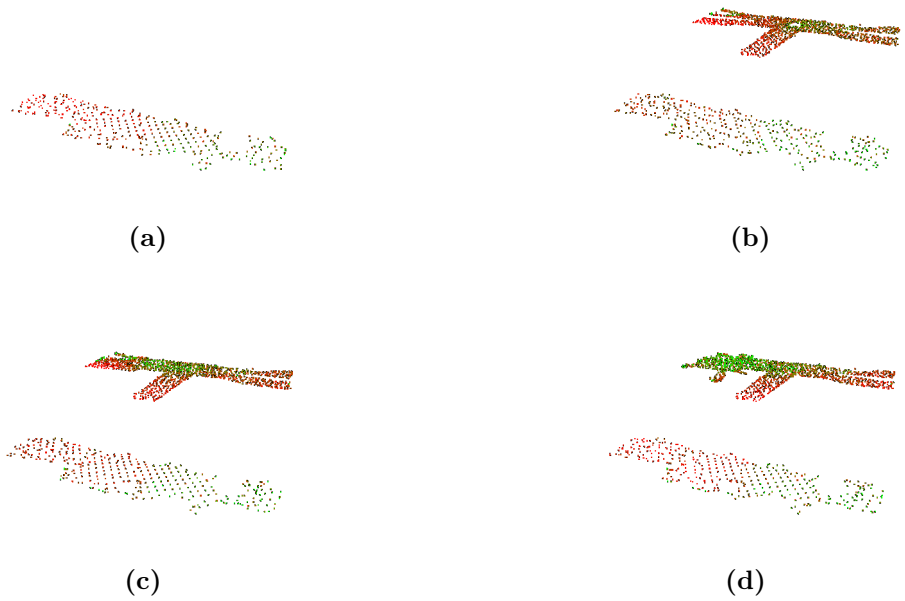


Figure 7.18: Representation of the potential next best view positions. The potential positions are selected from the detected floor voxels with addition of the scanning unit height. The points are color coded to represent the position with the highest information gain as red and the positions with the lowest information gain as green. This represents the calculated potential next scan positions from four scans.

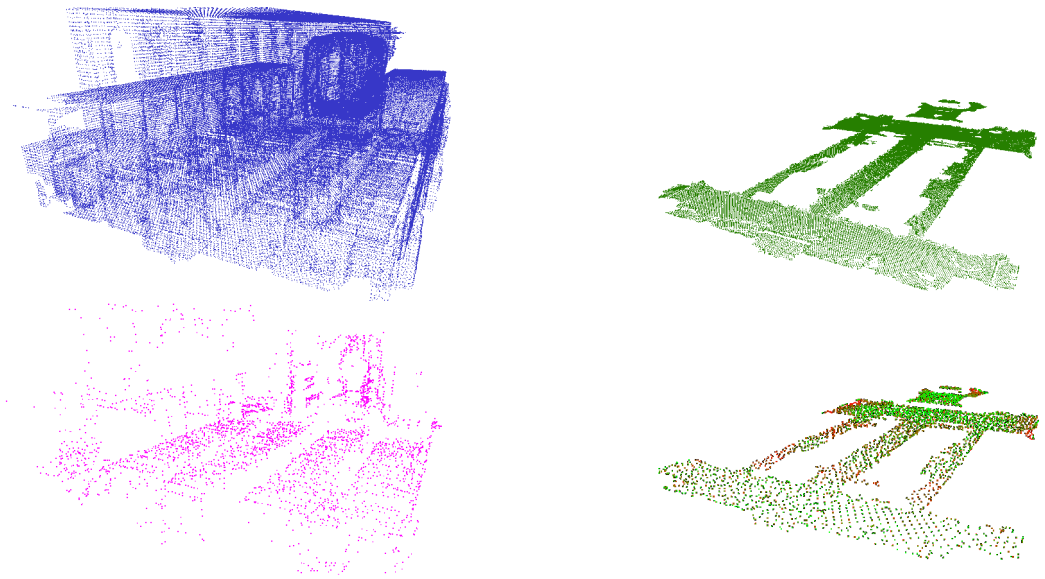


Figure 7.19: Representation of the status of the environment after accumulation of 21 scans. Top Left: Representation of the accumulated octree. Top Right: Representation of the accumulated floor voxels. Bottom Left: Representation of the accumulated hole voxels. Bottom Right: Representation of the calculated potential next scan position from the accumulated data set.

Figure 7.20: Color representation of the registered point clouds from the Saint Martin church in Wangen im Allgäu, Germany. The data set contains 21 scans captured from the inside of the church. The data is captured with Z+F IMAGER 5010C scanner. Each scan contains around 45 million points. The color information is captured with the help of the HDR camera of the scanner. The data is registered and a complete 3D model of the environment is generated.

Figure 7.19 represents the status of the environment after accumulation of 21 captured scans. The remaining hole voxels are mostly due to the rows of seats in the church. These narrow spaces are not visible from any scan position used for capturing the entire environment. The coverage of narrow spaces between each row of the church seats are not the focus of these experiments. More precisely this experiment focuses on the generation of the full accumulated model of the environment. The results present the utilisation of the hole detection and the NBV algorithm to guarantee the complete model generation of the environment with a minimum number of required scan positions (cf. Figure 7.20).

A similar experiment is carried out in office-like indoor environment. After capturing each scan the new scan is registered to the previous scans. The newly captured points are accumulated into the project accumulated octree and new holes and floor voxels are detected. Additionally the floor and hole information are accumulated. The NBV process is calculated at each scan position and the next scan position is selected from the suggested potential scanning positions. After selection of the next best scanning position a path is generated from the current scan pose to the next best scan pose. The ZF-Volksbot mobile robot is used to reach the new scan position. The robot follows the generated path to arrive at the goal position. The D* lite [128, 130] path planning method is used to generate this path on the 2D occupancy grid map representation of the environment. The path following method proposed by Indiveri [112, 113] is utilised for path following. The experiment contains seven captured scans with the Z+F IMAGER 5010C scanner on the mobile robot. The process was stopped by the operator after these scans.

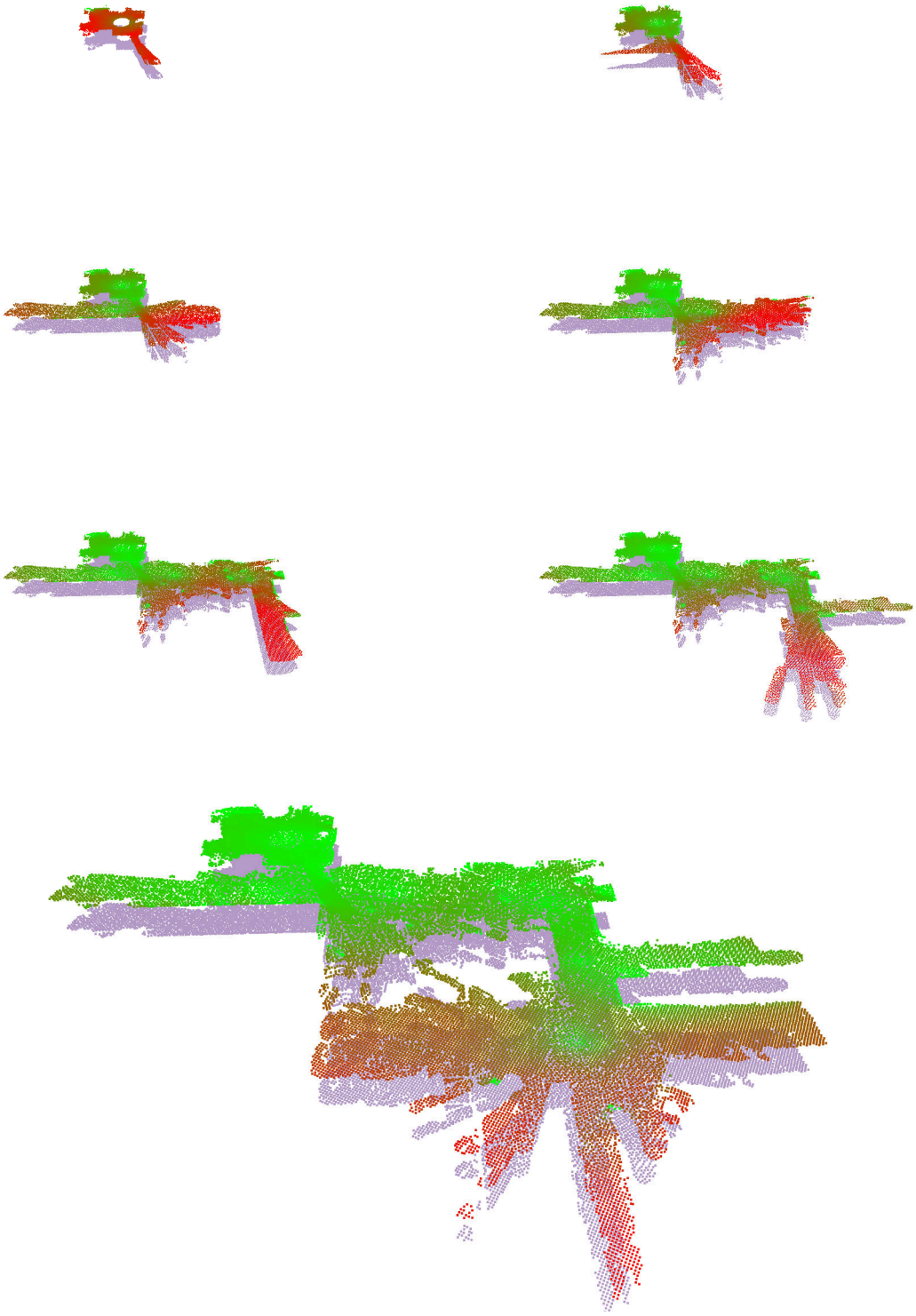


Figure 7.21: Representation of the potential next best view positions. The potential positions are selected from detected floor voxels with addition of the scanning unit height. The detected floor voxels are presented as grey points. The potential next best view points are color coded to represent the position with highest information gain as red and the positions with the lowest information gain as green. This figure presents the progress of the NBV algorithm for a sequence of scans in an indoor environment.

