



HAL
open science

Geometric modeling of indoor scenes from acquired point data

Sven Oesau

► **To cite this version:**

Sven Oesau. Geometric modeling of indoor scenes from acquired point data. Other [cs.OH]. Université Nice Sophia Antipolis, 2015. English. NNT : 2015NICE4034 . tel-01176721

HAL Id: tel-01176721

<https://theses.hal.science/tel-01176721v1>

Submitted on 15 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ NICE SOPHIA ANTIPOLIS
ECOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION
ET DE LA COMMUNICATION

PHD THESIS

to obtain the title of

PhD of Science

of the University of Nice - Sophia Antipolis

Specialty : COMPUTER SCIENCE

By

Sven OESAU

Geometric modeling of indoor scenes from acquired point data

Thesis Advisors : Pierre ALLIEZ and Florent LAFARGE

prepared at INRIA Sophia Antipolis, TITANE Team

to be defended on June 24th, 2015

Jury :

Reviewers : Renaud MARLET - École des Ponts ParisTech
Andrei SHARF - Ben-Gurion University of the Negev

Advisors : Pierre ALLIEZ - INRIA (TITANE)
Florent LAFARGE - INRIA (TITANE)

Examinators : Noela DESPRÉ - Airbus Space and Defense
Andreas FABRI - GeometryFactory

Résumé

La modélisation géométrique et la sémantisation de scènes intérieures à partir d'échantillon de points et un sujet de recherche qui prend de plus en plus d'importance. Le traitement d'un ensemble volumineux de données est rendu difficile d'une part par le nombre élevé d'objets parasitant la scène et d'autre part par divers défauts d'acquisitions comme par exemple des données manquantes, du bruit, ou un échantillonnage de la scène non isotrope. Cette thèse s'intéresse de près à de nouvelles méthodes permettant de modéliser géométriquement et efficacement un nuage de point non structuré et d'y donner de la sémantique et se répartie en trois axes : détection de forme, classification et reconstruction. Dans le chapitre 2, nous présentons deux méthodes permettant de transformer le nuage de points en un ensemble de formes. Nous proposons en premier lieu une méthode d'extraction de lignes qui détecte des segments à partir d'une coupe horizontale du nuage de point initiale. Puis nous introduisons une méthode par croissance de régions qui détecte et renforce progressivement des régularités parmi les formes planaires. Cette méthode utilise les régularités usuelles des environnements transformés par l'Homme, i.e. la coplanarité, le parallélisme et l'orthogonalité, cela afin de réduire la complexité du problème et d'améliorer le fittage de donné lorsqu'elles sont défectueuses. Dans la première partie du chapitre 3, nous proposons une méthode basée sur de l'analyse statistique afin de séparer de la structure de la scène les objets la parasitant. Dans la seconde partie, nous présentons une méthode d'apprentissage supervisé permettant de classifier des objets en fonction d'un ensemble de formes planaires. Nous introduisons dans le chapitre 4 une méthode permettant de modéliser géométriquement le volume d'une pièce (sans meubles). Nous commençons par partitionner l'espace en utilisant des formes élémentaires extraites de la structure inhérente à la pièce. Une formulation énergétique est ensuite utilisée afin de labelliser les régions de la partition comme étant intérieur ou extérieur de manière robuste au bruit et aux données manquantes.

Abstract

Geometric modeling and semantization of indoor scenes from sampled point data is an emerging research topic. Recent advances in acquisition technologies provide highly accurate laser scanners and low-cost handheld RGB-D cameras for real-time acquisition. However, the processing of large data sets is hampered by high amounts of clutter and various defects such as missing data, outliers and anisotropic sampling. This thesis investigates three novel methods for efficient geometric modeling and semantization from unstructured point data: Shape detection, classification and geometric modeling.

Chapter 2 introduces two methods for abstracting the input point data with primitive shapes. First, we propose a line extraction method to detect wall segments from a horizontal cross-section of the input point cloud. Second, we introduce a region growing method that progressively detects and reinforces regularities of planar shapes. This method utilizes regularities common to man-made architecture, i.e. coplanarity, parallelism and orthogonality, to reduce complexity and improve data fitting in defect-laden data.

Chapter 3 introduces a method based on statistical analysis for separating clutter from structure. We also contribute a supervised machine learning method for object classification based on sets of planar shapes.

Chapter 4 introduces a method for 3D geometric modeling of indoor scenes. We first partition the space using primitive shapes detected from permanent structures. An energy formulation is then used to solve an inside/outside labeling of a space partitioning, the latter providing robustness to missing data and outliers.

Keywords: Geometry processing, Shape detection, Indoor scene reconstruction, Scene understanding, 3D modeling, LiDAR data, Energy minimization, Graph cut

Contents

	Page
Contents	iii
List of Figures	vi
List of Tables	vii
1 Introduction	1
1.1 Point clouds	3
1.2 Geometric modeling of indoor scenes	7
1.3 Approaches	10
1.4 Motivation and contribution	21
1.5 List of publications	23
2 Shape detection	25
2.1 Feature sensitive line extraction	25
2.2 Planar shape detection and regularization in tandem	31
2.3 Summary	54
3 Classification	55
3.1 Statistical analysis	56
3.2 Object classification via planar abstraction	61
3.3 Conclusions	71
4 Geometric modeling of indoor space	73
4.1 Overview	74
4.2 Cell decomposition	74
4.3 Cell occupancy labeling with min-cut	75
4.4 Experiments	80
4.5 Summary	91
5 Conclusion	93
Bibliography	99

List of Figures

1.1	Terrestrial laser scanning	5
1.2	Multi-view stereo	7
1.3	Floor plan room segmentation	16
1.4	Reconstruction with edge and corner regularization	18
1.5	Reconstruction from Multi-view stereo	21
2.1	2D input point set	26
2.2	Line hypotheses at different scales	27
2.3	Multi-scale line hypothesis	29
2.4	Global clustering of line directions	30
2.5	Interleaved detection and regularization	33
2.6	Iterations	34
2.7	Growing error metric	35
2.8	Seed point selection	36
2.9	Constrained non-local refitting	38
2.10	2D Morton curve	42
2.11	Kahn building	44
2.12	Robustness	45
2.13	Kinect	46
2.14	Regularity vs. coverage	47
2.15	Road	48
2.16	Octree parameters	51
2.17	Synergy between regularization and detection	52
2.18	Detection on curved shapes	53
3.1	Cluttered and uncluttered distributions	56
3.2	Horizontal slicing applied to a synthetic scene	58
3.3	Clutter removal	59
3.4	Resampling for anisotropy removal	60
3.5	Multiscale Planar Abstraction	61
3.6	Area fragmentation under multiple scales	63
3.7	Pairwise Orientation	64
3.8	Princeton Shape Benchmark	67
3.9	Confusion matrices	68
3.10	Indoor objects	69
4.1	Reconstruction pipeline	75
4.2	3D space partitioning	76
4.3	Estimation of point coverage	78
4.4	Data term	79

4.5	Reconstruction of the Cory 5th floor	81
4.6	Reconstruction of the Euler building entry area	82
4.7	Reconstruction from a synthetic dataset	83
4.8	Reconstruction of Kinect-recorded indoor scene	84
4.9	Impact of parameters	86
4.10	Reconstruction at lower resolutions	88
4.11	Robustness to noise	89
4.12	Robustness to outliers	90
4.13	Failure case with stepped floors	91

List of Tables

2.1	Benchmark	49
3.1	Feature importance	70
3.2	Running times	70
4.1	Running times	87
4.2	Parameters and running times of lower resolution datasets	88

Introduction

Geometric modeling and semantization of indoor spaces is an emerging topic in research. While urban modeling has received much attention for more than a decade, indoor modeling surprisingly has been less explored, although it is of practical interest. The indoor space is shaped by humans for human interaction, thus we spend more time inside than outside.

Applying outdoor reconstruction methods to indoor scenes is not relevant as indoor modeling pose different challenges than outdoor modeling. The outside of a building can often be described by a single or few cuboids, and the amount of clutter hiding part of the geometry is rather low. In contrast, indoor spaces may exhibit fine geometric details and a high amount of clutter at various scales. The clutter in indoor scenes ranges from large piecewise linear furniture, such as closets or tables, to highly irregular objects, such as clothes or plants.

Recent advances in acquisition technologies provide high accuracy and sampling rate that allow for an effective measurement of entire insides of buildings within a handful of hours. At the same time low-cost handheld 3D scanners have become available, allowing for real-time acquisition of 3D objects or small indoor scenes. The current scientific challenge is to process and analyze these sheer amounts of so-produced indoor data to extract high-level information. The goal of this thesis is to explore novel methods for efficient geometric modeling and semantization of indoor scenes from point data.

Blueprints and precise models are often needed for the architecture, engineering and construction application domains. Due to construction tolerances and modifications performed afterwards the real geometry of a building often differs from the blueprints. Measuring the real geometry with a scanner provides precise physical measurements of the indoor space, but turning those measurements into an accurate and semantized Building Information Model (BIM) is often a manual or at best semi-automatic and labor-intensive process.

Exploring and mapping unknown indoor environments is a requirement not just for

robotic systems assisting elderly people in residential homes or industrial applications. In emergency management, e.g., in urban regions destroyed by an earthquake or a fire, fast location of casualties is crucial. However, areas are often not accessible by rescue workers. Robots can provide a survey of the site at low risk. Other applications are providing interactive maps, e.g. to extend Google Maps, or virtual visits, e.g. for museums.

In the entertainment industry real places are often modeled manually to visually replicate the original scenery. The 3D models are used to create special effects and to augment the cinematic in a visual coherent way. In computer games realistic 3D models of real world areas help providing an immersive environment. Allowing users to interact with the scene enriches the immersive experience, but requires a modeling selective to the meaning of objects.

The thesis is structured as follows:

The domain of indoor scene modeling is introduced in *Chapter 1*. We first describe the properties of the input point data and acquisition techniques. Hurdles imposed by the acquisition constraints and modalities of indoor spaces motivate the use of domain specific knowledge. The challenges in indoor scene reconstruction are categorized into abstraction and classification of point data, and geometric modeling. We review the state of the art in accordance with the identified challenges.

Chapter 2 details our two contributions for shape detection from raw points. First, we propose a line extraction method from 2D measured point data, which is sensitive to details in presence of noise and outliers. Second, we contribute a planar shape detection process that considers regularities of man-made environments to gain robustness against noise and inaccuracies due to faulty registration. The method is designed for parallel execution on GPU allowing to process millions of points within seconds.

We present methods for classification of point data in *Chapter 3*. Due to the high amount of clutter in indoor scenes the identification of structure in the input data plays an important role. We propose a machine learning method showing that planar shapes detected in point data provide sufficient information for classifying objects common to indoor environments.

Chapter 4 presents a method for reconstructing a watertight 3D model of permanent structures, such as walls, floors and ceilings, given a raw point cloud of an indoor scene. The main idea is a graph-cut formulation to solve an inside/outside labeling of a space partitioning. *Chapter 5* draws a conclusion from our proposed methods

and the perspectives for future work are discussed.

1.1 Point clouds

Point clouds are the most common and basic data type in surface reconstruction and urban modeling. A 3D point cloud is a set of spatial locations in \mathbb{R}^3 . Typically, they represent surface samples acquired through physical measurements. Additional properties such as color or normals indicating the orientation of the underlying surface may be provided. The most common techniques for acquisition are laser scanning, structured-light cameras like Kinect and multi-view imagery.

Depending on the acquisition target and used technique the data may come in two different formats: as a range image or as a point cloud. A range image provides additional information: a common acquisition origin and a neighborhood for each point based on that viewpoint. However, a recording of an indoor scene requires several scanning locations to provide sufficient coverage which does not allow for representation by a single range image.

Connectivity. Point clouds in general are not structured, i.e., there is no connectivity information between points revealing the surface associated to each point. In dense point clouds sampled on a simple object, spatial adjacency provides a good indication of topological adjacency. For indoor scenes that usually contain a large amount of diverse objects, however, the lack of connectivity information poses a major challenge for identifying single connected surfaces and objects.

Normal information. Many algorithms require normal information with the point data as it provides valuable information, e.g., to infer topological adjacency between points or to infer which side of a sample is empty/solid space. Often, the normal information is not available as not all acquisition technologies provide this data. There are different ways to estimate normal information from points. Hoppe *et al.* [HDD⁺92] proposes to fit a local tangent plane assuming that points in a local neighborhood belong to the same surface. A *principal component analysis* of the neighbored points provides the normal vector. A recent approach proposed by Boulch *et al.* [BM12] detects sharp features and can provide more accurate normal vectors by using a selective neighborhood for normal estimation. However, such estimation leads to unoriented normals. While estimated unoriented normals are

approximately orthogonal to the underlying surface, it is not defined whether they point to empty or solid space.

1.1.1 Terrestrial LiDAR scanner

Light detection and ranging (LiDAR) is an acquisition technique calculating distances by measuring the time of travel of light. Samples are taken sequentially from the surroundings measuring point after point. A typical LiDAR scanner can acquire points at a rate of up to 1 million samples per second generating dense point clouds. LiDAR scanners come in several variants: airborne LiDAR on a plane, terrestrial LiDAR mounted on a tripod, as well as smaller devices for distance sensors in mobile robots and cars. The accuracy of terrestrial LiDAR scanners is usually within a few millimeters or even below. This low level of noise renders LiDAR as a favorable candidate for reconstructing precise models for architectural planning. Measuring the distance based on the reflection of a signal, transparent surfaces, such as windows, are often only partially sampled or not sampled at all. Instead the opaque environment behind the transparent surface might be measured. Reflective surfaces, such as uncoated metal or even water at a certain incident angle, might reflect the signal in a different direction and thus result in artifacts. In case of mirrors, the scanner maps a part of the reflected scene behind the mirror. The scanning procedure of most laser scanners collects samples sending laser beams with a constant angular spacing. Surfaces facing the scanner are thus sampled uniformly with a sampling density decreasing quadratically with the distance to the scanner. The orientation of the surface relative to the scanner may lead to anisotropic sampling when the surface normal deviates from the incident laser beam, see Fig. 1.1. Obviously, a scanner can only measure samples from surfaces that can be seen from the scanning position. In particular for scenes with a high amount of clutter, many surfaces are hidden by other surfaces. An effective way to provide a more complete acquisition of a scene, several acquisitions can be integrated into one dataset. Combining several acquisitions requires knowledge about the registration of scanning positions, i.e., the relative position and orientation. There are two ways to obtain this information. First, the single scans can be aligned in software after the acquisition is done by using rigid registration methods like *Iterative Closest Point* [BSMW14]. A sufficient overlap between scans is required to match adjacent scans and provide a good alignment. An accumulation of error across several scans is likely.

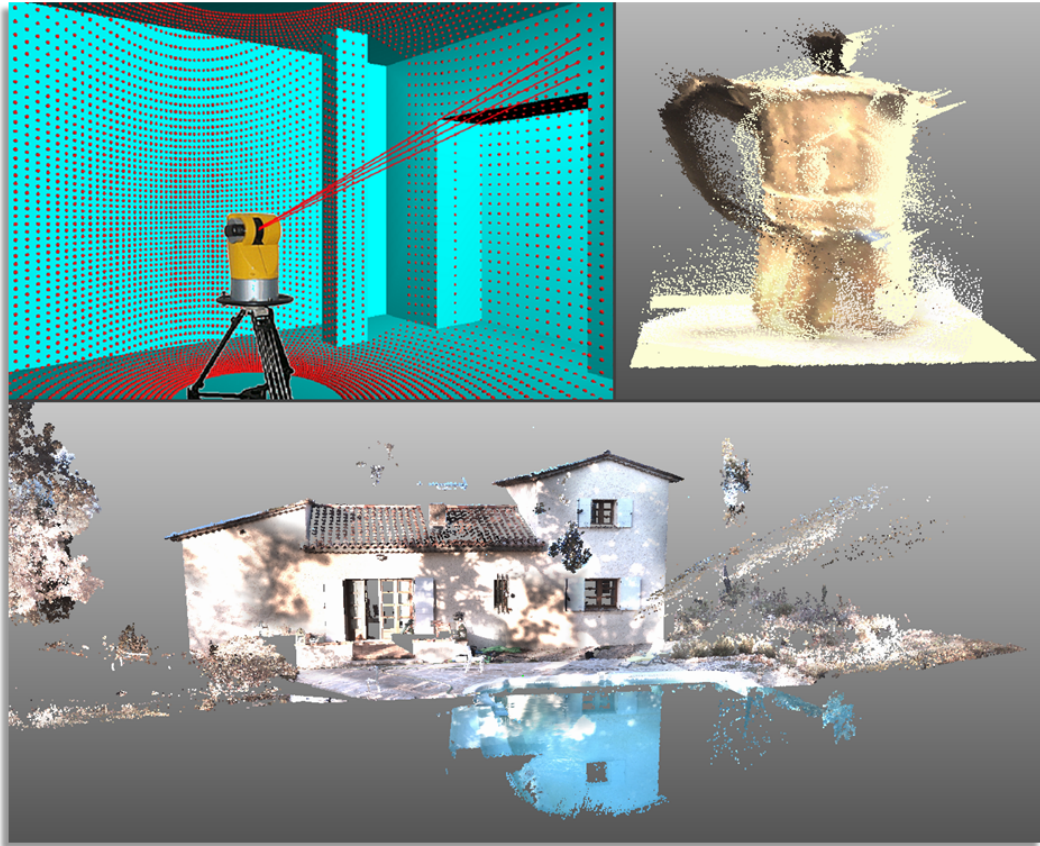


Figure 1.1: **Terrestrial laser scanning.** *Upper left:* Sampling process of an LiDAR scanner. Surfaces facing the scanner are sampled uniformly with a high density as, e.g., the wall in the back. However, surfaces sampled by an acute incident angle may lead to sparse and anisotropic sampling (*top right*). *Bottom:* Noise and artifacts caused by complex light/materials interactions in the scene. The upper right image shows artifacts and noise caused by the metallic coffee pot. In the lower image the water surface of the pool caused a reflection of the facade below ground level. Upper left image courtesy of ADWMainz [dWudLM15].

The second option is to set up physical targets during the acquisition process. For each scanning position the scanner needs to recognize at least two to three targets shared by other scanning positions. After acquisition the single scans are aligned by the target positions. In comparison the registration using targets provides the best precision, but the positioning of targets requires some planning time as well as the recognition of targets for each position. The physical alteration of the scene by setting up targets might be a drawback in some applications.

The high accuracy and sampling rate make LIDAR scanners the default choice for indoor acquisitions, however, due to its high cost it might not be affordable for small companies.

1.1.2 Kinect

In recent years an affordable hand-held 3D scanner showed up: Kinect. Originally designed as an entertainment device, it became popular for researchers due to its high affordability. It provides a real-time video stream together with a depth image. The depth is acquired by structured light. A pattern is projected into the scene using infrared (IR) light. The deformed pattern is recorded by an IR camera and the depth is calculated from the deformation.

The high affordability comes at the cost of precision and range. Meant to be used in front of the television to recognize humans the acquisition range is just a few meters. The quality of the depth information is subject to structured noise that cannot be simply filtered across several frames [KE12].

Newcombe *et al.* [NIH⁺11] and Izadi *et al.* [IKH⁺11] proposed methods for registering the acquired data in real-time, allowing for a 3D reconstruction and manipulation in real-time. These methods record a single point cloud, while moving the scanner slowly through a small scene. Similar to LIDAR scanning, acquiring scenes with a Kinect sensor suffers from missing data. The mobility of the sensor during scanning and the registration of the point cloud in real-time provides feedback to improve coverage.

Multi-view stereo Multi-view stereo (MVS) is a different approach to generate a 3D point cloud from a set of images. This approach is a generalization from stereo-vision, e.g., the human vision. In a first step feature points are detected within each image and clustered to find corresponding points common to multiple images. These points provide a means to recover the camera position for each image and the camera parameters. Depth maps for every view can be generated and integrated to create a single point cloud.

Multi-view stereo techniques in general provide a lower accuracy and a significant number of outliers compared to scenes acquired by LIDAR scanners. However, recent methods show high-quality 3D reconstructions from images. MVS is comparatively easy to use as the asset cost, a good digital camera, is affordable for consumers. However, in indoor scene reconstruction MVS techniques are less pop-



Figure 1.2: **Multi-view stereo.** *Left:* One of the 27 images taken from a calvary. *Right:* Point cloud reconstructed by Vu *et al.* [VKLP09], containing a significant amount of outliers. Images courtesy by Vu *et al.* [VKLP09].

ular, as the feature sparsity of indoor structure significantly hinders the acquisition process.

1.2 Geometric modeling of indoor scenes

Point clouds acquired from physical measurements, even with high accuracy, are difficult to use directly for architectural planning or robotic navigation. A suitable representation is required to turn the highly complex into a low complexity structure. A compact model provides accessible information, fast processing and consumes little memory. Geometric modeling is the process of fitting a mathematical model to the measured data. The type of model that is fitted to the data must be chosen according to the expected data. A simple model powerful enough to explain the measured data should be favored over a flexible complicated model. Data measured from a linear process for instance, should be modeled by a linear one instead of a quadratic model, as the parameters of the linear process are directly provided by the linear model while the quadratic one might adapt to additional noise. A typical application of geometric modeling is reverse engineering of man-made objects like mechanical parts.

1.2.1 Requirements.

The requirements towards an adequate geometric model for indoor scenes emerge from the needs of the applications. Recovering a blueprint or BIM requires an accurate modeling of the structures of an indoor scene. Depending on the application a rough 2D outline of the floor plan or a highly detailed volumetric model is needed, e.g., visitor guidance map vs. architectural planning. While a basic classification of structure into floor, ceiling and walls might be sufficient for some applications, others require consideration and classification of clutter to derive contextual information to, e.g., distinguish different types of rooms.

Currently, generating a BIM from measurement data requires manual definition of the geometric model or at least manual corrections following an automatic process. The manual process is labor-intensive, with increasing richness of details as well as the increasingly larger scale and complexity of scenes, such as acquisitions covering the interior as well as the outside of a building.

Space of human interaction. Indoor environments are the everyday interaction space of humans. Consequently, a variety of different objects of all shapes and many scales can be found in the interior. This wide range of clutter often covers a significant part of the permanent structures, hampering an accurate extraction. The appearance of clutter can be substantially different between indoor environments, such as residential homes and industrial sites.

Contrary to clutter, permanent structures often match simpler assumptions such as piece-wise planarity due to manufacturing reasons. The floor can be assumed to be horizontal with few exceptions, such as ramps for wheelchairs. Staircases may display a special case due to their small scale. Generally, walls are vertical and rooms usually comprise rectangular corners to allow for the efficient use of space. The assumption of two predominant orthogonal wall directions, i.e., a Manhattan world scene, is sometimes even extended across the urban area, especially in the USA. Guided by the wall layout, clutter on the inside, especially furniture, is often aligned accordingly. However, in addition to wall directions there are far more regularities found in buildings due to ease of construction and established standards, e.g. door and window sizes.

Acquisition modalities. The modalities of the acquisition impose further challenges to the modeling process. In addition to noise other kind of defects are generated during acquisition. Imprecise registration of scans leads to mis-aligned overlaps. This affects not only the accuracy of detected structures, but also challenges geometric modeling by causing ghosting surfaces. A similar but more difficult problem is due to artifacts caused by complex material properties. Reflective surfaces distort the acquisition signal response by deflecting the signal onto local surfaces in the vicinity. Distinguishing these structured artifacts from real structure is difficult as they mimic geometric properties of original objects or structure and require consideration of the context to be discarded.

The inverse problem, i.e. absence of data, is very frequent and one of the major challenges in geometric modeling. In the presence of clutter or even just complex architecture the problem of incomplete acquisition or missing data is not avoidable. However, providing an accurate geometric model of the structures requires finding a plausible estimation of hidden structures.

Variable sampling densities restrict the amount of information distant to the scanning origin. Anisotropic sampling, caused by a small incident angle during acquisition, adds further hurdles to the detection of underlying surfaces.

We group the challenges into three different categories:

Shape detection. Abstraction of the input data yields a reduction of complexity for further processing. Objects are in general composed of a few primitive shapes. The challenge of the shape detection step is to provide a maximal complexity reduction while covering a large part of the scene and maintaining fidelity to the physical scene. Robustness to noise and outliers are also important quality criteria due to the defective nature of measurements.

Classification. The large amount of clutter and diversity of indoor scenes hamper the geometric modeling of permanent structure. A separation of the input data into permanent structure and clutter is necessary. Whether false positive or false negative classification is less tolerable depends on the subsequent processing. High-level semantization requires segmentation and classification of clutter to gain contextual information.

