Chapter 3
Indoor 3D: Overview on Scanning and Reconstruction Methods

Ville V. Lehtola, Shayan Nikoohemat, and Andreas Nüchter

3.1 Introduction

Accurate three-dimensional (3D) data is called for creating accurate reconstructions of indoor spaces, i.e., application-suitable digital twins of these spaces. For instance, the global indoor 3D laser scanner market accounted for 3.79 billion in 2017 (Businesswire 2019). The purpose of reconstruction is roughly dividable into two types: schematic models for engineering purposes or visual models that are intended for a broader audience than just engineers. On the one hand, schematic applications include performing change detection between building information models (BIM) and as-built data, and the related planning and monitoring of construction processes and building conditions. On the other hand, visually-appealing virtual models are useful for facility management, supporting high-level decision making, real-estate brokering and marketing, displaying cultural and historical heritage, and other applications. The schematic and virtual properties of digital 3D models can also be combined. Indoor models of public buildings, e.g., airports and shopping malls, can be used to assist indoor navigation and location-based services. Concerning the public sector, construction permit processes may be sped up by applying automated model checkers into these digital models – before and after the construction. Furthermore, decision making on city planning can be facilitated when plans are made for indoor or underground public places. Such places are common, for example, near metro stations and in northern countries where winters are cold.

V. V. Lehtola (✉) · S. Nikoohemat
ITC Faculty, University of Twente, Enschede, The Netherlands
e-mail: ville.lehtola@iki.fi

A. Nüchter
Robotics and Telematics, University of Würzburg, Würzburg, Germany
e-mail: andreas@nuechti.de

M. Werner, Y.-Y. Chiang (eds.), *Handbook of Big Geospatial Data*,
https://doi.org/10.1007/978-3-030-55462-0_3

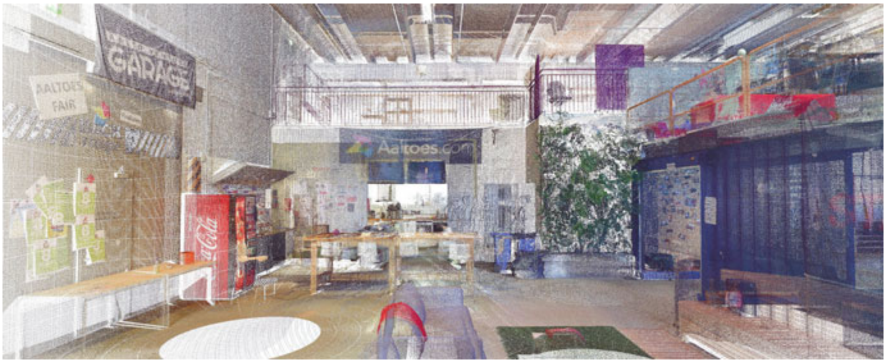


Fig. 3.1 Point cloud of Startup Sauna entrance at Aalto University. (Reproduced from Lehtola et al. 2017)

Perhaps surprising to a common man, the scanning and modeling of building interiors and exteriors are two different things. The reader may reflect on this while they proceed. The activities for reconstructing building exteriors were already well-known when the interest towards the indoor spaces was taking its first steps (Musialski et al. 2013).

The creation of indoor 3D models from scanned data was mainly a curiosity before 2010s, and was done without modern mobile mapping methods. The reconstruction of indoor models relied on 3D point clouds obtained from terrestrial laser scanning (TLS), see Fig. 3.1, or on classical photogrammetry, specifically, bundle adjustment. On one hand, TLS scanning required professional level high-cost equipment and post-processing software for the lidar data, which made the process impossible to automate. On the other hand, digital RGB images taken with calibrated cameras were employed to find similar features from images and then triangulate the 3D geometry from these images. After sparse matching, dense matching and reconstruction techniques were employed, e.g. those based on voxelization (Furukawa et al. 2009). Considering automated processing, this bundle adjustment-based technique required professional knowledge of camera calibration from the user and had problems with lighting, textures, and the complex geometry of the indoor environments (Lehtola et al. 2014). These initial techniques however brought an initial sense of success and with the development in miniaturization of sensors (lidars, MEMS INS), they sparked a boom of interest in indoor scanning and reconstruction.

The indoor scanning problem has been a hot topic throughout the 2010s, seeing many different scanning systems being designed (Lehtola et al. 2017). The problem itself was approached from several directions. First, the positioning procedure of the mobile mapping system that traditionally relied on global navigation satellite systems (GNSS) was 're-designed' to operate in interior spaces. This meant disabling the GNSS receiver and using only the inertial sensor to navigate. This so-called (pedestrian) dead reckoning[1] technique however results into a rapidly increasing uncertainty about the position of the sensor system, because the inertial sensor drift rate is an unknown function with respect to time. In other words, even if the inertial sensor drift is calibrated at an instant of time, that drift changes at a certain rate. Now, this change rate could also be measured and calibrated away, but because the change rate is unknown, the calibration does not last. With a navigation grade inertial measurement unit (IMU), i.e. equipment with a very small drift change rate, Trimble was able to design a pushcart system in 2012. This unit however remains to be a test system due to the high costs of such an IMU (>20,000 euros). Hence, the key in indoor scanning is the robustness of the positioning method. The position of the scanning system, when developed as a function of time, becomes the traversed path of motion, i.e. the trajectory.

After the bundle adjustment, the TLS, and the dead reckoning methods led into shortcomings, the research focus was intensified in mobile systems and trajectory-based methods. There, the basic idea is to track the position and heading (i.e. pose) of the sensor system as a function of time in 3D relative coordinates. The pose updates are done using the overlaps in optical data, that is for example keeping record on déjà-vu's, or technically, features that have been seen before. In robotics, this is known as the simultaneous localization and mapping (SLAM).

This book chapter is written as follows. We shall begin by considering the properties of indoor environments and what problems they pose for scanning and reconstruction (Sect. 3.2). Then we discuss how can these spaces can be understood by computers, i.e., map representations (Sect. 3.3). The development of the indoor scanning techniques are reflected on the introduced problems and we list some prominent mobile mapping methods (Sect. 3.4). Based on these and given an application, the reader should to be able to identify the scanning challenges related to that application and then be able to select a suitable indoor mobile mapping system for that application. Furthermore, in order to give the reader a basic understanding in how the indoor mapping systems perform simultaneous localization and mapping (SLAM), we describe the algorithm based on iterative closest points (ICP) (Sect. 3.5). This description (along with the cited works) allows – in principle – for the reader to construct their own indoor mobile scanning system.[2] We expect, however, that most readers do not construct their own systems but are instead interested in the functionality of the existing systems and their development in the creation of point clouds. The reconstruction of indoor spaces (Sect. 3.6) covers the necessary step of semantically segmenting the created point cloud and the following step on turning this labeled point cloud into a meshed model. Hence, by indoor 3D reconstruction we are referring to the process of generating a meshed model which is exportable to one of the standard formats such as IFC (industry

[1] The terms dead reckoning and pedestrian dead reckoning are used in the field of positioning and navigation.

[2] Note that ready open source SLAM codes are also available, e.g. https://github.com/googlecartographer/ (Hess et al. 2016).

foundation classes) or IndoorGML (Chen and Clarke 2017). In other words, the point cloud that is obtained from scanning is replaced by a mesh that consists of continuous geometrical shapes such as planes. In Sect. 3.7, we review applications. The book chapter ends with a discussion on future trends (Sect. 3.8) and a list of exercises for students (Sect. 3.9).

A common thread of this book chapter, as the reader will discover, is that the scanning trajectory is of critical importance in each of the steps towards the final 3D model, i.e. an application-suitable digital twin of the indoor space. The concept of trajectory, i.e. the path that the scanning system has traveled, is at the very core of mobile mapping and we highlight that it is important to understand what it stands for as it is exploited not only in scanning but also in reconstruction steps of the indoor spaces. To this end, we need to review some terminology.

3.1.1 Terminology

The development in indoor mobile mapping has heritage in multiple fields of science. Hence, there are several words that bear a similar or identical meaning. A systematic review of the scanning terminology is listed in Table 3.1. Additionally, there are some apparent ambiguities that need to be clarified. It is important to differentiate between *relative positioning*, where a map with an internal coordinate system is created,[3] and *absolute positioning*, where a map with geographic coordinates is created (typically using a GNSS receiver). Here, the term *map* follows from robotics (definition in Sect. 3.3) and does not refer to a cartographic map. When relative positioning such as SLAM techniques are used to create a map, this map can be *geo-referenced*. Geo-referencing is a surveying term that means that the internal coordinate system of a map or image is transformed into a geographic coordinate system, typically WGS82 (World Geodetic System). In other words, after an indoor space is mapped, the obtained map may be connected onto an outdoor map to form a seamless indoor-outdoor transition in the map. Finally, the range *precision* of

Table 3.1 Typical terminology related to (indoor) point cloud registration comes from different disciplines of science (term data registration is sometimes also used). The symbols $(xyz\theta\phi\kappa)$ correspond to 3 Cartesian coordinates and 3 Euler angles

Discipline	Equipment	Term	Mathematical equivalent
Laser scanning	lidar	(uses terms below)	
Photogrammetry	Digital camera	Orientation	$(xyz\theta\phi\kappa)$
Computer vision	Digital camera	External calibration	$(xyz\theta\phi\kappa)$
Robotics	Robot (or sensor)	Pose (or posture)	$(xyz\theta\phi\kappa)$
Navigation	GNSS receiver	Position, heading	(xyz), $(\theta\phi\kappa)$

[3] Typically, the origin is chosen to be at the start point of scanning, i.e. $(x, y, z) = (0, 0, 0)$.

lidars is around some millimeters, so positioning from dense scans using SLAM almost never results in issues with precision. Instead, there are problems related to erroneous scan registrations which are discussed in Sect. 3.5.5 and which we refer to with the word accuracy or accurate data.

3.2 Properties of Indoor Environments and Identification of Scanning and Reconstruction Problems

Indoor spaces may be dark or over-illuminated. They can be colorful or lacking texture. These properties of indoor spaces have an immediate impact on the functionality of sensors. Sensor capabilities with respect to different conditions in indoor environments are detailed in Table 3.2. Note that only the most commonly used sensors are included. The range of a sensor is important when large indoor facilities are scanned. In addition to sensors, there are further things to consider.

Every object and feature within an interior space has a specific purpose, as they have been designed by humans. These objects and features come in different sizes, see Fig. 3.2. There are, in fact, a lot of objects in which people do not normally pay attention, and some of these may have geometrically complex shapes. Some are small, such is the width of an electric wire, and some are big, such is a room. The magnitude of sizes varies from the order of one centimeter to dozens or even hundreds of meters. In other words, the characteristic length scale of interior spaces spans four orders of magnitude. We call this as a *multi-scale problem* (Lehtola et al. 2017).

The multi-scale problem sets apparently conflicting criteria to the design of the indoor mapping system. On one hand, the sampling resolution should be large to be able to account for the smallest details, but on the other it should be sparse to make covering large spaces computationally tractable. However, the fast accumulation of data from large resolution may be dealt with sophisticated data distillation techniques. Hence, an ideal system designed for three-dimensional (3D) indoor reconstruction has a sampling rate that can account for the smallest details, but is able to do efficient data distillation so that even the largest interior spaces may

Table 3.2 Optical sensor capabilities with respect to different conditions in indoor environments

Conditions	RGB (camera)	RGB-D (range camera)	lidar	RGB and lidar
Nominal	Y	Y	Y	Y
Weak textures	N	Y	Y	Y
Dark	N	N	Y	Y
Direct light or sunlight	N	N	Y	Y
Advantage	Textures (and geometry)	Textures and geometry	Geometry	Textures and geometry
Range	Unlimited	6–10 m	30–100 m	30–100+ m

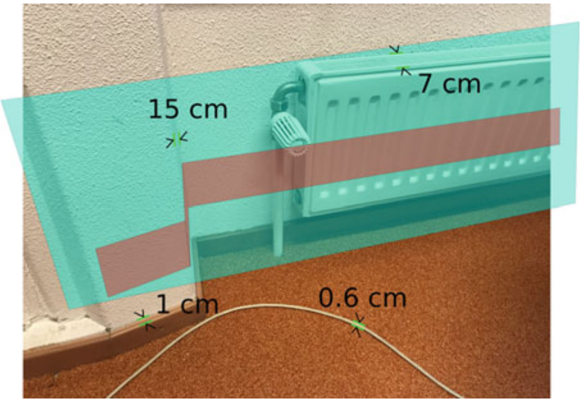


Fig. 3.2 Multi-scale problem. Each object and feature in indoor environments serves a specific purpose for which it has been put there. These span a multitude of length scales, for example, a network cable has 0.6 cm thickness, while the thickness of the radiator is one order of magnitude larger, i.e. 7 cm. The building itself can span a distance of hundreds of meters. Success in separating the objects of these different scales depends on the precision of the data and the models used. A coarse assumption of a rectangular room leads to the elimination of these features, depicted with a cyan plane. Instead, using a piece-wise planar model shown with red planes allows for the recovery of the different objects

be covered. Usually for applications, it is important that the level of detail stays the same regardless of the size of the building. Hence, in practice, the application determines the properties that the measurement system should fulfill.

The measuring geometry of indoor data is very different from traditional remote sensing, where the Earth is viewed from above, and from 3D scanning of single objects, since of two restrictions. First, scanning techniques must account for not being able to see the surrounding indoor space in one snapshot, as sensors typically have a field of view that does not cover 360 degree rotation around two directions. Second, the sensor trajectory is more restricted and difficult measuring geometries that may lead to registration problems are encountered for example in narrow doorways. In contrast, an air-borne scanning system can be freely flown above the Earth or a studio-system freely moved around the single object that is 3D scanned.

Indoor environments are highly convoluted spaces. In topological sense, they can be thought to resemble Swiss cheeses, i.e. bulks with multiple carved holes. Such a bulk can be discretized with an occupancy grid for optical ‘mining’ (see Sect. 3.3). One typical problem that is encountered is the difficulty in distinguishing the points captured from the two different sides of a thin wall. Another is distinguishing between an opening caused by missing data and an opening caused by an existing window.