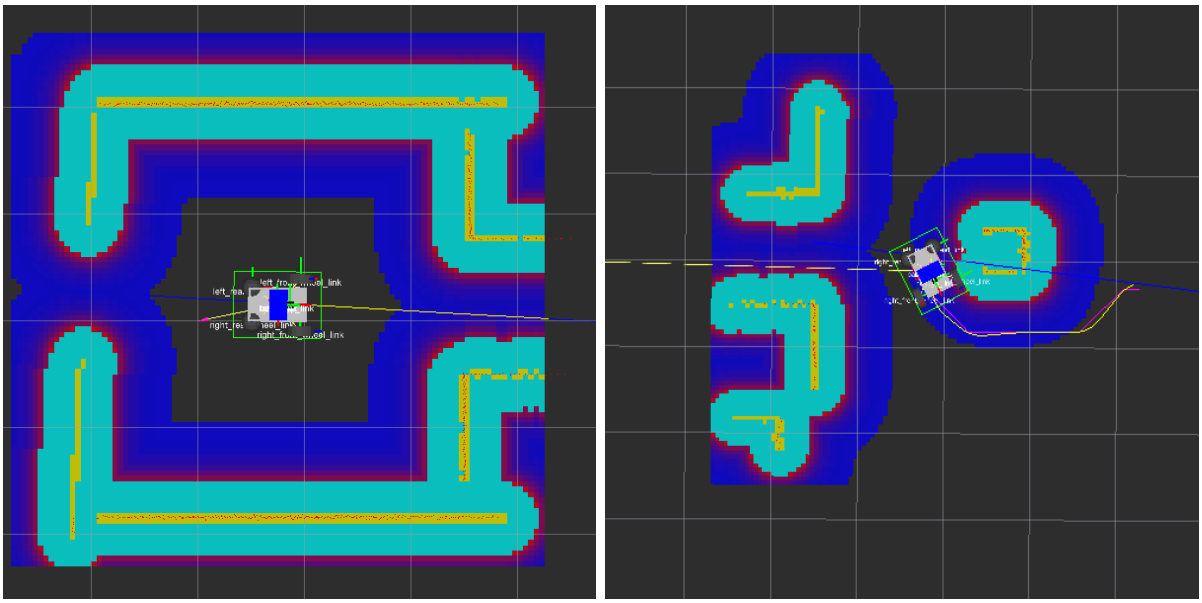


Figure 7.22: Left: Represents the 2D occupancy grid map of the environment. The map is generated with the cell accuracy of 5cm. The larger grid shown in the image is the representation of 1m grid cells. The inflated obstacles are presented with highest probability of collision at the obstacles with yellow color. The probability of collision reduces with increased distance to the obstacle. The planned path is presented as blue line and the predicted near future robot poses are presented as the yellow path. Right: Represents the temporary collision free generated path for the robot to avoid the obstacle. This method detects a free cell on the planed path after the detected obstacle (cf. the blue path). The blue path intentionally is chosen to intersect with the obstacle. The pink path is generated from the current robot pose to this intermediate goal position. The yellow path represents the prediction of the robot poses based on the current robot pose and the generated path for following. The path following algorithm is utilised to calculate this estimation.

Figure 7.21 presents the floor accumulation with the potential next scan position estimation. The sequence of scan positions are selected from the area with maximum information gain. These regions are presented with the red color points.

Figure 7.22 presents the proposed method for representation of 2D grid maps of the environment with inflated objects. The robot calculates local segments of the global path for following. The path following method is used to predict the near future poses of the robot during the path following process.

The planned path is presented as blue and the predicted near future robot poses are presented as yellow. The proposed obstacle avoidance method in this chapter is used to avoid collision with dynamic and static obstacles in the environment. In case of collision detection a temporary collision free path is generated for avoiding the obstacles. The algorithm selects a free grid cell on the planned path after the detected obstacle as an intermediate goal and a collision free path is generated from the current robot pose to this goal. The temporary collision free path is presented as pink.

Figure 7.23 presents the behaviour of the robot with respect to a change in the local occu-

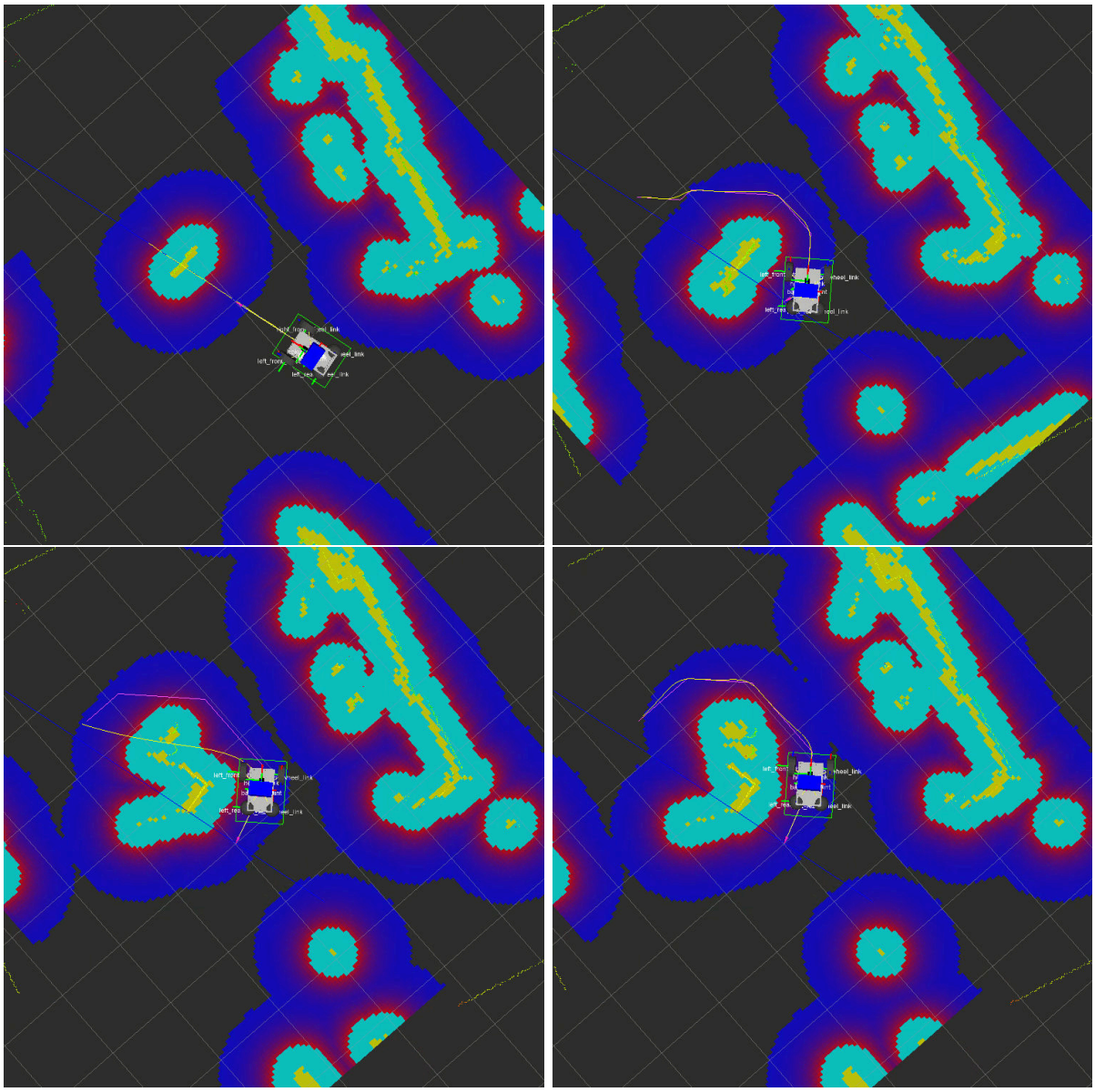


Figure 7.23: Representation of the dynamic obstacle avoidance. Top Left: The obstacle is detected on the robots planned path. Top Right: A temporary collision free path is generated for the robot to avoid the obstacle. Bottom Left: The local occupancy of the environment changes due to the dynamic obstacle. A collision is detected on the previously generated temporary collision free path. Bottom Right: A new intermediate goal position is selected and a revised collisions free path is generated.

pancy of the environment. During the path following process the robot continuously measures the local environment. These measurements are used to detect the local changes in the local occupancy and the possibility of collision. After the detection of a collision in the local environment a temporary collision free path is generated. This path is generated for the robot to

avoid the obstacle. This is presented as the pink path in Figure 7.23. The robot follows the newly generated path to arrive at a free point on the original path after the obstacle. If the dynamic obstacle moves and the original planned path is free of collision, the robot discards the temporary path and continues to follow the original planned path. Additionally, if the dynamic object moves and blocks the temporary path a new collision free path is generated for the robot.