Reconstruction. Extraction of an accurate 3D model requires considering a domain specific knowledge. Selecting a piecewise-planar model yields an effective modeling of permanent structure.

The challenge in reconstruction is to generate geometric models faithful to the real physical structure and to provide a plausible completion of missing data while maintaining low complexity. While some applications favor simplicity over high accuracy, detecting structural details without being sensitive to noise and outliers is difficult. High fidelity to the measurement data does not necessarily imply high fidelity to the physical scene as the acquisition is a defective process. A highly regular geometric model may instead be favored by an application over a model faithful to measured data.

1.3 Approaches

This section provides an overview of the state-of-the-art in geometric modeling of indoor scenes, grouped into the three motivated categories.

1.3.1 Shape detection.

The automated detection of primitive shapes is an instance of the general problem of fitting mathematical models to data. There is a wide variety of shapes in all dimensions, the simplest example in the early days of Computer Vision being the detection of 1D shapes such as line segments in 2D images. The rapid technological advances and affordability that characterize the acquisition devices have stimulated research for detecting 2D shapes in 3D point clouds. Furthermore, the detection of 3D shapes such as cuboids in images, see Xiao *et al.* [XRT12], has shown effective to understand the arrangement of 3D objects in indoor scenes.

RANSAC. The random sample consensus (RANSAC) [FB81] has been widely used for shape detection [SHFH]. Based on stochastic sampling and probabilities, it constructs iteratively many shape hypotheses from few samples and verifies them against the input data in order to select the shapes with highest number of inliers. In addition to being a non-deterministic algorithm, RANSAC only produces satisfactory results with a probability that depends on the number of iterations. The latter is potentially huge as it depends on, e.g., the number of triplets of points re-

quired to determine a 3D plane. The construction based on a minimal set of samples provides robustness against outliers and noise. RANSAC is suitable for detection of many different shapes. However, the types of shapes that require a small minimal set of samples to uniquely define them are favored over complex models. A general quadric surfaces already requires 10 samples resulting in a huge number of possible input sample combinations.

Schnabel *et al.* [SWK07] proposed a fast RANSAC-based method for detecting several types of primitive shapes in point-cloud data. They divide the input point data into subsets. Constructed shape hypotheses are then only tested against a subset to efficiently predict the number of inliers in all input data. Further evaluation on additional subsets is only performed for the shape hypotheses with the most predicted inliers. RANSAC in general does not consider spatial proximity of inliers inherently. Schnabel *et al.* added a spatial clustering in parameter space of the shapes by a user-specified world space distance. While being efficient it does not adapt to varying point density common to acquired point data.

This approach is satisfactory in particular when the shapes are heterogeneous in size, as it optimizes for the probability to not miss the largest shapes. It is however less efficient on large-scale urban scenes containing a large number of small shapes. Adjusting the parameters of Schnabel’s algorithm to detect primitives in indoor scenes is often a trial-and-error process. Another drawback for indoor scene reconstruction is that Schnabel’s algorithm is not robust to highly variable point density and strong anisotropy. Li *et al.* [LWC⁺11] build upon the efficient RANSAC approach from Schnabel *et al.* [SWK07]. They detect relationships from the set of detected primitives and perform global optimizations to regularize, by re-fitting these primitives to the detected relationships. They deal with several types of primitive shapes and consider relationships such as parallelism, orthogonality, co-axiality and positioning. This method performs regularization after complete detection, then re-start detection until only few points are left undetected or a maximum number of iterations is reached. They show satisfactory results for mechanical parts. However, the accuracy is limited and the processing time is within minutes for datasets of only single objects with up to 850k points.

Hypothesis-then-selection. As for RANSAC this approach generates many hypotheses from the input data. This can be done, e.g., by minimal sampling and construction, or by fitting shapes locally. In a subsequent step a labeling is per-

formed to assign one hypothesis to each input sample while minimizing a global energy designed to, e.g., favor regularity. Pham *et al.* [PCYS12] applied this approach to homography detection in images using graph-cut for energy minimization.

Accumulation space. Accumulation space methods rely upon voting in parameter space of the shapes sought after. Many primitive shape hypotheses are locally fitted to data samples and accumulated in parameter space. The final shapes are extracted via clustering of the corresponding density function in parameter space, through, e.g., mean shift [Che95]. The Hough transform [Hou62] is a common accumulation space method, designed to detect simple parametric shapes such as lines and circles in grayscale images [Dav05]. Such transform is to some extent robust to occlusions and noise. Another common accumulation space method is the Gaussian sphere mapping used for instance for pipeline detection from complex industrial areas [QZN14]. Local primitive shapes are mapped via their normal onto the unit sphere, so that, e.g., points on a cone accumulate as a ring on the unit sphere. However, there is no connectivity information between those points as only the normals and not the point locations are considered. In addition, accumulators are sensitive to discretization artifacts, which has motivated robust extensions through, e.g., randomization [BM12]. For more details on the Hough transform for plane detection we refer to Borrmann *et al.* [BELN11] who evaluate the performance and accuracy of different accumulation space layouts.

Region growing. Another popular method for shape detection originated from image processing is region growing. The main idea is to emit a local hypothesis by fitting a shape to an initial seed point, then consolidate this hypothesis by growing to neighboring points. Shapes are extracted sequentially after propagation terminates. Contrary to RANSAC and Hough transform based methods, region growing inherently detects parts that are connected. For 3D data provided as depth images fast region growing methods have been proposed by Holz *et al.* [HB12]. However, depth images differ from unorganized point clouds where adjacency information between samples is missing. In addition, the position of the acquisition device is in general not available, hampering the use of the empty space defined by the line of sight. Rabbani *et al.* [RvDHV06] propose a smoothness constraint for region growing. Instead of extracting parametric shapes they aim at clustering complex structure, such as non-straight pipe conduits, and favor undersegmentation. While this is a distinctive different idea for shape detection it does not translate well to

the common piecewise planar non-industrial indoor domain.

1.3.2 Classification

Separating structure from clutter in measured point data is necessary for geometric modeling of the structure or indoor space. Statistical methods are commonly applied to take advantage of the planarity and large extents of structures. Most previous work on indoor reconstruction assumes the knowledge of the up vector and vertical floors and ceiling as well as horizontal walls.

Classification of objects inside indoor scenes is indispensable for a wide range of applications such as robotics, reverse architecture or augmented reality. Common approaches for object classification define a set of distinctive features to describe objects. A supervised machine learning method is trained to differentiate between object classes based on those features.

Room segmentation. Ochmann *et al.* [OVW⁺14] propose a method for segmenting point data acquired from an indoor scene into separate rooms and doors. Initially acquisitions within the same room are merged semi-automatically. The affiliation of points to rooms is iteratively calculated by estimating the affiliation probability for each point to each room. The affiliation probability is estimated for each point by the visibility of points assigned to other rooms. To estimate the visibility, planar shapes are detected first using RANSAC [SWK07] then by testing for intersections of the line of sight between two points. Each iteration points are assigned the room where the largest number of points are visible. When convergence is reached and the points are clustered into rooms, the doors are detected from intersections between points and scanning origin with different room labels and planes of the detected planar shapes. Results are shown on a synthetic and few acquired datasets. While the method works well on datasets with convex rooms it may fail on non-convex rooms, especially on long branched corridors. Partially scanned rooms are not detected and often considered to be part of an adjacent room. A further hurdle is imposed by the required interaction for the initial merging of the scans that belong to the same room.

Feature extraction. Image processing and machine learning have long been concerned by object classification. Supervised machine learning classifiers are often trained to build a model from labeled training data, then to predict labels for new unknown instances. A popular method for detecting and describing key feature points (keypoints) in images is the scale-invariant feature transform (SIFT) [Low99, Low04]. Keypoints for feature extraction are first located by searching for the scale space of the image with high contrast. Features are then extracted from the neighborhood of each keypoint. Performing the feature extraction at the scale with highest signal range and extracting histograms aligned with the strongest signal peak provides invariance to rotation and scaling.

Several point-based features are used for object classification from point clouds. Rusu *et al.* propose the notion of *fast point feature histograms* (FPFH) [RMBB08, RBB09] to capture local geometric properties based on normal information. Johnson *et al.* [Joh97] introduced the *spin images* as a local point descriptor. Based on a point-normal pair the neighboring points are mapped onto a pose-invariant 2D histogram. Common approaches, e.g. [TM14], combine several local point descriptors at many keypoints. Based on the resulting labels the classification hypotheses are verified by registering meshes or point clouds of known objects with the scene [AMT⁺12, Ale12]. While these approaches achieve good recognition rates, they are in general compute-intensive and have limited capability to classify unknown object instances of a class.

Object classification. Nan *et al.* [NXS12] propose an indoor object recognition method by interleaving segmentation and classification as they identify this as a linked problem. As preprocessing a random forest classifier is trained from an object database using features calculated from horizontal slabs of the upward oriented bounding box of each object. The segmentation-classification is performed as region growing on an oversegmentation of the scene into small patches. Starting from random triplets of patches, they accumulate spatially neighbored patches based on highest classification likelihood. Interleaved with the growing, each set of patches is refined by non-rigid template fitting. They show satisfying results on a few synthetic and acquired scenes, as long as the upward orientation of objects is met.

Kim *et al.* [KMYG12] introduced a graph-based primitive matching approach to

classify objects in an indoor environment captured by a handheld scanner. During a learning phase, canonical geometric primitives (e.g., planes, boxes) are fitted to the training point data and a hierarchical primitive-joint graph is built from the data. A joint herein denotes the type of junction between the primitives. During recognition primitives are fitted to the query point data. Guided by the learned hierarchical graph, the query data are iteratively segmented into objects.

Mattausch *et al.* [MPM⁺14] introduce a unsupervised machine learning method for segmenting similar objects in indoor scenes. In a preprocessing step nearly planar patches are extracted from the input data. They define a reference coordinate system for each patch and categorize into vertical and horizontal patches based on the inclination. Six geometric features of the fitted rectangle and alpha shapes are used to describe each patch. Based on the Euclidean metric, the similarity is defined as being within the *k-nearest-neighbors*. A similarity matrix is constructed from pairwise similarity between joint configurations of a patch and a spatially neighbored patch. A diffusion embedding following a clustering yields clusters of similar patches ideally the same object. Their method generates satisfactory results on a set of scanned offices, with effective clustering of objects. However, there are a few limitations. In office buildings often very similar models of chairs, tables, etc. are used in most offices. It is unclear if the method can cluster different objects of the same class well. In addition, the method cannot handle tilted or lying objects; the upward direction is a strong requirement.

Golovinskiy *et al.* [GKF09] introduced a segmentation and shape-based classification method for objects in urban environments. On large data sets they localize and segment potential objects. A small set of basic features such as estimated volume and spin images, combined with contextual features such as “located on the street”, are used to discriminate the objects. After evaluating different machine learning methods they conclude that considering different segmentation methods and adding contextual information significantly improve the detection performance.

1.3.3 Reconstruction.

Surface reconstruction has been an active research topic for decades. Despite the wide variety of methods, they often perform unsatisfactorily for extracting a surface representation from indoor scenes. They commonly result in a single surface representation approximating the entire point cloud, which is unsatisfactory for

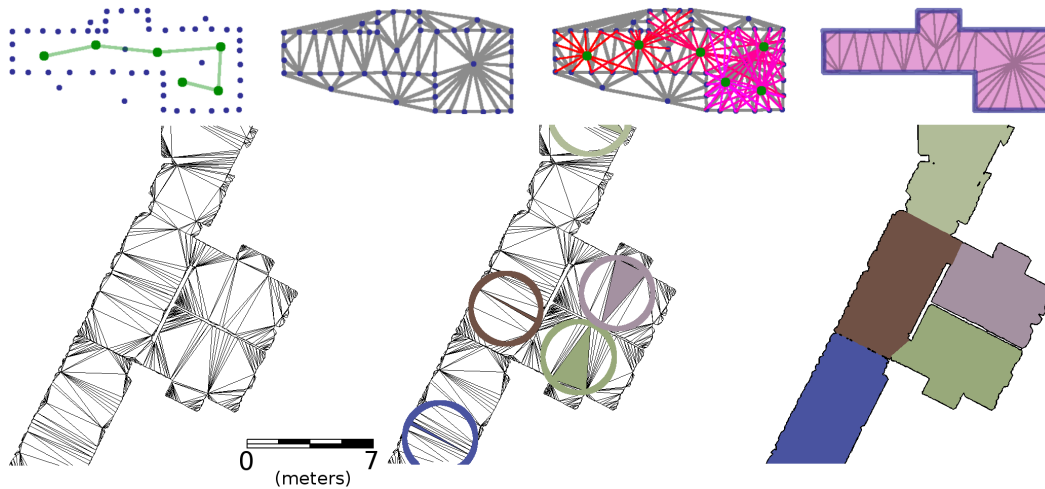


Figure 1.3: **Floor plan room segmentation.** Turner *et al.* [TZ14] label a Delaunay triangulation of a 2D point set into interior and exterior by tracing the line of sight from the scanner centers to the sampled points, see *upper row*. The *lower left* image shows the 'interior' triangles before the room segmentation. Seed triangles for the room segmentation, highlighted in the *lower mid* images, are chosen by the local maxima of the circumcircle radii. The final segmentation of rooms is shown on the *lower right*. Images courtesy by Turner *et al.* [TZ14].

further specialized processing such as semantic labeling into floors, ceilings and walls. A favored approach would provide a classification of the point cloud into permanent structures and clutter. General surface reconstruction methods instead assume that the point cloud is created from a single surface whereas indoor scenes are usually composed of planar parts and arbitrarily shaped clutter. For an in-depth review of surface reconstruction methods we refer to Berger *et al.* [BTS⁺14].

For the reconstruction of permanent structures from indoor scans, we advocate for considering a domain-specific knowledge. Common knowledge assumptions are piecewise planar permanent structures and Manhattan world scenes, i.e., exactly three orthogonal directions: two for the walls and one for floors and ceilings. We classify the existing approaches into four categories: *floor plan extraction*, *small scale reconstruction*, *scene reconstruction* and *Multi-view stereo*.

Floor plan extraction. The reconstruction of floor plans is a typical challenge for robot navigation. Okorn *et al.* [OXAH10] provide a model of the floor plan by detecting wall segments from a point cloud. They analyze the point density

along the known up direction to locate and remove floor and ceiling. A Hough transform is used to extract vertical wall segments from the remaining points. During extraction the wall segments are aligned with the two orthogonal major directions. Although the majority of walls are detected, they are often not captured in their full extent. The output of the algorithm is a set of unconnected wall segments that are used to classify the points into permanent structures and clutter. However, neither structural relations nor volumes of the indoor space are provided. A method for extracting a watertight floor plans is introduced by Turner *et al.* [TZ14]. They use point data from a horizontal 2D LIDAR scanner, a typical choice to aid navigation for mobile robots. In a first step a Delaunay triangulation of the input points is generated and the line of sight between the scanner origin and the sample points is used to label the triangles as interior or exterior. In a second step the triangles are segmented into rooms using Graph-cut. The number of rooms is guessed initially from the local maxima of circumcircle radii and the length of the shared edge between triangles is used as the regularity term favoring small edges between, e.g., doors. Analyzing the labeling allows for adjustment of the set number of rooms and the segmentation is repeated until the number of rooms converges. Afterwards a variant of Garland-Heckbert [GH97] is used to simplify the boundary. The segmentation into rooms is depicted by Fig.1.3.

Small scale reconstruction. Adan *et al.* [AH11] build on top of Okorn’s approach [OXAH10] for detecting walls and focus on modeling the shape of single wall surfaces. After wall detection each wall surface is voxelized and labeled as either occupied, empty or occluded for each scanning position. A supervised learning mechanism using a *support vector machine* (SVM) is used for segmenting and labeling wall surfaces into rectangular parts, either solid or open such as for windows and doors. The combined labels and several geometric properties are used as features during the learning process.

In parallel to the present work, Boulch *et al.* [BdLGM14] introduced a reconstruction method for modeling an indoor scene captured in a single scan while imposing regularity constraints on the model. A space partitioning is created from planar shapes extracted via region growing from the acquired range image. In a second step the cells of the space partitioning are labeled as empty or solid. The partitioning thus defines the space of possible solutions. Ghost primitives aligned with the

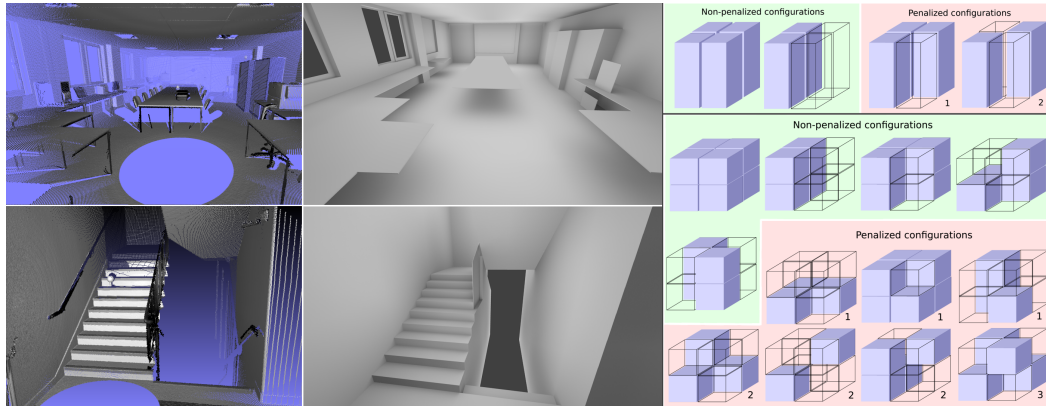


Figure 1.4: **Reconstruction with edge and corner regularization.** Boulch’s [BdLGM14] method focuses on the detailed reconstruction of an indoor scene by labeling the cells of a space decomposition. *Left:* Reconstruction of an indoor room and a staircase. The method minimizes the number of corners and the length of edges. *Right:* Constellations of cell labels around edges and corners that are penalized by the energy formulation. Images courtesy of Boulch *et al.* [BdLGM14].

detected primitives are inserted into parts of the space partitioning hidden from the scanner to generate plausible reconstructions on areas of missing data. The labels of the cells are determined via an energy formulation. The line of sight and the normal orientation of acquired points provides a label indication for the data term. The regularity term limits model complexity by minimizing surface area, length of edges and number of corners. The energy formulation is solved via LP relaxation as it contains high order terms. The results show a watertight surface with high level of detail, see Fig 1.4. Notice however that this approach is limited to processing single scans and does not differ between structure and clutter.

Scene reconstruction. Sanchez *et al.* [SZ12] recently proposed a method for modeling building interiors as an arrangement of planar polygons. They classify the points based on estimated normals by assuming a Manhattan world scene. The planar primitives are reconstructed by a region growing segmentation, least-squares fitting by employing RANSAC and alpha shapes [EKS] to reconstruct polygons. Their main contribution is a fitting of a staircase model to unsegmented points. In a first step they search the input data for eligible locations by a tolerant fitting of an inclined planar shape. The segmented points are then analyzed to estimate the parameters of the staircase model. They show results in form of polygons and

staircases detected from a multi-level building. However, there is no structure information such as adjacency of primitives.

Budroni *et al.* [BB10, BB09] introduced a reconstruction method based on sweeping methods. Manhattan world directions are detected by rotational sweeping, i.e., by rotating a vertical plane through a random input point around the up vector and counting the points close to the plane. Repeated at different points, the Manhattan world directions show up as peaks in the number of close points.

By horizontal and vertical sweeping, i.e., by moving the plane along the known directions, reveals floor, ceiling and wall positions. The floor plane is decomposed by the so-detected wall segments and marked as inside or outside based on the point density in each cell. The output is a watertight model extracted from the ground plan. However, resilience to missing data is not addressed.

Xiao *et al.* [XF12] introduce a *constructive solid geometry* (CSG) based method to reconstruct large-scale indoor environments for visualization purposes. The authors propose a greedy algorithm for creating a CSG model, guided by an objective function devised to both measure quality and control the level of detail. In order to reconstruct different floor and ceiling heights they first decompose the point cloud into horizontal slices and apply their method first in 2D, then in 3D to merge the 2D models. Although this approach can deal with more than two wall directions, it performs well on scenes mostly consisting of orthogonal or parallel structures. This limitation is mostly due to the primitive generation for the CSG model extraction: They detect linear wall segments and combine parallel or orthogonal ones for generating candidate primitives to be used by the CSG model.

Another reconstruction method of the indoor space based on merging primitives is proposed by Jenke *et al.* [JHS09] proposed an algorithm reconstructing the free space volume by merging cuboid primitives. This approach is based on Schnabel’s algorithm to detect planar primitives. The primitives are first structured in a graph, then a graph matching is applied to locate the cuboid candidates for the greedy reconstruction. Due to the fitting of cuboids their approach is limited to rectangular geometry. In presence of clutter or missing data the graph matching may fail as it requires at least five planar primitives to fit a cuboid.

Independently from our work, Mura *et al.* [MMJ⁺13, MMV⁺14] explored a diffusion process based approach to reconstruct indoor scenes with a segmentation of rooms. In order to construct a space partitioning for the diffusion process planar patches are extracted using region growing in single scans of the point cloud. The vertical

extent of patches considering occlusion is estimated and vertical patches with a certain minimal height are considered as wall segments. A 2D cell decomposition is created from clustered wall segments. The segmentation of rooms is solved via a heat diffusion and an iterative k-medoids clustering. A possible oversegmentation is solved in a post-processing step merging rooms. Watertight reconstructions from cluttered data are generated in less than a minute. However, the reconstructed models lack details of structure and cannot reconstruct different ceiling heights or staircases. Only point data in the format of range images is suitable for this method.

In addition, only Turner *et al.* [TCZ14] follow a different idea to reconstruct a watertight 3D model using a mobile laser scanner. The input data is first voxelized to reduce memory requirements. The indoor space is then carved within a voxel grid. Initialized as solid space, all voxels on the line of sight from each acquired point to its scanning origin is then marked as empty. To reduce memory requirements of voxel storage an adaptive data structure is devised to store only the boundary voxels of the solid space. Planar regions on the boundary between empty and solid space are then extracted via region growing and triangulated. The experiments show good fidelity to the input data, but the reconstruction does not differ between structure or clutter. Although the processing time is lower than other carving methods, it is still within the range of hours.

Newcombe *et al.* [NIH⁺11] and Izadi *et al.* [IKH⁺11] propose a surface reconstruction method from point clouds acquired by a low-cost consumer-grade handheld scanner. By designing their approach for parallel execution on GPUs they achieve real-time performance. Reconstructing an indoor scene via a general surface reconstruction method and generating the output as a surface mesh allows for a general and detailed representation. However, no semantic classification of the scene into wall, floor, ceiling or clutter is provided.

Reconstruction from images. Furukawa *et al.* [FCSS09a] propose a reconstruction method using Multi-view stereo from indoor images. They apply a Manhattan world model that allows them to reconstruct a piece-wise planar model. A first reconstruction of points from the images via a usual Multi-view stereo package is analyzed to detect the dominant axes. Based on the axes and reconstructed points they generate plane hypotheses. A depth image is retrieved by applying

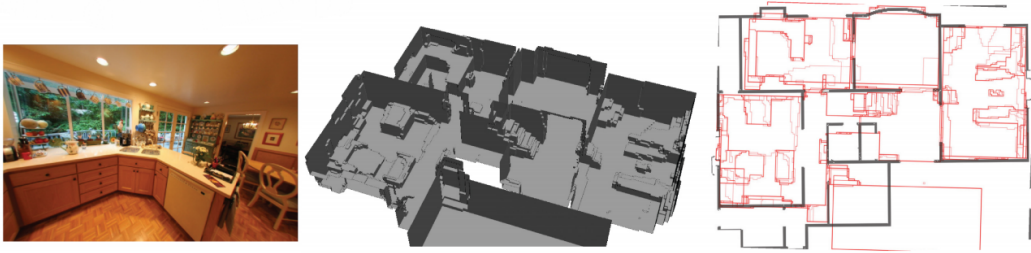


Figure 1.5: **Reconstruction from Multi-view stereo.** *Left:* One photograph used by Furukawa *et al.* [FCSS09b] during reconstruction. *Middle:* Final 3D model exhibits many details but also some blocky surface parts. *Right:* Comparison of the reconstructed floor plan (*red*) to the ground truth (*black*). Images courtesy by Furukawa *et al.* [FCSS09b].