Occlusions are abundant indoors, since often an object or one part of the area to be scanned blocks another part of the area to be scanned, see Fig. 3.3. Outdoors, when large platforms may be used, the problem can be alleviated by fusing sensor data from different platform locations (Schneider et al. 2010). Indoors, the sensor

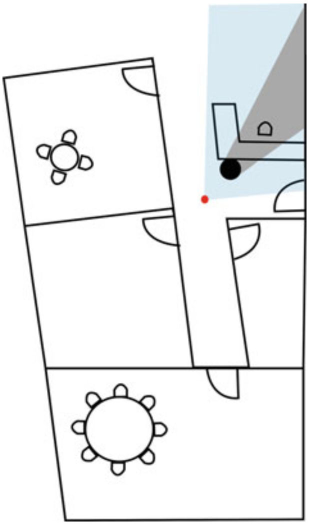


Fig. 3.3 Occlusion is a common problem indoors. Some parts of the scene occlude some other parts of the scene. Obvious examples of occluding objects include static constructs such as pillars and corners, but there are also dynamic objects such as furniture and doors. Here, a pillar is occluding a part of the view of the scanning system located at the red spot (visible area shown in blue, occluded area in grey)

systems are purposefully smaller and this approach is less feasible. These occlusions can be then overcome with footwork, i.e. a thorough scan of the indoor space, which is likely to require an online interface from which the operator can see what part of the areas need further scanning. This need for an interface has partly lead the development and design of commercial indoor scanning products.

Dynamic occlusions and static occlusions are two different things. If a scan is planned, it needs to be taken into account that indoor spaces are often full of people and objects. In technical sense, people that move around may be referred to as dynamic occlusions while objects that do not move present static occlusions. Dynamic occlusions can be detected by performing scans of the same environment at different instances of time and then comparing the obtained point clouds. If only one instance of time is used, then dynamic occlusions may not be easily detected. Static occlusions need to be treated in the reconstruction phase, for example, oftentimes a priori knowledge of the environment is used to fill in the gaps (Sect. 3.6). Different measures to detect changes in indoor MLS point clouds are discussed in Lehtola et al. (2017).

Reflection is the 'evil twin' of occlusion. Reflections of optical rays may occur from transparent surfaces, e.g. glass, or shiny surfaces, e.g. metal. With digital images, light sources are probably the most common cause for reflections from surfaces. As another simple example, the first return of a laser beam is backscattered from a window and the second one follows from the beam hitting something solid beyond that window. These are a typical cause of outliers in the indoor 3D data. One straightforward way of eliminating these is to use a threshold value to omit returns that have a low intensity value. However, this is not always feasible for automated methods, as the threshold value depends on multiple factors and hence may appear arbitrary.

Outliers in indoor 3D data may be considerably harder to eliminate than the ones present in 3D object data because objects have a simple (convex hull) topology while indoor spaces usually do not. In other words, while outlier points inside a 3D object are harmless, they are a problem inside a room. Also, airborne scanned laser data that forms a surface with height differences usually is easier to de-noise than an indoor 3D point cloud that contains empty spaces inside.

In indoor 3D scanning, all surfaces are explicit surfaces in contrast to object 3D scanning. When scanning separate objects, e.g. by moving a camera around them, it is typically assumed that the surface of that object does not contain any holes, i.e. that the surface is implicit. This assumption greatly facilitates the reconstruction, because then a coherent surface without holes is always recovered. However, this assumption must be relaxed for indoor spaces, because for example windows (or arbitrary decorations) form holes on the walls (or other surfaces). This, that all indoor surfaces are explicit, makes the reconstruction process significantly harder than what it is for single objects. Data that is missing due to scanning occlusions or due to incomplete scan coverage must then be identified, and dealt with. The identification of this missing data is plausible with e.g. machine learning techniques that can benefit from the consistence of the existing data to create an estimate for occluded shapes and textures (Sect. 3.6).

List of problems or challenges identified in indoor 3D scanning and reconstruction is then as follows

- Optical sensor challenges as in Table 3.2 (S)
- Multi-scale problem: objects of different size (S,R)
- Occlusions from the measurement geometry in a highly convoluted space (S)
- Dynamic occlusions (S) and static occlusions (R)
- Reflections and outliers (R)
- Convoluted space with explicit surfaces (S,R)

Note that some of these problems are typically solved in either the scanning (S) or the reconstruction (R) phase. This depends on the problem characteristics. The reader should keep these in mind when reading the following sections.

3.3 Map Representations

Computers (or robots) understand the indoor spaces differently than humans. In their memory, they form a map. How the map looks like is explained in a while. First, consider the following procedure where a mobile mapping system gathers optical data of the environment, while being propagated forward by a human operator:

Initially, the map does not exist. It is generated from optical observations. In this process, we can see that there are two important concepts, positioning and map expansion. These are intertwined. The captured optical data can be transformed to expand a map, if and only if the platform movement is known. That is, if the platform can be accurately positioned with respect to time. In outdoor

```
Result: Map (and other scanned data)
Initialize new map from the first scan;
while Scanning do
    Observe new data;
    if Match between new data and the stored map is found then
        Update the position of the system on the map;
        Expand the map with the new data (e.g. Figure 1.4) ;
    end
end
```

environments, global navigation satellite system (GNSS) receivers are typically employed to provide absolute positions for a mobile mapping system in a so-called PVT format (position, velocity, precise time).[4] However, as the GNSS signals are not available indoors, relative positioning methods must be employed. This means that the generated map is employed to localize the system on it. Therefore, this is called simultaneous localization and mapping (SLAM, Sect. 3.5). In other words, estimating the trajectory of the platform and estimating the map is the very same problem, that is, the problem has a dualistic nature.

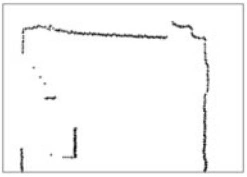
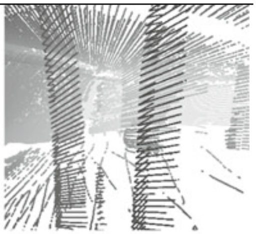
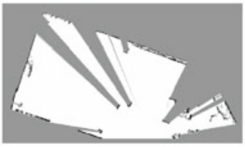
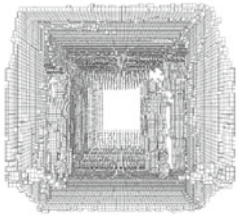
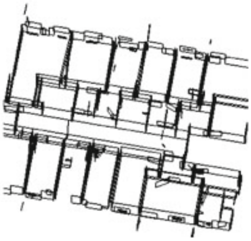
The map itself may have a variety of forms, see Table 3.3. A map can be 2D or 3D. However, the localization method used to construct it may be limited to 2D even if it outputs a 3D map. A good example of such system would be a multi-sensor system that utilizes a 2D lidar to perform the localization but has a 3D lidar or digital cameras to capture data. In some instances, these systems or methods are referred to output 2.5D maps. The 2.5D stands for two and half dimensions meaning that the outputted point clouds are 3D but there are some limitations in the method, e.g. restricting the mapping of two stories on top of each other into the same map.

Point-based maps are point clouds, such as Fig. 3.1, which are extended as the scanning continues. The benefits of these maps are that the point density can be allowed to vary from dense to sparse. The point density is stored and may be utilized later to evaluate the uncertainties in the scan result. Furthermore, point-based maps lack the discretization error that is present when the space is discretized into voxels or when planes are used to represent the space. Their limitation, however, is that they are not infinitely dense.

The voxel maps (or occupancy grids), such as Fig. 3.4, consist of cells of a given size, e.g. 5 cm^3, that are labeled as either occupied or unoccupied (or unexplored). For example, the map updating could go as follows: if during a scan a point is observed and that point resides in a voxel, then that voxel is marked as occupied. Also, the voxels residing along the line of sight to that point are also marked as unoccupied. A voxel map does not, however, have to be binary. It can also be probabilistic, for example see OctoMap (Hornung et al. 2013). Then the voxel cells are do not have binary states of being either occupied or unoccupied, but

[4] Navipedia of European Space Agency: https://gssc.esa.int/navipedia/

Table 3.3 Map representations used when scanning indoor environments

	2D	3D
Point-based		
Voxel		
Feature (plane)	N/A	

have a probability of being full (or empty). Note that in the reconstruction phase[5] (Sect. 3.6), the voxels are converted into a binary (occupied or empty) format, while here they may have unexplored or probabilistic states. Occupancy grids offer a straightforward way to represent the scanned space and are powerful in 2D, where they offer a computationally light way for keeping track of dense scans. In 3D, however, this beneficial property is severely countered by the rapidly increasing amount of (empty) voxels. The usefulness of a 3D voxel map hence easily suffers from the amount of memory required to span a large volume, because this demand increases as $O(N^3)$.

Planes are commonly used to represent floors, walls, and ceilings in indoor reconstruction (Sect. 3.6). Hence, planar features are beneficial in that they may allow for the SLAM algorithm to output models that are close to ones obtained from reconstruction (Grant et al. 2019; Karam et al. 2019). Further discussion related to SLAM algorithms is in Sect. 3.5. Other geometrical features may also be used instead or in conjunction with planar features.

[5] In reconstruction literature, voxel maps are also referred to as Manhattan world approximation.

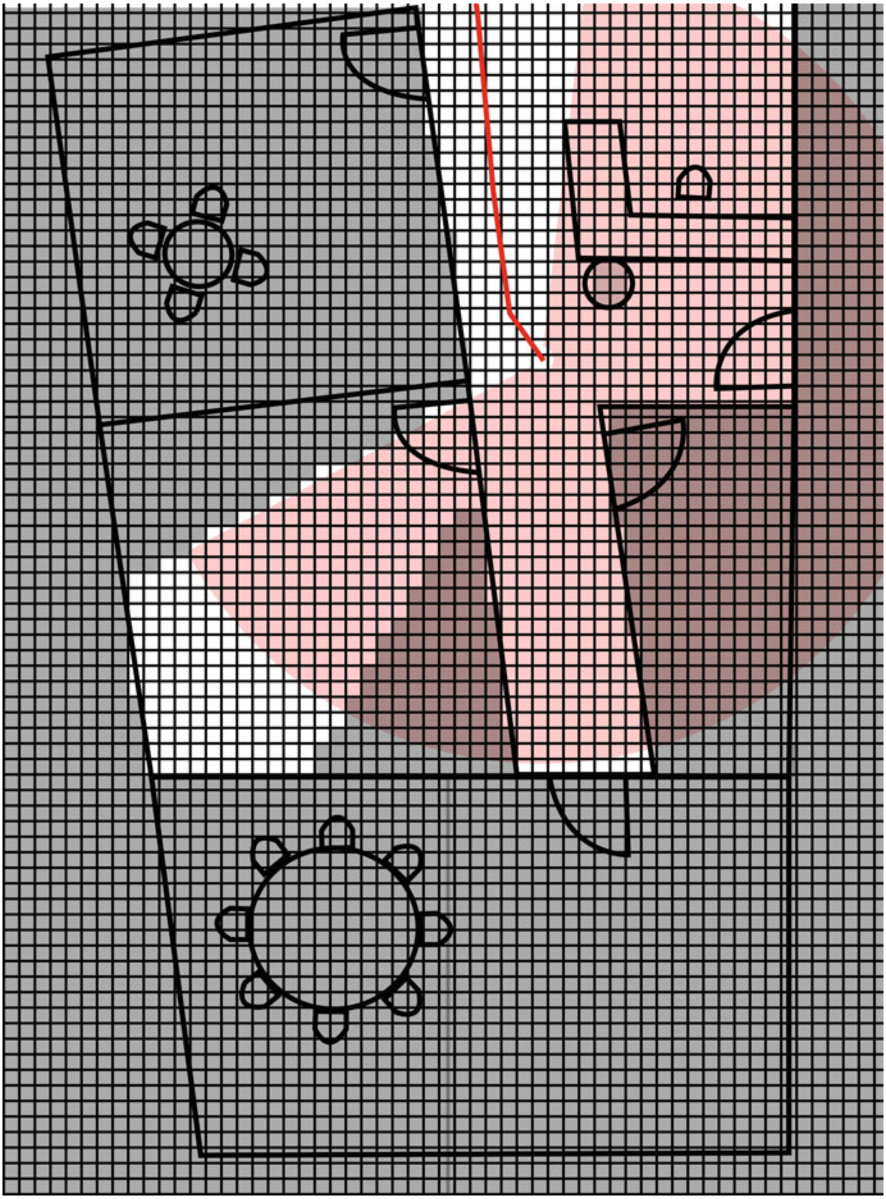


Fig. 3.4 Occupancy grid representation of an indoor environment: grey areas are not explored, white areas are empty, except those with lines. Red path is the past trajectory of the mobile mapping system. Red arc displays the field of view of the scanner. One room door is open and the scanner can partly see inside that room. Other doors are closed

Multiple maps may be created and benefited from. If the scanning system provides a map that is updated online while the operator is walking, it shows the operator which areas are yet unexplored. This is helpful, so that all the rooms and corners of the indoor environment get covered in the final detailed map that will be the output of the scanning.

3.4 Development of Indoor Scanning Systems

The history of the development of indoor scanning systems is briefly visualized in Fig. 3.5. First conscious attempts to capture whole 3D indoor environments were concluded using RGB cameras (Fig. 3.5c) and structure from motion techniques (Furukawa et al. 2009). Soon after, consumer-level depth cameras (RGB-D, Fig. 3.5a were found suitable for some limited mapping tasks but their weakness remains to be a very limited range[6] (Du et al. 2011). Almost simultaneously, a backpack platform with cameras and laser scanners was put together by Liu et al. (2010) to enable the capture of accurate geometry and textures.

The following wave of development consisted of improving the way of usage of one scanline (or 2D) lidars. A single 2D lidar was used in conjunction with an inertial sensor in a system called Zebedee (Bosse et al. 2012). Another 2D lidar system used a rotation encoder (Zhang and Singh 2014). Ultimately, a mobile 2D lidar was used without any other sensors to capture 3D indoor data (Fig. 3.5e, Lehtola et al. 2015, 2016). In other words, the a-priori model for the ego-motion required to start registering the data was successfully relaxed at a later stage of the SLAM processing. Lauterbach et al. (2015) (Fig. 3.5f) combined a 2D laser scanner in conjunction with a 3D scanner on a backpack system. Here, data from the 3D scanner could augment the 2D trajectory from an initial localization from the 2D scanner into full 3D.

The third wave of development consists of multi-line scanners (Fig. 3.5g). These relatively inexpensive but potent scanners enabled the emergence of multiple commercial systems. The systems based on these multi-line scanners include for example hand-held (Fig. 3.5h) and backpack (Fig. 3.5i) systems.

3.4.1 Single Sensor Methods and Multi-sensor Systems

Single sensor methods are important in understanding the possibilities and limitations of the sensors. Their study helps designing multi-sensor systems with optimal combinations of sensors that complement each other. In the following, we list selected state-of-the-art systems and methods.

[6] Today the commercial RGB-D cameras have a range of only up to 10 m.