7.8 Summary

This chapter presents an application toward a fully autonomous mobile mapping system for indoor environments. This presents the wheeled mobile robots to achieve this goal. The kinematics of differential drive mobile robots are presented and used for the design of the control system of the robot. This chapter demonstrates the four steps of perception, localisation, cognition and motion control. These are the main procedures of the operation cycle of an autonomous mobile robot. The robot perception considers the robot sensors. It utilises these sensors to obtain information about its internal status. Additionally, these sensors are used to capture the local environment of the robot. The ZF-Volksbot mobile robot platform used for the experiments of this chapter is equipped with wheel encoders and 2D/3D laser scanners. The wheel encoders provide the odometry information of the robot. The 2D range finder is used to measure the local environment of the robot for the 2D map generation and obstacle avoidance system of the robot. The 3D laser scanner on the robot captures the 3D model of the environment. Robot localisation produces the required pose information for map generation. The robot pose in the map is examined to determine the accomplishment of the goal. The generation of a map in an unknown environment requires a precise localisation. On the other hand a precise map of the environment is required for the robot to localise itself in the environment. Both localisation and mapping are achieved by solving the Simultaneous Localisation And Mapping (SLAM) problem. Two popular 2D SLAM methods namely GMapping [96, 97] and Hector [132] SLAM methods are presented in this chapter. GMapping method is used on the ZF-Volksbot mobile robot for generation of the 2D occupancy grid maps of the environment. Additionally, they are used for precise robot pose estimation. After the generation of the map and the localisation process an autonomous mobile robot gathers information from the environment and determines its next goal. The main contribution of this chapter is the proposed NBV algorithm that is based on the novel hole detection method. The aim of an indoor mobile mapping system is to generate a complete 3D model of the environment.

The cognition process of this system detects a sequence of scanning positions to maximise the information gain with a minimal number of scan positions. This chapter proposes a hole detection method to determine the obscured environment. This method detects the 3D border of unseen areas. The border of a hole is defined as an unseen voxel adjacent to an occupied voxel with minimum one empty neighbouring voxel. The status of voxels are determined with a ray tracing method. Afterwards, the proposed NBV method detects the floor voxels. This is used for the calculation of a set of potential next best scanning positions. The potential position with maximum visibility of hole voxels is selected as the next best scan position. This defines the next goal of the mobile robot. The motion control system of the robot calculates the shortest path to the goal from the robot pose. The D* lite [128, 130] algorithm is used for calculation

of the shortest path. Afterwards, the path following method proposed by Indiveri [112, 113] is used to follow the shortest path to the goal position for the next data acquisition. This chapter proposes an obstacle avoidance method that combines three procedures. First, inflated obstacles are mapped on a 2D occupancy grid map. The values of grid cells define the cost of robot traversal. Second, the path following method and robot pose is used to estimate the near future poses of the robot. The status of these poses are examined for detection of collision on the robot path. Third, the occupancy information of the local environment is updated with the onboard sensors. In case of collision detection the obstacle avoidance system tracks the planned path to a free cell after the obstacle. This cell is used as a temporary goal. A new collision free path is generated from the robot pose with the updated occupancy information of the local environment. The robot continues the path following until the goal position for scanning is reached. The entire process iterates to capture the complete 3D model of the environment.

Chapter 8

Conclusion

This thesis focuses on the exploitation of 3D point cloud processing methods for documentation and mapping of environments. First an introduction of related work and previous work of the author is presented. This is followed by short introductions on 3D point cloud representation and registration methods. Large point clouds are captured from the environment with the help of modern 3D measurement systems. Data management and processing of these large point clouds are challenging. The panorama generation methods proposed in this thesis present a data structure for captured point clouds. The data is presented in several panorama images to represent the range, reflectance and color information from the captured environment. The 2D representation of 3D point clouds introduces the conventional 2D image processing algorithms to the domain of 3D point clouds. Several projection methods are utilised for the generation of the panorama images. Experiments in this thesis present the Pannini and the Mercator projections as the most suitable projection methods for point cloud processing. Although they outperform the equirectangular projection mainly for feature-based registration method, the equirectangular projection produces satisfactory results. During the panorama generation a point map is used to accommodate the correspondences between the 3D points and the image pixels. The advantages and limitations of the proposed panorama images follow the definition of each method. Due to the limitation of some projection methods the point cloud is divided into sub sets with a horizontal field of view that is adequate for the utilised projection method. Vertical field of view is also an important attribute of the projection. Because of the introduced distortions of the Pannini projection with large vertical field of view, it is recommended to use the Pannini projection for outdoor environment that require smaller vertical field of view. On the other hand the Mercator projection is preferred for indoor environments that require larger vertical field of view. These panorama methods and maps are the basis of most of the algorithms and methods proposed in this thesis.

Full 3D models of environments consist of multiple scans from different viewing angles. The registration process is the calculation of the relative transformation between the scan poses. The convergence of registration methods especially the well-known ICP algorithm, depend on the initial estimates of scan poses. However, these information are not available all the time or the estimations are highly inaccurate. The feature-based registration method proposed in this thesis aims at the calculation of the initial pose estimates for already existing projects lacking

pose information. This is necessary for applications such as the CASTLE3D with the focus of capturing a few individual findings on the excavation. These projects generally lack the pose planning or pose estimates due to the limited time. However, when the focus is on capturing the entire environment the proposed NBV method is recommended. The combination of pose estimation and autonomous robot navigation increases the efficiency of the data acquisition. A survey and evaluation of the modern 2D feature detector and descriptor methods are presented for point clouds based on the generated panorama images. The matched features between two point clouds provide the required correspondences for the calculation of the relative transformation matrices. Several feature filtration and matching methods are presented. A RANSAC-like registration method calculates the relative coarse transformation matrix. This provides the initial pose information required for the fast convergence of the ICP algorithm. The combination of several projection methods and feature detector and descriptor methods is studied. A real world evaluation of these methods show the impact of the selected method on the registration results. The combination of SIFT features with the Mercator and Pannini projections are the most suitable methods for automatic registration of 3D point clouds. Additionally, a comparison of the feature-based registration method with other initial pose estimation methods is presented. The proposed method outperforms other pose estimation methods. The improvements are both on the generated result and the required effort for the calculation of the pose estimates. However, the drawback of the proposed feature-based registration method is the reliance on 2D features and panorama projection methods. Future work for the feature-based registration method includes the investigation into projection methods with smaller distortion or dynamic projections that calculates small image patches for the calculation of the descriptor of the feature. These image patches are generated by selecting the feature as the center of the projection and utilising a small field of view around the feature to minimise the distortion introduced by the projection. Furthermore, the further investigation into 3D features for the feature-based registration method is required.

Point cloud registration is a very time consuming process. Methods for point reduction is utilised to improve the required time for registration. Point cloud reduction methods based on panorama images further increase the benefits of panorama images for point cloud processing. Three reduction method are introduced based on panorama images generated from point clouds. The first method utilises the generated point correspondence map during the panorama generation process. The second method relies on the reverse equations of panorama projections. The third method uses the conventional 2D image scaling methods in addition to the aforementioned methods for further reduction of the point clouds. The performance of several panorama projection methods with the proposed reduction methods is evaluated. The presented experiments show the efficiency of reduction methods to drastically reduce the size of the point cloud while retaining all the necessary information for feature-based registration. Other reduction methods such as octree reduction method introduce limitations for the panorama generation method, thus rendering them as inadequate methods for feature-based registration methods.