Graph-cut to assign the pixels to those hypotheses. They show reconstructions of indoor scenes with more details than other reconstruction methods. However, the method does not differ between structure and clutter leading to complex geometry in presence of clutter. The method is restricted to reconstructions of single images. However, Furukawa *et al.* [FCSS09b] contributed a reconstruction pipeline building upon [FCSS09a] to reconstruct larger scenes from images. The depth information for each image is integrated into a single voxel representation. Each voxel is labeled as interior or exterior via an energy formulation solved via Graph-cut.

1.4 Motivation and contribution

Shape detection. For urban or indoor reconstruction, primitive shape detection is commonly performed as first step, e.g., see [MMV⁺14, JHS09, SZ12, BdLGM14, AH11, MWA⁺13], to guide the fitting of a geometric model. Primitive shapes are also amenable to the meaningful recovery of hidden or missing parts of objects. Due to practical reasons and manufacturing constraints, man-made objects and environments are often compositions of primitive shapes and exhibit a large number of regularities such as parallel, coplanar and orthogonal relationships. When dealing with defect-laden and missing data, an increasing number of reconstruction methods rely upon these regularities when deriving the surfaces from a collection of shapes [MMJ⁺13, FCSS09b, XF12]. Some shape detection approaches already consider domain-specific information

by, e.g., *favoring* approximate regularities. However, only the exact regularity of the detected shapes has substantial impact upon the complexity of the algorithms downstream the modeling pipeline. For reconstruction approaches proceeding by space partitioning in particular, coplanar and parallel primitives reduce significantly both the number of cells of the partition and the overall visual complexity of the reconstructed scenes [CLP10]. However, considering the regularities already at the detection stage does not only provide a higher regularity of the outcome, but also provides a means to improve detection in noise or sparse sampled areas and thus improve robustness against noise and outliers. Very few approaches detect and regularize altogether to minimize complexity. Li *et al.* [LWC⁺11] perform complete detection and regularization in alternation with adaption of parameters. Although this method allows for flexible optimization, the running times are in the order of minutes for point sets with less than a few 100k points and less than 100 primitives. Zhou *et al.* [ZN12] perform an iterative coarse-to-fine shape detection and regularization specific for modeling of buildings from airborne LIDAR. However, their method is domain specific and performs regularization iteratively only from coarse to fine scale without backward connection.

In *Chapter 2* we present a planar shape detection method discovering and reinforcing regularities within the input data during detection. The consideration of regularities during extraction proves to be helpful to gain robustness against noise, outliers and sparse sampling.

Classification. Many current approaches rely upon the knowledge of the up vector [MPM⁺14, KMYG12, NXS12]. Fu *et al.* [FCODS08] point out the central role of the up direction in man-made object and thus the importance for object classification. While the up vector helps simplifying the classification problem, it also restricts the detection to upward posed objects.

Keypoint based approaches typically do not require knowledge about the up direction. However, the locality requires a classification of features at each keypoint followed by a clustering into an object label. As pointed out by Alexandre [Ale12], the computational complexity is high. In addition, a point-based feature can only capture local shape properties and is therefore not easy to generalize from single object instances to object classes.

In *Chapter 3* we instead propose to classify objects based on global features derived

from planar shapes, themselves detected from the input point data. First, robust and efficient shape detection methods can abstract large point data into a set of planar shapes, at multiple scales. Second, the planar abstraction provides us with a means to extract more global information and capture common properties within object classes. Third, exploring the relationships between the planar shapes yields invariance to orientation and scale.

Reconstruction. Although there is a wide range of approaches to the geometric modeling of indoor scenes, none of them satisfies all requirements. Some exhibit a high accuracy or high level of detail [BdLGM14, ZK13], but are only applicable to small scenes and do not separate clutter. Xiao and Mura [XF12, MMV⁺14] introduce different approaches to reconstruct larger scenes while proving resilience to clutter. Although both of them are able to capture the indoor space of complex architecture, they lack structural details or require knowledge of the scanner position and the point data as range images [MMV⁺14].

We propose a reconstruction method [OLA14] by labeling an primitive-driven space decomposition. First, floor and ceiling are detected and the input data is separated into horizontal slices containing either floor, ceiling or wall segments, see Section 3.1. Second, feature-sensitive extraction of wall segments, see Section 2.1, identifies structural details and thus results in a detailed space decomposition. An energy formulation provides us with a mean to trade data faithfulness for regularity, the latter providing robustness to defect-laden data, see Section 4.

1.5 List of publications

The work in this thesis has led to the following publications:

- *Object Classification via Planar Abstraction.* Sven Oesau, Florent Lafarge, Pierre Alliez. Submitted to Symposium of Geometry Processing 2015.
- *Planar Shape Detection and Regularization in Tandem.* Sven Oesau, Florent Lafarge, Pierre Alliez. Computer Graphics Forum - conditionally accepted.
- *Indoor Scene Reconstruction using Feature Sensitive Primitive Extraction and Graph-cut.* Sven Oesau, Florent Lafarge, Pierre Alliez. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 90, April 2014.

- *Indoor Scene Reconstruction using Primitive-driven Space Partitioning and Graph-cut*. Sven Oesau, Florent Lafarge, Pierre Alliez. Eurographics workshop on Urban Data Modeling and Visualisation, 2013.

Shape detection

In this chapter we present two methods to extract geometric shapes from measured point data. Abstraction provides a reduction in complexity by turning a high number of points affected by noise, outliers and missing data into a high-level primitive representation suitable to subsequent steps along the indoor modeling pipeline. First, a multi-scale line extraction method from a 2D point set is performed. Considering several scales offers adaptivity to details. Global clustering via Hough transform aligns similar lines and thus reduces complexity, and provides robustness against noise and outliers.

Second, we introduce a planar shape detection method from 3D measured point data. Exploiting regularities within the input data, i.e. parallelism, co-planarity and orthogonality, common to man-made objects, improves accuracy and robustness to defect-laden data. Planar shapes are extracted via region growing in parallel in many locations in the point cloud. During extraction the shapes are realigned to reinforce regularities detected between shapes. Designed with GPU architecture in mind the method can process millions of points within seconds.

2.1 Feature sensitive line extraction

Efficient processing of data acquired from measurements requires reduction in complexity, typically by abstraction. In this section we expect as input a 2D point set measured from mostly piecewise linear surfaces. The target is expected to be man-made and therefore to exhibit a limited complexity as with, e.g., collinear segments. However, the input data may represent a non-manifold or non-linear boundary. We aim at extracting line segments from the input data without knowledge about the level of detail. The data may be hampered by various defects, such as noise, outliers, sparse sampling and missing data. A typical input data set is depicted in Fig. 2.1. The line detection and extraction is split into two steps: (1) *Local fitting*. A multi-scale line fitting method is used to generate a line hypothesis for

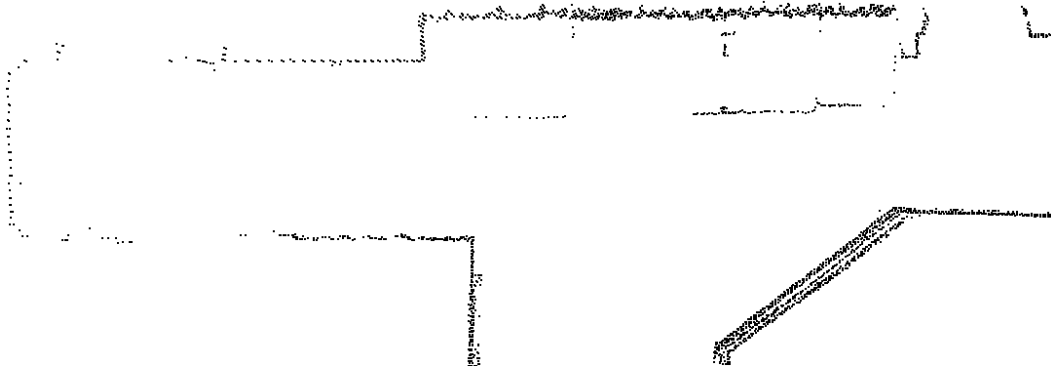


Figure 2.1: **2D input point set.** Excerpt of a typical input dataset featuring different kinds of defects. On the lower right several misaligned scans form a thick band of points leading to the possible detection of ghost primitives. A significant amount of noise with outliers is depicted in the mid top of the image. In the top right, the amount of noise is lower and the boundary exhibits a higher level of detail. A detection of a single long line is preferred for the noisy mid top section, whereas several small line segments are favored to represent the detailed part in the top right. Presence of noise and outliers especially impact sparsely sampled parts of the input data, as depicted on the left.

each point representing the local direction; (2) *Global clustering.* The points are locally clustered into line segments, and these segments are globally clustered into lines through a Hough transform.

Local fitting. A line at a point p_i can be estimated from the input points P via local fitting to a spatial neighborhood:

$$N_{p_i,r} := \{p \in P : \|p - p_i\|_2 < r\}. \quad (2.1)$$

However, there are unknowns: the local level of detail, i.e., the boundary geometry, and the scale of the neighborhood considered for fitting. For longer segments without details, the neighborhood for estimating the line should be as large as possible to provide stable results in presence of noise. However, for points close to corners or crossings the size of the neighborhood must be small and therefore results in imprecise estimation, see Figure 2.2 (left). Furthermore, a corner has a higher level of detail as it can not be described well with a single line.

In order to obtain a line segment adaptive to the local scale and level of detail, several hypotheses are made at multiple scales. For each point the hypotheses are

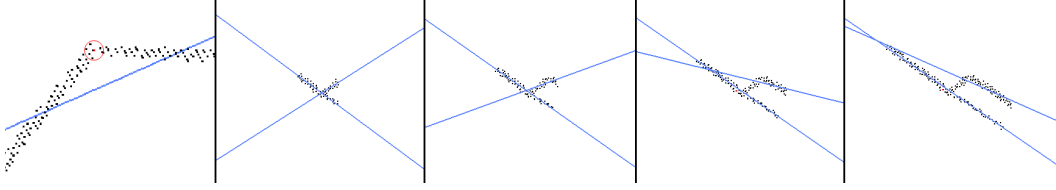


Figure 2.2: **Line hypotheses at different scales.** Left: Least-squares fitting of a line to the neighborhood of a point (red) located in a corner setting. Next four images: Fitting of two lines on four scales to a non-manifold setting. For the two smaller scales, see left and mid-left images, the neighborhood is well covered by the chosen lines. For larger scales however, see mid-right and right images, the geometry is too complex to be covered by two lines.

evaluated to find the one that best describes the local boundary geometry. The idea behind the line fitting is that the expected data from physical measurements can usually be explained by a piecewise linear model. Thus for each of the points there exists a small neighborhood in which the local boundary geometry consists of either one straight line or two line segments in a corner configuration. Even many non-manifold settings can be represented by two lines on a small scale. For each scale of each point two hypotheses are proposed: a single line hypothesis to describe simple straight segments and a two-line hypothesis for detecting corner or crossing configurations.

The two-line hypothesis is created via RANSAC. Two pairs of points are randomly selected from the neighborhood and used to construct two lines l_1 and l_2 .

The quality of the hypothesis is measured in the least-squares sense by:

$$S = \sum_i^n \min_{j \in \{1,2\}} d(p_i, l_j)^2, \quad (2.2)$$

where $d(p, l)$ is the ℓ^2 distance between point p and line l . After a series of samplings, the pair of lines with the lowest sum of squared residuals S is selected as the best pair for the current scale.

As the local level of detail is not known, several line hypotheses are established at multiple scales in order to select a level of detail. The scales, i.e., the ranges of the neighborhoods used for the hypotheses, are selected as multiples of the grid size τ . The minimum number of points in the neighborhood is set to five, as for a smaller number there always exists a perfect alignment of two lines. The parameter τ corresponds to the average spatial distance between points. It can either be extracted

from the data or provided if it is already known from preprocessing. The smallest scale considered relevant is $N_{p_i, 2\tau}$, as even smaller neighborhoods are unlikely to contain five or more points.

To find the proper line hypothesis for a point, the largest suitable scale for each type of hypothesis, single line and two lines, is first selected. This selection is performed by generating each type of hypothesis at increasing scales, starting with $N_{p_i, 2\tau}$. If a hypothesis matches a certain quality criterion, the scale is increased by doubling the range $N_{p_i, r}^j \rightarrow N_{p_i, 2r}^{j+1}$. The largest suitable scale is considered to be the largest scale before the quality criterion fails, see Figure 2.2. The quality of the hypothesis is measured by the maximum Euclidean distance from a point p in $N_{p_i, r}$ to the closest line. A parameter ε is introduced to control the quality criterion:

$$\max_{p \in N_{p_i, r}} \left(\min_{j \in \{1, 2\}} d(p, l_j)^2 \right) < \varepsilon^2, \quad (2.3)$$

where ε denotes the specified tolerance in Euclidean distance between a point and the closest line. A high tolerance value deals with high noise in the input data and vice versa.

Among the two types of hypothesis (single or two lines), the hypothesis with largest scale is selected. When the two scales are equal the single line hypothesis is favored. There is one exception to this. If the two line hypothesis consists of two almost collinear lines, we consider the local boundary configuration to be a single diffuse line, see Fig. 2.1, and discard the two line hypothesis.

Finally, a line l_i , is assigned to every point p_i by choosing the closest line of the selected hypothesis. The line estimation process is illustrated by Figure 2.3 (left).

Global clustering: For linear segments and corners the multi-scale line estimation is satisfactory, but for very small line segments the assigned lines might be heterogeneous, see Figure 2.3 (middle). For robust clustering of points into line segments, we adopt an idea from [HWG⁺13] to sharpen and separate the assigned line directions of the points. Through bilateral filtering, originally introduced for signal processing by [SB97], the points are classified around sharp features into disjoint clusters by their normals. We apply the bilateral filter as an iterative filter updating the line direction of one point with the weighted average of the line directions of the surrounding points. The scale s , selected during the previous step, is used for determining the neighborhood:

$$n'_i = \sum_{p_j \in N_{p_i, s}} n_j \theta(\text{dist}(p_i, l_j)) \psi(n_i, n_j). \quad (2.4)$$

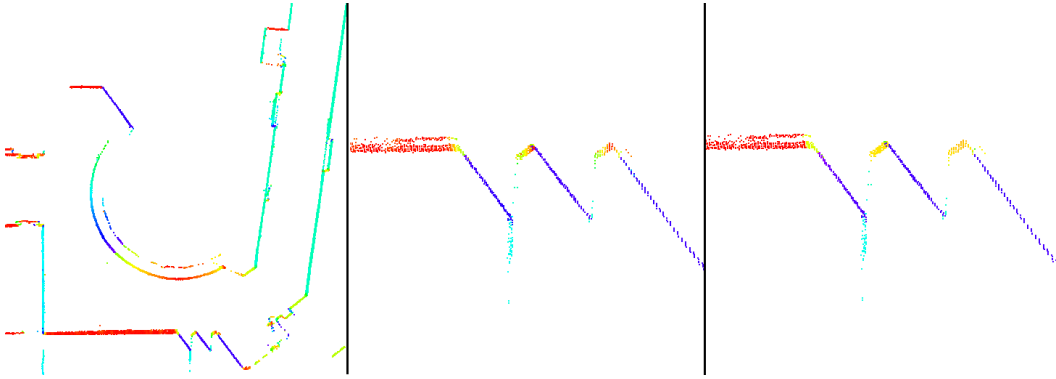


Figure 2.3: **Multi-scale line hypothesis.** Fitted line directions depicted by color in measured data. Left: Excerpt of a scene containing a corridor. The circular wall section is matched accurately. Middle: Some smaller segments on a detailed wall section have an unsharp diffuse fitting. Right: Segments are separated after filtering.

The normals n_i and n_j in the original formulation are replaced by the direction of the assigned line. θ is the spatial weighting function, providing a high weight for points that are close to the chosen line l_i as they are likely to be part of the same line segment. A Gaussian weighting function is chosen, parameterized by the ℓ^2 distance of the current point p_i to the line l_j fitted to neighboring point p_j :

$$\theta(d) = e^{-\frac{d^2}{\sigma_{spatial}^2}} \quad \text{with} \quad \sigma_{spatial} = 2\varepsilon. \quad (2.5)$$

We then define a similarity function ψ favoring points with similar line direction. The similarity function from [HWG⁺13] is adapted by using the line directions instead of the normal vectors:

$$\psi = e^{-\left(\frac{1-n_i^T n_j}{1-\cos(\sigma)}\right)^2} \quad (2.6)$$

with σ set to 15° . We next extract the line segments through region growing. The bilateral filtering and region growing clusters neighboring points into line segments. As many line segments are nearly collinear, we use a Hough transform for global clustering of the segments into lines, see Figure 2.4.

The Hough transform [Dav05] is used for robust 2D line extraction through accumulation and extraction of local maxima in a discretized parameter space, denoted as Hough Accumulator. In our framework the angle of each line with respect to the x axis and the distance between the line and the center of the bounding rectangle

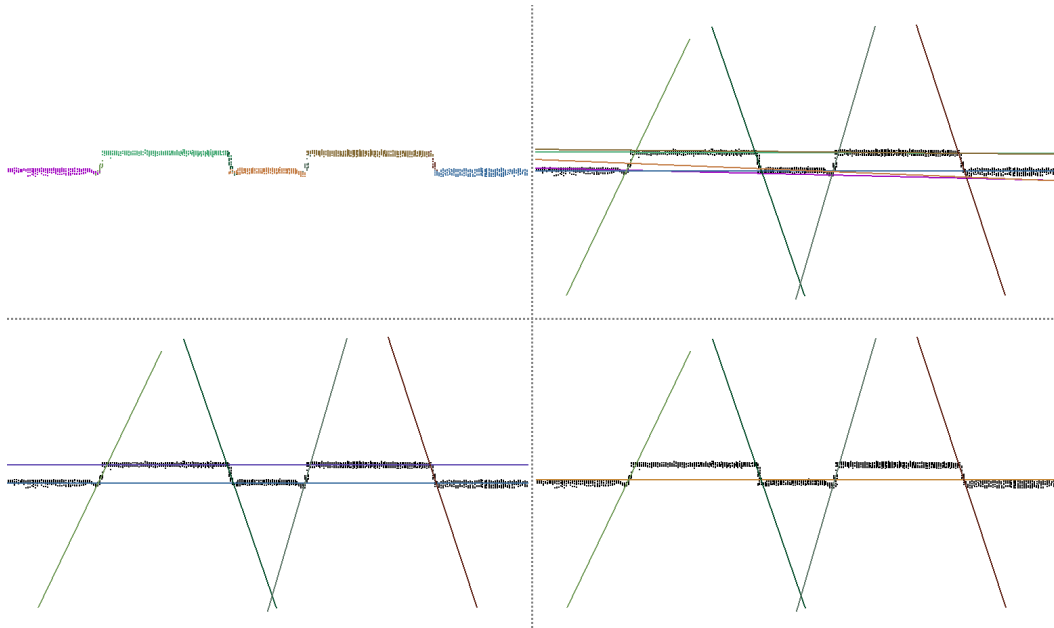


Figure 2.4: **Global clustering of line directions.** Upper left: Clustered line segments. Upper right: Lines extracted without global clustering. Lower left: Global clustering using Hough transform leads to reduction of the number of lines. Lower right: Over-simplification induced by coarse resolution for Hough Accumulator and a high value for ϵ .

of the scene are chosen as parameters. Each cluster is added to the Hough Accumulator with a fitted least-squares line and a number of votes corresponding to the number of points in the cluster.

The main lines are so extracted one by one, starting with the global maximum in the Hough Accumulator. For each extracted maximum a line segment is fitted to cover the span of all points contained in the corresponding cell of the Hough Accumulator. All other clusters that match the line are removed from the Hough Accumulator and the next peak is detected. A cluster is considered matching a line when its points have a maximum distance ϵ to the line. This procedure is repeated until all clusters have been extracted from the Hough Accumulator.

2.2 Planar shape detection and regularization in tandem

Extraction of shapes from measured point data is the first step for many applications. Further processing usually operates on the set of shapes instead of the point data. Therefore, inaccuracies in the extracted shapes have direct impact on the quality of the final outcome. Some methods [OXAH10, MMV⁺14, OLA14] perform a regularization of shapes afterwards without considering the original input data. While this effectively reduces the complexity of the set of shapes, the accuracy with respect to the input data may be impaired. In reconstruction, regularity is often used to compensate for missing data or to reduce complexity of the solution space. However, only very few approaches consider regularity during primitive extraction. We incorporate regularity in the input data into the detection process. This not only yields high regularity while maintaining data fidelity, but also improves robustness against noise and missing data. Our method is feasible for GPU implementation to cope with the increasing amounts of data generated by acquisition technologies.

2.2.1 Overview.

Our algorithm takes as input a raw point set and proceeds as follows. A set of seeds are distributed uniformly over the input points via the cells of a hierarchical space decomposition - an octree. From these seeds we start detecting primitive shapes through region growing (Section 2.2.2). During growing we repeatedly interrupt the shape detection process in order to detect non-local relationships between the shapes that have been detected so far (Section 2.2.3). The shapes are regularized according to these relationships (Section 2.2.3), and we iterate until complete detection, i.e., until no more points can be assigned. We provide below a pseudo-code of the algorithm 1, and Figure 2.5 depicts the overall process.

Our motivation for such methodology stems from the following observation: Knowledge about dominant directions and non-local relationships between a preliminary set of shapes detected from the input data can aid further detection by guidance. The key is thus to derive such relationships from the input data early during the detection process, which is possible only if a sufficient number of shapes are already detected. To provide many shape hypotheses early in the detection process and to achieve short running times, we detect a high number of indepen-

Algorithm 1 Interleaved detection and regularization

```

generate octree
compute kNN

c ← leafCells
repeat
  for all ci do
    if NOT hasActiveShape (ci) then
      findSeed (ci)
      growShapes (ci)

  g ← detectRelations
  for all groups of shapes gi do
    regularization (gi)
    adjustCoplanarity (gi)
until no new points assigned

```

dent shapes in parallel. As parallel methods are efficient only when the amount of synchronized access is minimized, we favor a region growing approach that operates locally by expanding the borders of a connected region. In addition, region growing is an incremental process, providing knowledge about a primitive shape before it has been entirely detected.

Note that after each region growing performed in parallel, each shape is refitted to its associated points to improve data fidelity. The regularities are detected and turned into a graph of relationships between shapes which represent a set of regularity hypotheses. To reinforce the detected relationships between shapes we simulate non-local fitting from these hypotheses, and verify fidelity of the regularized shapes with respect to the associated points. Albeit outliers must be accepted when seeking for outlier robustness, the shape is not regularized and we rollback to the former detected shape when a too large fraction of the points does not fit well. Such rollback provides us with resilience to bad decisions taken in the early steps of the detection process, and to imperfect choices of seed points. Detaching points that are no longer faithful to the shape provides guaranteed data fidelity, i.e., maximum Euclidean distance between input points and extracted shapes.

Compared to region growing a Hough transform might look more appealing for regu-

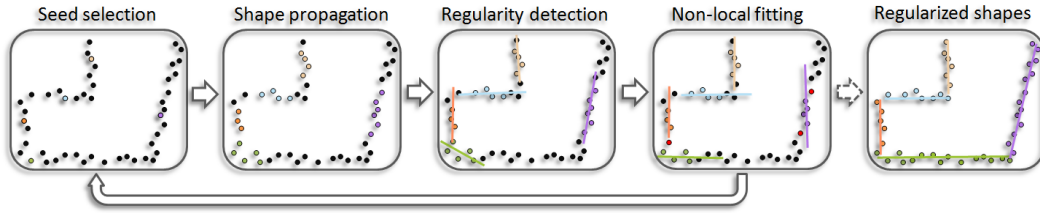


Figure 2.5: **Interleaved detection and regularization.** Our method operates a concurrent region growing process for detecting shapes, depicted by different colored points in the first step. The region growing process is interleaved with a regularization of the shapes. Relationships between shapes are detected from locally fitted planes and reinforced by non-local fitting of the shapes during detection. Fidelity to the input points is verified by checking the regularized shape against the associated points. A small number of outliers is acceptable to allow for resilience to outliers and noise (see green shape). A major deviation of a regularized shape from the input points triggers a fallback to local fitting for further propagation (see purple shape).

larization, as they map the input data onto the parameter space where regularization is easier through, e.g., quantification. In the former section a Hough transform was used to cluster line segments into lines. However, the computational complexity and memory consumption depend on the number of degrees of freedom of the primitives sought after. A line in 2D space can be represented by 2 parameters, e.g. angle and distance to the origin. Searching for planes in 3D requires at least 3 dimensions in accumulator parameter space (2 for orientation, 1 for distance to origin), leading to high memory consumptions and running times. Clustering a density function with, e.g., points in accumulator space inherently yields planes that are coplanar. However, we observe that such clustering algorithm is sensitive to the choice for the origin, as moving the origin changes the neighborhood in parameter space. Finally, and albeit the accumulation in the parameter space might be done in parallel, the extraction remains sequential, thus accumulation space methods are less suitable for our approach than a collaborative region growing and regularization method.