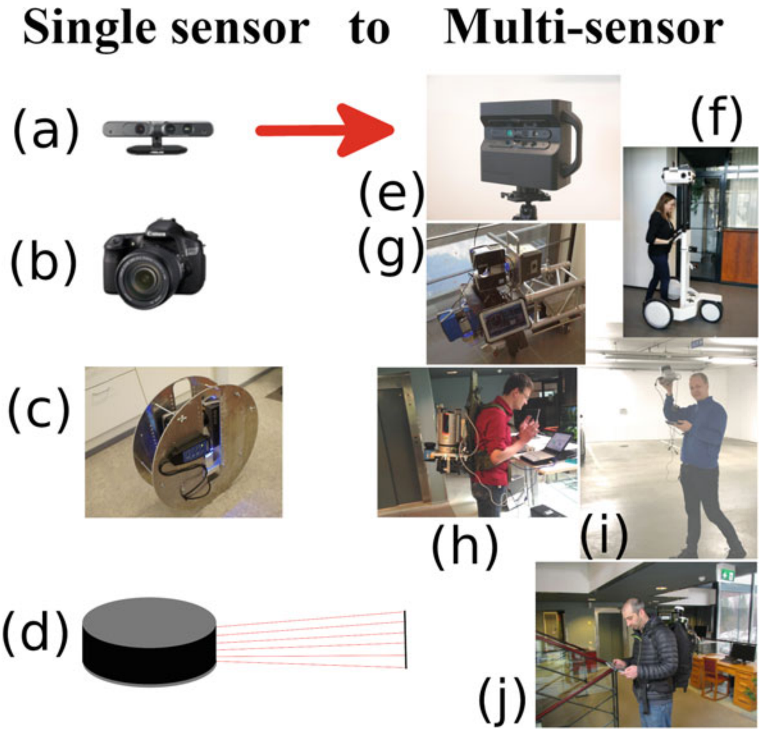


Fig. 3.5 Development of indoor 3D scanning platforms, from single sensor systems (**a**, **b**, **c**, **d**) to multi-sensor systems (**e**, **f**, **g**, **h**, **i**, **j**). These selected example systems are (**a**) RGB-D camera (Kinect), (**b**) RGB camera, (**c**) rolling 2D scanner (Aalto VILMA), (**d**) a multi-line laser scanner, (**e**) 3x RGB-D cameras (Matterport), (**f**) a multi-sensor trolley (NavVis), (**g**) a multi-sensor pushcart (FGI Slammer), (**h**) backpack with a 2D scanner and a 3D scanner (From University of Würzburg), (**i**) hand-held system (Kaarta stencil), and (**j**) multi-sensor backpack (Leica Pegasus). Many more systems exist. (**e**, **f**, **g**, **h**, **i**, **j**) are reproduced from Lehtola et al. (2017) (CC)

Single sensor methods operate on the data from

- Digital RGB and RGB-D (depth) cameras: visual SLAM, see e.g. Mur-Artal and Tardós (2017) and review by Taketomi et al. (2017). Individual frames from a video feed are matched so that the movement of the camera can be estimated.
- 2D lidar (Lehtola et al. 2016). Note that it is very challenging to reconstruct a 3D model out of 2D lidar data, and therefore these lidars are usually employed in a multi-sensor system.
- Multi-line lidar (Moosmann and Stiller 2011; Grant et al. 2019). The multi-line lidar outputs a scan that already has some 3D geometry, and the overlap from different scans can be employed in 3D scan registration for SLAM (see Sect. 3.5).

Multi-sensor systems are common in mobile mapping of indoor spaces. Our careful estimate is that there are dozens of different multi-sensor systems. Those who are interested in quantitative measures should refer to Lehtola et al. (2017), where the performance of eight different systems has been compared. Three of these are further analyzed in Tucci et al. (2018). Here, we divide multi-sensor systems into human-carriable systems and mobile platforms.

3.4.1.1 Carriable Systems

The simplest multi-sensor systems consist of a 2D lidar and an inertial sensor, e.g. Zebedee scanner (Bosse et al. 2012), or a 2D lidar and an angular decoder, e.g. LOAM (Zhang and Singh 2014). These additional sensors provide digital a-priori knowledge about the motion of the scanner that can be utilized in a prediction algorithm to make registration of the data feasible. The prediction step is especially important in providing information about the extra dimension, when a 3D reconstruction is attempted using the data from a 2D scanner. It is also worthwhile in keeping track of fast rotations when using a 3D scanner that has a low frequency (of some 10–20 Hz) in capturing scan lines, see e.g. Velas et al. (2018).

RGB cameras are also used in minimalist systems. Using an inertial measurement unit (IMU) in conjunction with a RGB camera allows for solving the absolute scale of the camera network (Nützi et al. 2011) and it provides robustness against sudden rotations where the camera system would otherwise lose track (Concha et al. 2016).

Lidar backpack systems can mount several sensors and have a combination of lidars, cameras, and IMUs. One of the first backpack systems had 3 cameras, 3 Hokyuo lidars, and one IMU (Liu et al. 2010). Since then, backpacks are seeing more 3D lidars such as Riegl VZ-400 (Lauterbach et al. 2015) and multi-line Velodyne scanners (Blaser et al. 2018).

3.4.1.2 Mobile Platforms

Mobile platforms roll on wheels, having space to mount multiple sensors to ensure a full capture of the environment. Also, they offers some advantages related to the predictability of the platform movement. For example, the localization may be conducted in 2D. Numerous experimental pushcart or trolley platforms have been assembled. For example, Radler is an instrumented surveyor's wheel that uses low-cost sensors, a 2D laser scanner and an IMU, to create 3D point clouds (Borrmann et al. 2018), while the FGI scanner (Kaijaluoto et al. 2015) (Fig. 3.5g) is more cumbersome to move, but consists of solely state-of-the-art high-end sensors. One of the known commercial platforms is NavVis (2016), see Fig. 3.5f.

3.4.1.3 Micro Aerial Vehicles

Micro Aerial Vehicles (MAVs) offer maneuverability and flexibility in mapping indoor spaces. They have, for example, been used in inventing warehouses (Eudes et al. 2018) and mapping caves (Kaul et al. 2016). One captivating scene of autonomous MAV mapping can be found in the movie Prometheus. In buildings, however, the use of MAVs is limited by that they cannot open doors and therefore are restricted by closed doors.

3.5 Iterative Closest Point SLAM

The positioning of indoor mobile mapping systems is performed using simultaneous localization and mapping (SLAM) techniques, since satellite-based positioning is unavailable indoors. We offer an example on SLAM techniques in the form of ICP-based SLAM, but the reader should be aware that there are other SLAM techniques as well. The ICP algorithm is used for matching new observations against the stored map, after which the map expansion can be done and the system position may be updated.

3.5.1 The ICP Algorithm

The ICP algorithm is the de-facto baseline for all other algorithms. The complete algorithm was invented at the same time in 1991 by Besl and McKay, by Chen and Medioni and by Zhang. The method is called the *Iterative Closest Points (ICP) algorithm.*

Given two independently acquired sets of 3D points, $\hat{M}$ (model set) and $\hat{D}$ (data set) which correspond to a single shape, we want to find the transformation $(\mathbf{R}, \mathbf{t})$ consisting of a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$ which minimizes the following cost function:

$$E(\mathbf{R}, \mathbf{t}) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{m}_i - (\mathbf{R}\mathbf{d}_i + \mathbf{t})||^2 , \tag{3.1}$$

All corresponding points can be represented in a tuple $(\mathbf{m}_i, \mathbf{d}_i)$ where $\mathbf{m}_i \in M \subset \hat{M}$ and $\mathbf{d}_i \in D \subset \hat{D}$. Two things have to be calculated: First, the corresponding points, and second, the transformation $(\mathbf{R}, \mathbf{t})$ that minimizes $E(\mathbf{R}, \mathbf{t})$ on the basis of the corresponding points. The ICP algorithm uses closest points as corresponding points. A sufficiently good starting guess, i.e. that the matched point sets are quite similarly oriented already, enables the ICP algorithm to converge to the correct minimum, see Fig. 3.6.

Current research in the context of ICP algorithms mainly focuses on fast variants of ICP algorithms (Rusinkiewicz and Levoy 2001). If the input are 3D meshes then a point-to-plane metric can be used instead of Eq. (3.1). Minimizing using a

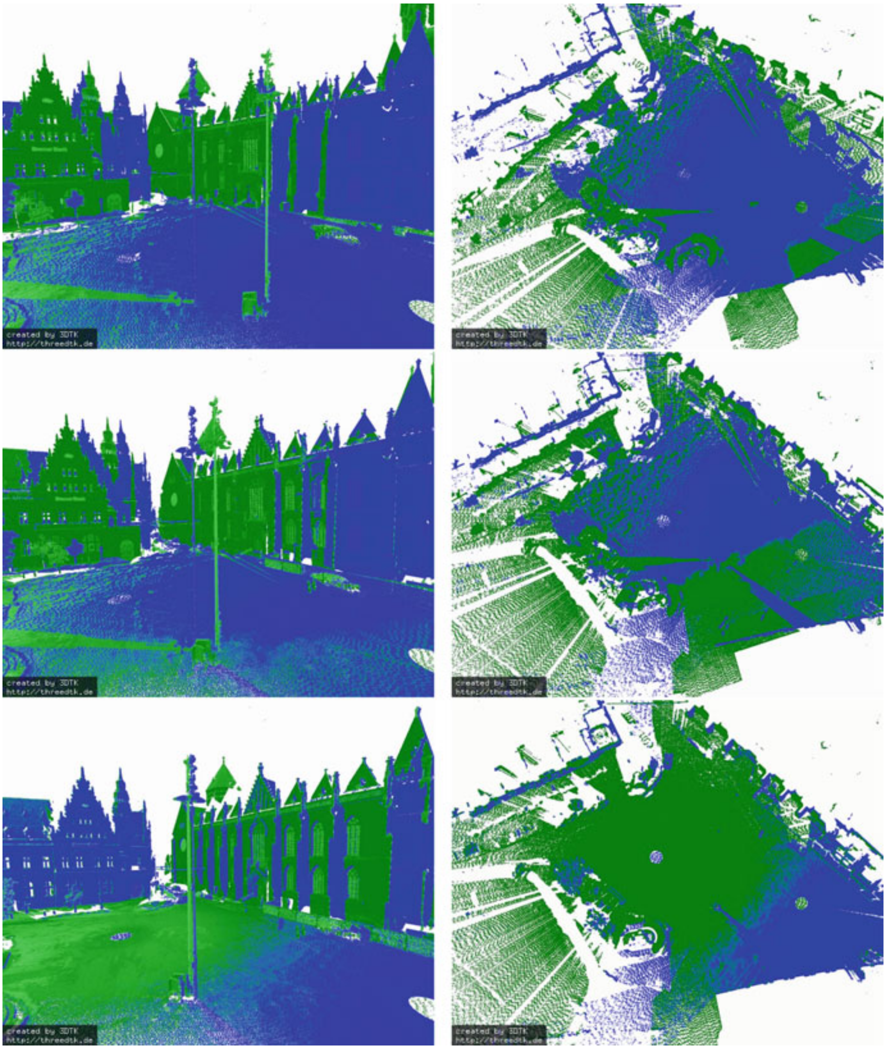


Fig. 3.6 Registration of 3D scans. The scanned scene shows the Domshof in Bremen. Left: 3D point cloud, Right: Bird eye's view. Top: Initial registration based on rough estimates. Middle: Result after 5 iterations of ICP. Below: Final registration after ICP has terminated

point-to-plane metric outperforms the standard point-to-point one, but requires the computation of normals and meshes in a pre-processing step.

The computation of closest points is the most expensive step in the ICP algorithm. Using the optimized k-d trees the cost for finding the closest point to a given query point is at average in the order of $O(\log N)$ (Friedman et al. 1977), thus the overall cost is $O(N \log N)$ (expected time). Note: N can be very large (Elseberg et al. 2013). Improvements to k-d tree search have been presented by Elseberg et al. (2012). They include approximate k-d tree search (Greenspan and Yurick 2003), registration using $d2$-trees (Mitra et al. 2004) and cached k-d tree search (Nüchter et al. 2007).

3.5.2 Computing Optimal Poses

Four algorithms are currently known that solve the error function (3.1) in closed form (Lorusso et al. 1995). The difficulty of this minimization is to enforce the orthonormality constraint for the rotation matrix **R**. Three of these algorithms separate the computation of the rotation **R** from the computation of the translation **t**. These algorithms compute the rotation first and afterward the translation is derived using the rotation. For this separation, two point sets M' and D' have to be computed, by subtracting the mean of the points that are used in the matching:

$$\mathbf{c}_m = \frac{1}{N}\sum_{i=1}^{N}\mathbf{m}_i, \qquad \mathbf{c}_d = \frac{1}{N}\sum_{i=1}^{N}\mathbf{d}_i \tag{3.2}$$

and

$$M' = \{\mathbf{m}'_i = \mathbf{m}_i - \mathbf{c}_m\}_{1,\dots,N}, \qquad D' = \{\mathbf{d}'_i \ = \mathbf{d}_i \ - \mathbf{c}_d\}_{1,\dots,N}. \tag{3.3}$$

After replacing Eqs. (3.2) and (3.3) in the error function, $E(\mathbf{R}, \mathbf{t})$ Eq. (3.1) becomes:

$$\begin{aligned} E(\mathbf{R}, \mathbf{t}) &= \frac{1}{N}\sum_{i=1}^{N}||\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i - \underbrace{(\mathbf{t} - \mathbf{c}_m + \mathbf{R}\mathbf{c}_d)}_{=\tilde{\mathbf{t}}}||^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}||\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i||^2 - \frac{2}{N}\tilde{\mathbf{t}} \cdot \sum_{i=1}^{N}\left(\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\right) + \frac{1}{N}\sum_{i=1}^{N}||\tilde{\mathbf{t}}||^2 . \end{aligned} \tag{3.4}$$

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

$$\mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d.$$

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

$$E(\mathbf{R}, \mathbf{t}) \propto \sum_{i=1}^{N} \left|\left|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\right|\right|^2. \tag{3.5}$$