The use of panorama images for point cloud processing is extended to point cloud compression methods. The captured measurements from the environment are usually stored in vendor specific binary file formats. An ASCII representation is the preferred format for data exchange. An ASCII representation of the point cloud generates very large files with limited parsing capabilities. Data compression is crucial for transmission and archiving of point clouds. LAS and

LAZ file formats are proposed by ASPRS as a binary and compressed binary file format for point clouds. Panorama images from point clouds introduce the 2D image compression methods to point clouds. 3D point clouds are mapped onto range, reflectance and color panorama images. Range encoding methods are proposed to encode the measured range information from floating points onto 24 bit integer values. This value is mapped either on a three channel image or three separate grey scale images. Several lossless compression methods are evaluated. The comparison of image compression methods with octree based compression and industry standard LAS and LAZ methods are presented. The results show a high compression rate with conventional image compression methods, especially the PNG compression method outperforms all the other lossless compression methods. It also outperforms the lossy JPEG method with the maximum quality setting. Furthermore, the effects of the lossy compression methods such as the JPEG compression method on point clouds are studied. The lossy compression methods are not recommended for the data archiving due to the loss of data. However, they are applicable for fast data transmission and some visualisation methods. The range encoding method based on three separate grey scale images is proposed to avoid the difference in compression ratio on chrominance and luminance of color images. A filtration method is presented to reduce the number of outliers from the recover point cloud. The lossy methods are able to generate images with higher compression ratio however, the generated artefacts are unavoidable with these methods. The introduced error results in a very imprecise recovered point cloud. Finally, the effects of several panorama projection methods on the compression methods are evaluated. The only lossless projection method is the equirectangular projection when the scan resolution is used for the generation of the panorama image. Other projection methods introduce some data loss due to the mapping of two or more 3D points onto one image pixel. The future work for point cloud compression methods includes the investigation of other 2D compression methods on panorama images and also developing a new binary format that contains the entire compressed point cloud based on PNG compression method to avoid several compressed images for range, reflectance and color.

Several point cloud processing methods are proposed based on the generated panorama images from the captured point clouds. Point cloud processing methods are used to introduce a framework for the documentation of archeological excavation sites. This framework improves the process of archeological excavation and documentation. The aim of the framework is to substitute the conventional documentation methods. The framework is used to combine the 2D images, 3D point clouds, archeological notes and CAD models for the presentation of the excavation site. 3D measurement systems capture the environment with a very high rate thus, generating very dense point clouds representing the environment. These scans are often only used as the snapshot of the excavation. They are not used for the data analysis and the generation of the knowledge. The core of the presented framework is a Computer Aided SysTem for Labelling archeological Excavations in 3D (CASTLE3D). This is a novel approach for generating the digital documentation of the excavation sites. The results of the experiment demonstrate the ability of CASTLE3D for the on-site semantic mapping of the archeological excavation sites. It increases the accuracy of the generated data and offers the access to the original data for further studies. CASTLE3D also presents a practical use case for point cloud processing based on panorama images and especially the compression of the data. The feature-based registration method is required for the registration of scans from few positions on the excavation site. CASTLE3D

in combination with Adiuvabit as the back-end database of the framework assist archeologist to execute the workflow of the proposed framework. Further developments and studies are required to incorporate the Adiuvabit data base into the CASTLE3D. The introduction of a web based service and a centralised data base for the framework is necessary. This introduces the possibility for generation of a social community around the digital documentation of the archeological excavations.

To further assist the data acquisition process an application toward a fully autonomous mobile mapping system for indoor mapping environments is presented. A novel sensor placement method is proposed to reduce the number of required scans for capturing an environment completely in 3D. The method relies on the detected holes in the environment for the calculation of the next best scanning position. The holes represent the obscured environment in the point clouds. They are used to maximise the information gain with each scan. The holes are defined as the border between empty and unseen areas. The border of the hole in a voxelised environment is defined as the unseen voxel adjacent to an occupied voxel with minimum one empty neighbouring voxel. The next best scanning position is selected from the reachable floor positions of the environment. The position with the maximum visibility of hole voxels is selected as the next best scanning position. The autonomous navigation system of the robot utilises the next best scanning position to plan and execute the path from the current robot position. The proposed obstacle avoidance method is used to reach the goal position without any collision. The robot platform provides a very good pose estimate that are used for the registration process. The acquisition process continues until the entire environment is captured. The sensor placement method is designed as an generic algorithm to both assist the mobile mapping system and the archeologist on-site during the data acquisition process to accelerate this process and to provide a measure of completeness of the captured data. Use of NBV method is recommended especially when the focus is on capturing an entire environment. In comparison to feature-based registration method that focuses on the registration of the existing projects or projects with few selective scan positions the NBV method provides a complete set of processes including data acquisition, planning, navigation and registration of the entire environment. To further improve the NBV method distances to the previous scan positions and the overlapping area between the new position and previous scans have to be considered during the selection of the next best scan position. Furthermore, reachability of the NBV position need to be investigated during the selection process.

Finally to summarise, this thesis presents several 3D point cloud processing methods for documentation and mapping of environments. Some of these methods rely on the panorama images generated from the point clouds. The presented projection methods are examined and evaluated with these processing methods. To register the 3D point clouds without initial pose estimates, due to the lack of the data during the acquisition process on existing projects or the limitation of time for planning or inaccurate estimation of the initial pose, the feature-based registration method is introduced. The feature-based registration method outperforms other initial pose estimation methods without the requirement of the installation of markers or additional setup. Point cloud processing is very time consuming process. Point cloud reduction methods reduce the processing time for the registration methods. The panorama images generated for the feature-based registration are exploited for the reduction of the point clouds. The drastically reduced point clouds contain the required information for the feature-based registration.

To further utilise the panorama images, point cloud compression methods based on panorama images are investigated. The image based point cloud compression methods outperform the industry standard LAZ compression method. The compressed images are the most efficient method for transmission and archiving of the large point cloud data sets. All these methods are utilised to increase the efficiency of the documentation of the archeological excavation sites. The proposed framework is able to use these processing methods for the organisation and registration of the captured data from the excavation. CASTLE3D is the suitable enhancement to the conventional documentation methods. The back-end database of the framework assist archeologist during the data analysis process. To increase the efficiency of the data acquisition process, especially when the focus is on capturing an entire environment, an application towards a fully autonomous indoor mobile mapping system is introduced. The mobile mapping system and the NBV method improve the data acquisition process by minimising the required number of scans. The NBV method provides additional information to assist the operator during the planning process. The combination of all the proposed methods in this thesis demonstrate the required steps for documentation and mapping of environments with the help of 3D point clouds.

Bibliography

[1] M. Agrawal, K. Konolige, and M. R. Blas. CenSurE: Center surround extremas for realtime feature detection and matching. In *Proceedings of the 10th European Conference on Computer Vision*, volume 5305 of *Lecture Notes in Computer Science*, pages 102–115, October 2008.

[2] D. G. Aguilera, P. R. Gonzalvez, and J. Lahoz. An automatic procedure for co-registration of terrestrial laser scanners and digital cameras. *ISPRS Journal of Photogrammetry and Remote Sensing*, pages 308–316, 2009.

[3] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino. Closed loop steering of unicycle like vehicles via lyapunov techniques. *IEEE Robotics Automation Magazine*, 2:27–35, March 1995.

[4] P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. AVENUE: Automated site modeling in urban environments. In *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling (3DIM '01)*, pages 357–364, 2001.

[5] American Society of Photogrammetry and Remote Sensing. ASPRS LAS format, Specification for LASer exchange file format. http://www.asprs.org/.

[6] G. Antonelli, S. Chiaverini, and G. Fusco. Exciting trajectories for mobile robot odometry calibration. In *Proceedings of the 7th IFAC Symposium on Robot Control*, pages 429–434, Wroclaw Poland, September 2003.

[7] G. Antonelli, S. Chiaverini, and G. Fusco. An odometry calibration method for mobile robots based on the Least-Squares technique. In *Proceedings of the American Control Conference*, pages 3429–3434, June 2003.

[8] G. Antonelli, S. Chiaverini, and G. Fusco. A calibration method for odometry of mobile robots based on the Least-Squares technique: Theory and experimental validation. *IEEE Transactions on Robotics*, 21:994–1004, October 2005.

[9] F. I. Apollonio, M. Gaiani, and B. Benedetti. 3D reality-based artefact models for the management of archaeological sites using 3D GIS: A framework starting from the case study of the Pompeii Archaeological area. *Journal of Archaeological Science*, 39:1271–1287, May 2012.

[10] A. Astolfi. Exponential stabilization of a wheeled mobile robot via discontinuous control. *Journal of Dynamic Systems, Measurement, and Control*, March 1999.

[11] J. E. Banta, Y. Zhien, X. Z. Wang, G. Zhang, M. T. Smith, and M. A. Abidi. Best-next-view algorithm for three-dimensional scene reconstruction using range images. In *Proceedings of the SPIE (Intelligent Robots and Computer Vision XIV: Algorithms)*, volume 2588, pages 418–429, 1995.

[12] S. Barnea and S. Filin. Keypoint based autonomous registration of terrestrial laser point-clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1):19–35, 2008.

[13] J. Bauer, K. Karner, K. Schindler, A. Klaus, and C. Zach. Segmentation of building models from dense 3D point-clouds. In *Proceedings of the 27th Workshop of the Austrian Association for Pattern Recognition*, pages 253–259, 2003.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (SURF). *Journal of Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[15] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision*, pages 404–417, May 2006.