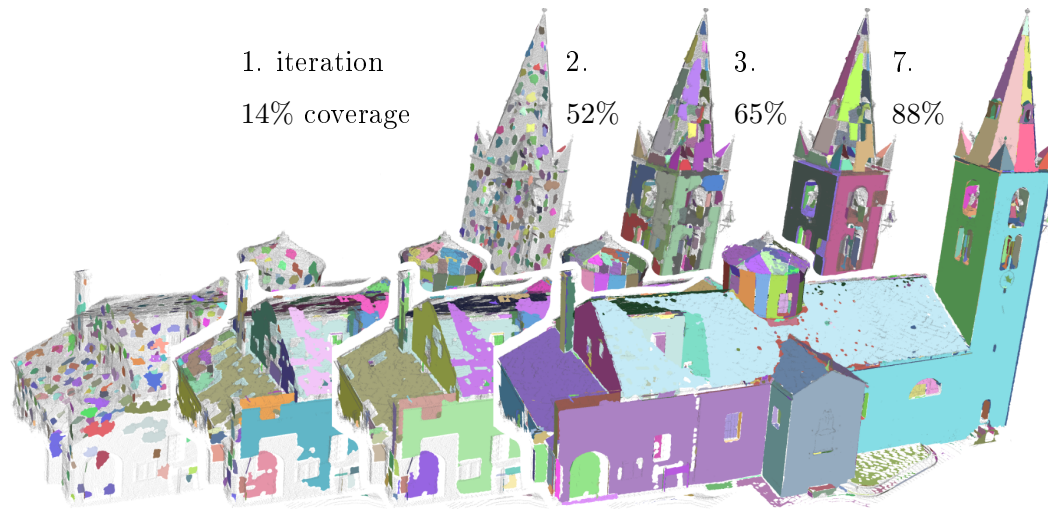


Figure 2.6: **Iterations.** After iteration 1, more than 1.000 shapes are detected in parallel. After iteration 2, 52% of the input points are assigned to 250 larger shapes. After iteration 3, 65% of the input points are assigned to 140 even larger shapes. Upon termination (iteration 7), 86 shapes cover 88% of the input points.

2.2.2 Region growing

Primitive shapes are detected through region growing. A shape is represented as a set of points and associated fitting plane. Growing is achieved either by adding neighbor points to shapes in parallel, or by hierarchical pairwise merging of shapes when they are detected as being both adjacent during growing, and coplanar during the regularity detection step. However, to avoid the need for synchronized access, we restrict the growing of each shape to its cell. As we deal with unstructured point clouds and growing in parallel on GPU, several key ingredients need to be defined: a local neighborhood, an error metric to decide propagation and the criteria to best select seed points.

Local neighborhood. Images naturally provide neighborhood information due to the arrangement of pixels. This allows for an efficient access during the growing process. For unstructured point clouds however this information is not available. One solution is the range search to determine the neighbors, e.g., a spherical neighborhood. However for point clouds acquired by laser scanners this is impractical due to highly variable density. We rely instead upon a K-Nearest Neighbor (kNN) graph data structure to determine point-neighborhood during growing, as it better adapts to variable point density.

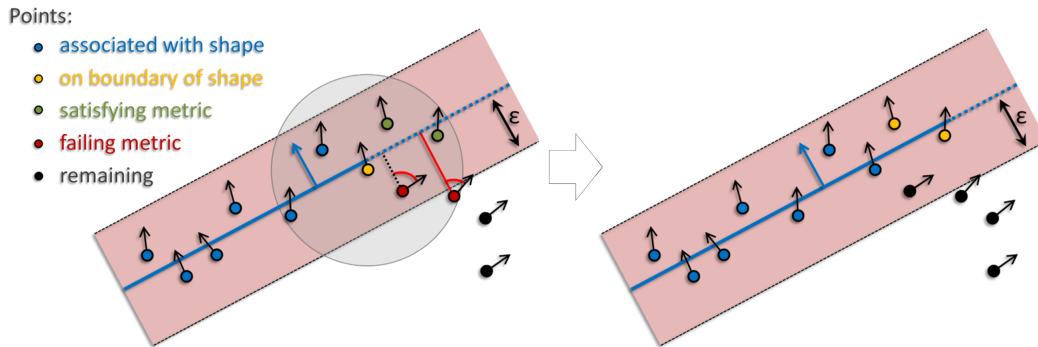


Figure 2.7: **Growing error metric.** We grow a shape by adding neighbor points, indicated by a gray circle, to the boundary points, depicted as yellow. Each neighbor point within the ε -domain around the shape and with a small deviation of the normal to the shape normal is associated to the shape. Two points in the neighborhood, marked as green, match this condition. Two other points, depicted as red, do not match the condition by a major deviation of the normal or a position outside of the ε -domain. Growing is carried out for the neighborhood of the two newly associated points, depicted in yellow in the right picture.

Growing error metric. The error metric used to decide whether a neighbor point well fits a shape for growing involves two error tolerance parameters: ε defines the maximum Euclidean distance between a point and the plane of a shape, and α defines the maximum angle deviation between the normal of a point and the normal of the plane of a shape. The shape propagation is illustrated by Figure 2.7.

Seed point selection. The choice of seed points to initialize parallel region growing has some impact on the quality of results and running times. We define two criteria: planarity of neighborhood and minimal distance to the cell center. For planarity we favor seeding points with a high number of unassigned neighbors (out of the kNN) that well fit a plane, according to the error metric. Neighbors already assigned to another shape are not considered for fitting. Such planarity criterion indicates the presence of a planar structure, favorable to growing, but our experiments showed that seeding closer to cell centers avoids considering already assigned neighbor points and hence supports faster growing. However, we give a strict priority to the planarity criterion and use the distance to the cell center as a second priority, as the first does not lead to a unique choice in general. Seed points are repeatedly selected during the detection process, when no shape can grow further in this cell

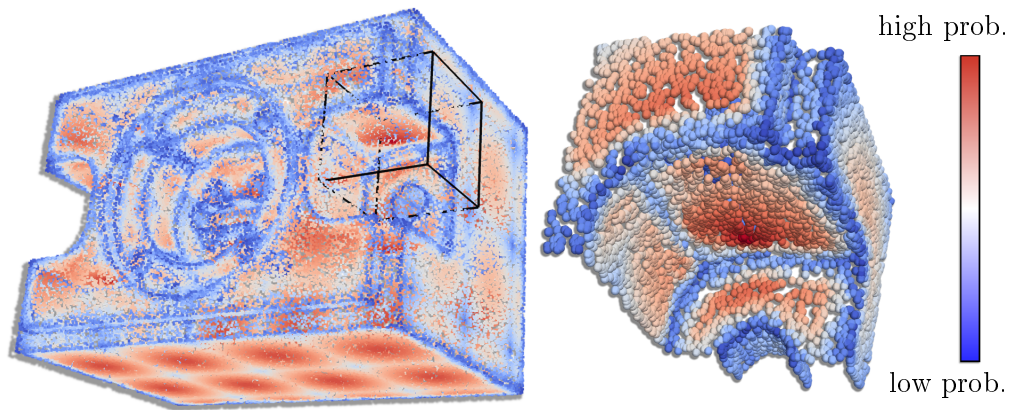


Figure 2.8: **Seed point selection.** The point set of a mechanical part is colored by the probability of a point to be chosen as a seed point for the region growing process. Points close to sharp features are assigned a low probability as their neighborhood is non-planar. On flat parts of the surface points closer to the cell center of the space decomposition are favored as they allow for a faster expansion not being bounded by the borders of the cell.

and when there are unassigned points left. Region growing is restricted to operate within one cell of the space partitioning. With one exception however: When the growing shape hits the boundary of the cell, neighbor points from other cells fitting well the shape are marked as best candidate seed points. Figure 2.8 depicts a point set with probability function to be selected as seed points.

Hierarchical pairwise merging. The space partitioning is created during pre-processing without knowledge about the structure of the input point cloud. A planar part may have been split into several cells and hence detected in parallel in different cells. Two shapes belonging to the same planar surface are detected as being coplanar during the regularity detection step. They are merged into one shape if they are also detected as being adjacent during growing.

Finalization. A shape is considered to be finalized, i.e., not active, if no more neighbor points can be assigned.

2.2.3 Regularization

We consider three types of relationships between shapes: *parallel*, *orthogonal* and *coplanar*. Given a configuration of shapes with associated points the regularity

detection step constructs a conflict-free graph of relationships. The nodes of this graph represent groups of parallel shapes that are connected to groups of orthogonal shapes. This graph is later used during the regularization step. Parallel shapes are detected via mean shift clustering in the Gauss normal map, while orthogonal relationships between clusters of parallel shapes are greedily added in a second step by comparing their mean directions. Coplanarity relationships are detected later and are not represented through the graph. For each regularity detection step the relationships are re-learned from all shapes and the graph is rebuilt entirely.

Parallelism. We first generate clusters of parallel shapes. A parameter β is used to specify the tolerance angle deviation between two shape normals. A simple pairwise comparison is not sufficient as a constellation of three shapes a, b and c may already conflict if $|n_a \cdot n_b| \geq \cos \beta$ and $|n_b \cdot n_c| \geq \cos \beta$, but $|n_a \cdot n_c| < \cos \beta$. Instead we perform a Gauss-map clustering. Each shape is projected onto the unit sphere by its normal and assigned a weight equal to the number of associated points. As we consider unoriented normals we also consider the mirrored point on the sphere. The peaks are extracted via mean-shift, restricted to one hemisphere, using a Gaussian kernel with $\sigma = \beta$. All shapes within one peak are considered to be parallel facing in the direction of the peak.

Orthogonality. In a next step we add orthogonal relationships to the graph, by pairwise comparison of the directions of the parallel clusters. Initially we put all clusters into a pool, then perform a greedy selection, starting by removing the cluster with highest number of points, as this cluster is expected to provide the highest confidence. Two clusters c_1 and c_2 are considered to be orthogonal if $|n_{c_1} \cdot n_{c_2}| < \cos(\frac{\pi}{2} - \beta)$. All clusters orthogonal to the current one, are connected to the current one by an *orthogonal* edge and removed from the pool. We repeat this greedy selection by choosing the next cluster with the highest number of points remaining in the pool. Upon termination the graph consisting of several disconnected subgraphs. Given an indoor scan of a Manhattan world environment for instance, all shapes of the structures are ideally contained in one subgraph. The center-node of the subgraph represents the largest number of points associated to parallel shapes, in this case probably the direction of floor and ceiling. Two nodes are connected to the center-node, each containing the shapes of one wall direction.

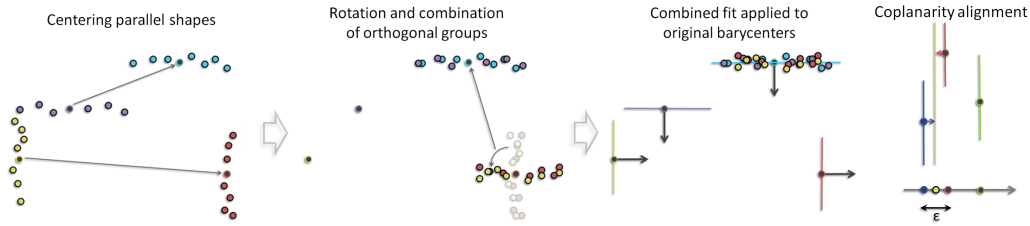


Figure 2.9: **Constrained non-local refitting.** The orientation of regularized shapes is defined in one fitting step to guarantee exact parallelism and orthogonality between regular shapes. The points of parallel shapes are translated by moving the center of mass into a single location (left). Points of mutually orthogonal shape clusters are combined by rotating the points by $\frac{\pi}{2}$ around the cross product of the mean direction of each group (middle). The normal of the least-squares fitting plane is the best fit orientation for the parallel shapes and the inverse rotated normal for the orthogonal groups respectively (right). The coplanarity regularization is done in a subsequent step within groups of parallel shapes. The shapes are projected onto a line along their normal, illustrated by the gray arrow (right). Shapes in one ε -cluster are considered to be coplanar, depicted by the blue and red point, and are shifted to the mean of the cluster, depicted by the yellow point.

Coplanarity. Coplanarity relationships are detected only after reorienting the shapes in the regularization step (Section 2.2.3), as shapes detected to be approximately parallel have already been regularized to be exactly parallel. For each group of parallel shapes we perform a clustering based on the distance between the planes. More specifically, the (clustered) shapes are projected onto points on the line defined by their shared normal through the origin. Each shape is weighted by the number of points associated to the shape. We then cluster the shapes through mean-shift applied to these projections, through a Gaussian kernel parameterized with $\sigma = \varepsilon$.

Constrained fitting Provided a set of shapes with associated points and a graph of relationships, we regularize the shapes by performing constrained plane re-fitting where constraints are set in accordance to the graph. Plane fitting is performed through principal component analysis (PCA), extended to fit multiples planes with a fixed relative orientation.

The best least squares fitting plane to a set of points of a shape S_i passes through the center of mass μ_i . Its normal orientation is aligned with the vector with minimal variance of the points. PCA provides an orthogonal basis aligned to the principal variation of the data by extracting the eigenvectors of the covariance

matrix cov_i . The elements of cov_i are covariances $\sigma(x, y)$ of pairwise coordinates and the variances $\sigma^2(x)$ on the diagonal.

$$\mu_i = \frac{1}{|S_i|} \sum_{p \in |S_i|} p \quad (2.7)$$

$$\sigma_i(x, y) = \sum_{p \in |S_i|} (p_x - \mu_{i,x})(p_y - \mu_{i,y}) \quad (2.8)$$

$$cov_i = \begin{pmatrix} \sigma_i^2(x) & \sigma_i(x, y) & \sigma_i(x, z) \\ \sigma_i(y, x) & \sigma_i^2(y) & \sigma_i(y, z) \\ \sigma_i(z, x) & \sigma_i(z, y) & \sigma_i^2(z) \end{pmatrix} \quad (2.9)$$

The eigenvectors of cov_i denote the orthogonal directions of variation and the corresponding eigenvalues quantify the amount of variation. Only the eigenvector \vec{e}_3 of the smallest eigenvalue needs to be determined as this corresponds to the orientation of the plane normal. As we perform iterative re-fitting with reliable initial guesses and perform all computations in parallel on GPU we find it more efficient to compute the smallest eigenvalue through the power iteration method applied to the inverted covariance matrix.

The basic idea of the power iteration is that each vector can be written as a linear combination of eigenvectors. By multiplying the covariance matrix cov_i with a vector v each term gets scaled by its eigenvalue:

$$v = ae_1 + be_2 + ce_3, \quad (2.10)$$

$$cov_i \cdot v = a\lambda_1 e_1 + b\lambda_2 e_2 + c\lambda_3 e_3. \quad (2.11)$$

Therefore, the portion of the largest eigenvalue is the most amplified. By iterative multiplication with the covariance matrix and normalization, the vector converges to the eigenvector of the largest eigenvalue. The convergence is fast when the starting value is already well aligned. It also depends on the ratio of the largest to the second largest eigenvalue: $\frac{\lambda_2}{\lambda_1}$. In our case we are looking for the smallest eigenvalue. In case of planar shapes the first two eigenvalues are large compared to the third one. Therefore we perform the power iteration on the inverse covariance matrix:

$$v_{n+1} = \frac{cov_i^{-1} \cdot v_n}{\|cov_i^{-1} \cdot v_n\|}. \quad (2.12)$$

We extend the common plane least squares fitting to the fitting of clusters of planes with a fixed relative orientation (either parallel or orthogonal) by combining the

covariances matrices into one a single matrix. The covariance matrix measures the variance relative to the centered data and is therefore translation invariant. For a cluster of parallel shapes the covariance matrices cov_i of each shape's point set are thus simply added. For clusters of parallel shapes that are mutually orthogonal we choose as "master" shape the one with largest number of points and all clusters of orthogonal shapes connected in the graph are rotated around their center of mass to match the master shape. Rotation R_i is specified by using the cross-product between the normals of S_i and of the master shape as axis and $\frac{\pi}{2}$ as rotation angle. We weight the influence of each set of points by multiplying their covariance matrix by a weight set to the number of points:

$$cov = \sum_i (|S_i| \cdot cov_i). \quad (2.13)$$

The regularized planes of the shapes are then given by the backward transformed direction and the individual barycenter μ_i or the mean barycenter for coplanar shapes, see Fig. 2.9:

$$(R_i^\top \vec{x})p - (R_i^\top \vec{x})\mu_i = 0. \quad (2.14)$$

2.2.4 Implementation in CUDA

Our algorithm is implemented in C++ and in CUDA using compute capability 1.1. We use the CGAL library for fast normal estimation when not provided with the input points. On GPU the processing is structured into blocks of threads. The GPU is triggered by a single function call, executed on a number of blocks of threads specified at call-time. Threads of one block are processed simultaneously following a SIMD principle (single instruction multiple data). Blocks, however, may be handled in any order chosen by the driver and many blocks are processed in parallel. To achieve satisfactory performance, synchronizing operations and branching must be minimized and the work should be distributed over the threads as evenly as possible. Memory accesses of threads within one block are aligned linearly to optimize the cache usage.

We choose an octree for space partitioning as it provides a decomposition into compact cells and can be implemented very efficiently on the GPU. The input points are reordered in memory to be linear and continuous within each octree cell. This allows for efficient memory access on the GPU.

The typical way to construct an octree data structure on CPU is top-down. Starting

from the smallest cube containing the bounding box as the root node, each node is recursively split into its contained octants. The recursion is typically bound by a minimum number of contained points, a minimum size of a node or simply a maximal depth. However, this does not suit the GPU architecture well which requires an even work distribution among all GPU cores. One ideally would have to write three different methods for subdividing nodes. (1) A first method splitting a single or few nodes cooperatively with many blocks of threads, used in the beginning splitting few but big nodes. (2) A second method for handling one node per block. (3) A third method for splitting many nodes at once per block.

We follow a different approach proposed by Karras *et al.* [Kar12]. They first assign a label to each point based on the Morton curve. The Morton curve is a space filling curve, that travels a compact grid by visiting subsequently adjacent cells see Fig. 2.10. Ordering data along the Morton curve has been identified by Pascucci *et al.* [PF01] to provide a cache-oblivious structuring. Karras uses the 3D variant of the Morton curve to assign to each point the position on the curve. This can be done efficiently in parallel as each point is processed independently. Afterwards the points are sorted efficiently by radix-sort using the CUDA implementation by [MG11].

The neighborhood information of an input sample is accessed very frequently during region growing. We use precomputed approximate K Nearest Neighbors (kNN) to allow for efficient access on the GPU during region growing. Due to the fixed number of neighbors, the work per point on the border during region growing is evenly distributed. To compute the kNN we rely on the octree data structure created to distribute the work with a fine partitioning. In a first step we compute all k nearest neighbors within each octree cell. The search is extended to the adjacent cells only for the points that are closer to the octree boundary than their $\frac{k}{2}$ th neighbor.

The region growing step on GPU is designed to minimize the number of synchronized operations. Common implementations of region growing methods on unstructured data such as point sets keep track of the region boundary through a list of points. Managing a list of points in a highly parallelized setting, however, involves synchronizing. We store instead one flag per point to manage its state: *not visited*, *to be visited*, *associated* and *rejected*. In each region growing step, a range of points inside the octree cell is processed. The range is split equally between all threads, by processing every i th index by the i th thread to allow for optimal cache usage. Only points marked as *to be visited* are processed and the state changes respectively. If

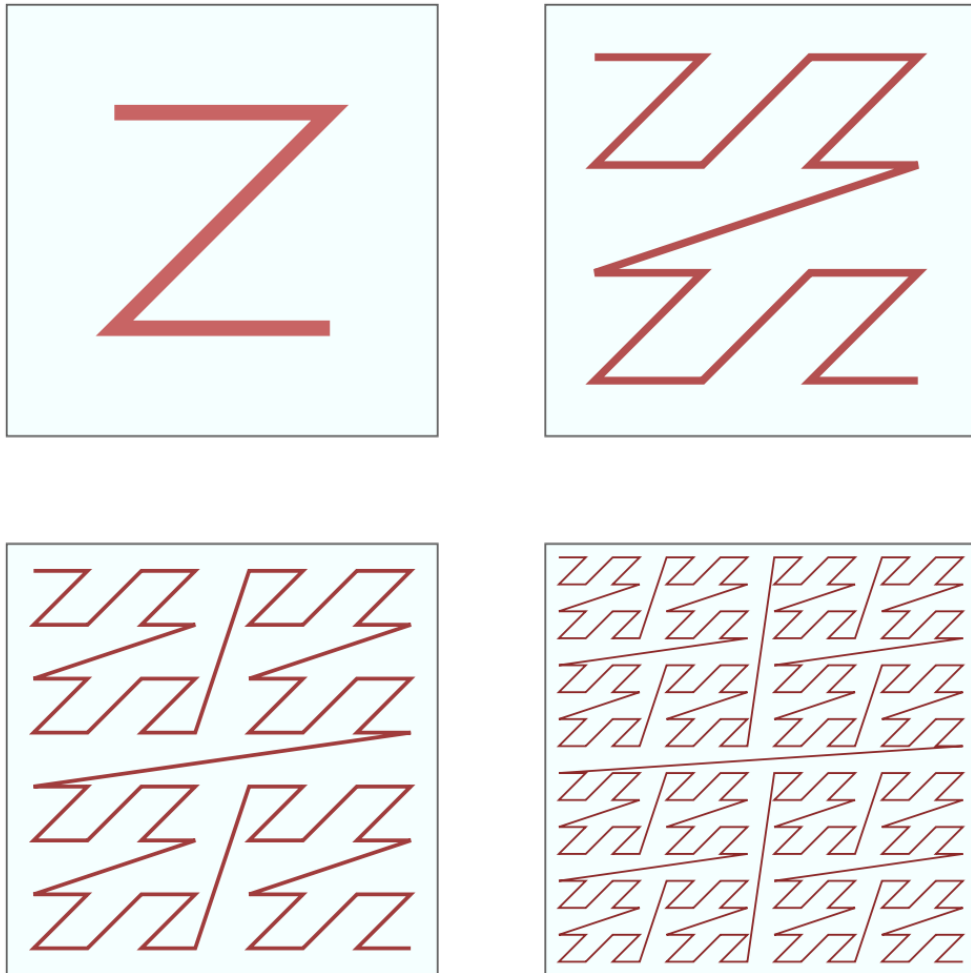


Figure 2.10: **2D Morton curve.** The image depicts the Morton or Z-order curve in 2D on four different levels. The curve follows a depth-first search in a quad-tree. Using the 3D variant of this curve yields a depth-first search in an octree. Image courtesy by *David Eppstein*.

a point satisfies the growing error metric (Section 2.2.2), of the shape all neighbors with state *not visited* are set to *to be visited*. Each point is handled only once by a single thread within one region growing step, such that no synchronization is required for setting a flag.

Each thread keeps track of the minimal and maximal index of the point that has been flagged as *to be visited*. The minimal and maximal index of all threads are used as the range for the next growing step. The range in the initial step contains only the seed point in order to restrict the processing to the minimal necessary range while providing a memory-aligned access pattern and minimal need for synchronization. We perform a mean shift clustering of normals on the unit sphere (Gauss map) to detect parallelism between shapes. As the input normals are unoriented the clustering is performed on a half-sphere. More specifically, each bin of a discretized Gauss map records a list of all contained shapes to allow for quick access. We chose β as discretization angle: the user-specified parameter for tolerance angle deviation. Mean shift clustering is then performed independently within each thread, starting from the initial location of the thread, homogeneously distributed across the Gauss map. If no shapes are found within the kernel size β , the initial location is instead relocated to the closest non-empty bin. Overlapping peaks are resolved by retaining the clusters with the largest number of points assigned to their associated shapes. The detection of orthogonal relationships between the clustered groups of parallel shapes is then performed sequentially starting from the cluster with the largest number of points, and by comparing it with all other clusters.

2.2.5 Experiments

Benchmark. Surface reconstruction and shape detection have common topics of research, hence a set of common criteria for evaluation have been established [BLN⁺13]. Depending on the type of input data and defects such as missing data or outliers, some criteria such as the Hausdorff distance may not be relevant. To measure geometric fidelity we choose the mean distance of a detected shape to its associated points. The coverage, i.e., percentage of points assigned to primitive shapes, is used as indicator for completeness of the detection. The running times listed in Table 2.1 include all preprocessing steps such as octree generation and kNN. For evaluation we used a MacBook Pro laptop with Core i7 4850HQ and a GeForce GT 750M graphics card.