1. The first method was developed in 1987 by Arun, Huang, and Blostein. The rotation $\mathbf{R}$ is represented as an orthonormal 3 × 3 matrix. The optimal rotation is calculated by $\mathbf{R} = \mathbf{V}\mathbf{U}^T$. Here the matrices $\mathbf{V}$ and $\mathbf{U}$ are derived by the singular value decomposition $\mathbf{H} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T$ of a cross correlation matrix $\mathbf{H}$. This 3 × 3 matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \sum_{i=1}^{N} \mathbf{m}_i'^T \mathbf{d}'_i = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix}, \tag{3.6}$$

   where $S_{xx} = \sum_{i=1}^{N} m'_{x,i} d'_{x,i}$, $S_{xy} = \sum_{i=1}^{N} m'_{x,i} d'_{y,i}$, ….
2. The second method is similar to the previous method and was independently developed in 1988 by Horn, Hilden and Negahdaripour. Again, a correlation Matrix $\mathbf{H}$ according to Eq. (3.6) is calculated. Afterwards a so-called polar decomposition is computed, i.e., $\mathbf{H} = \mathbf{P}\mathbf{S}$, where $\mathbf{S} = (\mathbf{H}^T\mathbf{H})^{1/2}$. For this polar decomposition, Horn et al. (1988) define a square root of a matrix. Let $\mathbf{H}$, $\mathbf{S}$ and $\mathbf{P}$ the matrices as described above. Then the optimal rotation is given by

$$\mathbf{R} = \mathbf{P} = \mathbf{H}\left(\frac{1}{\sqrt{\lambda_1}}\mathbf{u}_1\mathbf{u}_1^T + \frac{1}{\sqrt{\lambda_2}}\mathbf{u}_2\mathbf{u}_2^T + \frac{1}{\sqrt{\lambda_3}}\mathbf{u}_3\mathbf{u}_3^T\right),$$

   where $\{\lambda_i\}$ are the eigenvalues and $\{\mathbf{u}_i\}$ the corresponding eigenvectors of the matrix $\mathbf{H}^T\mathbf{H}$ (Horn et al. 1988).
3. The third method finds the transformation for the ICP algorithm by using unit quaternions. This method was invented in 1987 by Horn. The rotation represented as unit quaternion $\dot{\mathbf{q}}$, that minimizes Eq. (3.1), corresponds to the largest eigenvalue of the cross covariance matrix $\mathbf{N} =$

$$\begin{pmatrix} (S_{xx} + S_{yy} + S_{zz}) & (S_{yz} + S_{zy}) & (S_{zx} + S_{xz}) & (S_{xy} + S_{yx}) \\ (S_{yz} + S_{zy}) & (S_{xx} - S_{yy} - S_{zz}) & (S_{xy} + S_{yx}) & (S_{zx} + S_{xz}) \\ (S_{zx} + S_{xz}) & (S_{xy} + S_{yx}) & (-S_{xx} + S_{yy} - S_{zz}) & (S_{yz} + S_{zy}) \\ (S_{xy} + S_{yx}) & (S_{yz} + S_{zy}) & (S_{zx} + S_{xz}) & (-S_{xx} - S_{yy} + S_{zz}) \end{pmatrix}.$$

4. The fourth solution method for minimizing Eq. (3.1) uses so-called dual quaternions. This method was developed by Walker et al. in 1991. Unlike the first three methods covered so far the transformation is found in a single step. There is

no need to apply the trick with centroids to compute the rotation in a separate fashion. Here, the optimal transformation consisting of a rotation and translation is again a solution of the eigenvalue problem of a 4×4 matrix function that is built from corresponding point pairs.

The closed-form solutions discussed so far are all non-linear, since they need an eigenvector/eigenvalue solver, e.g., in case of using the third method, a quartic equation must be solved (Horn 1987).

For SLAM applications it is necessary to have a notion of the uncertainty of the poses calculated by the registration algorithm. The following is the extension of the probabilistic approach first proposed by Lu and Milios (1997) to 6 DoF. This extension is not straightforward, since the matrix decomposition, i.e., Eq. (3.8) cannot be derived from first principles. For a more detailed description of the extension refer to Borrmann et al. (2008a,b). In addition to the pose $\mathbf{X}$, the pose estimate $\bar{\mathbf{X}}$ and the pose error $\Delta\mathbf{X}$ are required.

The positional error of a scan at its pose $\mathbf{X}$ is described by:

$$E = \sum_{i=1}^{m} \|\mathbf{X} \oplus \mathbf{d}_i - \mathbf{m}_i\|^2 = \sum_{i=1}^{m} \|\mathbf{Z}_i(\mathbf{X})\|^2$$

Here, $\oplus$ is the compounding operation that transforms a point $\mathbf{d}_i$ into the global coordinate system. For small pose errors $\Delta\mathbf{X}$, E can be linearized by use of a Taylor expansion:

$$\begin{aligned} \mathbf{Z}_i(\mathbf{X}) &\approx \bar{\mathbf{X}} \oplus \mathbf{d}_i - \mathbf{m}_i - \nabla\mathbf{Z}_i(\bar{\mathbf{X}})\Delta\mathbf{X} \\ &= \mathbf{Z}_i(\bar{\mathbf{X}}) - \nabla\mathbf{Z}_i(\bar{\mathbf{X}})\Delta\mathbf{X} \end{aligned}$$

Utilizing the matrix decomposition $\mathbf{M}_i\mathbf{H}$ of $\nabla\mathbf{Z}_i(\bar{\mathbf{X}})$ that separates the pose $\mathbf{X}$, which is contained in $\mathbf{H}$ from the points $\mathbf{m}_i$ and $\mathbf{d}_i$, which are contained in $\mathbf{M}_i$:

$$\mathbf{Z}_i(\mathbf{X}) \approx \mathbf{Z}_i(\bar{\mathbf{X}}) - \mathbf{M}_i\mathbf{H}\Delta\mathbf{X}$$

Appropriate decompositions are given for the Euler angles, quaternion representation and the Helix transform in the following paragraphs. Because $\mathbf{M}_i$ is independent of the pose, the positional error E is approximated as:

$$E \approx (\mathbf{Z} - \mathbf{M}\mathbf{H}\Delta\mathbf{X})^T(\mathbf{Z} - \mathbf{M}\mathbf{H}\Delta\mathbf{X}),$$

where $\mathbf{Z}$ is the concatenation of all $\mathbf{Z}_i(\bar{\mathbf{X}})$ and $\mathbf{M}$ the concatenation of all $\mathbf{M}_i$'s.

E is minimized by the ideal pose:

$$\bar{\mathbf{E}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{Z}$$

and its covariance is given by

$$\mathbf{C} = s^2(\mathbf{M}^T\mathbf{M}),$$

where s^2 is the unbiased estimate of the covariance of the identically, independently distributed errors of $\mathbf{Z}_i$:

$$s^2 = (\mathbf{Z} - \mathbf{M}\bar{\mathbf{E}})^T(\mathbf{Z} - \mathbf{M}\bar{\mathbf{E}})/(2m - 3). \tag{3.7}$$

Note that $\bar{\mathbf{E}}$ is the minimum for the linearized pose $\mathbf{H}\Delta\mathbf{X}$. To obtain the optimal $\mathbf{X}$ the following transformation is performed:

$$\begin{aligned} \mathbf{X} &= \bar{\mathbf{X}} - \mathbf{H}^{-1}\bar{\mathbf{E}}, \\ \mathbf{C} &= (\mathbf{H}^{-1})\mathbf{C}(\mathbf{H}^{-1})^T. \end{aligned}$$

The representation of pose $\mathbf{X}$ in Euler angles, as well as its estimate and error is as follows:

$$\mathbf{X} = \begin{pmatrix} t_x \\ t_y \\ t_z \\ \theta_x \\ \theta_y \\ \theta_z \end{pmatrix}, \bar{\mathbf{X}} = \begin{pmatrix} \bar{t}_x \\ \bar{t}_y \\ \bar{t}_z \\ \bar{\theta}_x \\ \bar{\theta}_y \\ \bar{\theta}_z \end{pmatrix}, \Delta\mathbf{X} = \begin{pmatrix} \Delta t_x \\ \Delta t_y \\ \Delta t_z \\ \Delta\theta_x \\ \Delta\theta_y \\ \Delta\theta_z \end{pmatrix}$$

The matrix decomposition $\mathbf{M}_i\mathbf{H} = \nabla\mathbf{Z}_i(\bar{\mathbf{X}})$, i.e., the Jacobian, is given by:

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & \bar{t}_z\cos(\bar{\theta}_x) + \bar{t}_y\sin(\bar{\theta}_x) & \bar{t}_y\cos(\bar{\theta}_x)\cos(\bar{\theta}_y) - \bar{t}_z\cos(\bar{\theta}_y)\sin(\bar{\theta}_x) \\ 0 & 1 & 0 & -\bar{t}_z & -\bar{t}_x\sin(\bar{\theta}_x) & -\bar{t}_x\cos(\bar{\theta}_x)\cos(\bar{\theta}_y) - \bar{t}_z\sin(\bar{\theta}_y) \\ 0 & 0 & 1 & \bar{t}_y & -\bar{t}_x\cos(\bar{\theta}_x) & \bar{t}_x\cos(\bar{\theta}_y)\sin(\bar{\theta}_x) + \bar{t}_y\sin(\bar{\theta}_y) \\ 0 & 0 & 0 & 1 & 0 & \sin(\bar{\theta}_y) \\ 0 & 0 & 0 & 0 & \sin(\bar{\theta}_x) & \cos(\bar{\theta}_x)\cos(\bar{\theta}_y) \\ 0 & 0 & 0 & 0 & \cos(\bar{\theta}_x) & -\cos(\bar{\theta}_y)\sin(\bar{\theta}_x) \end{pmatrix}. \tag{3.8}$$

and

$$\mathbf{M}_i = \begin{pmatrix} 1 & 0 & 0 & 0 & -d_{y,i} & -d_{z,i} \\ 0 & 1 & 0 & d_{z,i} & d_{x,i} & 0 \\ 0 & 0 & 1 & -d_{y,i} & 0 & d_{x,i} \\ \end{pmatrix}.$$

As required, $\mathbf{M}_i$ contains all point information while $\mathbf{H}$ expresses the pose information. Thus, this matrix decomposition constitutes a pose linearization similar to those proposed in the preceding sections. Note that, while the matrix decomposition is arbitrary with respect to the column and row ordering of **H**, this particular description was chosen due to its similarity to the 3D pose solution given by Lu and Milios (1997).

3.5.3 Marker and Feature-Based Registration

Sometimes the ICP algorithm does not properly converge from the starting guess and is attracted into a local minimum. To avoid these issues with starting guess in the ICP framework, marker based registration uses defined artificial or natural landmarks as corresponding points. This manual data association ensures that by minimizing Eq. (3.1) the scans are registered at the correct location. Iterations are no longer required. Feature based algorithms, like using SIFT features, automatically extract the 3D position of natural features and do not need any iterations nor manual interference for registration (Böhm and Becker 2007).

While registering several 3D data sets using the ICP algorithm or marker and feature-based registration techniques, errors sum up. These errors are due to imprecise measurements and small registration errors. Therefore, globally consistent scan matching algorithm aim at reducing these errors.

3.5.4 ICP-Based SLAM

Chen and Medioni (1992) aimed at globally consistent range image alignment when introducing an incremental matching method, i.e., all new scans are registered against the so-called metascan, which is the union of the previously acquired and registered scans. This method does not spread out the error and is order-dependent.

Bergevin et al. (1996), Stoddart and Hilton (1996), Benjemaa and Schmitt (1997); Benjemaas and Schmitt (1998), and Pulli (1999) present iterative approaches. Based on networks representing overlapping parts of images, they use the ICP algorithm for computing transformations that are applied after all correspondences between all views have been found. However, the focus of research is mainly 3D modeling of small objects using a stationary 3D scanner and a turn table; therefore, the used networks consist mainly of one loop (Pulli 1999), where the loop closing has to be smoothed. These solutions are locally consistent algorithms that stick to the mentioned analogy of the spring system (Cunnington and Stoddart 1999), whereas true globally consistent algorithms minimize the error function in one step. A probabilistic approach was proposed by Williams et al. (1999), where each scan point is assigned a Gaussian distribution in order to model the statistical errors made by laser scanners. This causes high computation time due to the large amount of data in practice. Krishnan et al. (2000) presented a global registration algorithm that minimizes the global error function by optimization on the manifold of 3D rotation matrices.

The n-scan registration using linearization allows us to compute global optimal poses in one step given point correspondences between adjacent scans. These scans are given by a graph, where each link, $j \to k$ denotes a set of point pairs, i.e., closest points. Following the notation of ICP, scan j serves as the model set, while scan k serves as data set. Next we present four novel linear methods for the parameterization of the rotation.

For an uncertainty-based global point cloud registration method or SLAM method, the 2-scan case, discussed above is extended. Under the assumption that two poses $\mathbf{X}'_j$ and $\mathbf{X}'_k$ are related by the linear error metric $\mathbf{E}'_{j,k}$ we wish to minimize the Mahalanobis distance that describes the global error of all the poses:

$$\begin{aligned} W &= \sum_{j \to k} (\bar{\mathbf{E}}_{j,k} - \mathbf{E}'_{j,k})^T \mathbf{C}^{-1}_{j,k} (\bar{\mathbf{E}}'_{j,k} - \mathbf{E}'_{j,k}) \\ &= \sum_{j \to k} (\bar{\mathbf{E}}_{j,k} - (\mathbf{X}'_j - \mathbf{X}'_k)) \mathbf{C}^{-1}_{j,k} (\bar{\mathbf{E}}'_{j,k} - (\mathbf{X}'_j - \mathbf{X}'_k)). \end{aligned} \tag{3.9}$$

The error between two poses is modeled by the Gaussian distribution $(\bar{\mathbf{E}}_{j,k}, \mathbf{C}_{j,k})$. In matrix notation, W becomes:

$$W = (\bar{\mathbf{E}} - \mathbf{H}\mathbf{X})^T \mathbf{C}^{-1} (\bar{\mathbf{E}} - \mathbf{H}\mathbf{X}).$$

Here $\mathbf{H}$ is the signed incidence matrix of the pose graph, $\bar{\mathbf{E}}$ is the concatenated vector consisting of all $\bar{\mathbf{E}}'_{j,k}$ and $\mathbf{C}$ is a block-diagonal matrix comprised of $\mathbf{C}^{-1}_{j,k}$ as submatrices. Minimizing this function yields new optimal pose estimates. The minimization of $\mathbf{W}$ is accomplished via the following linear equation system:

$$\begin{aligned} (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H}) \mathbf{X} &= \mathbf{H}^T \mathbf{C}^{-1} \bar{\mathbf{E}} \\ \mathbf{B}\mathbf{X} &= \mathbf{A}. \end{aligned}$$

The matrix $\mathbf{B}$ consists of the submatrices

$$\mathbf{B}_{j,k} = \begin{cases} \sum_{k=0}^{n} \mathbf{C}^{-1}_{j,k} & (j = k) \\ \mathbf{C}^{-1}_{j,k} & (j \neq k). \end{cases}$$

The entries of $\mathbf{A}$ are given by:

$$A_j = \sum_{\substack{k=0 \\ k \neq j}}^{n} \mathbf{C}^{-1}_{j,k} \bar{\mathbf{E}}_{j,k}.$$

In addition to $\mathbf{X}$, the associated covariance of $\mathbf{C}_X$ is computed as follows:

$$\mathbf{C}_X = \mathbf{B}^{-1}$$

The actual positional error of two poses $\mathbf{X}_j$ and $\mathbf{X}_k$ is not linear:

$$E_{j,k} = \sum_{i=1}^{m} \left\| \mathbf{X}_j \oplus \mathbf{d}_i - \mathbf{X}_k \oplus \mathbf{m}_i \right\|^2 = \sum_{i=1}^{m} \left\| \mathbf{Z}_i(\mathbf{X}_j, \mathbf{X}_k) \right\|^2 .$$