[16] J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, Washington, DC, USA, 1997.

[17] G. H. Bendels, P. Degener, R. Wahl, M. Körtgen, and R. Klein. Image-based registration of 3D-range data using feature surface elements. In *Proceedings of the 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 115–124, 2004.

[18] G. H. Bendels, R. Schnabel, and R. Klein. Detecting holes in point set surfaces. *Journal of WSCG*, February 2006.

[19] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975.

[20] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239 – 256, February 1992.

[21] P. Biber and W. Strasser. The normal distributions transform: A new approach to laser scan matching. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2743–2748, 2003.

[22] J. M. Biosca and J. L. Lerma. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63:84–98, 2008.

[23] P. S. Blaer and P. K. Allen. Data acquisition and view planning for 3-D modeling tasks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 417–422, October 2007.

[24] J. Böhm and S. Becker. Automatic marker-free registration of terrestrial laser scans using reflectance features. In *Proceedings of the 8th Conference on Optical 3D Measurment Techniques*, pages 338–344, Zurich, Switzerland, July 2007.

[25] O. Booij, B. Terwijn, Z., and B. Krose. Navigation using an appearance based topological map. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3927–3932, Rome, Italy, May 2007.

[26] J. Borenstein. The CLAPPER: A dual-drive mobile robot with internal correction of dead-reckoning errors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3085–3090, May 1994.

[27] J. Borenstein, H. Everett, and L. Feng. *Navigating mobile robots: Systems and techniques*. A K Peters, February 1996.

[28] J. Borenstein and L. Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, 12(6):869–880, December 1996.

[29] J. Borenstein and Y. Koren. Motion control analysis of a mobile robot. *Journal of Dynamic Systems, Measurement, and Control*, 109(2):73–78, 06 1987.

[30] J. Borenstein and Y. Koren. High-speed obstacle avoidance for mobile robots. In *Proceedings of the IEEE International Symposium on Intelligent Control*, pages 382–384, August 1988.

[31] J. Borenstein and Y. Koren. The Vector Field Histogram - Fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7:278–288, 1991.

[32] J. Borenstein. Intemal correction of dead-reckoning errors with a dual-drive compliant linkage mobil robot. *Journal of Robotic Systems*, 12(4):257–273, 1995.

[33] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally consistent 3D mapping with scan matching. *Journal of Robotics and Autonomous Systems (JRAS)*, 56(2):130–142, 2008.

[34] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter. The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. *Journal of 3D Research*, 2:1–13, March 2011.

[35] D. Borrmann, A. Nüchter, M. Dakulovic, I. Maurovic, I. Petrovic, D. Osmankovic, and J. Velagic. A mobile robot based system for fully automated thermal 3D mapping. *Journal of Advanced Engineering Informatics, Elsevier*, 28:425–440, October 2014.

[36] H. Boulaassal, T. Landes, P. Grussenmeyer, and F. Tarsha-Kurdi. Automatic segmentation of building facades using terrestrial laser data. In *Proceedings of the ISPRS Workshop on Laser Scanning and SilviLaser*, pages 65–70, Espoo, Finland, 2007.

[37] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[38] G. Bradski and A. Kaehler. *Learning OpenCV, Computer Visoin with OpenCV library*. O' Reilly Media, first edition, 2008.

[39] C. Brenner, C. Dold, and N. Ripperda. Coarse orientation of terrestrial laser scans in urban environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1):4–18, 2008.

[40] C. Brenner. *Dreidimensionale Gebäuderekonstruktion aus digitalen Oberflächenmodellen und Grundrissen*. PhD thesis, Universität Stuttgart, Fakultät für Bauingenieur- und Vermessungswesen, 2000.

[41] W. Burgard, A. Derr, D. Fox, and A. Cremers. Integrating global position estimation and position tracking for mobilerobots: The Dynamic Markov Localization approach. In *Proceedings of the IEEE/RSJ InternationalConference on Intelligent Robots and Systems*, 1998.

[42] W. Burgard, D. Fox, and D. Hennig. Fast grid-based position tracking for mobile robots. In *Proceedings of the 21th German Conference on Artificial Intelligence*, pages 289–300, 1997.

[43] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer vision: Part IV*, volume 6314 of *ECCV'10*, pages 778–792, 2010.

[44] S. Campana and F. Remondino. Fast and detailed digital documentation of archaeological excavations and heritage artifacts. In *Proceedings of the 35th CAA Conference (Computer Applications and Quantitative Methods in Archaeology)*, pages 36–42, Berlin, Germany, April 2007.

[45] G. Campion, G. Bastin, and B. d'Andréa Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1):47–62, 1996.

[46] P. Chalmovianský and B. Jüttler. *Mathematics of Surfaces: Proceedings of the 10th IMA International Conference*, chapter Filling Holes in Point Clouds, pages 196–212. Springer Berlin Heidelberg, 2003.

[47] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Journal of Image and Vision Computing*, 10(3):145–155, 1992.

[48] F. Chenavier and J. L. Crowley. Position estimation for a mobile robot using vision and odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2588–2593, May 1992.

[49] K. S. Chong and L. Kleeman. Accurate odometry and error modelling for a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 2783–2788, April 1997.

[50] R. Colbaugh, E. Barany, and K. Glass. Adaptive control of nonholonomic robotic systems. *Journal of Robotic Systems*, 15:365–393, 1998.

[51] J. Cosmas, T. Itagaki, D. Green, E. Grabczewski, F. Weimer, L. J. V. Gool, A. Zalesny, D. Vanrintel, F. Leberl, M. Grabner, K. Schindler, K. F. Karner, M. Gervautz, S. Hynst, M. Waelkens, M. Pollefeys, R. DeGeest, R. Sablatnig, and M. Kampel. 3D MURALE: A multimedia system for archaeology. In *Proceedings of the Conference on Virtual Reality, Archeology, and Cultural Heritage*, pages 297–306, 2001.

[52] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1989.

[53] J. L. Crowley. Asynchronous control of orientation and displacement in a robot vehicle. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1277–1282, 1989.

[54] M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1828–1833, Rome, Italy, May 2007.

[55] M. Cummins and P. Newman. Accelerated appearance-only slam. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1828–1833, Pasadena, CA, USA, May 2008.

[56] C. de Wit Canudas, G. Bastin, and B. Siciliano. *Theory of Robot Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1996.

[57] C. de Wit Canudas, H. Khennouf, C. Samson, and O. Sordalen. Nonlinear control design for mobile robots. In *Recent Trends in Mobile Robots*, volume 11 of *World Scientific Series in Robotics and Automated Systems*. World Scientific, 1993.

[58] C. de Wit Canudas and O. J. Sordalen. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 37:1791–1797, 1992.

[59] denkmalDaten Winkler KG. Adiuvabit. http://www.adiuvabit.de, 2008.

[60] H. J. V. der Hardt, R. Husson, and D. Wolf. An automatic calibration method for a multisensor system: Application to a mobile robot localization system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3141–3146, May 1998.

[61] P. Deutsch. DEFLATE compressed data format specification version 1.3. RFC Editor, May 1996.

[62] O. Devillers and P. M. Gandoin. Geometric compression for interactive transmission. In *Proceedings of the IEEE Conference on Visualization*, pages 319 – 326, 2000.

[63] G. Dissanayake, H. F. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1009–1014, 2000.

[64] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17:229–241, 2001.

[65] N. Doh, H. Choset, and W. K. Chung. Accurate relative localization using odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1606–1612, September 2003.

[66] C. Dornhege and A. Kleiner. A frontier-void-based approach for autonomous exploration in 3D. *Journal of Advanced Robotics*, 27:459–468, 2013.

[67] A. Doucet, N. de Freitas, K. P. Murphy, and S. J. Russell. Rao-Blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 176–183, San Francisco, CA, USA, 2000.

[68] G. Dudek and M. Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, 2000.

[69] M. K. H. Eggert. *Prähistorische Archäologie - Konzepte und Methoden (Uni-Taschenbücher M)*. UTB, Stuttgart, 3., aktualisierte und erweiterte auflage. edition, May 2008.

[70] A. Ekman, A. Törne, and D. Strömberg. Exploration of polygonal environments using range data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 27:250–255, 1997.

[71] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1135–1142, 2003.

[72] J. Elseberg, D. Borrmann, and A. Nüchter. Full wave analysis in 3D laser scans for vegetation detection in urban environments. In *Proceedings of the 23rd International Symposium on Information, Communication and Automation Technologies*, Sarajevo, Bosnia, October 2011.

[73] J. Elseberg, D. Borrmann, and A. Nüchter. One billion points in the cloud - An octree for efficient processing of 3D laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76:76–88, February 2013.

[74] H. R. Everett. *Sensors for Mobile Robots: Theory and Application*. A. K. Peters, Ltd., Natick, MA, USA, 1995.

[75] M. Everingham, A. Zisserman, C. K. Williams, and L. V. Gool. The PASCAL visual object classes challenge results. Technical report, University of Oxford, University of Edinburgh, KU Leuven, 2006.