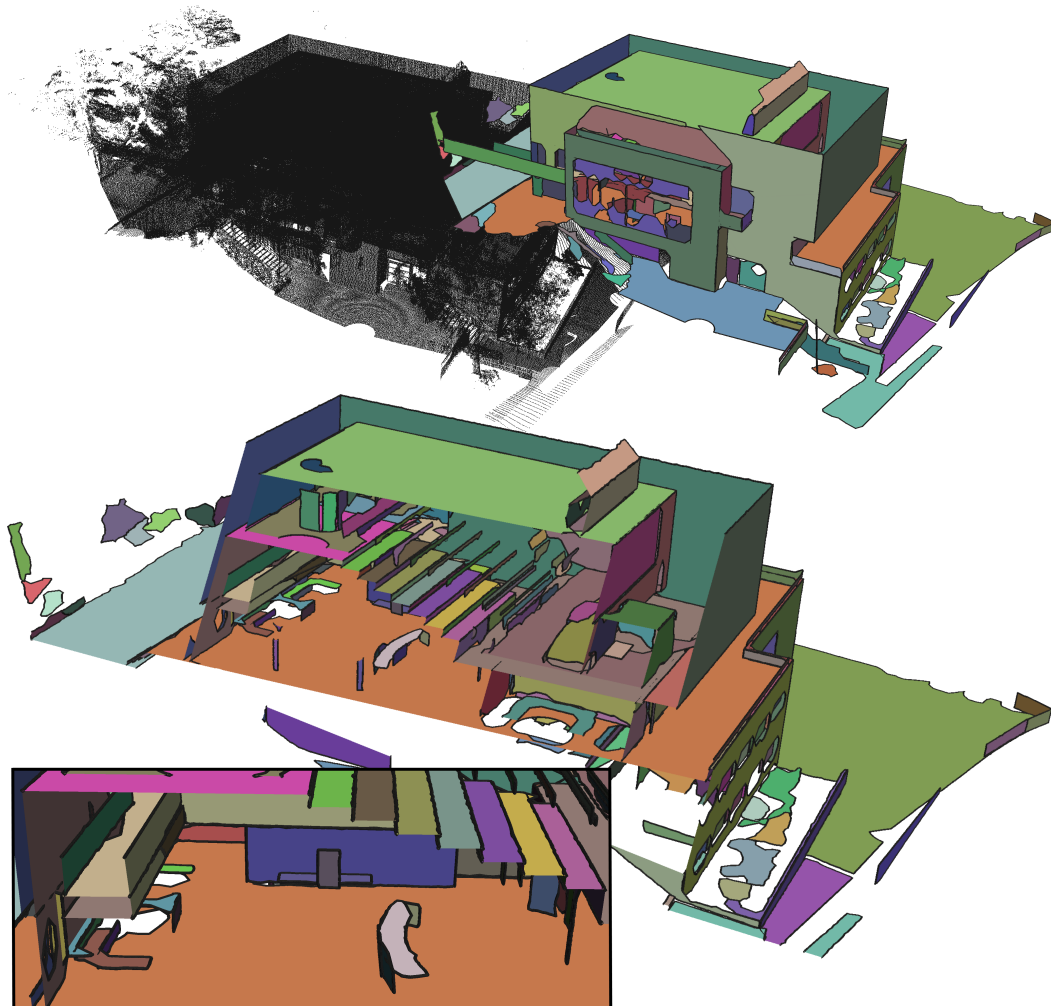


Figure 2.11: **Kahn building**. The input point set (5.2M points) has been acquired via a LIDAR scanner, from the inside and outside of a physical building. 200 shapes have been detected, aligned with 12 different directions in 179 different planes. The cross section depicts the auditorium in the upper floor and the entrance hall in the lower floor. The closeup highlights the steps of the auditorium which are made up of perfectly parallel and orthogonal planes.

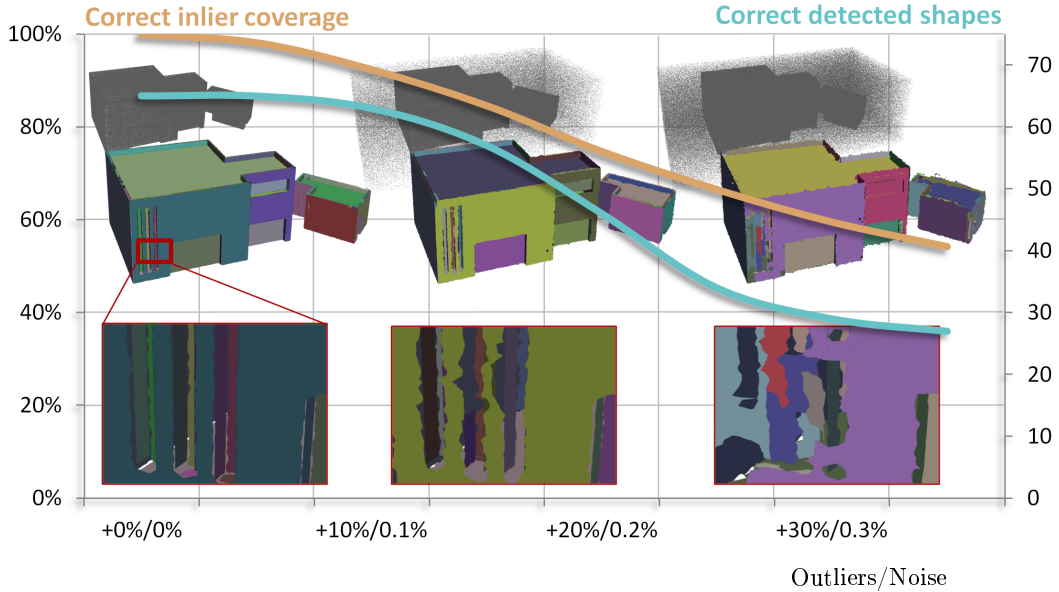


Figure 2.12: **Robustness.** We sample the model uniformly and evaluate our method while increasingly adding outliers and Gaussian noise. With an increasing amount of noise and outliers small shapes are more difficult to detect. The closeup depicts a section with narrow windows that is affected by the increase of noise. However, the coarse structures are still detected for a strong amount of defects, depicted by the full set of detected shapes in the upper row.

Evaluating the regularity of a set of primitive shapes is still quite unexplored. We measure the complexity of a set of planar shapes by the degrees of freedom of the corresponding planes. High complexity refers to low regularity and vice-versa. A plane has three degrees of freedom: two for the orientation and one for the signed distance to the origin. for a group of parallel shapes we count only two degrees of freedom for orientation. For two or more pairwise orthogonal set of shapes we consider three degrees of freedom in total for the orientation. Coplanar shapes count for one.

We compare our method against three other methods to evaluate the shape detection performance and the regularity: a region growing method (*RegGrow*) for detecting planar shapes in unstructured point data [LM], the efficient RANSAC-based shape detection method (*RANSAC*) proposed by Schnabel *et al.* [SWK07], and *GlobFit* [LWC⁺11], an iterative regularization method relying upon the aforementioned *RANSAC* method. For rigorous comparison we set the parameters of these methods to provide results with similar mean errors and minimum number of points per shape. While *RANSAC* and *GlobFit* can handle other types of shapes

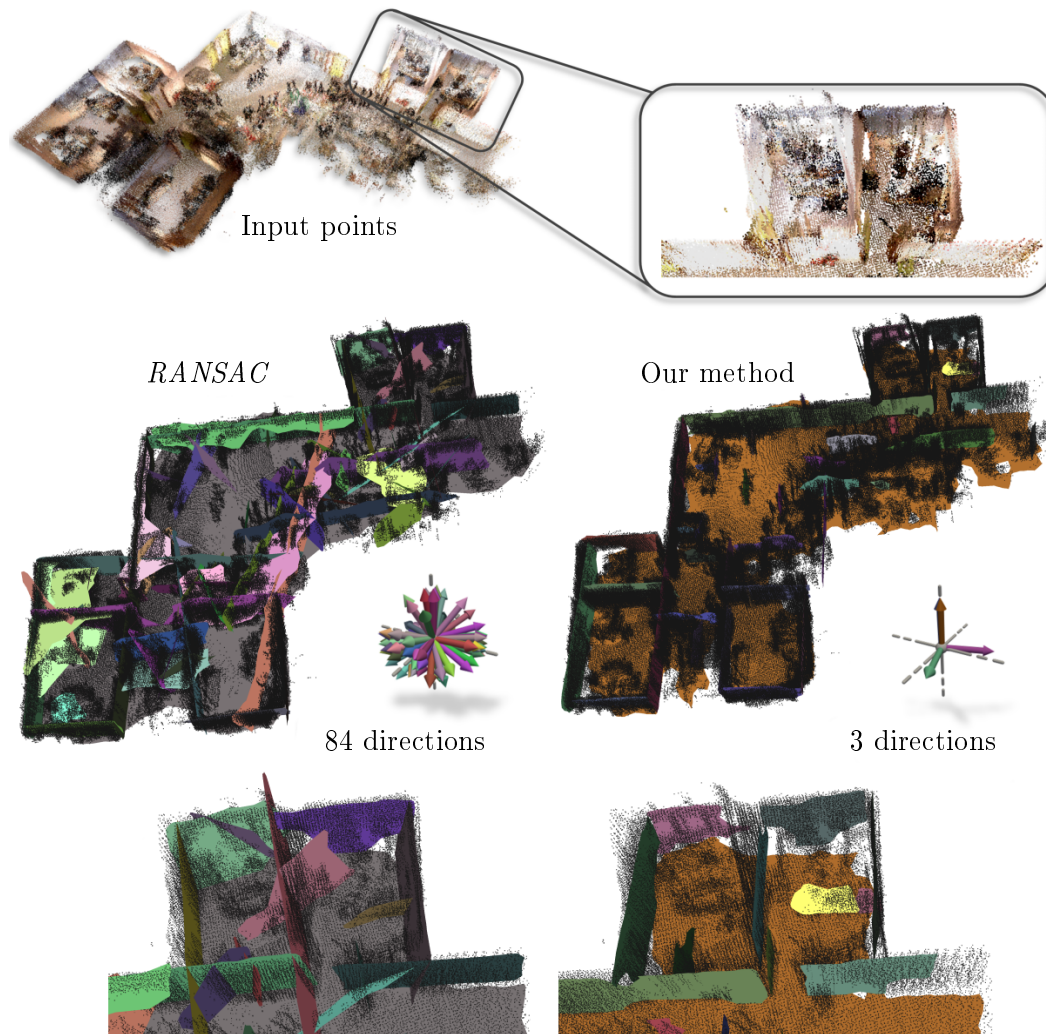


Figure 2.13: **Kinect**. This Manhattan world scene has been acquired by a mobile robot using Kinect sensors and registered into a single point set. Schnabel’s method detects many shapes but they are not very well aligned due to the high amount of noise and clutter and imprecise registration. Our methods identifies the main directions of the Manhattan world and aligns the shapes with those directions during detection. This allows for compensation of the imprecise registration, see lower image row for comparison with Schnabel’s method.

we restrict the methods to planar shapes for comparison.

The datasets used for evaluation range from architectural scenes acquired via laser scanners and Kinect sensors to a point set acquired by Multi-View Stereo (MVS) using the approach from Vu et. al [VKLP09]. The Kinect point set covers an indoor Manhattan world scene [GMRFM14]. On defect-laden inputs our experiments show that the interleaved regularization improves the shape detection step through detected relationships, see Figure 2.13.

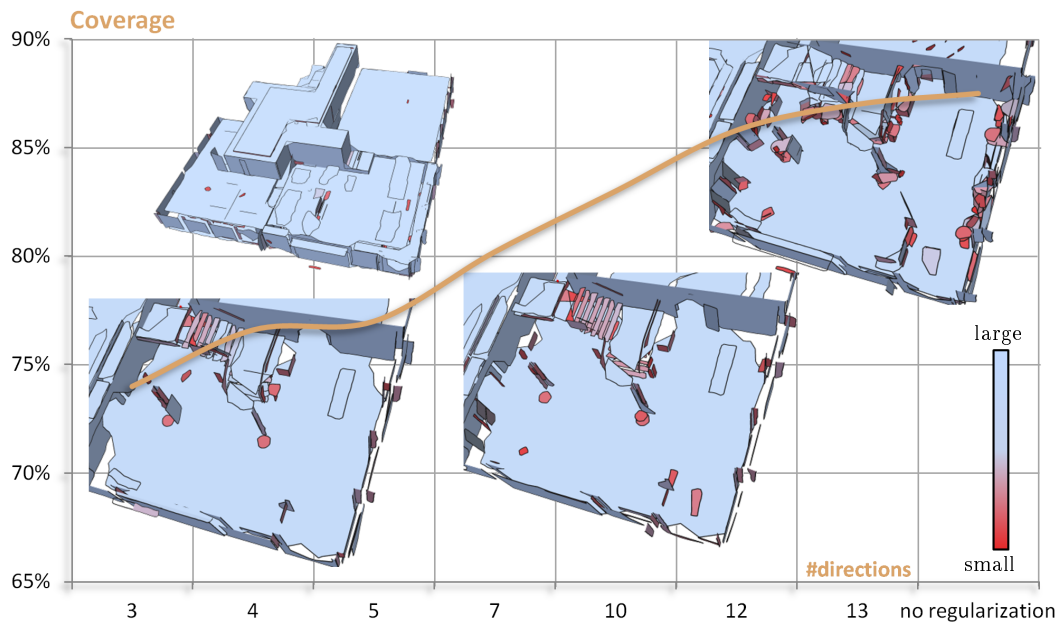


Figure 2.14: **Regularity vs. coverage.** Favoring strong regularity may come at the cost of coverage. We applied our method to the LIDAR dataset 'Euler' with different parameters. Colored scale refers to the shape area. The full reconstruction with low regularity is shown in the upper left. A high amount of clutter in the entrance hall can be seen on the upper right closeup with a low regularity, i.e. 388 different directions. The two closeups in the lower figure show the variation in detection of irregular shapes, clutter, while favoring a higher regularity, i.e., 5 and 10 different directions. Many small shapes on clutter are detected while the shapes on structural parts remain mostly untouched.

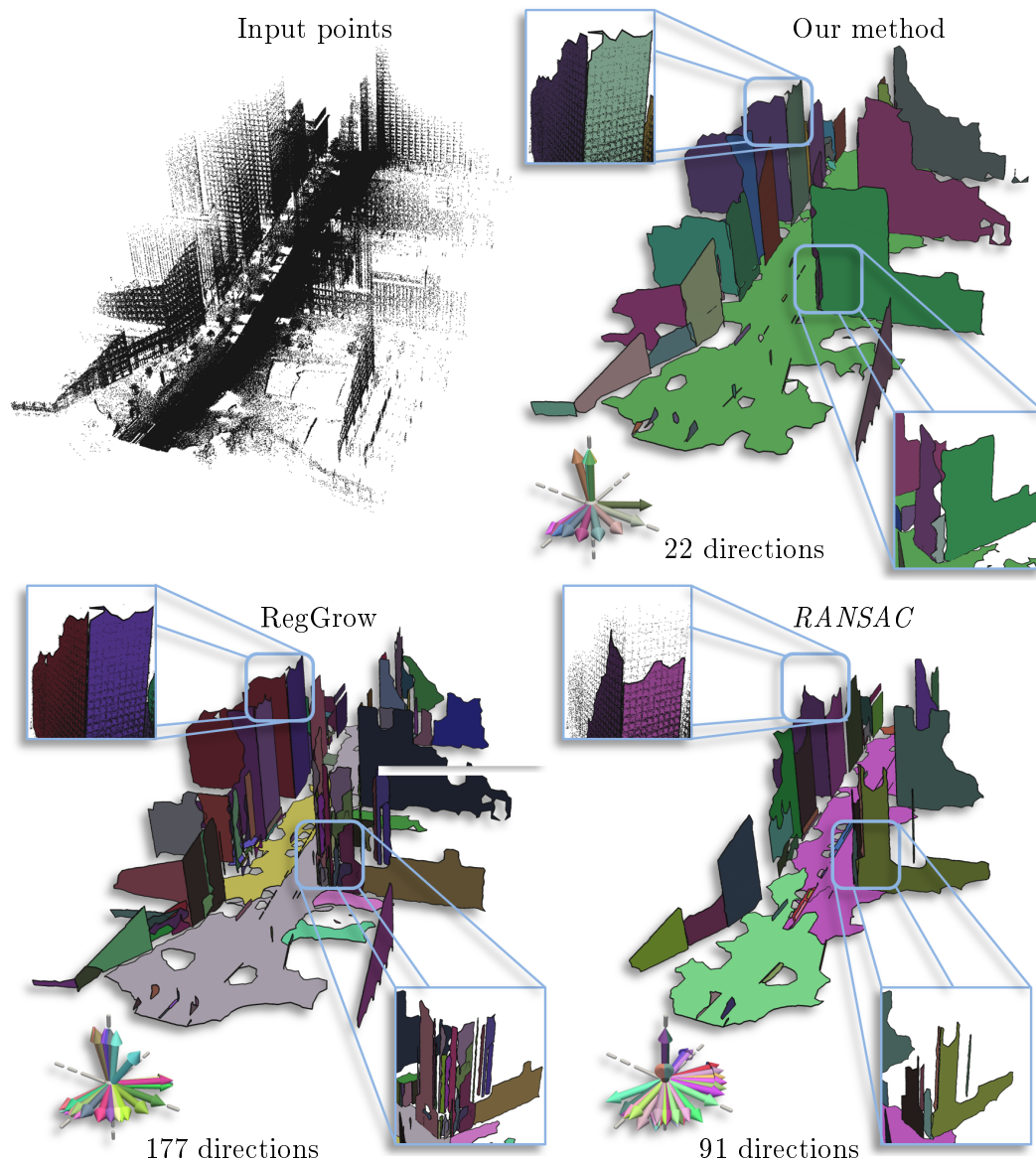


Figure 2.15: **Road**. This dataset was acquired from a car moving along a street. The point density strongly varies from the road towards the top of the buildings. Our method (top right) and the *RegGrow* method (bottom left) exhibit resilience against varying point density and are able to detect details close to the road as well as shapes in sparse areas. The *RANSAC* method (bottom right) does not adapt well to the sparse sampling and is not able to detect the upper parts of the buildings, see excerpts in upper part. This leads to a lower coverage 79.3% vs. 87.8% (our method) and 88.7% (*RegGrow*), see Table 2.1. The *RegGrow* method tends to generate a higher number of primitives compared to the other methods, shown in the lower excerpt showing a curved part of a building. The result of our method shows a stronger regularity compared to *RegGrow*.

datasets	methods	evaluation criteria				output complexity		
		\emptyset error	coverage	time	complexity	#shapes	#dirs	#planes
Kahn 5.2M pts Fig. 2.11	Ours	0.045	72,0%	8.7s	0.33	200	12	179
	<i>RANSAC</i>	0.041	72,4%	34.9s	0.98	200	195	200
	RegGrow	0.044	76,6%	348s	0.94	295	272	290
Euler 3.9M pts Fig. 2.14	Ours	0.014	81,0%	5.1s	0.37	133	14	120
	<i>RANSAC</i>	0.011	81,8%	26,2s	0.99	232	228	232
	RegGrow	0.012	87,1%	379,1s	0.97	284	273	284
Kinect 0.3M pts Fig. 2.13	Ours	0.22	43,6%	2.2s	0.30	47	3	39
	<i>RANSAC</i>	0.26	84,8%	12.6s	1.0	84	84	84
	Globfit	0.24	50,5%	185s	0.34	48	3	46
Road 1.4M pts Fig. 2.15	Ours	1.6	87,8%	5.2s	0.49	73	22	63
	<i>RANSAC</i>	1.6	79,3%	28.3s	1.0	91	91	91
	RegGrow	1.52	88,7%	102s	0.96	185	177	181
MVS alley 1.1M pts	Ours	0.031	83,3%	3.2s	0.42	69	11	67
	<i>RANSAC</i>	0.028	82,2%	8.5s	1.0	55	55	55
	RegGrow	0.028	83,2%	29s	0.99	133	132	133

Table 2.1: **Benchmark.** The Kahn and Euler datasets represent buildings acquired by a laser scanner. The Kinect dataset shows a small indoor scene of several registered Kinect scans. The Road dataset shows a urban scene recorded by a laser scanner mounted on a car. A multi-view stereo dataset, MVS alley, was used to test the methods on another kind of data featuring stronger noise, irregular and incomplete sampling.

The *RANSAC* method shows a comparatively low computation time compared to *RegGrow*, but is significantly slower than our GPU-based algorithm. While the coverage of the data is similar to ours, *RANSAC* does not perform any regularization and therefore exhibits low regularity. The coverage of *RANSAC* is similar to the one produced by our method, albeit the process devised to ensure connected components is not adaptive to variable point density and has impact on the running times. Choosing a small tolerance for connectivity leads to a separate detection of details in densely sampled areas and to absence of detection in sparse areas. A high tolerance for connectivity leads to loss of details in dense areas, but yields reconstruction in sparse areas (Figure 2.15).

The *RegGrow* method achieves a higher coverage in almost all experiments compared to the two other methods, but the number of detected primitives is higher while all methods are set to use the same minimal number of points per shape. The comparison with our region growing mechanism shows, that in some cases the regularity of shapes may come at the cost of coverage, see Fig. 2.14.

For evaluating *GlobFit* we used the implementation provided by the authors, and rely upon the output of *RANSAC* as in the original publication. It can optimize for wider range of relationships, but is both memory- and compute-intensive. This renders the method unsuitable for datasets at the scale of urban scenes, and we could compare on a single dataset due to excess of memory consumptions. On the *Kinect* dataset the regularization yields high regularity by enforcing a Manhattan world providing a similar result to our method.

Robustness. To evaluate the robustness of our method we manually designed a model of a house in *Trimble Sketchup* and generated a defect-laden point set. Such designed model provides us with a ground truth: we can distinguish between points sampled from a shape and added outliers, and thus correctly measure the coverage and fidelity to the ground truth, see Figure 2.12.

The constantly recurring detection and reinforcement of non-local regularities makes the method resilient against outliers, noise and sparse sampling. By jointly fitting parallel shapes the accuracy in sparsely sampled noisy areas is reinforced by parallel shapes from densely sampled areas, see Fig. 2.13. The region-growing method inherently provides to some extent outlier robustness due to its local propagation behavior.

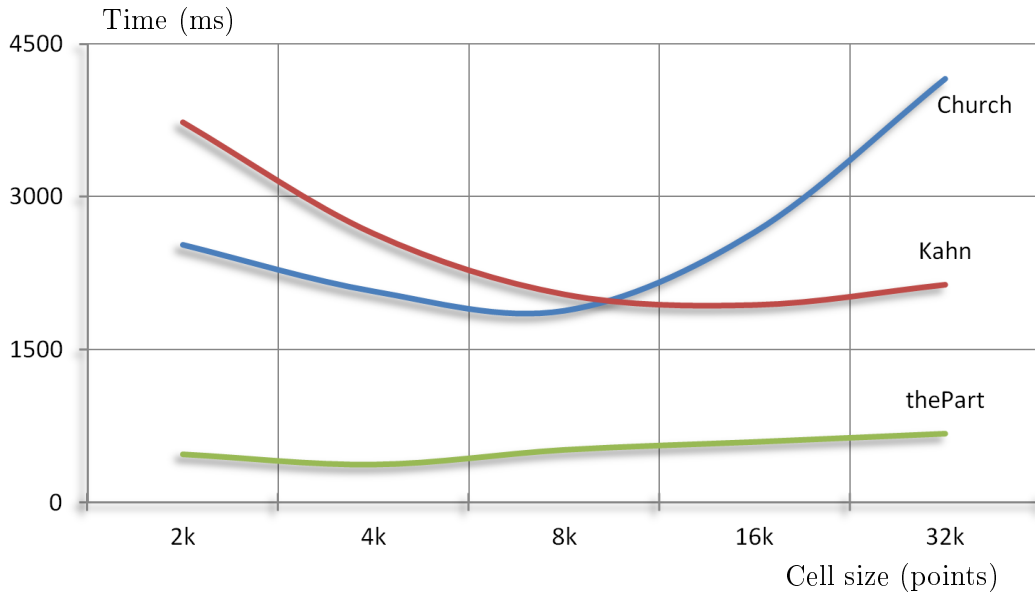


Figure 2.16: **Octree parameters.** We depict the impact of the chosen cell size of the octree upon the running time. The Kahn dataset is shown in Fig. 2.11, Church in Fig. 2.6 and thePart in Fig. 2.8. The Kahn dataset has been acquired by a laser scanner and features a high point density with low noise. A larger cell size enables fast shape growing not limited by the space decomposition. The church dataset instead exhibits higher noise, lower point density and less points per shape. A smaller cell size enables detecting all shapes with fewer iterations, as only one shape per cell and iteration can be detected.