Analogous to the simple 2-scan case the linearized pose difference $\mathbf{E}'_{j,k}$ is obtained by use of a Taylor expansion of $\mathbf{Z}_i(\mathbf{X}_j, \mathbf{X}_k)$:

$$\mathbf{Z}_i(\mathbf{X}_j, \mathbf{X}_k) \approx \mathbf{Z}_i(\bar{\mathbf{X}}_j, \bar{\mathbf{X}}_k) - \left(\nabla_{\mathbf{X}_j} \mathbf{Z}_i(\bar{\mathbf{X}}_j, \bar{\mathbf{X}}_k) \Delta \mathbf{X}_j - \nabla_{\mathbf{X}_k} \mathbf{Z}_i(\bar{\mathbf{X}}_j, \bar{\mathbf{X}}_k) \Delta \mathbf{X}_k\right).$$

Here, $\nabla_{\mathbf{X}_j}$ refers to the derivative with respect to $\mathbf{X}_j$. Utilizing the same matrix decomposition $\mathbf{M}_i\mathbf{H}$ of $\nabla \mathbf{Z}_i(\bar{\mathbf{X}})$ as in the 2-scan case $\mathbf{Z}_i(\mathbf{X}_j, \mathbf{X}_k)$ is approximated as:

$$\mathbf{Z}_i(\mathbf{X}_j, \mathbf{X}_k) \approx \mathbf{Z}_i(\bar{\mathbf{X}}_j, \bar{\mathbf{X}}_k) - \mathbf{M}_i \mathbf{E}'_{j,k},$$

where $\mathbf{E}'_{j,k}$ is the linear error metric given by:

$$\begin{aligned} \mathbf{E}'_{j,k} &= (\mathbf{H}_j \Delta \mathbf{X}_j - \mathbf{H}_k \Delta \mathbf{X}_k) \\ &= (\mathbf{X}'_j - \mathbf{X}'_k). \end{aligned}$$

$\mathbf{E}'_{j,k}$ is linear in the quantities $\mathbf{X}'_j$ that will be estimated by the algorithm. Again, the minimum of $\mathbf{E}'_{j,k}$ and the corresponding covariance are given by

$$\begin{aligned} \bar{\mathbf{E}}_{j,k} &= (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{Z} \\ \mathbf{C}_{j,k} &= s^2(\mathbf{M}^T\mathbf{M}). \end{aligned}$$

Here $\mathbf{Z}$ is the concatenated vector consisting of all $\mathbf{Z}_i = \bar{\mathbf{X}}_j \oplus \mathbf{d}_i - \bar{\mathbf{X}}_k \oplus \mathbf{m}_i$.

Note that the results have to be transformed in order to obtain the optimal pose estimates, just like in the 2-scan case.

$$\begin{aligned} \mathbf{X}_j &= \bar{\mathbf{X}}_j - \mathbf{H}_j^{-1}\mathbf{X}'_j, \\ \mathbf{C}_j &= (\mathbf{H}_j^{-1})\mathbf{C}_j^X(\mathbf{H}_j^{-1})^T. \end{aligned}$$

Note that SLAM techniques have not been developed for indoor applications only, as applications for SLAM exist also in undersea, space, underground, and forest environments. We recommend interested readers to get acquainted with probabilistic techniques such as Kalman and particle filtering in SLAM (see e.g. Thrun et al. 2005).

3.5.5 Assessing the SLAM Errors

The Mahalanobis distance of Eq. (3.9) offers a formulation for a computational error that can be minimized to match all poses. However, the minimized residual of this error can not be straightforwardly interpreted to assess the quality of the final point cloud. For example, it does not take a stance on whether the ICP algorithm has been attracted into local minima leading to failed pose matching, which in turn may lead into serious distortions in the final point cloud. In other words, the residual from Eq. (3.9) can be very small even if the final point cloud is nonsense. This is because of the dualistic nature of the SLAM problem: the pose errors are transformed onto errors in the observed 3D shape of the environment. The errors in the final point cloud are thus a function of all data (as these were used to estimate the poses), and include sensor errors, system calibration errors, and quality and extent of observation overlap. For the case of any SLAM (also ICP), the errors can therefore be

- quantitatively assessed only if reference data is available, from the trajectory or from the point cloud
- qualitatively assessed with the human eye, which is commonly used in 3D visualization, from the point cloud
- assessed against a-priori knowledge, e.g. geometric rules that require all walls to be planar or such that require all corners to be straight.

Point cloud to point cloud comparisons can be conducted by using a measure for control points, point subsets, or whole point clouds (Lehtola et al. 2017). Different measures are summarized in Table 3.4. The choice depends on the properties of the scanned object, i.e. whether it can change shape, and whether the point cloud has already been smoothed, e.g. filtered for outliers. Shape change is a property often related to human 3D body scanning but a room with swinging doors could also be considered with these measures. Smoothing is usually an integrated part of commercial products, meaning that if the data is captured with such a product, the output is a smoothed point cloud.

Once a model is reconstructed (see Sect. 3.6) from the scanned point cloud, the assessment becomes more straightforward. It is then a model to model comparison. The straightforwardness follows from that the measures such as completeness, that the model covers the reference, and correctness, that the model does not contain anything extra with respect to the reference, can be defined (Tran et al. 2019). However, the cave-at here is that then the total errors are a function of not only the SLAM process but also the reconstruction process. This may make the assessment appear oblique if the cause of errors is of interest. In industry, the reconstructed

Table 3.4 Metrics for point cloud to point cloud comparison. DEM stands for digital elevation models. (Adapted from Lehtola et al. 2017)

	Smoothed point cloud	Non-smoothed point cloud
Rigid object	L_p norms Hausdorff measure Examples: Scanned 3D objects, DEM	L_1 and L_2 norms with cutoff radius Examples: Raw point clouds
Non-rigid object	Gromov-Hausdorff Gromov-Wasserstein Examples: Shape changing objects	N/A

models are in standard formats (see Sect. 3.6) and are validated with commercial model checkers.

In mobile mapping, a real time map from an online SLAM is sometimes used to direct the operator when data is gathered. However, pose errors are typically larger for online SLAM than offline SLAM, since less data is used for overlap computation in the online versions to keep the computational load tractable. In turn, the offline, or post-processing, SLAM algorithms can optimize over all data. Such is the minimization of the Mahalanobis distance of Eq. (3.9) and such are also the so-called graphSLAM techniques that are based on graphs representing all observations (Grisetti et al. 2010).

3.6 Indoor 3D Reconstruction

By indoor 3D reconstruction we are referring to the process of generating a mesh model which is exportable to one of the standard formats such as IFC (industry foundation classes) or IndoorGML (Chen and Clarke 2017). In other words, the point clouds are replaced by a mesh that consists of continuous geometrical shapes such as planes and boundary representations (B-Rep). During a reconstruction process, a successful composition of walls is the most important factor because it defines the main layout of the interiors. However, some approaches are contented by providing a volumetric model of the interiors without an explicit representation of walls. The reconstruction process here includes the data segmentation step, where the point cloud is divided into rooms and subspaces. Note that some of the room segmentation methods explained next directly result in a final mesh model (e.g. cell decomposition), while some just assign labels to points (e.g. mathematical morphology) and require another method for the creation of the mesh.

3.6.1 Space Subdivision and Room Segmentation

Space subdivision is referred to the problem of dividing the space into semantic subspaces. Another term used in the literature for space subdivision is room segmentation. However, there are slight differences between the concepts of a room and a subspace. A room is separated from other rooms by permanent structures such as walls, floors and ceilings and there should be an opening (e.g. door) to connect two rooms. A subspace can represent a room or part of a room, for example a meeting area which is separated from the rest of that room by temporary partitions. When spaces are physically separated by permanent structures, space subdivision is equivalent to room segmentation. Several important remarks need to be considered when dealing with space subdivision:

1. The space subdivision can be done in 2D (Bormann et al. 2016), in 2.5D (Ikehata et al. 2015) and in full 3D (Mura et al. 2016).
2. The space subdivision can be done with Manhattan-World assumptions (Khoshelham and Díaz-Vilariño 2014) or without it (Ochmann et al. 2019). In Manhattan-World assumption, walls are assumed to be perfectly vertical and perpendicular to each other.
3. The trajectory of the acquisition device, in case of mobile laser scanners, can be a valuable data source for the space subdivision (Elseicy et al. 2018; Nikoohemat et al. 2018).

In the following, the most common space subdivision methods in the literature are presented along with their limitations.

Mathematical morphology The input data is converted into a 2D grid, which is essentially an image (with pixels), and or into a 3D voxel grid. The pixels (or voxels) are labeled as occupied (not accessible) or empty (accessible). A morphological erosion is applied on empty pixels which causes the occupied pixels (e.g. walls) to grow and the empty pixels to either vanish or get separated, if they had a weak connection. Then a connected component analysis is run, which identifies all connected segments of empty space. Each empty segment represents a room candidate. Finally, a morphological dilation is applied on the generated room segments to grow until the border of the room meets the occupied space (Bormann et al. 2016; Nikoohemat et al. 2018). See Fig. 3.7. The limitation of the morphology approach is that one has to make a selection for the pixel size to provide a good trade-off between computational cost and accuracy of the room topology. Obviously, smaller pixel size represents a better accuracy of the room topology but becomes computationally expensive and needs more iteration to converge to the correct number of rooms.

Delaunay Triangulation/Voronoi Diagram (Bormann et al. 2016; Turner et al. 2014): For this method, the input is either a set of points representing wall samples or a 2D grid of occupied and empty pixels (similar to the morphology case). Delaunay triangulation is run on the input producing a set of triangles that connect

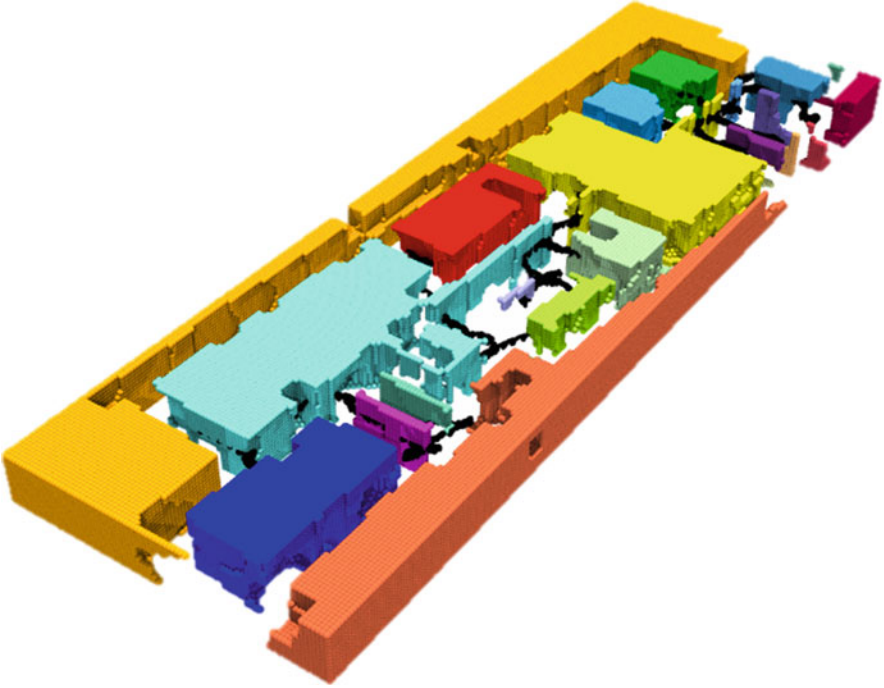


Fig. 3.7 Room segmentation by mathematical morphology. Each color is one segment. (Reproduced from Nikoohemat et al. 2018)

the wall sample points of the input data, see Fig. 3.8. Then the triangles are labeled as inside or outside using the line of sight established from the scanner trajectory. If an intersection between the line of sight and the triangle is encountered, then the triangle is labeled as inside. Inside triangles are used as room seeds. For each triangle, a circumcircle is generated (i.e. the unique circle that passes through each corner point of that triangle). It is assumed that the circumcircles with the highest overlap belong to the same room, and only one of these is stored. The initial set of room seeds then equals to the largest remaining circumcircles. The result is a rough location of each room and an initial number of rooms. Finally, candidate rooms are merged under two conditions: (i) if they share a large perimeter with another room and (ii) if they share a border which is too large to be a door. This, however, results in over-segmentation in the long corridors. Delaunay triangulation is mainly implemented in 2D and then extended to 2.5D. It is not an ideal method for true 3D modeling.

Cell decomposition is perhaps the most used approach in the literature. It consists of three steps, see Fig. 3.9. (i) The input data is converted into a set of lines (2D) or planes (3D). (ii) Lines or planes are elongated so that they intersect with the bounding box limits of the modeled space. This process generates a 2D or 3D cell complex, where each cell is represented by a piece-wise planar polyhedron

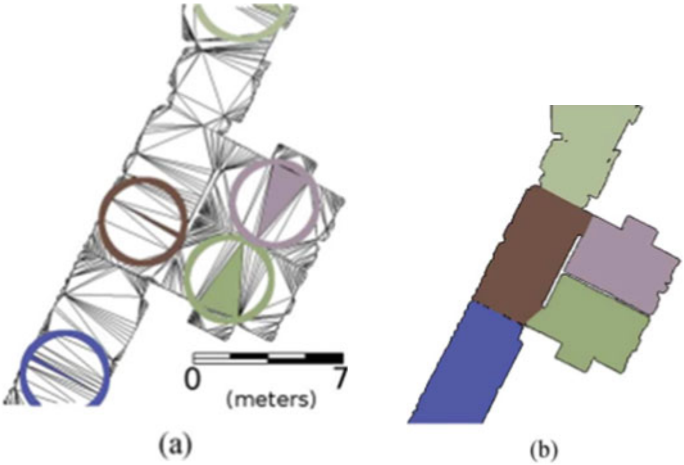


Fig. 3.8 (**a**) Delauney triangulation. The shown circles with the longest circumcircles form seeds for rooms. (**b**) Obtained room segmentation. (Reproduced from Turner et al. 2014 with permission)

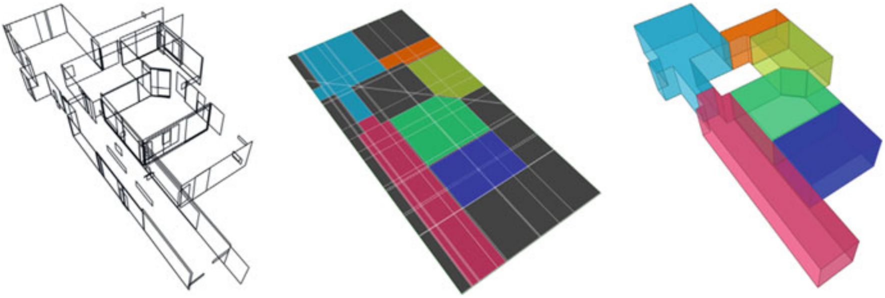


Fig. 3.9 Room segmentation by cell decomposition. Steps in (i) left, (ii) middle, and (iii) right image explained in text are visualized here. (Reproduced from Mura et al. 2014)

or a convex cell. Additionally, each cell is labeled as an inside cell or an outside cell. Typically, this is done using line-of-sight ray tracing techniques. (iii) Cells with the inside label are clustered and merged to form individual rooms. This is the step where methods show most differences, especially in the way room seeds are clustered, for example, with Markov clustering (Mura et al. 2016), Integer Linear Programming (Ochmann et al. 2019), or graph cut optimization (Oesau et al. 2014). This that there has been multiple different approaches to the optimization of the cell clustering step tells that it is not a straightforward problem, which may be considered as a limitation.