[76] M. A. Fischler and R. C. 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, June 1981.

[77] A. Flint, A. Dick, and A. J. van den Hengel. Thrift: Local 3D structure recognition. In *Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, pages 182–188, 2007.

[78] J. Folkesson and H. Christensen. Outdoor exploration and SLAM using a compressed filter. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 11, pages 419–426, September 2003.

[79] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centers of circular features. In *Proceedings of the ISPRS Intercommission Workshop on Fast Processing of Photogrammetric Data*, pages 281–305, 1987.

[80] W. Förstner, T. Dickscheid, and F. Schindler. Detecting interpretable and accurate scale-invariant keypoints. In *Proceedings of the 12th IEEE International Conference on Computer Vision*, pages 2256–2263, Kyoto, Japan, 2009.

[81] D. Fox. Adapting the sample size in particle filters through KLD-sampling. *International Journal of Robotics Research*, 22, 2003.

[82] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Journal of Robotics and Automation*, 4, 1997.

[83] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo Localization: Efficient position estimation for mobile. In *Proceedings of the National Conference on Artificial Intelligence*, pages 343–349, 1999.

[84] U. Frese and G. Hirzinger. Simultaneous localization and mapping - A discussion. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 17–26, Seattle, USA, 2001.

[85] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.

[86] C. Fröhlich, M. Mettenleiter, C. Heldand, D. Blersch, and S. Kurz. Surveying by means of both 3D geometry with HDR colour and thermal imagery. In *Proceedings of the Automatisierungstechnik*, April 2015.

[87] C. Früh and A. Zakhor. 3D model generation for cities using aerial photographs and ground level laser scans. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 31–38, 2001.

[88] P. M. Gandoin and O. Devillers. Progressive lossless compression of arbitrary simplicial complexes. *ACM Transactions on Graphics*, 21(3):372 – 379, July 2002.

[89] P. Ghamisi, A. Mohammadzadeh, M. R. Sahebi, and F. Sepehrband. A novel real time algorithm for remote sensing lossless data compression based on enhanced DPCM. In *International Journal of Computer Applications*, volume 27, August 2011.

[90] A. Gilboa, A. Tal, I. Shimshoni, and M. Kolomenkin. Computer-based, automatic recording and illustration of complex archaeological artifacts. In *Journal of Archaeological Science*, volume 40, pages 793–1448, February 2013.

[91] D. Girardeau-Montaut, M. Roux, R. Marc, and G. Thibault. Change detection on points cloud data acquired with a ground laser scanner. In *Proceedings of the ISPRS Workshop on Laser Scanning*, pages 30–35, September 2005.

[92] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme. Robust localization using relative and absolute position estimates. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1134–1140, 1999.

[93] H. H. González-Baños and J.-C. Latombe. Navigation strategies for exploring indoor environments. *International Journal of Robotics Research*, 21:829–848, 2002.

[94] E. Grabczewski, J. Cosmas, P. V. Santen, D. Green, T. Itagaki, and F. Weimer. 3D MURALE: Multimedia database system architecture. In *Proceedings of the Conference on Virtual Reality, Archeology, and Cultural Heritage*, pages 315–322, 2001.

[95] S. Gray, S. Chitta, V. Kumar, and M. Likhachev. A single planner for a composite task of approaching, opening and navigating through non-spring and spring-loaded door. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013.

[96] G. Grisetti, C. Stachniss, and W. Burgard. Improving Grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2443–2448, 2005.

[97] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 23:34–46, February 2007.

[98] A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Journal of Mathematische Annalen*, 69:331–371, 1910.

[99] D. Hähnel, W. Burgard, D. Fox, and S. Thrun. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[100] D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Journal of Robotics and Autonomous Systems*, 44:15–27, 2003.

[101] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1557–1563, 2003.

[102] P. Hansen, P. Corke, W. W. Boles, and K. Daniilidis. Scale-invariant features on the sphere. In *Proceedings of the International Conference on Computer Vision*, 2007.

[103] P. Hansen, P. Croke, W. Boles, and K. Daniilidis. Scale invariant feature matching with wide angle images. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2007.

[104] E. C. Harris. *Principles of Archaeological Stratigraphy*. Academic Press, second edition edition, 1989.

[105] J. Hertzberg, K. Lingemann, and A. Nüchter. *Mobile Roboter: Eine Einführung aus Sicht der Informatik*. Springer Vieweg, Berlin, Germany, 2012.

[106] Hokuyo Automatic CO. LTD. Hokuyo scanner. http://www.hokuyo-aut.jp.

[107] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, April 1987.

[108] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Journal of Autonomous Robots*, 2013.

[109] Y. Huang, J. Peng, C. C. J. Kuo, and M. Gopi. Octree-based progressive geometry coding of point clouds. In *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics*, pages 103 – 110, 2006.

[110] D. Huber. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, Carnegie Mellon University, 2002.

[111] D. A. Huffman. A method for the construction of minimum redundancy codes. In *Proceedings of the Institute of Radio Engineers*, pages 1098 – 1101, September 1952.

[112] G. Indiveri. Kinematic time-invariant control of a 2D nonholonomic vehicle. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 3, pages 2112–2117, Phoenix, USA, December 1999.

[113] G. Indiveri and M. L. Corradini. Switching linear path following for bounded curvature car-like vehicles. In *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, July 2004.

[114] M. Isenburg. LASzip: Lossless compression of LiDAR data. European LiDAR Mapping Forum, 2012.

[115] F. G. Jean-Emmanuel Deschaud. A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. In *Proceedings of the 3D Processing, Visualization and Transmission Conference*, Paris, France, May 2010.

[116] D. Joho, C. Stachniss, P. Pfaff, and W. Burgard. Autonomous exploration for 3D map learning. In *Proceedings of the Conference on Autonome Mobile Systeme*, pages 22–28. Springer, October 2007.

[117] M. Kaess, R. Arkin, and J. Rossignac. Compact encoding of robot-generated 3D maps for efficient wireless transmission. In *Proceedings of the IEEE International Conference on Advanced Robotics*, pages 324–331, Coimbra, Portugal, June 2003.

[118] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach. Real-time compression of point cloud streams. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2012.

[119] Z. Kang, J. Li, L. Zhang, Q. Zhao, and S. Zlatanova. Automatic registration of terrestrial laser scanning point clouds using panoramic reflectance images. *Journal of Sensors*, pages 2621–2646, 2009.

[120] A. Karmacharya, C. Cruz, F. Marzani, and F. Boochs. Industrial archaeology: Case study of knowledge management for spatial data of findings. In *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, July 2008.

[121] M. Katsianis, S. Tsipidis, K. Kotsakis, and A. Kousoulakou. A 3D digital workflow for archaeological intra-site research using GIS. *Journal of Archaeological Science*, 35:655–667, March 2008.

[122] L. Kavraki, P. Svestka, J. claude Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 566–580, 1996.

[123] K. Kawashima, S. Yamanishi, S. Kanai, and H. Date. Finding the next-best scanner position for as-built modeling of piping systems. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40, 2014.

[124] M. Khatib and R. Chatila. An extended potential field approach for mobile robot sensor-based motions. In *Proceedings of the Conference on Intelligent Autonomous Systems*, pages 490–496, March 1995.

[125] M. Khatib, H. Jaouni, R. Chatila, and J.-P. Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2920–2925, 1997.

[126] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5:90–98, April 1986.

[127] K. Klasing, D. Wollherr, and M. Buss. A clustering method for efficient segmentation of 3D laser data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4043–4048, May 2008.

[128] S. Koenig and M. Likhachev. Improved fast replanning for robot navigation in unknown terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002.

[129] S. Koenig and M. Likhachev. Incremental A*. In *Proceedings of the Conference on Neural Information Processing Systems*, 2002.

[130] S. Koenig and M. Likhachev. D* Lite. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 476–483, Menlo Park, CA, USA, 2002.

[131] S. Koenig and R. G. Simmons. Xavier: A robot navigation architecture based on partially observable markov decision process models. In *Proceedings of the Conference on Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, pages 91–122. MIT Press, 1998.

[132] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. A flexible and scalable SLAM system with full 3D motion estimation. In *Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics*, pages 155–160, Kyoto, Japan, November 2011.

[133] K. Komoriya and E. Oyama. Position estimation of a mobile robot using optical fiber gyroscope (OFG). In *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems, 'Advanced Robotic Systems and the Real World'*, volume 1, pages 143–149, September 1994.

[134] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. In *Proceedings of the Conference on Robotics: Science and Systems*, pages 941–957, Seattle, USA, 2009.

[135] K. Köser and R. Koch. Perspectively invariant normal features. In *Proceedings of the International Conference on Computer Vision*, pages 14–21, 2007.

[136] R. Krishnamurthy, B.-B. Chai, H. Tao, and S. Sethuraman. Compression and transmission of depth maps for image-based rendering. In *Proceedings of the International Conference on Image Processing*, 2001.

[137] S. C. Kuzminsky and M. S. Gardiner. Three-dimensional laser scanning: Potential uses for museum conservation and scientific research. In *Journal of Archaeological Science*, volume 39, pages 2744–2751, August 2012.