Parameters. The algorithm requires selecting few parameters: ε , the Euclidean tolerance error distance between shape and input points, α , the normal tolerance deviation between shape and input samples, and the minimum number of points per shape are common among shape detection methods. β is the maximum angle deviation used to consider two planes as parallel during the detection of regularities. The chosen cell size for octree creation determines the number of generated leaf cells and therefore the degree of parallelism in execution. Per iteration of the method at most one seed point is chosen per cell and therefore at most one shape per cell is detected. A separation into few large cells allows further expansion of single shapes, but requires more iterations for detecting all shapes and leads to inefficient load balancing. Many small cells, however, lead to better load balancing, but might add some overhead due to the increase of shapes for regularization and fitting. A graph

evaluating the impact of cell size on performance is shown in Fig. 2.16. For point sets mainly consisting of large planar shapes, e.g. architecture captured by a laser scanner, a high cell size leads to a higher performance. An upper cell size limit is imposed by the hardware specification of the GPU. Choosing a cell size leading to fewer cells than the GPU can handle in parallel will not use the full capacity of the GPU. Otherwise, for more detailed geometry a smaller cell size is preferable. However, in our experiments we found a cell size of 8k suitable in most cases.

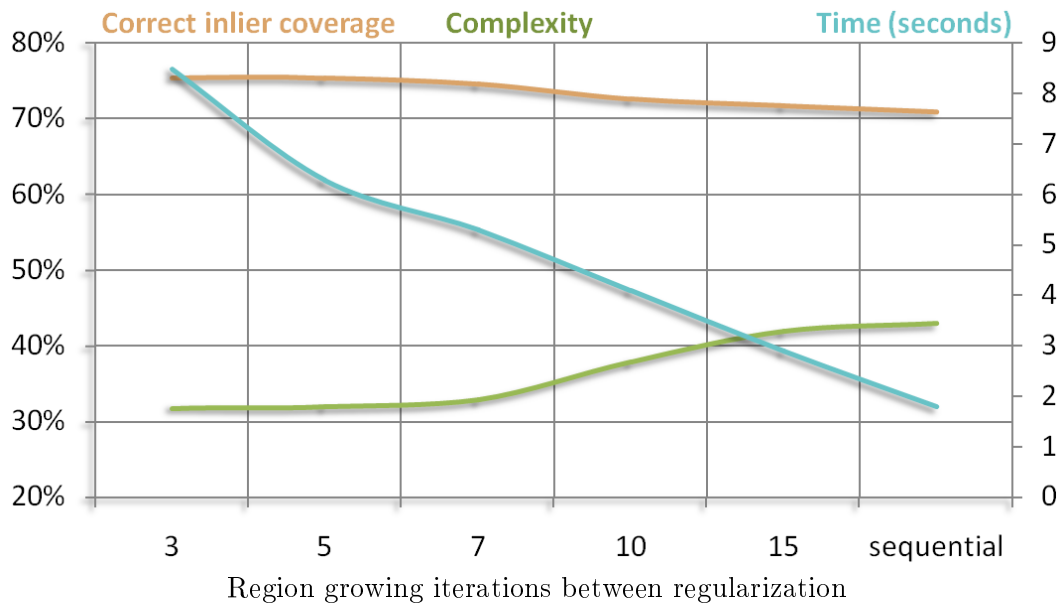


Figure 2.17: **Synergy between regularization and detection.** We evaluate the mutual benefit between detection and regularization by varying the alternating frequency. A progressive detection and regularization yields higher detection rate and regularity, at the cost of increased computational time due to additional regularization, compared to a less progressive or even purely sequential process. A highly frequent alternation, however, provides no additional gain for the additional invested computations.

The chosen number k of nearest neighbors impacts the propagation speed of shapes and the ability to handle anisotropic data. However, the k NN are stored for each input sample in memory on GPU and imposes a restriction to the maximum number of points that can be processed at once. In our implementation for each input sample the position, the normal and two integer flags are stored on GPU. This leads to a memory consumption of $32 \text{ Bytes} + k \times 4 \text{ Bytes}$ per point. For a common

choice such as $k = 20$, each sample point consumes 112 Bytes. For a GPU with 1GB of memory the maximum number of points to process at once is around 8-9M considering a few other memory structures (GPUs with 12GB memory are available but not yet routine).

Synergy of regularization and detection. A distinctive property of our algorithm is the interleaved detection and regularization. We evaluate our method with respect to sequential approaches by varying the frequency between detection, i.e. region growing, and regularization. We use as input our sampled ground truth model with added noise and outliers (resp. 0.2% and 20%) and measure both coverage (in percent with respect to the sampled points) and regularity, see Fig. 2.17. This experiment shows that regularization provides a guidance during detection leading to a higher fidelity to the sampled points. Notice that a high frequency yields no further benefit and increase computational times. A very low frequency or even a purely sequential approach leads to shapes with large spatial extend. The regularization potential of these large shapes is limited as, in general, a change in orientation implies a large deviation from its assigned points.

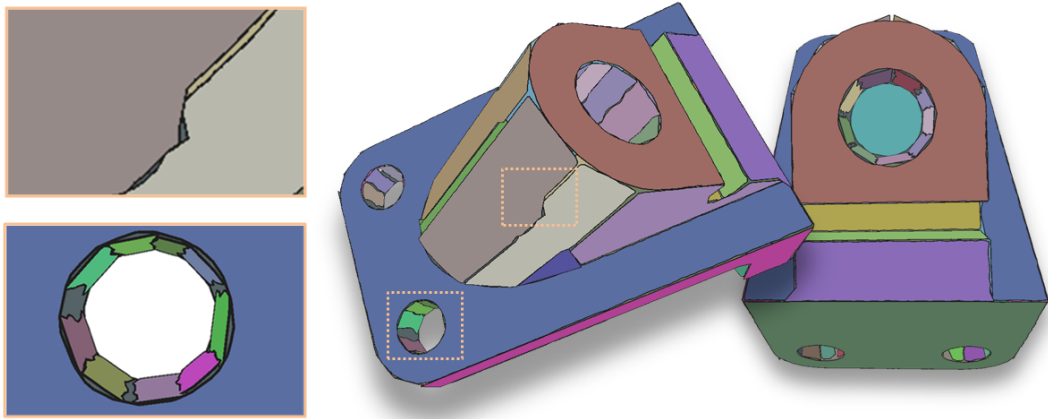


Figure 2.18: **Detection on curved shapes.** This mechanical part contains several cylindrical surface parts at different scale as well as planar parts. The regularization of planar shapes on curved surfaces may lead to the detection of irregular interfaces between shapes or over-segmentation, depicted in the lower left close-up. Some small shapes approximating the cylindrical parts are aligned to the larger planar shapes (right). Note however that the regularization of the larger planar parts is not contaminated by the cylindrical parts.

Limitations. While providing fast detection and alignment of planar structures our method is not designed for the reconstruction of free-form shapes. Our algorithm approximates curved surfaces by planar patches. However, due to the confinement of the region growing within one cell the orientation of the space partitioning is likely to impact the detected shapes on curved parts, see Fig. 2.18.

Processing data on the GPU provides the benefit of highly parallelized processing. However, it comes with a memory restriction limiting the size of the datasets that can be processed. This is partially due to the kNN for the shape propagation as k indices must be stored for each input sample.

2.3 Summary

We presented two methods for detecting the relevant shapes in point sets for indoor reconstruction, i.e., linear and planar shapes. In the first part a multi-scale line fitting approach is proposed to accurately extract line segments from detailed 2D point sampled geometry. Although aimed at the modeling of piecewise linear boundary, circular parts are covered by several line segments.

Our second contribution is a method for detecting and regularizing planar primitive shapes from unorganized 3D point clouds acquired on man-made physical scenes. A novel aspect of our method consists in interleaving shape detection and regularization so as to make the two processes mutually cooperate. Such approach is shown to improve detection and robustness, in particular when dealing with defect-laden data. Another contribution is to design all data structures and algorithm components with an eye on constraints of modern GPU architectures. Our experiments on a variety of point clouds demonstrate the added value of our approach in terms of efficiency, detection quality and regularity. The main parameters of our algorithm provide us with a means to trade coverage for regularity.

Classification

Indoor scenes exhibit a wide variety of different structures and objects depending on the environment. While office buildings might contain many similar rooms with only a few variants of chairs, desks, etc., residential homes and industrial sites will often exhibit very different configurations and amounts of structure and clutter. Structure is often piecewise planar for manufacturing reasons, whereas clutter can exhibit any shape, from very regular and piecewise planar, e.g. a box-shaped wardrobe, to very irregular, e.g., plants and cloth.

In this chapter we present two methods for semantizing point clouds acquired from indoor scenes. Reconstructing indoor scenes requires processing specialized to the underlying type of surface. A reconstruction of the indoor space often assumes piecewise linear structures. We thus rely on planar shape detection.

Our first contribution provides a separation of data collected from structures from other data like outliers and clutter. The goal of this method is to provide a separation into permanent structure and clutter, facilitating the extraction of geometric shapes from sampled walls, floor and ceiling. A statistical analysis is performed to detect ceiling and floor in the input data. The input data is partitioned into horizontal slices containing mainly ceiling, floor or walls. This allows for further individual processing of each element.

Our second method aims at the classification of indoor objects from planar shapes. Object classification is an important facet of the scene understanding problem and has a wide range of applications such as robotics or augmented reality. Our main idea is to abstract planar shapes from point data on several scales and to derive global features from the relationship between the shapes. A supervised machine learning method is trained to solve the multiclass classification problem. Compared to typical keypoint based features, which capture the local geometry, considering the global relationships allows a better generalization from single object instances to object classes.

3.1 Statistical analysis

The high amount of clutter in indoor scenes challenges reconstruction methods not just by occluding the structures to be acquired, but also by appearing in the point clouds itself. A geometric modeling of the architecture requires distinguishing between structure and clutter, outliers and artifacts.

For separating the structure from the other measured data, we exploit the general piecewise-planarity of structure. However, compared to other methods our algorithm does not require a Manhattan world scene. Walls are assumed to be vertical and perpendicular to floor and ceiling. We assume that walls are piecewise linear along the vertical direction. Although residential homes might contain a high amount of clutter, large parts of the planar structure are typically visible to the acquisition device. Especially ceiling and floor are represented by a high amount of samples in most data sets. Additionally, they are horizontal and planar resulting in a high amount of points at certain heights in the input data. At every height where the permanent structures change, e.g. windows or doors, we assume the presence of horizontal planar structures, i.e., floor or ceiling. To detect these horizontal structures, we extend an idea from Okorn *et al.* [OXA10]. As horizontal structures result in high numbers of samples sharing similar heights, they appear as peaks in the point distribution along the vertical axis. The presence of clutter challenges the detection as it appears as noise in the distribution. We remove all points from the distribution whose normals are not parallel to the vertical direction. This cleans the histogram and facilitates the detection of peaks, see Fig. 3.1. If normals are not available, they are estimated through local principal component analysis (PCA) applied to a local neighborhood. The point cloud is vertically partitioned into horizontal slices containing important horizontal structures and vertical slices

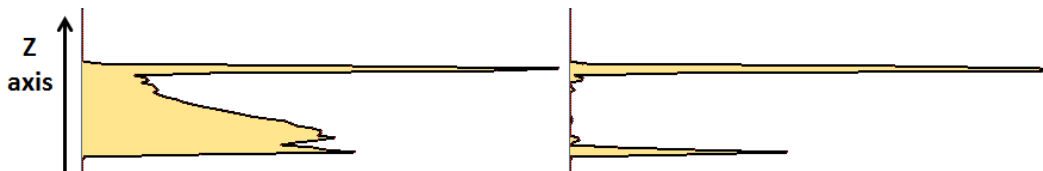


Figure 3.1: **Cluttered and uncluttered distributions.** Comparing point distributions along vertical direction n_z without filtering by normals (left) and with filtering by normals (right, $|n_i \cdot n_z| > 0.98$). The chosen bin size is 8 cm and the histogram is split into 69 bins.

containing vertical structures, i.e., walls. For extracting the peaks from the point distribution we create a histogram, where the bin size is user-specified (in range 5-10 cm by default). The peaks in the histogram are located through mean shift [Che95]. Mean shift is an iterative algorithm similar to gradient descent for detecting maxima in density functions. For discretized spaces, mean shift is able to detect the interpolated maxima and yields a higher precision than the discretization of the underlying space. Before locating the peaks in the histogram via mean shift, a Gaussian smoothing is applied. We use a flat kernel for mean shift. By choosing a kernel and bin size, close peaks in the histogram are inherently merged by mean shift. Each bin in the histogram is used as a starting point for the mean shift. We iterate until either convergence or a maximum of 10 iterations has been reached. To ensure a minimum distance between two peaks, close peaks within a small distance h are clustered. We choose a default value of twice the bin size for h . The z coordinate of each maximum is denoted by $m_i, i \in \{1, \dots, N_m\}$.

The point cloud is now split at points around the peaks into horizontal structure-slices, containing the peaks and representing floor and ceiling, and into wall-slices, covering the remaining parts representing the walls and remaining clutter. The split points are thus selected based on the gradient of the histogram. The number of points in the bins of the histogram is assumed to drop significantly around the horizontal structures. The split points are located by walking through the bins in the histogram in both directions from the peak until the gradient is significantly smaller than at the first step adjacent to the peak. An example of selected split points is depicted by Figure 3.2.

If the split points are not located within a certain range around the peak, the peak is considered to be clutter and removed. As every local maximum in the histogram is detected, peaks that are not significant, i.e., that do not stand out to their neighborhood by a certain ratio, are removed. As close peaks are merged by mean shift and splitting points are chosen within a maximum range, it is guaranteed that split points of adjacent peaks do not overlap. The type of slice, wall-slices or horizontal structure-slices, is thus alternating along the vertical direction. For the following steps only the wall-slices are considered. However, both types of slices are used for the graph-cut optimization used for model extraction.

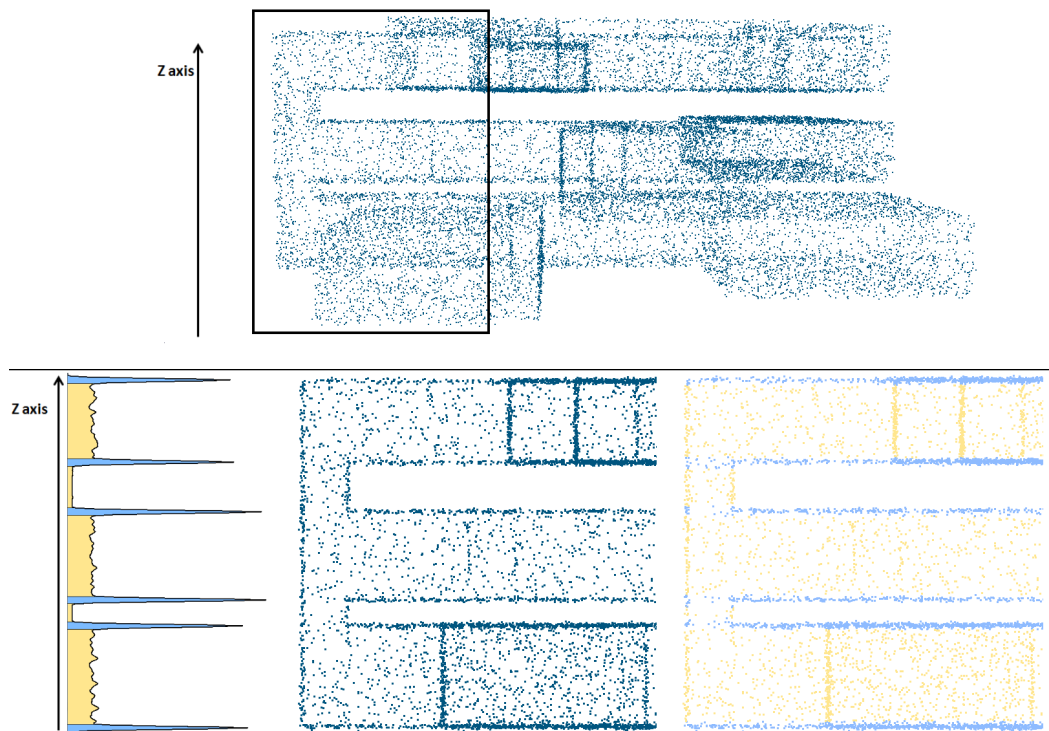


Figure 3.2: **Horizontal slicing applied to a synthetic scene.** Top: Input point cloud. The excerpt shown below is highlighted by a black box. Left: Distribution along vertical direction. The point cloud is split horizontally within a small range around the peaks. Ranges for horizontal structure-slices are depicted in blue, and yellow for wall-slices. Middle: Excerpt of original point cloud in side-view. Right: Side-view of excerpt colored by slice type.

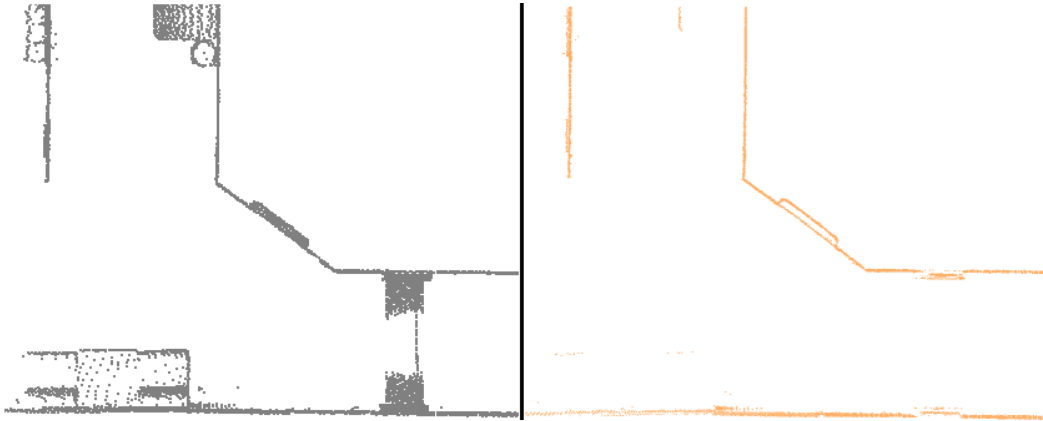


Figure 3.3: **Clutter removal.** Left: Wall-slice of a real dataset showing a corridor scene in top-view. The scene contains clutter that challenges the wall detection. Right: Most of the clutter gets removed by filtering the points by their normals.

Filtering: The detection of wall segments in indoor scenes is hampered by the presence of clutter: The clutter occludes the sight of the scanning device so that parts of the wall geometry are not sampled. When detecting permanent structures in the point cloud, clutter may be classified as permanent structure or vice versa. To avoid these cases, clutter is removed by filtering the points of the wall-slices by their normals. As walls are assumed to be vertical, points with a non-horizontal normal are filtered. More specifically, each point p_i in the wall-slices with normal n_i is removed when $|n_i \cdot n_z| < t$, where n_z denotes the unit vector along the vertical axis. A threshold of $t = 0.02$ corresponds to a deviation of a few degrees of the normal from the horizontal plane and provides a strict filtering of clutter while preserving samples on wall geometry, see Figure 3.3.

In order to gain robustness to missing data caused by occlusion and to reduce the problem to 2D, each wall-slice is projected vertically into the horizontal plane. Since walls are nearly vertical they get projected into line segments. Depending on the acquisition method the point distribution may exhibit a strong anisotropy. To remove anisotropy and speed up the following steps, we perform downsampling through an *occupancy grid* with uniform grid-size τ , provided as a parameter. In an occupancy grid each grid cell is labeled either empty or occupied. Downsampling is performed via replacing all points inside the cell by the averaged point location. This restricts the spatial point density to the grid size and approximates the original point coverage, see Figure 3.4. A small grid size τ allows for preservation of details in highly sampled areas, whereas a large grid size allows for the reconstruction of

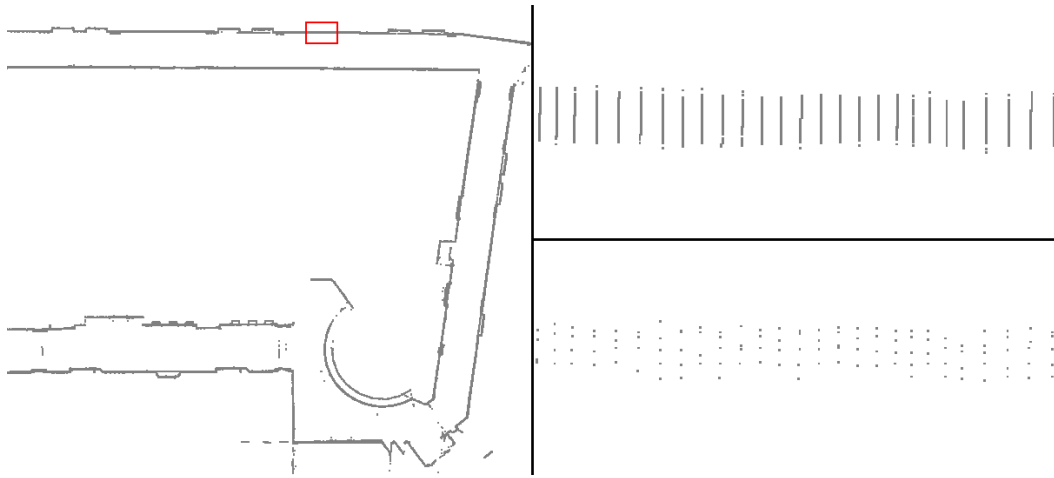


Figure 3.4: **Resampling for anisotropy removal.** Left: Overview of wall-slice of real dataset. Upper right: Original point distribution of wall section in red box. Strong anisotropy is caused by the acquisition method of the scanning device. Lower right: The anisotropy is significantly reduced after downsampling.

sparsely sampled areas.

3.2 Object classification via planar abstraction

Our approach is to use a supervised machine learning method, random forest, to distinguish between object classes. The goal is to create a classifier that is trained for different object classes and can predict the class for an unknown object. For training and classification each object is represented by an object descriptor, i.e., a feature vector. We extract a global feature vector from a set of planar shapes for each object.

Our method takes as input a set of point clouds with unoriented normals, sampled from objects. When normal attributes are not available we estimate them using a principal component analysis in a local neighborhood. For training and evaluation of the classifier a set of ground-truth object labels of the input point clouds is required. We assume that the scene has already been segmented into objects and focus on the classification of objects. Some previous works perform segmentation of objects in a 3D scene [SHKF12] or perform clustering in feature space in order to segment similar objects in an indoor scan [MPM⁺14].

Our method generates as output a classifier, ready to predict a trained object class from a feature vector. Our method comprises three main steps:

- Preprocessing (multiscale planar abstraction and adjacency detection)
- Feature computation
- Training



Figure 3.5: **Multiscale Planar Abstraction.** Left: Input point cloud of a goblet with outliers and noise. Middle left to far right: Planar abstraction with varying fitting tolerance from coarse to fine: 1%, 0.5% and 0.25% of bounding box diagonal.

3.2.1 Multiscale Planar Abstraction

The input point data are abstracted by planar shapes using an efficient RANSAC approach [SWK07], with three distinct fitting tolerances to capture the variation of the extracted shapes at different scales. The feature vector, computed in following step, aggregates all scales. More specifically, the largest fitting tolerance ε is chosen as 2% of the longest bounding box diagonal, then each following scale ε is halved. The main reasons for proceeding in a multi-scale fashion are the following. A detailed abstraction by a large number of small planar shapes obfuscates the dominant surfaces of the object. Conversely, choosing a large fitting tolerance captures well the dominant shapes but obfuscates the details. In addition, curved objects behaves differently, as the abstractions differ for each value fitting tolerance, see Fig. 3.5.