MLS trajectory-based method Mobile laser scanner data is registered into a 3D point cloud using SLAM techniques. From that process, for each point in the point cloud, we can obtain the point on the trajectory from which that particular

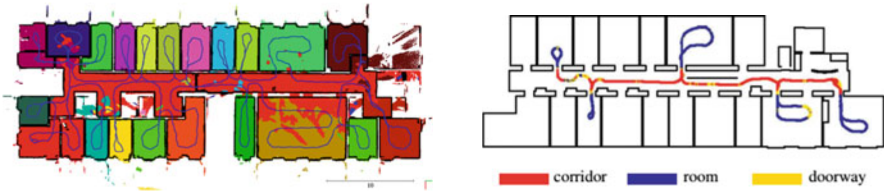


Fig. 3.10 Room segmentation by using the trajectory of a mobile laser scanning system. This method relies on doorway (opening) detection. Left: Elseicy et al. (2018), right: Mozos (2010, p.31)

measurement was done. In order to do room segmentation, first the trajectory is divided to segments using the locations, where a door is detected from the data, as division points. The location of the doors can be extracted by intersecting the trajectory with wall candidates, which are lines in 2D and planes in 3D. Obviously, only doors which were entered during the scanning can be identified and used to segment the trajectory. In addition, one can consider that each loop in the trajectory is a possible room candidate, see Fig. 3.10. By applying both criteria (door location and loops) the trajectory is segmented more robustly in that it takes into account also the cases where the entryway is larger or otherwise different than a standard door. Then the subset of the point cloud captured from that trajectory segment can be collected (using e.g. the time attribute in both point clouds and the trajectory). One limitation of this approach is the existence of openings, meaning that from the scanner positions residing inside a room parts of other spaces can also be scanned if there are openings to those spaces. This causes the room topology to become inaccurate near doors and windows. Another limitation concerns the correct detection of loops, for example if the scanning operator enters one room and exits from another door a loop is not formed. Similarly, if the operator makes an unnecessary loop in a big hall or a corridor it results in an over-segmentation of the space (Elseicy et al. 2018; Mozos 2010).

Machine learning methods bring in the possibility of naming the rooms by their functions. This so-called semantic labeling of rooms goes one step further than plain room segmentation. Machine learning methods such as random forest, adaBoost and conditional random field are used to cluster the detected planes or super segments into rooms with a function (e.g. corridor, kitchen, bedroom, etc.) (Bassier and Vergauwen 2019; Bormann et al. 2016; Mozos 2010). For this kind of methods, however, training data needs to be created and that training data should represent different types of rooms. Another novel approach is using deep learning (Convolutional Neural Network), which needs an even larger training dataset but in return is able to produce, for example, floor plans from a large set of RGB-D samples (Liu et al. 2018). Hence, the effort of creating a method is partly replaced by an effort to create a set of labeled data to be used for training. On the other hand, if the method can take sensor data as input, in a so-called end-to-end network fashion, this can be seen advantageous.

Other approaches for room segmentation such as graph-based methods, enclosure of the spaces, and shape grammar are referred to in the bibliography (Murali et al. 2017; Nikoohemat et al. 2019; Tran et al. 2018).

3.6.2 *Reconstruction of Walls*

Walls can be reconstructed as infinitely-thin planar planes, see Fig. 3.9, or as volumetric objects that have a non-zero thickness. Oftentimes the choice depends on the application, for example, in BIM models walls are volumetric objects. Given a point cloud, there are several approaches to separate walls form the furniture and to create the correct wall arrangement. Some of these approaches were explained in Sect. 3.6.1 on room segmentation, because these two problems are related. Room segmentation determines the room layout and consequently the wall arrangement. Furthermore, there is another approach that is based on constructive solid geometry (CSG) (Xiao and Furukawa 2014). It begins by slicing the point clouds to horizontal layers and looking for primitives such as rectangles in 2D or cuboids in 3D. Rectangle primitives are the most common shapes in the architecture. By exploiting a so-called free-space constraint and optimizing an objective function, the best arrangement of the primitives is selected, which means that some of the rectangles are merged into one another. Finally, the 2D CSG models are elongated in the third dimension to make them volumetric, and the objects stacked on top of each other form the sought-after 3D model with the correct wall arrangement. This approach generates volumetric walls but it does not create a topology for rooms. That is, the geometrical objects are created in their respective positions but the relations between these objects, e.g. how they form rooms, need to be established with additional means.

3.6.3 *Grammar Approach*

Grammar represents the rules of how things should be expressed. Ill-defined grammar or bad grammar results in ambiguities and misunderstandings. A natural language, English for example, would not function without grammar, because it would not make sense. Grammar is also widely applied in programming languages (see e.g. Chomsky 1959). Another types of grammar, ones intended for 3D models, can also be defined. For example, shape grammars have been used as modeling techniques in architecture, computer graphic and engineering (Stiny and Gips 1971). These were decades before the introduction of 3D scanning data. Later, grammar approaches that exploit the regularity and repetitive pattern of architectures were employed to generate models that are not based on real data, for urban designers or for computer games (Wonka et al. 2003).

Considering scanned 3D data, data-based reconstruction activities started from building façade reconstruction for city models (e.g. Müller et al. 2006; Musialski et al. 2013). Soon after, grammar was harnessed to benefit indoor 3D reconstruction (Becker et al. 2013; Boulch et al. 2013) with the founding idea that data-based indoor 3D reconstruction would yield BIM models (see Sect. 3.7). Simultaneously, automatically generated 2D floor plans could also be extracted (Ikehata et al. 2015). Floor plan generation is important, for example, in Americas as there most homes lack floor plans. Latest studies present BIM-aimed approaches where grammar rules are used iteratively to fit the best representations (parametric models with semantics) into the data (Becker et al. 2015; Tran et al. 2018). Point clouds have been established as the main input data format for grammar based indoor reconstruction.

The reconstruction of walls, or segmentation of rooms, is the goal also when we are using grammar. A grammar is a set of components such as terminals, non-terminals, rules, axioms and attributes. Rules define how the non-terminals (e.g. shape) should be transformed to terminals during the model creation. Rules generally are defined by an expert and this is the challenging part of the grammar. Some of the common rules in shape grammar are merge, split and a set of transformations. Attributes can be color, texture, material and labels. Attributes are not always part of the grammar unless we are using an attribute grammar.

One basic idea to apply grammar in indoor reconstruction is to define a primitive shape such as a cuboid as an axiom and fit the cuboid to the data (e.g. a point cloud). For this purpose a parametric shape can be defined and placed aligned with the axis in the data. If the data is axis-aligned then a Manhattan World can be assumed and fitting the cuboid to the data becomes easy. By placing, scaling, and transforming more cuboids the model can be reconstructed. Then, based on the fact that whether there are enough points to support the faces of each cuboid as valid walls, the cuboids can be merged or split to form rooms. Useful information may additionally be obtained from the locations of doorways which can be used to justify whether some cuboid faces should be added to split the cuboids (e.g. one cuboid per room). Similar approach is used by Ikehata et al. (2015) and Khoshelham and Díaz-Vilariño (2014). One limitation of using grammar lies in assuming Manhattan-World structures, as the rooms are considered to be in a grid. Another limitation is the need of an expert who defines the rules. However, as a line of research, it is possible to attempt to learn the rules form a large training data.

3.6.4 Detection and Reconstruction of Openings

Detection of openings (e.g. doors and windows) from the scanned data is important so that these openings can be successfully reconstructed. This adds an important level of detail to the model. It turns out that in most of indoor environments the structural surfaces are actually occluded (see Sect. 3.2), and as a consequence there are holes, for example, on the walls, which necessarily do not represent openings. The challenge then is to discriminate between these holes caused by occlusion and

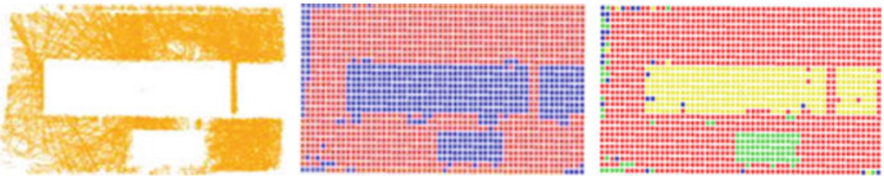


Fig. 3.11 Left: A window in a point cloud. Middle: Ray casting divides the voxels into occupied (red) and unoccupied (blue) classes. Right: Results after Eq. (3.10) include also occluded (green) and opening (yellow) classes. (Reproduced from Nikoohemat et al. 2017)

the holes which represent windows or doorway openings, for instance. We discuss two types of opening detection methods. First, ray casting can be used generally for opening detection. Second, trajectory-based methods are used for door detection.

classes

1. Ray casting is illustrated in Fig. 3.11. Initially, an image is generated for each wall surface, defined by a surface plane. The optical axis of the image is taken parallel to the normal vector of the surface plane. The image is bounded by a bounding box encompassing the wall surface. All the pixels are labeled by an initial label L_0. The objective then is to label the pixels on the surface to opening, occluded or occupied. A ray is cast from each scanner position S_i to each point P_i in the point cloud and the ray intersects with a surface at the intersection point I_j. Then two distances can be calculated from these three points: $D_{S_i-P_i}$, which is the distance between the scanner position and the measured point. $D_{S_i-I_j}$, which is the distance between scanner position and the intersection point. By comparing these two distances, we have

$$\text{Label} = \begin{cases} \text{Occlusion,} & \text{if} D_{Si-Pi} < D_{s_i-I_j} \\ \text{Occupied,} & \text{if} D_{Si-Pi} = D_{s_i-I_j} \\ \text{Opening,} & \text{if} D_{Si-Pi} > D_{s_i-I_j} \end{cases} \tag{3.10}$$

Note that some of the pixels remain with the label L_0 and in the image on the right they are shown in blue. After labeling each surface, the openings can be distinguished from occlusions and the borders of the openings can be extracted. Adan and Huber (2011) use a support vector machine (SVM) method to further reconstruct the border of the opening in each surface. Additionally, note that ray casting can also be done in 3D by using a set of voxels, which enables the treatment of more complex geometries for openings. Finally, a common drawback of any occlusion reasoning method is that when an opening is partially occluded it is hard to reconstruct the border between the opening and the occlusion.

2. Using the Trajectory: When indoor data is collected with a mobile laser scanning system, the trajectory of the laser scanner intersects doorways, see Fig. 3.12. Nikoohemat et al. (2017) exploits this fact to detect the doorways. Given a point cloud, the 3D space is voxelized and the voxels are labeled as empty and occupied (depending if there are points in the voxel). For doorway detection, the

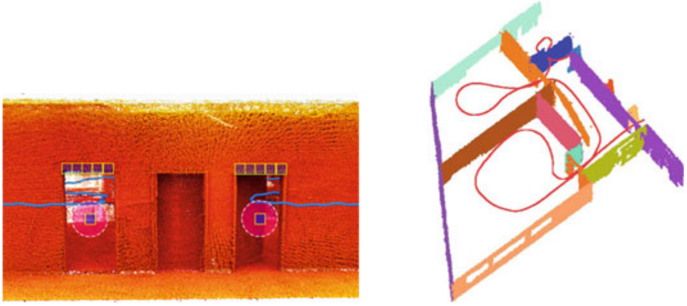


Fig. 3.12 (**a**) Snapshot from inside a point cloud with the trajectory from a mobile mapping system shown with a line. (**b**) Reconstruction of openings with the help of a known trajectory. ((**a**) is reproduced from Nikoohemat et al. 2018)

goal is to find the voxels at the center and on the top of the door frame, which gives us an approximate location of the doorway and its orientation. Note that this approach does not require any knowledge about the walls. Three criteria are checked for each voxel. It represents the center of a doorway if (i) above it, there are several occupied voxels, (ii) it resides close to the trajectory, e.g. distance is within 15 cm, and (iii) it is surrounded by empty voxels when considering a short radius (of e.g. 30 cm). The first criterion scouts for the top of the door frame. The second one is self-explanatory. The third criterion implies that the door center is in the middle of an opening (i.e. not a door that was closed). When identifying the voxel candidates for the door center, the voxels on top of the door center (considering a standard door height) are similarly identified as top of the door. For identifying closed doors, the empty voxels in the third criterion are replaced by occupied voxels, because we expect the center of a closed door is occupied by points. One of the benefits of this approach is that the doorways can be identified regardless of whether the doors are being opened or closed during the measurement. This approach has later been extended by others (Elseicy et al. 2018; Staats et al. 2019).

This method can be simplified if the wall surfaces are known. For example, each place where the trajectory intersects a wall is a probable doorway candidate. Also, the orientation of the door can be derived from the wall normal vector. Obviously, the limitation in using the trajectory knowledge is that there is no guarantee that any untraversed doorways would be detected (e.g. with closed doors).