[138] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[139] S. M. Lavalle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Proceedings of the Conference on Algorithmic and Computational Robotics: New Directions*, pages 293–308, 2000.

[140] D. Lee. *The Map-Building and Exploration Strategies of a Simple Sonar-Equipped Mobile Robot*. Cambridge University Press, Cambridge, UK, 1996.

[141] K. Lee, H. Woo, and T. Suk. Data reduction methods for reverse engineering. *The International Journal of Advanced Manufacturing Technology*, 17(10):735–743, 2001.

[142] Y. J. Lee, D.-Y. Kim, and M. J. Chung. Feature matching in omnidirectional images with a large sensor motion for map generation of a mobile robot. *Journal of Pattern Recognition Letters*, 25(4):413–427, March 2004.

[143] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7:376–382, June 1991.

[144] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1150 – 1157, Washington, DC, USA, 1999.

[145] D. G. Lowe. Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, 2004.

[146] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Journa of Autonomous Robots*, 4:333 – 349, April 1997.

[147] V. J. Lumelsky and T. Skewis. Incorporating range sensing in the robot navigation function. *IEEE Transactions on Systems, Man, and Cybernetics*, 20:1058–1069, September 1990.

[148] V. J. Lumelsky and A. A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Journal of Algorithmica*, 1987.

[149] M. Magnusson, H. Andreasson, A. Nüchter, and A. J. Lilienthal. Automatic appearance-based loop detection from 3D laser data using the normal distributions transform. *Journal of Field Robotics, Special Issue on Three-Dimensional Mapping*, 26:892–914, November/December 2009.

[150] M. Magnusson, T. Duckett, and A. J. Lilienthal. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24:803–827, October 2007.

[151] A. Mandow, J. L. Marti-nez, A. J. Reina, and J. Morales. Fast range-independent spherical subsampling of 3D laser scanner points and data reduction performance evaluation for scene registration. *Journal of Pattern Recognition Letters*, 31:1239–1250, 2010.

[152] P. S. Maybeck. The Kalman Filter: An introduction to concepts. In *Autonomous Robot Vehicles*, pages 194–204. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[153] D. Meagher. Geometric modeling using Octree encoding. *Journal of Computer Graphics and Image Processing*, 19:129–147, 1982.

[154] N. Meierhold, M. Spehrb, A. Schilling, S. Gumhold, and H.-G. Maas. Automatic feature matching between digital images and 2D representations of a 3D laser scanner point cloud. In *Proceedings of the Commission V Symposium, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 38, pages 446–451, Newcastle upon Tyne, UK, 2010.

[155] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 128–142, 2002.

[156] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision*, pages 525–531, 2001.

[157] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, October 2004.

[158] D. Mongus and B. Zalik. Efficient method for lossless LiDAR data compression. *International Journal of Remote Sensing*, 32(9):2507–2518, May 2011.

[159] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 593–598, Menlo Park, CA, USA, 2002.

[160] F. Moosmann, O. Pink, and C. Stiller. Segmentation of 3D LiDAR data in non-flat urban environments using a local convexity criterion. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 215–220. IEEE, 2009.

[161] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 116–121, May 1985.

[162] P. Morin and C. Samson. Time-varying exponential stabilization of a rigid spacecraft with two control torques. *IEEE Transactions on Automatic Control*, 42:528–534, 1997.

[163] P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. In *Proceedings of the International Symposium of Robotics Research*, 1989.

[164] P. Moutarlier and R. Chatila. An experimental system for incremental environment modelling by an autonomous mobile robot. In *Proceedings of the Experimental Robotics I: The First International Symposium*, pages 327–346, Montreal, Canada, 1989.

[165] K. P. Murphy. Bayesian map learning in dynamic environments. In *Proceedings of the Conference of Neural Information Processing Systems*, pages 1015–1021, 2000.

[166] J. D. Murray and W. vanRyper. *Encyclopedia of graphics file formats*. O'Reilly and Associates, 2 edition, 1996.

[167] F. Nenci, L. Spinello, and C. Stachniss. Effective compression of range data streams for remote robot operations using H.264. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, USA, 2014.

[168] M. Nieuwenhuisen, J. Stückler, and S. Behnke. Improving indoor navigation of autonomous robots by an explicit representation of doors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4895–4901, 2010.

[169] Nikon Metrology. iSpace - Portable metrology system user manual and startup guide. Webpage, July 2014. `http://www.nikonmetrology.com`.

[170] N. J. Nilsson. *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill Pub. Co., 1971.

[171] I. R. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH - An office-navigating robot. *AI Magazine*, 16:53–60, 1995.

[172] A. Nüchter, R. B. Rusu, D. Holz, and D. Munoz. Editorial: Semantic perception, mapping and exploration. *Journal of Robotics and Autonomous Systems*, 2013.

[173] A. Nüchter. *3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom*. Springer Publishing Company, Incorporated, 2009.

[174] A. Nüchter, J. Elseberg, P. Schneider, and D. Paulus. Study of parameterizations for the rigid body transformations of the scan registration problem. *Journal of Computer Vision and Image Understanding*, 114:963 – 980, 2010.

[175] A. Nüchter, K. Lingemann, and J. Hertzberg. Cached k-d tree search for ICP algorithms. In *Proceedings of the IEEE 6th International Conference on Recent Advances in 3D Digital Imaging and Modeling*, pages 419–426, 2007.

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

[177] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM-3D Mapping outdoor environments: Research articles. *Journal of Field Robotics*, 24:699–722, August 2007.

[178] A. Nüchter, H. Surmann, and J. Hertzberg. Planning robot motion for 3D digitalization of indoor environments. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 222–227, 2003.

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

[180] Open Source Robotics Foundation. Robot operating system (ROS). http://www.ros.org.

[181] L. Parida, D. Geiger, and R. Hummel. Junctions: Detection, classification and reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:687–698, 1998.

[182] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga. Fast registration based on noisy planes with unknown correspondences for 3D mapping. *IEEE Transactions on Robotics*, 26(3):424–441, 2010.

[183] 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 *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5725–5730, Taipei, Taiwan, October 2010.

[184] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.

[185] J. Peng and C.-C. J. Kuo. Octree-based progressive geometry encoder. In *Proceedings of the Conference on Internet Multimedia Management Systems IV*, pages 301 – 311, 2003.

[186] J. Peng and C. C. J. Kuo. Geometry-guided progressive lossless 3D mesh coding with Octree (OT) decomposition. *ACM Transactions on Graphics*, 24:609 – 616, July 2005.

[187] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak. Fast plane detection and polygonalization in noisy 3D range images. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 3378–3383, 2008.

[188] C. Potthast and G. S. Sukhatme. A probabilistic framework for next best view estimation in a cluttered environment. *Journal of Visual Communication and Image Representation*, 25:148–164, January 2014.

[189] T. Rabbani, F. A. van den Heuvel, and G. Vosselmann. Segmentation of point clouds using smoothness constraint. In *Proceedings of the ISPRS Commission V Symposium: Image Engineering and Vision Metrology*, 2006.

[190] J. D. Reu, P. D. Smedt, D. Herremans, M. V. Meirvenne, P. Laloo, and W. D. Clercq. On introducing an image-based 3D reconstruction method in archaeological excavation practice. *Journal of Archaeological Science*, 41:251 – 262, January 2014.

[191] RIEGL Laser Measurement Systems GmbH. Riegl scanner. http://www.riegl.com.

[192] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1508–1511. Springer, October 2005.

[193] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 430–443, May 2006.

[194] E. Rosten, R. Porter, and T. Drummond. FASTER and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:105 – 119, 2010.

[195] H. Rua and P. Alvito. Living the past: 3D models, virtual reality and game engines as tools for supporting archaeology and the reconstruction of cultural heritage - the case-study of the Roman villa of Casal de Freiria. *Journal of Archaeological Science*, 38:3296–3308, December 2011.

[196] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2564–2571, Barcelona, November 2011.

[197] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[198] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[199] R. B. Rusu and et al. Point Cloud Library. http://www.pointcloud.org.

[200] D. Santa-Cruz, T. Ebrahimi, J. Askelöf, M. Larsson, and C. Christopoulos. JPEG 2000 still image coding versus other standards. In *Proceedings of the 23rd Conference on Applications of Digital Image Processing*, volume 4115, pages 446 – 454, August 2000.

[201] A. D. Sappa and M. Devy. Fast range image segmentation by an edge detection strategy. In *Proceedings of the 3rd international conference on 3D digital imaging and modeling*, pages 292–299, 2001.

[202] A. E. Savakis. Evaluation of lossless compression methods for gray scale document images. In *Proceedings of the International Conference on Image Processing*, 2000.

[203] R. Schnabel and R. Klein. Octree-based point-cloud compression. In *Proceedings of the Symposium on Point-Based Graphics*, July 2006.

[204] D. Schulz and W. Burgard. Probabilistic state estimation of dynamic objects with a moving mobile robot. *Journal of Robotics and Autonomous Systems*, 34, 2001.