3.2.2 Features

Classification through machine learning requires a meaningful description of an object represented by a feature vector:

$$x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n, \quad (3.1)$$

where n denotes the dimension, similar for all feature vectors. In our approach we compute one feature vector per object, and the features are derived solely from the planar shapes. The main rational behind our choice of feature vectors is that the function of an object, the class in our context, constrains the shape. As the number of planar shapes detected from a single object depends on the object and detection parameters, we represent *distributions* of features computed for the whole set of planar shapes detected for each object. Each bin of the distribution represents one element of the feature vector, and the distributions are normalized to ensure comparability. Most features describe distributions: areas, orientations, and relationships between pairs of shapes: pairwise orientation, pairwise orientation restricted to adjacent shapes, transversality. We also add feature elements measuring the global aspect ratio of the object. Prior to computing the feature vectors we compute for each shape a planar polygon derived from the 2D alpha-shape of the associated point cloud, projected in the detected plane. A planar polygon facilitates computation of geometric properties such as areas and pairwise orientation. Note that the random forest approach is oblivious to the relations between the elements of the feature vector, so that a series of elements that belong to the same distribution is unknown to the classifier. In general each element of the feature vector is compared to the same

element of other feature vectors. The number of bins of the distributions is thus kept low to avoid increasing the sensitivity of the classifier and to separate objects of the same type. We detail next the features used for training and classification.

Area Fragmentation. We compute the distribution of shape areas, normalized to sum up to 1. More specifically, we accumulate the shape area within each bin of the distribution, instead of counting the shapes within a specific area range. The fragmentation of shape areas reflects whether the surface of an object is composed of few large shapes or many smaller planar shapes, or of anything in-between such as for a curved surface with a wide range of curvatures, see Fig. 3.6. We observed that using a linear scale for the bins of the distribution leads to a poor discriminative capability for the shapes with small areas: We have in general either very few large shapes, or many small shapes. We thus use a logarithmic scale of base 2 to provide a higher resolution for the small area bins.

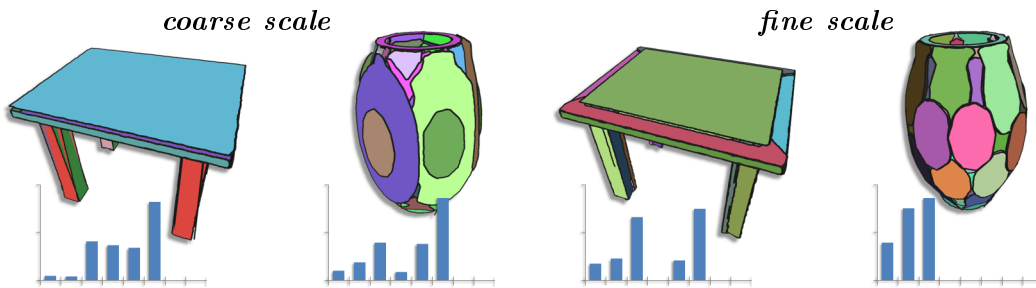


Figure 3.6: **Area fragmentation under multiple scales.** Left: Planar shapes detected from two point clouds with a large fitting tolerance. The area fragmentation distribution exhibits a high contribution of large shapes to the total shape area. Right: Using a small fitting tolerance for shape detection strongly changes the shape composition and hence distribution of the vase, while large shapes in the distribution for the table remain and only medium and small sized shapes change.

Pairwise Orientation. Assuming the pose of an object is known, the orientation of the parts is judged very discriminant by the random forest algorithm. When the pose is unknown however, the pose must be normalized to ensure bin-to-bin comparability by the machine learning method. In the SIFT operator [Low99] rotation-invariance is achieved by aligning the distribution with the reference direction derived from the largest signal peak in the neighborhood of a keypoint. We compute instead the distribution of angles between all pairs of planar parts, as this

does not require any reference direction. More specifically, we consider the range of angles $[0, \frac{\pi}{2}]$ as the normals are unoriented, and split this range evenly among the bins of the distribution. We then accumulate in each bin the product of areas of the corresponding pair of planar shapes. The distribution is normalized such that all bins sum up to 1. The discrimination capability for the pairwise orientation is depicted by Fig. 3.7.

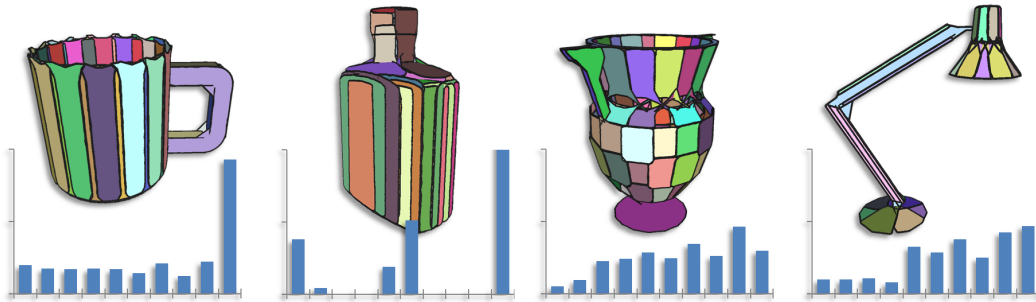


Figure 3.7: **Pairwise Orientation.** The distribution of pairwise orientation helps distinguishing different curved objects. The cylindrical shape of the mug is translated into a mostly uniform distribution with a peak owing to the bottom. The orientation distribution for the vase (middle right) reflects the bulgy body by a broader range of angles compared to the lamp (far right).

Adjacent Pairwise Orientation. In addition to the global pairwise orientation we compute the distribution of relative orientations of planar parts that are adjacent, as they reflect the sharpness of creases. Two planar shapes are considered adjacent if their respective alpha-shapes are closer than a user-specified distance, normalized by the longest bounding box diagonal. We first compute the bounding box of each shape and insert them in a hierarchical data structure (AABB tree) to accelerate the distance computations.

Orientation. The absolute orientation of planar parts plays an important discriminant role to determine the class of an object. Absolute orientation herein refers to a reference upward direction, which is unknown. We thus estimate a reference direction for each object by fitting an object-oriented bounding box. To infer a reference direction we proceed as follows. If the axis of the box with largest extent is unique we chose it as reference direction. Conversely, if the two major axes have comparable extend, we switch to the direction of minor axis. We then compare for

each planar shape its projected area with respect to the reference direction, and accumulate these areas in a distribution, with a range of angles $[0, \frac{\pi}{2}]$. In addition to the orientation distribution, we add to the feature vector the aspect ratio of the oriented bounding box computed as the length of the major axis divided by the length of the longest diagonal.

Transversality. Transversality is a notion that describes how shapes intersect. In our context transversality also reflects the structure of an object. A compact object, such as a drawer or a bottle, exhibits a low transversality while a bookshelf exhibits a high transversality. We compute the transversality of planar shapes by recording the relative positioning of all pairs of shapes that are adjacent. Two adjacent shapes that do not meet at their boundary are considered transverse. Given two adjacent planar shapes A and B , we compute the transversality $T(A, B)$ as the (smallest) ratio of areas of A on both sides of the supporting plane of B . For each pair of shapes (A, B) we compute the maximum transversality between $T(A, B)$ and $T(B, A)$. We then compute a transversality distribution with range $[0, \frac{1}{2}]$, and accumulate in the bins the normalized products of areas for all pairs of adjacent shape. We opt for a small number of bins to avoid confusing low transversality with detection inaccuracies.

3.2.3 Random Forest

Classification via supervised machine learning is performed in two phases. In the training phase a set of feature vectors with associated class labels is used to train a classifier. We choose random forests as machine learning approach, as it is general and effective on many classification problems. It is fast in training as well as in classification and can be parallelized. We use the implementation provided by OpenCV [Bra00]. Random forests operate by constructing a multitude of decision trees. Decision trees are built by choosing the most discriminative feature, i.e., the element in the feature vector, as a node to separate the training data according to their known class labels. Decision trees are known to overfit, i.e., to adapt to small variations and noise in the training data. Random forests overcome this issue by creating a large number of decision trees. For each decision tree a random subset of the training data is chosen and on each node only a random subset of the features are used. Additionally, the maximum depth of the trees can be limited. The classification is performed as a voting. The feature vector of an unknown object is evaluated on each tree and the predicted label corresponds to the most voted label.

Random forests aim at providing the highest prediction performance for the training data set. Choosing an imbalanced training set, where the number of training samples for each object class varies, can lead to a poor prediction performance for the underrepresented classes. The classifier sometimes can achieve a higher prediction performance by neglecting the minority classes. There are different ways to improve the performance for all classes. A common and effective way is to downsample over-represented classes instead of upsampling the minority classes as this may increase noise [CLB04].

3.2.4 Experiments

We implemented our approach in C++ using the CGAL Library [CGAL15], OpenCV [Bra00] and the efficient RANSAC approach implemented by Schnabel [SWK07]. The size of the feature vectors are as follows: 8 bins for the area fragmentation distribution, 10 bins for the pairwise orientation and pairwise adjacent orientation distributions and 5 bins for the orientation and transversality distributions. We observed the best results for different 3 scales. This sums up to a feature vector size of dimension 115, including the oriented bounding box ratio.

Object Databases. We perform the evaluation of our classifier on a subset of the Princeton Shape Benchmark [ben04], see Fig. 3.8. A subset of the full dataset is used as many objects do not belong to the indoor environment. We select 100 objects from 8 different object classes that are common to indoor scenes: Bottle, Chair, Couch, Lamp, Mug, Shelf, Table and Vase. Each model in the object database is sampled into a point cloud by ray shooting and oriented into a random direction to test orientation invariance. The calculated set of features is split into two sets: 60% for training and 40% for evaluation. To avoid a bias toward overrepresented classes, we remove samples until every class is represented evenly. On the benchmark we achieve a precision of 82,5%. The confusion matrix records which are predicted for the objects of one class. Misclassification occurs more often among the objects with curved surfaces. However, the classification of furniture is precise.


Our method is also evaluated from scanned indoor objects, see Fig. 3.10. Contrary to the previous experiment, the input point clouds are incomplete and suffer from anisotropy, noise and outliers due to acquisition constraints. 20 objects from two different classes, i.e., chair and non-chair, are considered. The training was performed on the scanned indoor objects from randomly chosen 60%, i.e., 12 samples.



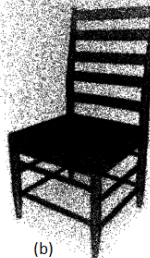
Figure 3.8: **Princeton Shape Benchmark.** The objects from the Princeton Shape Benchmark cover a variety of indoor objects with different shapes. Four tabletop object classes are used: Bottle, Lamp, Mug and Vase. We also select four furniture object classes common to indoor scenes: Chair, Couch, Shelf and Table. Before processing each object is point sampled by random ray shooting, scaled to fit into the unit cube and rotated into an arbitrary pose.

(a)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase	(c)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	2	0	0	0	1	0	0	2	Bottle	4	0	0	0	0	0	0	1
Chair	0	4	1	0	0	0	0	0	Chair	0	2	1	1	0	1	0	0
Couch	0	0	5	0	0	0	0	0	Couch	0	1	4	0	0	0	1	0
Lamp	0	1	0	3	1	0	0	0	Lamp	0	0	1	3	0	0	0	0
Mug	0	0	0	0	4	0	0	1	Mug	2	0	0	0	1	0	0	2
Shelf	0	0	0	0	0	5	0	0	Shelf	0	0	0	0	0	5	0	0
Table	0	0	0	0	0	0	5	0	Table	0	0	0	0	0	0	5	0
Vase	0	0	0	0	0	0	0	5	Vase	0	0	0	1	0	0	0	4

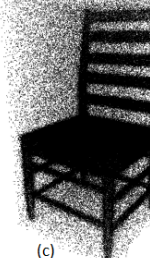
(b)	Bottle	Chair	Couch	Lamp	Mug	Shelf	Table	Vase
Bottle	2	0	0	0	1	0	0	0
Chair	1	4	0	0	0	0	0	0
Couch	0	1	3	0	0	0	1	0
Lamp	0	0	0	4	1	0	0	0
Mug	0	0	0	0	4	0	0	1
Shelf	0	0	0	0	0	5	0	0
Table	0	0	0	0	0	0	5	0
Vase	1	0	0	0	0	0	0	4



(a)



(b)



(c)

Figure 3.9: **Confusion matrices.** Performance of the method trained and tested on different subsets of the Princeton Shape Benchmark [ben04] with different added amounts of noise and outliers. (a): Sampled dataset without noise and outliers. The precision of the class prediction is 82,5%. The classifier is not very reliable for the classification of bottles, which are often mislabeled as vases. (b): Added 10% outliers and 0.5% noise. Compared to the noise-free version the precision slightly drops to 77.5%. (c): Added 20% outliers and 1% noise. The method maintains a precision of 70% for this level of noise. Bottom: A sample of the input point clouds is shown with increasing amounts of noise/outliers.

The classification of the remaining 8 objects predicted correct labels for all chairs and misclassified one non-chair object. The overall precision is 87,5%.



Figure 3.10: **Indoor objects.** We acquired 20 indoor objects with a Leica Scanstation P20 laser scanner. The sampling of the objects is heterogeneous and partly anisotropic. The lower 10 objects are labeled as chairs whereas the upper ten objects are labeled as non chairs.

Feature importance. Random forests record the importance of each feature after training. The importance translates the relevance of the feature for separating the class labels during the training process. Table 3.1 shows the feature importance for evaluation with the Princeton Shape Benchmark. The most relevant feature is the pairwise orientation histogram. The least meaningful feature for the Princeton Shape Benchmark is the transversality, yet it improves the precision. The importance for each scale shows that the multiscale approach provides a significant advantage for classification. The shape detection at fine scale, i.e., with a small fitting tolerance, often yields the largest number of shapes, but contributes the most to the precision. Note that all scales contribute to the classification performance, increas-

ingly from coarse to fine. The transversality at coarse scale provides no significant contribution. Using a high fitting tolerance for non-simple objects leads to overlapping and intersection of detected shapes and induces meaningless transversality.

Scale	Area fragmentation	Pairwise orientation	Adjacent pairwise orientation	Orientation	Transversality	Total
Coarse	3.3%	9.6%	5%	1.9%	0.3%	20.1%
Medium	5.8%	14.3%	6.6%	2.4%	2.1%	31.2%
Fine	6.4%	15.1%	6.8%	4.7%	4.1%	37%
All	15.5%	39%	18.4%	9%	6.5%	88.4%

Table 3.1: **Feature importance.** Contribution of different features to the classifier performance by using the Princeton Shape Benchmark. In addition to the histogram features per scale there is the oriented bounding box ratio as a single scalar feature with importance 11.6%.

Robustness. To evaluate the robustness of our method, we use the Princeton Shape Benchmark as before, and add noise and outliers before performing the multiscale shape detection. The performance under addition of high noise is recorded as confusion matrices in Fig 3.9. We performed two experiments and added 10% (20%) outliers and 0.5% (1%) noise w.r.t. bounding box diagonal, see lower images in Fig. 3.9. The precision of our classifier is 77, 5% and 70% respectively.

Performance. As recorded by Tab.3.2, feature computation is the most computationally intensive operation of our approach. The computational times are overall moderate as only a few minutes are required to compute all the features of the hundred objects of the Princeton dataset, which represent a total of 25M input points. The timings for learning and testing phases are negligible.

	Feature computation	Learning	Testing
Princeton (8 classes)	232.4	0.11	$< 10^{-3}$
Indoor (2 classes)	1.82	0.02	$< 10^{-3}$

Table 3.2: **Running times (in seconds).**

Limitations. Our method assumes that the input objects have been preliminarily extracted from the environment. Although the object segmentation problem has been explored in depth in the literature, there is still no general solution that separates objects from scanned scenes with a 100% correctness. In terms of robustness, our method is less resilient to missing data than to noise, outliers and heterogeneous sampling.

3.3 Conclusions

In this chapter we presented two methods for classification of point data. Geometric modeling of point clouds acquired from indoor scenes requires a separation of points sampled from structure and clutter. We presented a method exploiting the piecewise linear assumption of permanent structures. Knowledge about the upward direction facilitates the identification of horizontal structures like floors and ceiling. The result is a set of horizontal slices containing horizontal structures or walls filtered from clutter.

We further introduced a novel method for classifying objects from sampled point data. Departing from previous approaches, our method exploits a planar abstraction to discriminate the different classes of interest. Planar shapes are easy to detect and manipulate, and allow for a compact object representation, typically a few dozen planar shapes instead of hundred thousands of points. This approach offers a real added value in terms of (i) robustness, (ii) orientation and scale invariance, and (iii) low computational complexity.

Geometric modeling of indoor space

In this chapter we present a method to reconstruct volumetric models of indoor spaces from dense point clouds acquired on buildings. While some recent works show satisfactory levels of details [BdLGM14, NIH⁺11, IKH⁺11] they are only applicable to scans from a single position or small scenes.

Our method performs a fully automated reconstruction of multi-level architectures and with high fidelity to small details. With respect to other methods [XF12, MMV⁺14, BdLGM14] our method is more flexible and does not require knowledge of the scanning origins for each point or structured data such as range images. Knowledge of the scanning origins facilitates the reconstruction process by indicating the solid/empty side of a surface through the line of sight. However, if the scanning origins are known our method can benefit from it.

The main challenges for indoor space reconstruction are posed by moderate to high amounts of clutter. First, the input data only provides point locations but no further information about the underlying surfaces. Under the main assumption of piecewise planar structures our method separates structures from clutter and extracts wall segments from the input data by applying methods introduced in former chapters. Second, due to occlusion the scanning process is not complete and parts of the permanent structure are not sampled. The core methodology behind our method is an energy minimization providing robustness to missing data and outliers by yielding a plausible watertight volume.

Our method for volumetric modeling of indoor spaces provides the following contributions:

- **Arbitrary wall directions:** The model is not restricted to the Manhattan-world geometry and deals with planar wall detection for arbitrary vertical directions. The only assumption is that floors and ceilings are horizontal.

- **Multi-level buildings:** Our approach reconstructs an entire building with multiple levels in a single optimization step, without requiring *a priori* knowledge about the levels.
- **Missing and outlier data:** 3D space partitioning into volumetric cells and labeling of the cells by a global energy minimization provides resilience to missing data and outliers.
- **Raw data:** To be as general and applicable as possible, only dense raw point sets and knowledge about vertical direction are required. Nevertheless, when oriented normals or knowledge about the scanning device position are provided, they are utilized used to further improve robustness.

4.1 Overview

Our method takes as input a point cloud $P = \{p_1, \dots, p_n\} \in R^3$ and consists of two main steps depicted in Figure 4.1:

1. **Space partitioning:** The bounding box of P is partitioned into volumetric cells by using detected permanent structures as splitting planes. We use the statistical analysis method described in Section 3.1 to split the input data into horizontal slices containing either horizontal or vertical structures. The wall structures contained in each horizontal slice are extracted using the feature-sensitive line extraction detailed in Section 2.1.
2. **Surface extraction:** The volumetric cells created in previous step are labeled into either solid or empty space, respectively for permanent structures (walls, floors, ceilings) or for outside. The final reconstructed surface is then deduced from the labeled cells.

4.2 Cell decomposition

We partition the bounding box by first splitting the horizontal cross section of the bounding box into a single 2D cell decomposition and stacking copies of that 2D cell decomposition vertically to yield the 3D space partitioning. For constructing the 2D cell decomposition we use the line segments extracted from the Hough Accumulator. In architecture, wall directions are often shared within and even across different levels of the building. To consolidate for wall segments that were

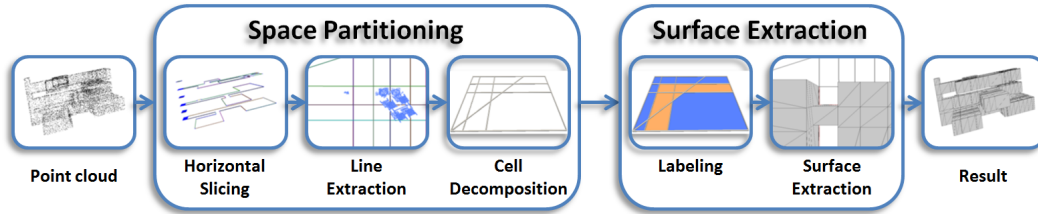


Figure 4.1: **Reconstruction pipeline.** The bounding box is split into cells using detected permanent structures. The final surface is obtained through labeling the cells in empty or solid space, and extracting the interfaces between empty and solid cells.

not detected on all floors, we combine line segments from all levels to create the 2D cell decomposition. During combination of lines from different levels very similar ones are clustered as they represent the same wall direction.

We consider line segments with a certain extent to be a shared wall direction and extend them to lines crossing the complete bounding box. We then use an arrangement data structure [AS98] for partitioning the horizontal plane by these lines into a 2D cell decomposition.

Line segments with a smaller extent are considered as local details and are only used to split cells in the decomposition locally. A default value for the minimum line extent of 1m has proven to be suitable.

The 3D space partitioning is then created by vertically stacking copies of the 2D cell decompositions, one for each wall-slice. Each copy is vertically extruded at the height of the corresponding peak in the distribution of the associated wall-slice, see Figure 4.2.

4.3 Cell occupancy labeling with min-cut

The final model is extracted from the 3D cell decomposition through labeling the cells as either empty or solid space. We formulated this binary labeling problem as a global energy minimization, solved through a graph-cut algorithm [BVZ01]. Such global minimization provides robustness against defect-laden data.

Graph-cuts are used to minimize certain energy formulations. It is popular especially in computer vision to solve low-level tasks such as segmentation or

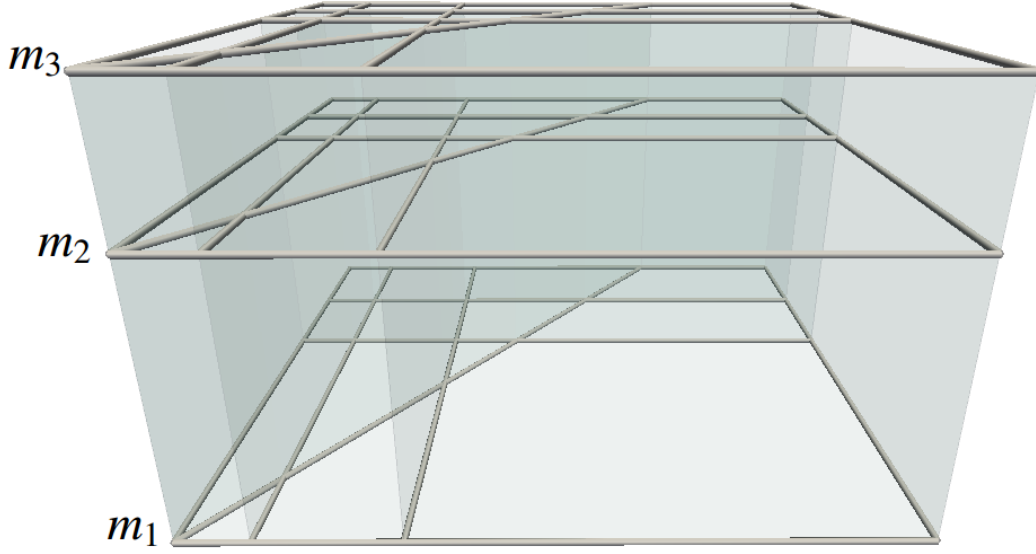


Figure 4.2: **3D space partitioning.** The 2D cell decomposition is extended in 3D through stacking and vertical extrusion. m_i denote the height of the detected peak during the horizontal slicing step.

stitching. Formulating a problem as an energy minimization provides a means to trade data fidelity for regularity. In case of defect-laden data hampered by noise, outliers or occlusion, one cannot rely solely on the output of basic segmentation or extraction methods. Incorporating priors like regularity makes it possible to find an optimal solution while trading data fidelity for complexity.

Solving the labeling problem via graph-cut requires an embedding of a graph G into the space decomposition. An undirected graph G is defined by a set of vertices V and a set of edges E :

$$G := (V, E) \quad \text{with} \quad E \in V^2. \quad (4.1)$$

Each vertex of G is associated to a volumetric cell. The edges of G connect all pairs of cells that share a vertical or horizontal face of the cell decomposition. We then use min-cut to minimize the following energy:

$$\min_{l \in \{0,1\}^{|V|}} \sum_{i \in V} D_i(l_i) + \alpha \sum_{(i,j) \in E} R_{i,j}(l_i, l_j), \quad (4.2)$$

where V denotes the vertices and E denotes the edges of G . The label l_i of a cell is either set to 0 for solid space or to 1 for empty space. D_i denotes the data term used

for each label l_i assigned to the cell i in order to favor data fidelity. $R_{i,j}$, referred to as the regularization term, represents a pairwise cost for connected cells. The parameter α is used to trade regularity for data fidelity. The data and regularization terms are determined with the horizontal structure-slices and wall-slices.