3.6.5 Reconstructing Occluded Data by Machine Learning

When an object is between the measurement instrument and another object, the first object occludes the other object (see Sect. 3.2). In a point cloud, this is seen as missing data (or holes) that are shaped as a shadow of the first object. For 2D images, inpainting is a well-known technique to restore missing pixels (so-

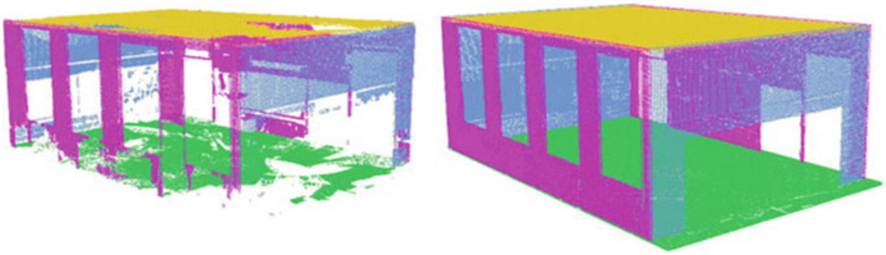


Fig. 3.13 Left: Point cloud with occlusion. Right: Disocclusion by inpainting. (Reproduced from Xiong et al. 2013)

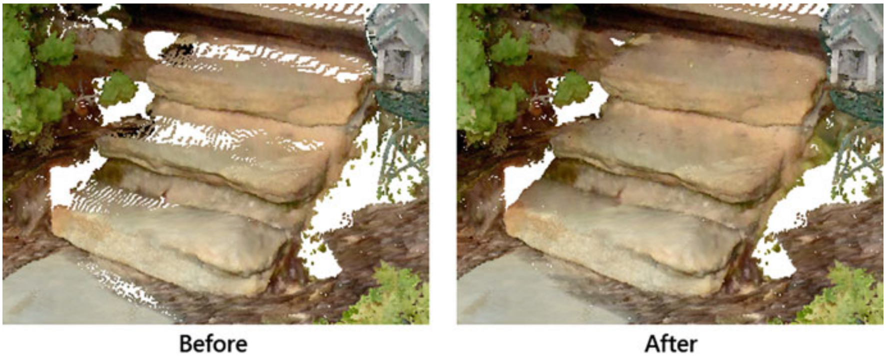


Fig. 3.14 Inpainting a 3D point cloud separately for geometry and colors and then combining the results. (Reproduced from Väänänen and Lehtola 2019)

called disocclusion in computer vision). For 3D data, a common approach is to take snapshots from the point cloud and then apply some known 2D techniques to fill the missing data. Xiong et al. (2013) use a 3D Markov Random Field inpainting algorithm for disocclusion, shown in Fig. 3.13. Essentially, the algorithm uses planar patches that are labeled as wall, ceiling, floor, or clutter, and the characteristics of opening shape and location for each patch class are learned using machine learning techniques. This learning allows for enough prior knowledge in order to distinguish between occlusion and openings, so that occlusions are filled but openings are left as they are.

Arbitrary geometries and textures can also be patched up, see Fig. 3.14. Väänänen and Lehtola (2019) train the patches separately for geometry and colors with a generative adversarial network (GAN) for each pre-defined class in the point cloud. Their method is noteworthy in that there is no need for external data, i.e. each point cloud can be inpainted per se, and the inpainting is independent of both the occlusion shape and cause. There are limitations as well. The patch size is essentially the size of the band-aid that is laid on the occluded holes, and covering large areas well is difficult.

3.7 Applications

As we have seen, indoor 3D data is important as reconstruction material and that reconstruction aims for industry standardized formats (see Sect. 3.6) in building information modeling (BIM). Figure 3.15 displays applications in facility management, asset management, and construction. Other applications for BIM include real estate brokering and various planning activities. The planning, building, and operating actions can be thought to form a so-called BIM cycle. The BIM-supported decisions relate either to the construction (or renovation) phase, during which the indoor spaces are physically modified, or to the operational phase, during which the indoor spaces are being used. The cycle is closed when the planning is revisited with the scanned as-built data and other data gathered while operating functions inside the building.

In the construction phase, for example, making spaces that are energy efficient brings savings in the fixed costs. This can be achieved through the integration of thermal data (Lagüela et al. 2013) onto 3D models which allows for insulation planning and energy conservation. Also, daylight simulations on the other hand allow for optimization of the electrical illumination (Díaz-Vilariño et al. 2014).

On the other hand, designing the spaces so that they facilitate the activities increases the efficiency of operations, which brings savings in the operational costs. Importantly, 3D models allow for detailed modeling of future operations, which in turn enables better advance planning before construction. As a specific example, hospitals are interested in the smooth flow of people and equipment and in tracking the room occupancy data. Indoor routing for pedestrians can be planned with the help of navigation graphs (Flikweert et al. 2019). Also, BIM helps with inventory data such as equipment serial numbers and make and model data so that it can be connected to room data and accessed when needed. See Fig. 3.15.

Automated model checkers are by themselves a commercial application.[7] Model checkers bring economic savings in altering the ways that problems are handled in construction. One classic example is the overlap of pipelines, meaning that a pipeline is planned to run in a space occupied already for another purpose such as another pipeline. Conventionally, such overlaps were detected in the field and solved by ad hoc methods in the order of occurrence. With 3D planning and model checking, such problems are detected already in the planning phase, and therefore better solutions are plausible. Concerning the public sector, these model checkers can be programmed to test and validate whether e.g. private 3D building plans comply with state regulations.

[7] For example, https://www.solibri.com/how-it-works

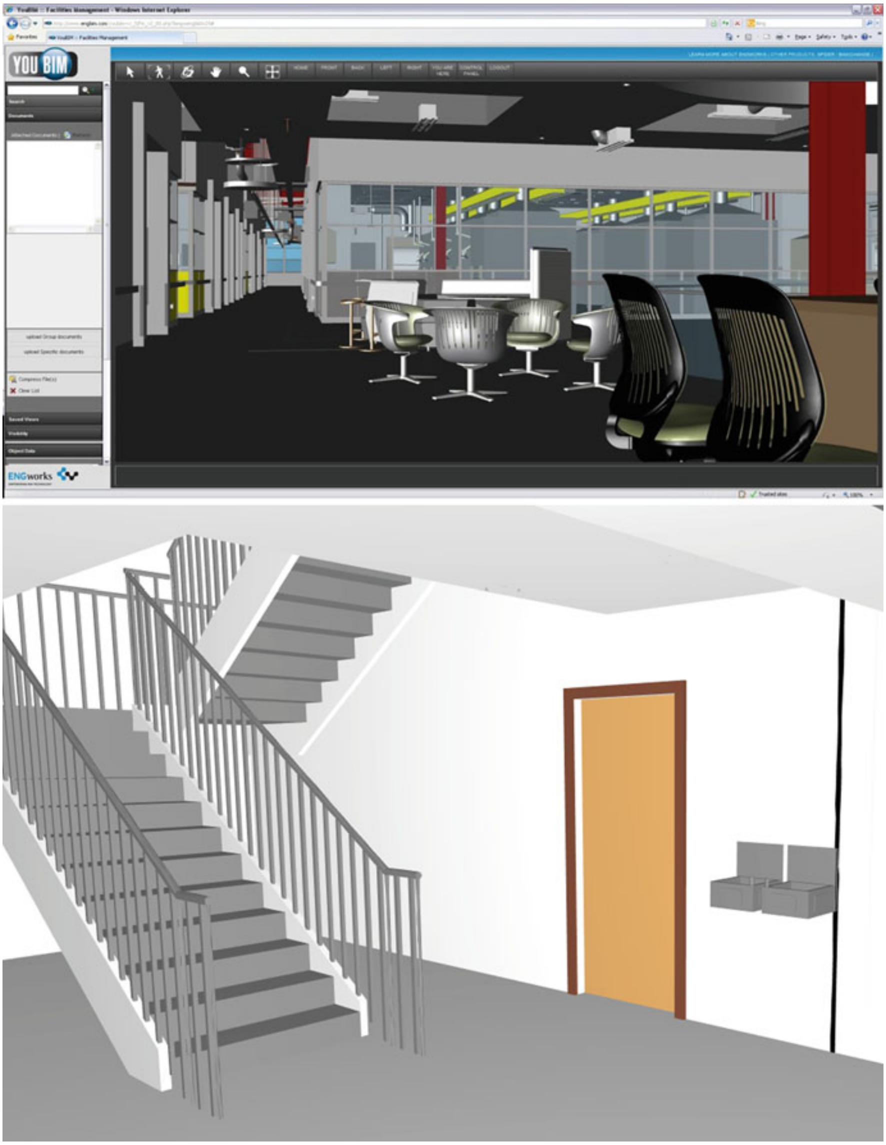


Fig. 3.15 3D BIM models for industry applications, e.g. facility and asset management (top, courtesy of Engworks) and construction (bottom, courtesy of Youbim, youbim.com). Building information modeling (BIM) is important also for various other planning applications

Imagine that your company would possess a large real-estate portfolio. To make most profit out of it, your company would benefit from having a detailed understanding of this ownership. Once the company has the understanding, it can be matched to meet the needs of clients. Further questions may then emerge: are the indoor spaces adjustable for different functions? What are the cost estimates for the adjustments? What functions would provide the most income? Is the portfolio missing something? Asset management is also important. Example assets are lights, fire extinguishers, tables, chairs, etc. Your company is asking: are all the assets that are supposed to be there in place? Are there some synergies inside the portfolio related to the procurement of new assets? Can maintenance be optimized?

Concerning the public sector, consider the vast amount of indoor spaces owned by state ministries, health services, schools, public transportation systems, and even universities. Same questions apply.

3.8 Future Trends

At the time of writing, we see the future trends in scanning, in reconstruction, and in society, as follows.

In scanning, several indoor mobile mapping systems (e.g. commercial systems Kaarta Stencil, Paracosm PX-80, Leica Pegasus: backpack) are relying on lidars that capture multiple scan lines simultaneously. This is because these multiline scanners offer robustness in the SLAM registration process. The authors expect that these multiline scanners will be upgraded into solid-state lidars which have no micro-mechanical moving parts and therefore offer more robustness in mobile use and further miniaturization possibilities for these platforms. New applications are likely to emerge. Single photon techniques are also interesting, since they require less energy to operate than traditional pulsed beams. Their cave-at is background illumination, mainly from sunlight, which may not manifest as overly restrictive in indoor environments (Lehtola et al. 2019).

In reconstruction, one of the main problems encountered is that the trouble in the capture of point clouds often leads into imperfections of the scanned surfaces. Some of these imperfections follow from the scanning geometry and from visual occlusions and manifest themselves as holes and missing observations. The authors expect that machine learning based methods that learn from the intact surfaces to tailor covering patches for the holes and missing observations shall become even more popular in the near future. Also, note that the BIM models in Fig. 3.15 are reconstructed from mobile mapping data and abide industry standards. The difference between these standardized BIM models and the reconstructed models shown in Sect. 3.6 is that there is an extra step in between. The reconstruction into a standardized BIM format typically requires manual effort as it is very hard to automate reliably, but this is about to change.

In society, the shift from 2D to 3D will eventually be completed. At the time of writing, 3D cadaster registration has been started, for example in Northern Europe, to allow for non-surface (underground) properties. In planning, private businesses are leading the way. Building information modeling (BIM) is widely used for construction planning by large construction companies, especially in Northern Europe. One major hindrance still standing in front of these models are the official processes of the state and the municipalities, such as construction permit admission processes. Oftentimes these processes cannot be set in motion with the 3D models, even if they abide an industrial standard, but the official process requires that floorplans need to be extracted into a conservative 2D form and stored in portable document format (PDF). The work on standards is therefore important (e.g. Zlatanova et al. 2016).

Finally, the authors hope that more open data sets would become available for indoor method testing in addition to the ISPRS dataset (Khoshelham et al. 2017).

3.9 Exercises for Students

The following exercises require the reading of this book chapter, some of the cited work, and – some thinking.

- How does the scanning process differ for building interiors and exteriors?
- How does the reconstruction process differ for building interiors and exteriors?
- Write a definition for the trajectory of an indoor mobile mapping system, using mathematical expressions when necessary. Let t denote time.
- An indoor space has been scanned. Give an example on how the available trajectory can be utilized to benefit the reconstruction of an indoor model.
- What are occlusions and how can they be avoided and/or dealt with to obtain watertight models?
- Give two examples on how machine learning can be benefited from in indoor scanning and/or reconstruction.
- Which representation contains more information about the indoor environment: a scanned point cloud or a reconstructed meshed model? Why?

References

Adan A, Huber D (2011) 3D reconstruction of interior wall surfaces under occlusion and clutter. In: 2011 international conference on 3D imaging, modeling, processing, visualization and transmission. IEEE, pp 275–281

Arun KS, Huang TS, Blostein SD (1987) Least square fitting of two 3-d point sets. IEEE Trans Pattern Anal Mach Intell 9(5):698–700

Bassier M, Vergauwen M (2019) Clustering of wall geometry from unstructured point clouds using conditional random fields. Remote Sens 11(13):1586

Becker S, Peter M, Fritsch D, Philipp D, Baier P, Dibak C (2013) Combined grammar for the modeling of building interiors. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 4:1–6

Becker S, Peter M, Fritsch D (2015) Grammar-supported 3D indoor reconstruction from point clouds for "as-built" bim. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 2:17–24

Benjemaa R, Schmitt F (1997) Fast global registration of 3D sampled surfaces using a multi-Z-buffer technique. In: Proceedings IEEE international conference on recent advances in 3D digital imaging and modeling (3DIM'97), Ottawa

Benjemaas R, Schmitt F (1998) A solution for the registration of multiple 3D point sets using unit quaternions. In: Computer vision – ECCV'98, vol 2, pp 34–50

Bergevin R, Soucy M, Gagnon H, Laurendeau D (1996) Towards a general multi-view registration technique. IEEE Trans Pattern Anal Mach Intell (PAMI) 18(5):540–547

Besl P, McKay N (1992) A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell (PAMI) 14(2):239–256

Blaser S, Cavegn S, Nebiker S (2018) Development of a portable high performance mobile mapping system using the robot operating system. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 4(1):13–20

Bormann R, Jordan F, Li W, Hampp J, Hägele M (2016) Room segmentation: survey, implementation, and analysis. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 1019–1026

Borrmann D, Jörissen S, Nüchter A (2018) Radler-a radial laser scanning device

Borrmann D, Elseberg J, Lingemann K, Nüchter A, Hertzberg J (2008a) Globally consistent 3D mapping with scan matching. J Robot Auton Syst (JRAS) 56(2):130–142. https://robotik.informatik.uni-wuerzburg.de/telematics/download/ras2007.pdf

Borrmann D, Elseberg J, Lingemann K, Nüchter A, Hertzberg J (2008b) The efficient extension of globally consistent scan matching to 6 DoF. In: Proceedings of the 4th international symposium on 3D data processing, visualization and transmission (3DPVT'08), Atlanta, pp 29–36. https://robotik.informatik.uni-wuerzburg.de/telematics/download/3dpvt2008.pdf

Bosse M, Zlot R, Flick P (2012) Zebedee: design of a spring-mounted 3-D range sensor with application to mobile mapping. IEEE Trans Robot 28(5):1104–1119

Boulch A, Houllier S, Marlet R, Tournaire O (2013) Semantizing complex 3D scenes using constrained attribute grammars. In: Proceedings of the eleventh eurographics/ACMSIGGRAPH symposium on geometry processing. Eurographics Association, pp 33–42

Böhm J, Becker S (2007) Automatic marker-free registration of terrestrial laser scans using reflectance features. In: Proceedings of 8th conference on optical 3D measurment techniques, Zurich, pp 338–344