[205] L. Sciavicco, B. Siciliano, and B. Sciavicco. *Modelling and Control of Robot Manipulators*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2nd edition, 2000.

[206] W. Scott, G. Roth, and J. F. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Journal of Computing Surveys*, 35:64–96, 2003.

[207] V. Sequeira, J. G. Gonçalves, and M. Ribeiro. 3D environment modelling using laser range sensing. *Journal of Robotics and Autonomous Systems*, 16:81 – 91, 1995.

[208] V. Sequeira, K. C. Ng, E. Wolfart, J. G. M. Goncalves, and D. C. Hogg. Automated 3D reconstruction of interiors with multiple scan views. In *Proceedings of the SPIE 11th Annual Symposium on Electronic Imagin*, pages 106–117, 1998.

[209] T. K. Sharpless, B. Postle, and D. M. German. Pannini: A new projection for rendering wide angle perspective images. In *Proceedings of the Conference on Computational Aesthetics in Graphics, Visualization, and Imaging*, 2010.

[210] T. C. Shermer. Recent results in art galleries (geometry). *Proceedings of the IEEE*, 80:1384–1399, September 1992.

[211] R. Siegwart and I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Bradford Company, Scituate, MA, USA, 2004.

[212] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[213] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainly. *International Journal of Robotics Research*, 5:56–68, 1987.

[214] J. P. Snyder and P. M. Voxland. An album of map projections. Report, USGS Numbered Series 1453, USGS Publications Warehouse, 1989.

[215] O. J. Sordalen and C. de Wit Canudas. Exponential control law for a mobile robot: Extension to path following. *IEEE Transactions on Robotics and Automation*, 9:837–842, 1993.

[216] L. Spinello, R. Triebel, and R. Siegwart. Multimodal detection and tracking of pedestrians in urban environments with explicit ground plane extraction. In *Proceedings of The IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1823–1829, September 2008.

[217] C. Stachniss and W. Burgard. Mobile robot mapping and localization in non-static environments. In *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, PA, USA, 2005.

[218] I. Stamos and P. Allen. 3-D model construction using range and image data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 531–536, USA, June 2000.

[219] B. Steder, G. Grisetti, and W. Burgard. Robust place recognition for 3D range data based on point features. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1400 –1405, May 2010.

[220] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3310 – 3317, May 1994.

[221] A. Stentz. The focussed D* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, August 1995.

[222] H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Journal of Robotics and Autonomous Systems*, 2003.

[223] D. Suter and E. Lim. Conditional Random Field for 3D point clouds with Adaptive Data Reduction. *Journal of Cyberworlds*, 2007.

[224] A. Swadzba, A.-L. Vollmer, M. Hanheide, and S. Wachsmuth. Reducing noise and redundancy in registered range data for planar surface extraction. In *Proceedings of the International Conference on Pattern Recognition*, pages 1–4, 2008.

[225] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20, 2001.

[226] S. Thrun. Robotic mapping: A survey. In *Exploring artificial intelligence in the new millennium*. Morgan Kaufmann, 2002.

[227] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 321–328, April 2000.

[228] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. The MIT Press, 2005.

[229] S. Thrun, W. Burgard, D. Fox, H. Hexmoor, and M. Mataric. A probabilistic approach to concurrent mapping and localization for mobile robots. *Journal of Machine Learning*, pages 29–53, 1998.

[230] B. Tovar, R. Murrieta-Cid, and S. M. Lavalle. Distance-optimal navigation in an unknown environment without sensing distances. *IEEE Transactions on Robotics*, 23:506–518, 2007.

[231] K. Trovato. Differential A*: An adaptive search method illustrated with robot path planning for moving obstacles and goals, and an uncertain environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 4:245–268, 1990.

[232] I. Ulrich and J.Borenstein. VFH+: reliable obstacle avoidance for fast mobile robots. In *Proceedings of the International Conference on Robotics and Automation*, pages 1572–1577, Leuven, Belgium, May 1998.

[233] I. Ulrich and J. Borenstein. VFH*: local obstacle avoidance with look-ahead verification. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 24–28, May 2000.

[234] C. Valgren and A. J. Lilienthal. SIFT, SURF & Seasons: Appearance-based long-term localization in outdoor environments. *Journal of Robotics and Autonomous Systems*, 58:157–165, February 2010.

[235] W3C. Portable Network Graphics (PNG) Specification. http://www.w3.org/TR/PNG/, November 2003.

[236] G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, pages 30 – 44, 1991.

[237] M. Wand, A. Berner, M. Bokeloh, A. Fleck, M. Hoffmann, P. Jenke, B. Maier, D. Staneker, and A. Schilling. Interactive editing of large point clouds. In *Proceedings of the Symposium on Point-Based Graphics*, pages 37 – 46, 2007.

[238] X. Wang, C. Toth, D. Grejner-Brzezinska, and H. Sun. Integration of terrestrial laser scanner for ground navigation in GPS-challenged environments. In *Proceedings of the 21st ISPRS Congress: Commission V, WG 3*, pages 513–518, 2008.

[239] Z. Wang and C. Brenner. Point based registration of terrestrial laser data using intensity and geometry features. In *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 37, pages 583–590, 2008.

[240] M. A. Wani and H. R. Arabnia. Parallel edge-region-based segmentation algorithm targeted at reconfigurable multiring network. *The Journal of Supercomputing*, 23:43–62, 2003.

[241] J. W. Weingarten and R. Siegwart. EKF-based 3D SLAM for structured environment reconstruction. In *Proceedings of the IEEE/ RSJ International Conference on Intelligent Robots and Systems*, pages 3834–3839, 2005.

[242] M. Weinmann and B. Jutzi. Fully automatic image-based registration of unorganized TLS data. In *Proceedings of the ISPRS Workshop on Laser Scanning*, 2011.

[243] M. Weinmann, M. Weinmann, S. Hinz, and B. Jutzi. Fast and automatic image-based registration of TLS data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66:62–70, October 2011.

[244] T. A. Welch. A technique for high-performance data compression. *Journal of Computer*, 17:8 – 19, June 1984.

[245] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3D model matching with viewpoint-invariant patches (VIP). In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[246] 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 *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[247] T. Wüst, S. Nebiker, and R. Landolt. Applying the 3D GIS DILAS to archaeology and cultural heritage projects - Requirements and first results. In *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, volume 35, pages 407 – 412, July 2004.

[248] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, July 1997.

[249] M. Y. Yang and W. Förstner. Plane detection in point cloud data. In *Proceedings of the 2nd International Conference on Machine Control Guidance*, volume 1, pages 95–104, Bonn, Germany, January 2010.

[250] F. Zacharias, C. Borst, and G. Hirzinger. Capturing robot workspace structure: Representing robot capabilities. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3229–3236, October 2007.

[251] Z. Zhang. Iterative point matching for registration of free-form curves. Technical Report RR-1658, INRIA-Sophia Antipolis, Valbonne Cedex, France, 1992.

[252] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23:337 – 343, 1977.

[253] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24:530 – 536, September 1978.

[254] Zoller + Fröhlich GmbH. Z+F Scanner. http://www.zf-laser.com.

Proclamation

Hereby I confirm that I wrote this thesis independently and that I have not made use of any other resources or means than those indicated.

Würzburg, February 2017

Die Schriftenreihe

wird vom Lehrstuhl für Informatik VII: Robotik und Telematik der Universität Würzburg herausgegeben und präsentiert innovative Forschung aus den Bereichen der Robotik und der Telematik.

Die Kombination fortgeschrittener Informationsverarbeitungsmethoden mit Verfahren der Regelungstechnik eröffnet hier interessante Forschungs- und Anwendungsperspektiven. Es werden dabei folgende interdisziplinäre Aufgabenschwerpunkte bearbeitet:

- Robotik und Mechatronik: Kombination von Informatik, Elektronik, Mechanik, Sensorik, Regelungs- und Steuerungstechnik, um Roboter adaptiv und flexibel ihrer Arbeitsumgebung anzupassen.

- Telematik: Integration von Telekommunikation, Informatik und Steuerungstechnik, um Dienstleistungen an entfernten Standorten zu erbringen.

Anwendungsschwerpunkte sind u.a. mobile Roboter, Tele-Robotik, Raumfahrtsysteme und Medizin-Robotik.

Lehrstuhl Informatik VII
Robotik und Telematik
Am Hubland
D-97074 Wuerzburg

Tel.: +49 (0) 931 - 31 - 86678
Fax: +49 (0) 931 - 31 - 86679

schi@informatik.uni-wuerzburg.de
http://www7.informatik.uni-wuerzburg.de

ISSN: 1868-7474 (online)
ISSN: 1868-7466 (print)
ISBN: 978-3-945459-14-0 (online)

Zitation dieser Publikation

HOUSHIAR, H. (2017). Documentation and mapping with 3D point cloud processing. Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 12. Würzburg: Universität Würzburg.
URN: urn:nbn:de:bvb:20-opus-144493