Businesswire (2019) Global indoor 3D laser scanner market outlook, 2017–2026. https://www.businesswire.com/news/home/20191009005303/en/Global-Indoor-3D-Laser-Scanner-Market-Outlook

Chen J, Clarke KC (2017) Modeling standards and file formats for indoor mapping. In: GISTAM, pp 268–275

Chen Y, Medioni G (1991) Object modelling by registration of multiple range images. In: Proceedings of the IEEE conference on robotics and automation (ICRA'91), Sacramento, pp 2724–2729

Chen Y, Medioni G (1992) Object modelling by registration of multiple range images. Image Vis Comput 10(3):145–155

Chomsky N (1959) On certain formal properties of grammars. Inf Control 2(2):137–167

Concha A, Loianno G, Kumar V, Civera J (2016) Visual-inertial direct slam. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 1331–1338

Cunnington S, Stoddart A (1999) N-view point set registration: a comparison. In: Proceedings of the 10th British machine vision conference (BMVC'99), Nottingham. citeseer.nj.nec.com/319525.html

Díaz-Vilariño L, Lagüela S, Armesto J, Arias P (2014) Indoor daylight simulation performed on automatically generated as-built 3D models. Energy Build 68:54–62

Du H, Henry P, Ren X, Cheng M, Goldman DB, Seitz SM, Fox D (2011) Interactive 3D modeling of indoor environments with a consumer depth camera. In: Proceedings of the 13th international conference on ubiquitous computing. ACM, pp 75–84

Elseberg J, Magnenat S, Siegwart R, Nüchter A (2012) Comparison on nearest-neigbour-search strategies and implementations for efficient shape registration. J Softw Eng Robot (JOSER) 3(1):2–12. https://robotik.informatik.uni-wuerzburg.de/telematics/download/joser2012.pdf

Elseberg J, Borrmann D, Nüchter A (2013) One billion points in the cloud – an octree for efficient processing of 3D laser scans. ISPRS J Photogramm Remote Sens (JPRS) Special Issue Terr 3D Model 76:76–88. https://robotik.informatik.uni-wuerzburg.de/telematics/download/isprs2012.pdf

Elseicy A, Nikoohemat S, Peter M, Elberink S (2018) Space subdivision of indoor mobile laser scanning data based on the scanner trajectory. Remote Sens 10(11):1815

Eudes A, Marzat J, Sanfourche M, Moras J, Bertrand S (2018) Autonomous and safe inspection of an industrial warehouse by a multi-rotor mav. In: Field and service robotics. Springer, Cham, pp 221–235

Flikweert P, Peters R, Díaz-Vilarino L, Voûte R, Staats B (2019) Automatic extraction of a navigation graph intended for indoorgml from an indoor point cloud. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 4(2/W5):271–278

Friedman JH, Bentley JL, Finkel RA (1977) An algorithm for finding best matches in logarithmic expected time. ACM Trans Math Softw 3(3):209–226

Furukawa Y, Curless B, Seitz SM, Szeliski R (2009) Reconstructing building interiors from images. In: 2009 IEEE 12th international conference on computer vision, pp 80–87, iD: 1

Grant WS, Voorhies RC, Itti L (2019) Efficient velodyne slam with point and plane features. Auton Robots 43(5):1207–1224

Greenspan M, Yurick M (2003) Approximate K-D tree search for efficient ICP. In: Proceedings of the 4th IEEE international conference on recent advances in 3D digital imaging and modeling (3DIM'03), Banff, pp 442–448

Grisetti G, Kummerle R, Stachniss C, Burgard W (2010) A tutorial on graph-based slam. IEEE Intell Transp Syst Mag 2(4):31–43

Hess W, Kohler D, Rapp H, Andor D (2016) Real-time loop closure in 2D lidar slam. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 1271–1278

Horn BKP (1987) Closed–form solution of absolute orientation using unit quaternions. J Opt Soc Am A 4(4):629–642

Horn BKP, Hilden HM, Negahdaripour S (1988) Closed–form solution of absolute orientation using orthonormal matrices. J Opt Soc Am A 5(7):1127–1135

Hornung A, Wurm KM, Bennewitz M, Stachniss C, Burgard W (2013) Octomap: an efficient probabilistic 3D mapping framework based on octrees. Auton Robots 34(3):189–206

Ikehata S, Yang H, Furukawa Y (2015) Structured indoor modeling. In: Proceedings of the IEEE international conference on computer vision, pp 1323–1331

Kaijaluoto R, Kukko A, Hyyppä J (2015) Precise indoor localization for mobile laser scanner. ISPRS Int Arch Photogramm Remote Sens Spat Inf Sci XL-4/W5:1–6. https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-4-W5/1/2015/

Karam S, Vosselman G, Peter M, Hosseinyalamdary S, Lehtola V (2019) Design, calibration, and evaluation of a backpack indoor mobile mapping system. Remote Sens 11(8):905

Kaul L, Zlot R, Bosse M (2016) Continuous-time three-dimensional mapping for micro aerial vehicles with a passively actuated rotating laser scanner. J Field Robot 33(1):103–132

Khoshelham K, Díaz-Vilariño L (2014) 3D modelling of interior spaces: learning the language of indoor architecture. Int Arch Photogramm Remote Sens Spat Inf Sci 40(5):321

Khoshelham K, Vilariño LD, Peter M, Kang Z, Acharya D (2017) The ISPRS benchmark on indoor modelling. Int Arch Photogramm Remote Sens Spat Inf Sci 42:367–372

Krishnan S, Lee PY, Moore JB, Venkatasubramanian S (2000) Global registration of multiple 3D point sets via optimization on a manifold. In: Eurographics symposium on geometry processing

Lagüela S, Díaz-Vilariño L, Martínez J, Armesto J (2013) Automatic thermographic and rgb texture of as-built bim for energy rehabilitation purposes. Autom Constr 31:230–240

Lauterbach H, Borrmann D, Heß R, Eck D, Schilling K, Nüchter A (2015) Evaluation of a backpack-mounted 3D mobile scanning system. Remote Sens 7(10):13753–13781

Lehtola VV, Kurkela M, Hyyppä H (2014) Automated image-based reconstruction of building interiors–a case study. Photogramm J Finl 24(1):1–13

Lehtola VV, Virtanen J-P, Kukko A, Kaartinen H, Hyyppa H (2015) Localization of mobile laser scanner using classical mechanics. ISPRS J Photogramm Remote Sens 99(0):25–29. http://www.sciencedirect.com/science/article/pii/S0924271614002585

Lehtola VV, Virtanen J-P, Vaaja MT, Hyyppä H, Nüchter A (2016) Localization of a mobile laser scanner via dimensional reduction. ISPRS J Photogramm Remote Sens 121:48–59

Lehtola V, Kaartinen H, Nüchter A, Kaijaluoto R, Kukko A, Litkey P, Honkavaara E, Rosnell T, Vaaja M, Virtanen J-P et al (2017) Comparison of the selected state-of-the-art 3D indoor scanning and point cloud generation methods. Remote Sens 9(8):796

Lehtola V, Hyyti H, Keränen P, Kostamovaara J (2019) Single photon lidar in mobile laser scanning: the sampling rate problem and initial solutions via spatial correlations. Int Arch Photogramm Remote Sens Spat Inf Sci 42:91–97

Liu T, Carlberg M, Chen G, Chen J, Kua J, Zakhor A (2010) Indoor localization and visualization using a human-operated backpack system. In: 2010 international conference on indoor positioning and indoor navigation. IEEE, pp 1–10

Liu C, Wu J, Furukawa Y (2018) Floornet: a unified framework for floorplan reconstruction from 3D scans. In: Proceedings of the European conference on computer vision (ECCV), pp 201–217

Lorusso A, Eggert D, Fisher R (1995) A comparison of four algorithms for estimating 3-D rigid transformations. In: Proceedings of the 4th British machine vision conference (BMVC'95), Birmingham, pp 237–246. citeseer.nj.nec.com/lorusso95comparison.html

Lu F, Milios E (1997) Globally consistent range scan alignment for environment mapping. Auton Robots 4:333–349

Mitra NJ, Gelfand N, Pottmann H, Guibas L (2004) Registration of point cloud data from a geometric optimization perspective. In: Scopigno R, Zorin D (eds) Eurographics symposium on geometry processing, pp 23–32

Moosmann F, Stiller C (2011) Velodyne slam. In: 2011 IEEE intelligent vehicles symposium (IV). IEEE, pp 393–398

Mozos ÓM (2010) Semantic labeling of places with mobile robots, vol 61. Springer, Berlin/Heidelberg

Müller P, Wonka P, Haegler S, Ulmer A, Van Gool L (2006) Procedural modeling of buildings. In: ACM transactions on graphics (Tog), vol 25. ACM, pp 614–623

Mur-Artal R, Tardós JD (2017) Orb-slam2: an open-source slam system for monocular, stereo, and RGB-D cameras. IEEE Trans Robot 33(5):1255–1262

Mura C, Mattausch O, Villanueva AJ, Gobbetti E, Pajarola R (2014) Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. Comput Graph 44:20–32

Mura C, Mattausch O, Pajarola R (2016) Piecewise-planar reconstruction of multi-room interiors with arbitrary wall arrangements. In: Computer graphics forum, vol 35. Wiley Online Library, pp 179–188

Murali S, Speciale P, Oswald MR, Pollefeys M (2017) Indoor scan2bim: building information models of house interiors. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 6126–6133

Musialski P, Wonka P, Aliaga DG, Wimmer M, van Gool L, Purgathofer W (2013) A survey of urban reconstruction. Comput Graph Forum 32(6):146–177

NavVis (2016) Digitizing indoors – NavVis. http://www.navvis.com. Accessed: 20 Oct 2016

Nikoohemat S, Peter M, Elberink SO, Vosselman G (2017) Exploiting indoor mobile laser scanner trajectories for semantic interpretation of point clouds. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 4:355–362

Nikoohemat S, Peter M, Oude Elberink S, Vosselman G (2018) Semantic interpretation of mobile laser scanner point clouds in indoor scenes using trajectories. Remote Sens 10(11):1754

Nikoohemat S, Diakité A, Zlatanova S, Vosselman G (2019) Indoor 3D modeling and flexible space subdivision from point clouds. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 4:285–292

Nüchter A, Lingemann K, Hertzberg J (2007) Cached *k*-D tree search for ICP algorithms. In: Proceedings of the 6th IEEE international conference on recent advances in 3D digital imaging and modeling (3DIM'07), Montreal, pp 419–426. https://robotik.informatik.uni-wuerzburg.de/telematics/download/3dim2007.pdf

Nützi G, Weiss S, Scaramuzza D, Siegwart R (2011) Fusion of IMU and vision for absolute scale estimation in monocular slam. J Intell Robot Syst 61(1–4):287–299

Ochmann S, Vock R, Klein R (2019) Automatic reconstruction of fully volumetric 3D building models from oriented point clouds. ISPRS J Photogramm Remote Sens 151:251–262

Oesau S, Lafarge F, Alliez P (2014) Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. ISPRS J Photogramm Remote Sens 90:68–82

Pulli K (1999) Multiview registration for large data sets. In: Proceedings of the 2nd international conference on 3D digital imaging and modeling (3DIM'99), Ottawa, pp 160–168

Rusinkiewicz S, Levoy M (2001) Efficient variants of the ICP algorithm. In: Proceedings of the third international conference on 3D digital imaging and modellling (3DIM'01), Quebec City, pp 145–152

Schneider S, Himmelsbach M, Luettel T, Wuensche H-J (2010) Fusing vision and lidar-synchronization, correction and occlusion reasoning. In: 2010 IEEE intelligent vehicles symposium. IEEE, pp 388–393

Staats B, Diakité A, Voûte R, Zlatanova S (2019) Detection of doors in a voxel model, derived from a point cloud and its scanner trajectory, to improve the segmentation of the walkable space. Int J Urban Sci 23(3):369–390

Stiny G, Gips J (1971) Shape grammars and the generative specification of painting and sculpture. In: IFIP congress (2), vol 2

Stoddart A, Hilton A (1996) Registration of multiple point sets. In: Proceedings of the 13th IAPR international conference on pattern recognition, Vienna, pp 40–44

Taketomi T, Uchiyama H, Ikeda S (2017) Visual slam algorithms: a survey from 2010 to 2016. IPSJ Trans Comput Vis Appl 9(1):16

Thrun S, Burgard W, Fox D (2005) Probabilistic robotics. The MIT Press, Cambridge, MA

Tran H, Khoshelham K, Kealy A, Díaz-Vilariño L (2018) Shape grammar approach to 3D modeling of indoor environments using point clouds. J Comput Civil Eng 33(1):04018055

Tran H, Khoshelham K, Kealy A (2019) Geometric comparison and quality evaluation of 3D models of indoor environments. ISPRS J Photogramm Remote Sens 149:29–39

Tucci G, Visintini D, Bonora V, Parisi E (2018) Examination of indoor mobile mapping systems in a diversified internal/external test field. Appl Sci 8(3):401

Turner E, Cheng P, Zakhor A (2014) Fast, automated, scalable generation of textured 3D models of indoor environments. IEEE J Sel Top Signal Process 9(3):409–421

Väänänen P, Lehtola V (2019) Inpainting occlusion holes in 3D built environment point clouds. Int Arch Photogramm Remote Sens Spat Inf Sci 42:393–398

Velas M, Spanel M, Hradis M, Herout A (2018) Cnn for IMU assisted odometry estimation using velodyne lidar. In: 2018 IEEE international conference on autonomous robot systems and competitions (ICARSC). IEEE, pp 71–77

Walker MW, Shao L, Volz RA (1991) Estimating 3-D location parameters using dual number quaternions. CVGIP Image Underst 54:358–367

Williams J, Bennamoun M (1999) Multiple view 3D registration using statistical error models. In: Vision modeling and visualization

Wonka P, Wimmer M, Sillion F, Ribarsky W (2003) Instant architecture, SIGGRAPH '03: ACM SIGGRAPH 2003 Papers July 2003 pp 669–677 vol 22. ACM Headquarters. https://doi.org/10.1145/1201775.882324

Xiao J, Furukawa Y (2014) Reconstructing the worlds museums. Int J Comput Vis 110(3):243–258

Xiong X, Adan A, Akinci B, Huber D (2013) Automatic creation of semantically rich 3D building models from laser scanner data. Autom Constr 31:325–337

Zhang Z (1992) Iterative point matching for registration of free–form curves. Technical Report RR-1658, INRIA–Sophia Antipolis, Valbonne Cedex. citeseer.nj.nec.com/zhang92iterative.html

Zhang J, Singh S (2014) Loam: lidar odometry and mapping in real-time. In: Robotics: science and systems conference (RSS), vol 2, p 9

Zlatanova S, Van Oosterom P, Lee J, Li KJ, Lemmen C (2016) Ladm and indoorgml for support of indoor space identification. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 4:257–263