Regularization term The regularization term $R_{i,j}$ is defined to favor a final model with low complexity. The penalty for different labels between adjacent cells is thus set to be proportional to the area of the shared face, see Eq.4.3. $A_{i,j}$ denotes the surface area of the shared face between cells i and j . For scale normalization the area is divided by the area of the horizontal cross section of the bounding box. As observed by [XF12], approaches that penalize the surface area tend to miss thin details such as walls.

This problem is referred to as the *shrinking bias*. Energy formulations solvable by graph-cuts, i.e., equation 4.2, favor a compact set of graph nodes by minimizing the summed edge length of edges between different labeled nodes. They are thus less suitable for labeling thin structures such as blood vessels in medical images. Although there are methods to overcome this problem [VKR08], they are not applicable in our case as they introduce connectivity priors and are not practical due to high computational costs.

Instead for preserving thin structures, we introduce a weight in order to lower the cost of different labels between adjacent cells where permanent structures are expected. The weight is added as a factor to the regularization term while still favoring a low complexity.

The weights between vertically adjacent cells lower the cost where horizontal structures are expected in the vicinity of the face. The presence of horizontal structures is estimated by the presence of points close to the face in the horizontal structure-slices. However, due to noise as well as imprecise cell alignment and varying scanning resolution, taking only the point density close to the face into account is ineffective, see Figure 4.3. Our experiments show that the coverage of the face provides a better solution. We use an occupancy grid to evaluate the point coverage. The grid size is chosen as 2τ , where τ is the size of the occupancy grid filter used for downsampling during line extraction. The weight $\omega_{i,j}$ is then defined via the ratio of occupied grid cells to the total number of grid cells.

The weights of horizontally adjacent cells are determined similarly. The shared face

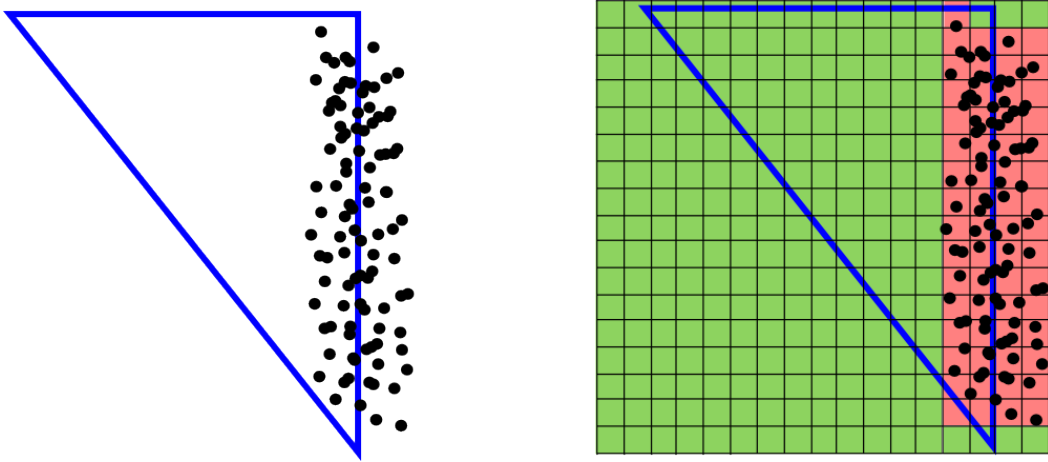


Figure 4.3: **Estimation of point coverage.** Left: Number of points in face is high, but coverage is low. Right: Usage of occupancy grid. Ratio of $\#$ occupied to $\#$ total grid cells provides an approximation of the coverage.

is vertically projected into an edge. The coverage of the edge by the points of the corresponding wall-slice is determined by discretizing the edge into bins of size 2τ . The weight $\omega_{i,j}$ is then defined as the ratio of number of occupied bins over total number of bins. This results into the following regularization term in cells i and j :

$$R_{i,j}(l_i, l_j) = \begin{cases} 0, & l_i = l_j \\ (1 - \omega_{i,j}) \cdot A_{i,j}, & l_i \neq l_j \end{cases} \quad (4.3)$$

However, for thin cells multiple faces might be close to the same points. As each face is generated by one specific line extracted during the line extraction step, only points of clusters fitted to this line are considered for that face. The cost for different labels might thus be zero only when the face or edge is fully covered. Such regularization term matches the condition of submodularity stated by [KZ04], see eqn. 4.4.

$$R_{i,j}(0, 0) + R_{i,j}(1, 1) \leq R_{i,j}(1, 0) + R_{i,j}(0, 1). \quad (4.4)$$

Data term To provide a cost for each combination of cells and labels, we estimate for each cell whether it belongs either to the empty or solid space. If the scanning device position is provided, this can easily be estimated by making use of the *free space* between each scanned point and its associated scanning origin. The problem of identifying inside and outside is recurrent for surface reconstruction [LA13].

Our rationale is that a ray cast from a point has an odd number of intersections with the geometry if the point is in an empty space, and an even number if it is in a solid

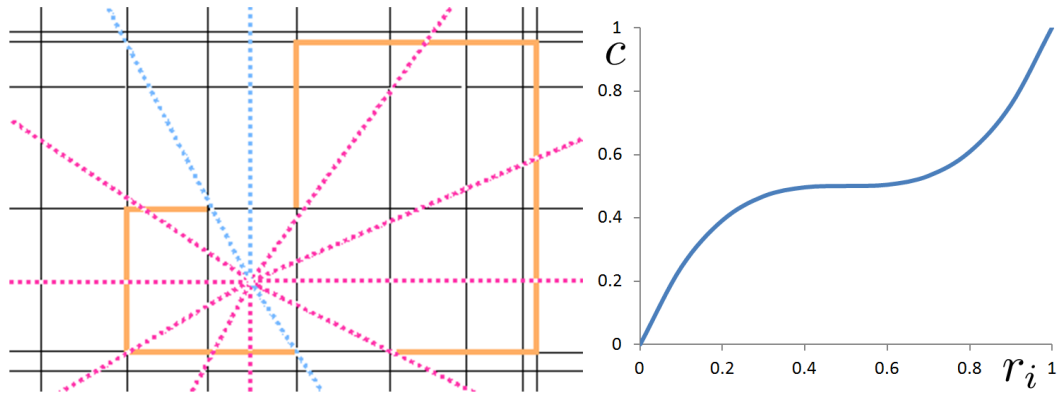


Figure 4.4: **Data term.** Left: Ray-casting to predict empty or solid label for a cell. The cell decomposition is depicted in black and edges with $\omega_{i,j} > \frac{1}{2}$ are depicted in orange. Rays indicating empty, i.e., odd number of intersections, are depicted in pink versus blue. Right: Inverse sigmoid function f turning ratio r_i of rays indicating empty space to total number of rays into c .

space. For counting intersections between a ray and the point cloud one solution would be to define a surface from the point cloud. However, clutter and missing data are likely to interfere and distort the number of intersections. The weights used for the regularization term are used as they are meant for indicating the presence of permanent structure. Rays are thus cast from each cell center and each intersection with a face with an assigned weight $\omega_{i,j} > \frac{1}{2}$ is counted as an intersection with a permanent structure, see Figure 4.4.

To improve stability at the cost of computation time, we shoot a higher number of rays. The ratio of rays r_i for cell i is defined as the ratio of rays with an even number of intersections to the total number of rays. It is mapped with an inverse sigmoid function $f : [0, 1] \rightarrow [0, 1]$, see Figure 4.4. f is chosen so that a value of r_i is mapped conservatively to a cost c . Ratios not clearly indicating either empty or solid space are mapped to the same cost for both labels. In this way the label depends on the regularization term.

Due to the different sizes of cells, larger cells receive a higher penalty from the regularization term as they have a larger surface area. In order to eliminate this bias the cost of the data term is scaled by V_i , defined as the volume of the cell i that, for scale normalization, is divided by the volume of the bounding box. This

leads to the final data term function:

$$D_i(l_i) = \begin{cases} c \cdot V_i, & l_i = 0 \\ (1 - c) \cdot V_i, & l_i = 1 \end{cases} \quad (4.5)$$

After labeling, every cell is marked as either empty or solid space. The final 3D model is extracted as the set of faces between different labeled adjacent cells.

4.4 Experiments

We evaluate our method on a synthetic multi-level dataset, two measurement datasets obtained by two types of laser scanners and one dataset acquired with a Kinect sensor. The algorithm is implemented in C++ using the Computational Geometry Algorithms Library [CGAL15]. For energy minimization we use the Graph Cut Library [BVZ01].

Cory 5th floor. In the measurement dataset¹, the point cloud is sampled on a hallway exhibiting non Manhattan-world geometry, including curved walls and archways. Some data are missing as several ways and rooms have partially been scanned without being entered. The scene also contains many fine details such as doors and tilted windows, and clutter such as couches and curtains. To reduce the memory footprint, the point cloud is downsampled by selecting every other point. The reconstructed model covers the structure of the hallway and partly captures details like doors, see Figure 4.5. The lintels of the doors are not sampled densely enough to appear as peaks during height analysis, and hence are not detected as horizontal structures. The height of the doors is extended to the next significant horizontal structure, here the ceiling, see lower right excerpts in Figure 4.5. The archways are not reconstructed as our method is not suited to reconstruct surfaces that are non-planar, or neither vertical nor horizontal. The circular wall is approximated with planar surfaces and only exhibits minor artifacts, although our method is not designed to detect and reconstruct non-planar walls, see upper right excerpts in Figure 4.5. The reconstruction in sparsely sampled areas, i.e., rooms and side-ways, is incomplete. This is mostly due to the absence of walls, as it impacts both the space partitioning and the energy minimization.

1. <http://www-video.eecs.berkeley.edu/research/indoor/>

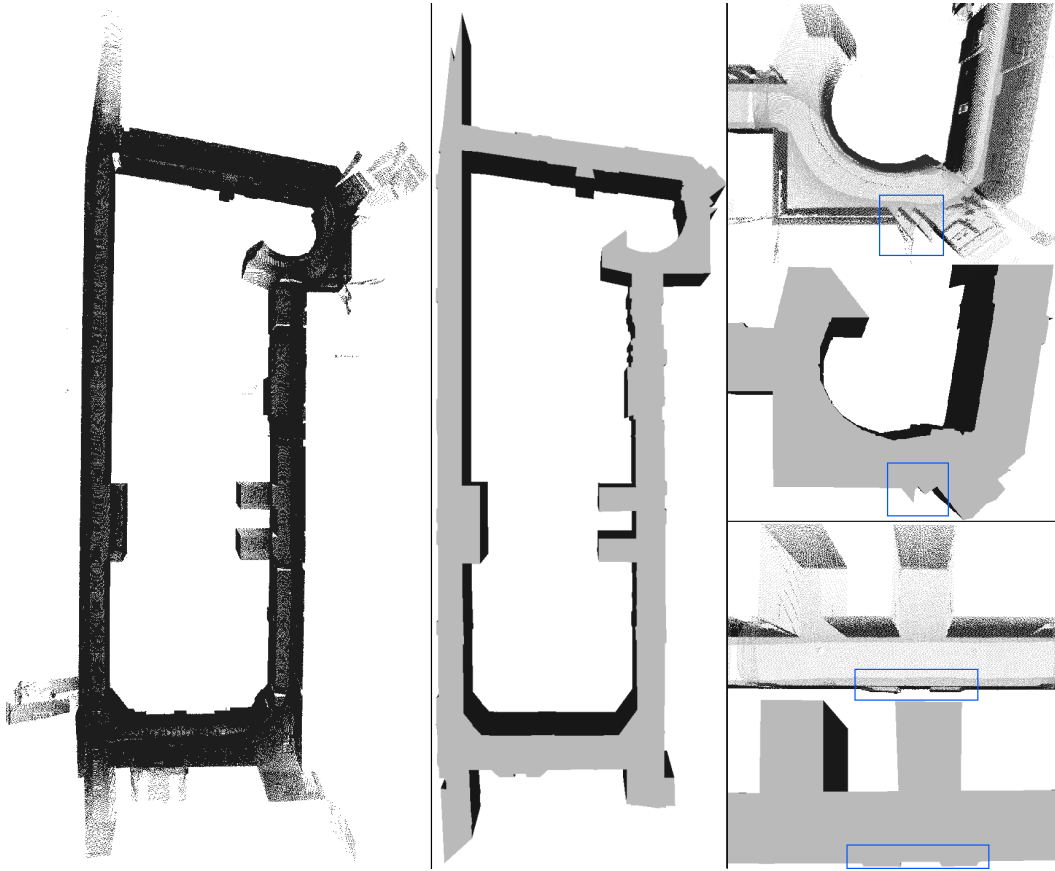


Figure 4.5: **Reconstruction of the Cory 5th floor.** Left: Input point cloud. Middle: Reconstruction of the indoor space. Upper right: Circular wall segment and details in the outer part. Points colored by estimated normals. Lower right: Another excerpt of the input point cloud, sampled on doors in the corridor and on an archway.

Inria Euler building. Using a Leica Scanstation P20, we performed an acquisition in the Euler building at Inria Sophia Antipolis. The datasets consists of 6 registered scans showing several adjacent rooms including the entrance, a conference room, a lecture room and a small corridor. It features walls scanned from both sides, clutter like chairs and tables as well as vegetation captured outside. The input data and the reconstructed model are depicted by Figure 4.6. The thin walls are reconstructed, while the clutter outside does not interfere with the reconstruction.

Synthetic dataset. For evaluation we use as ground truth a synthetic dataset created using Trimble Sketchup. We then use Meshlab for sampling a point cloud

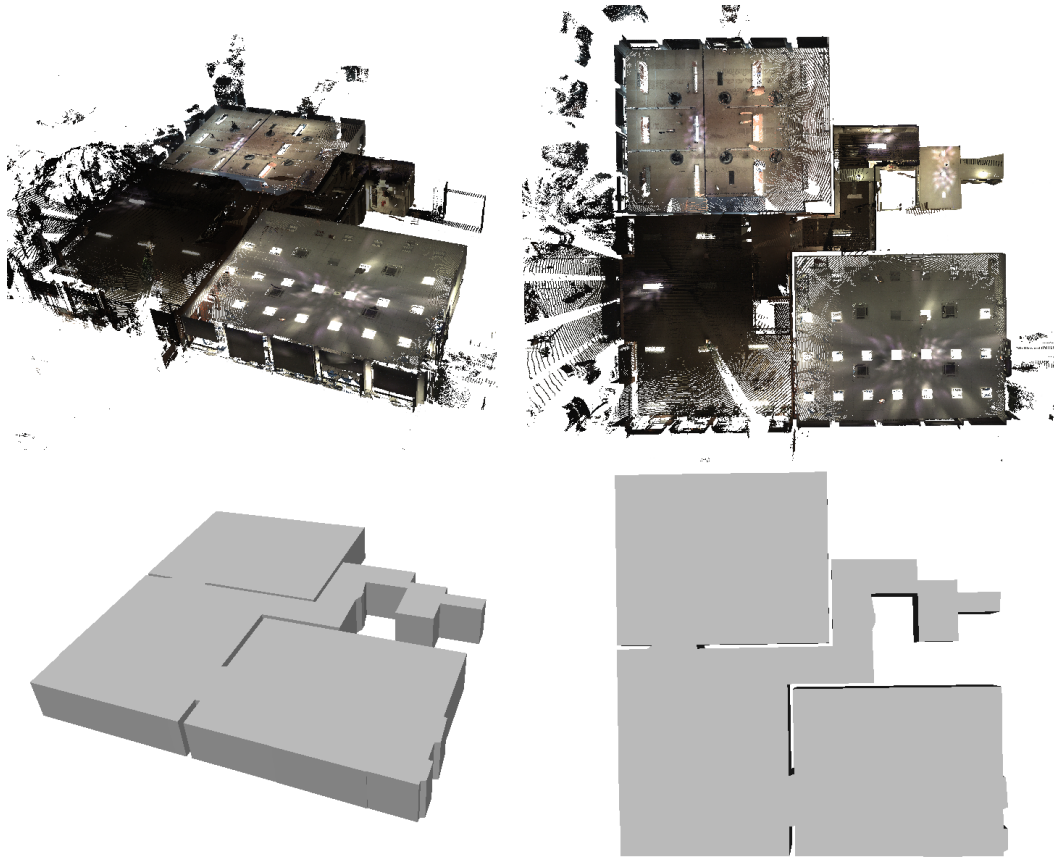


Figure 4.6: **Reconstruction of the Euler building entry area.** Top row: Input point cloud in perspective and top view. Bottom row: Reconstructed model in perspective and top view.

from the exported triangle mesh. A small amount of uniform noise (5 mm) is added after sampling. The ground truth model features a multi-floor, non Manhattan-world scene. The reconstructed model shows that different wall directions are reconstructed accurately, see Figure 4.7. Reconstruction of non-orthogonal wall directions is depicted by the upper two close-ups. The reconstruction of different height levels is shown by the lower two close-ups. The mesh is colored by the Hausdorff distance from the result to the ground truth showing a maximum distance of 2.3 cm. A small deviation in the reconstructed wall directions is due to the global clustering step in the Hough transform space. Such merging incurs a small deviation visible in the two red colored corners of the reconstructed model. The chosen parameters and timings are recorded in Table 4.1.

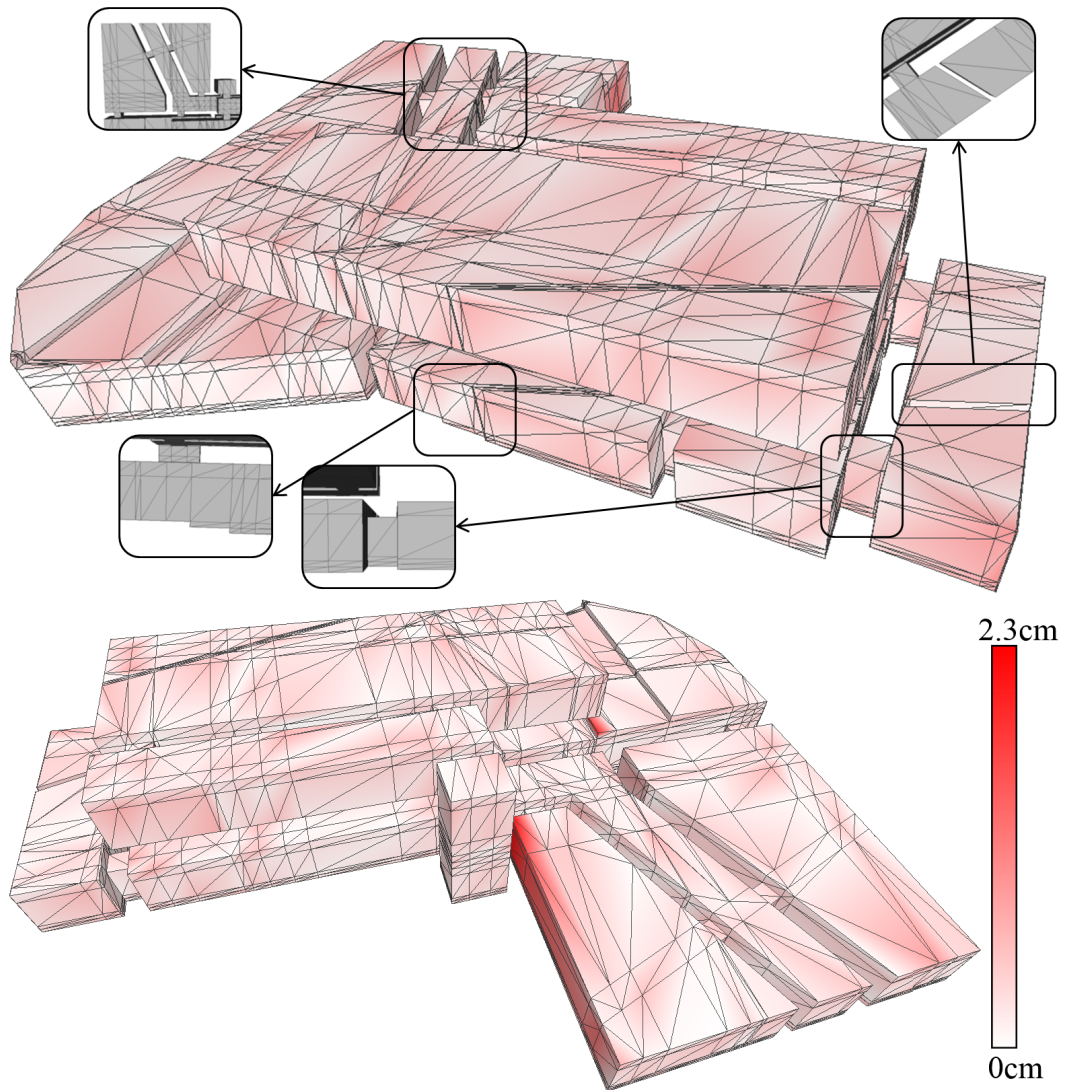


Figure 4.7: **Reconstruction from a synthetic dataset.** Two views of the same reconstruction, colored by the Hausdorff distance from the result to the ground truth.

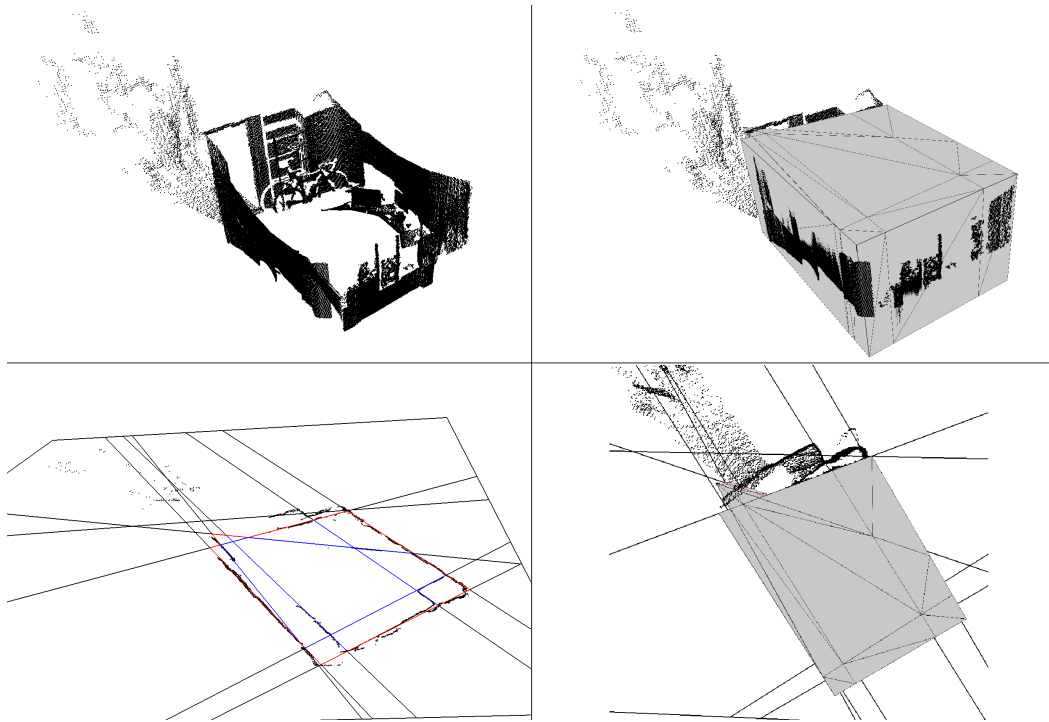


Figure 4.8: **Reconstruction of Kinect-recorded indoor scene.** Upper left: Input point cloud featuring an office room. Upper & lower right: Reconstructed model. Lower left: The labeled cell decomposition exhibits a good alignment of the room with the exception of the occluded corner.

Kinect dataset. Point clouds acquired by the consumer-grade Kinect sensor are challenging our algorithm as they contain many defects [KE12]. The scene exhibits a single office room containing clutter such as a desk, some cupboards and a bicycle. As the scan covers only a small part of the floor and no ceiling at all, no significant peaks show up in the horizontal distribution and the scene is processed as a single level. Compared to commercial-grade laser scanners the high level of noise requires a high tolerance value ($\varepsilon = 6$ cm) for clustering of the wall directions. The input data and reconstructed model are shown by Figure 4.8. Most of the room is correctly reconstructed except a corner entirely occluded by a bicycle and a cupboard. The corridor is not reconstructed due to very sparse sampling.

Parameters. The parameters have a direct impact on the quality and level of detail of the reconstructed model. The bin size for the horizontal slicing affects the detection of horizontal structures. It should be larger than the typical amount of noise, but still allow for several bins between horizontal structures in the height his-

togram. In general a value of 5-10 cm is appropriate. However, for scenes containing just ceilings and floors the method is stable to the choice of the bin size.

τ relates to the point density in the scanned data and is used for the downsampling. A high value of τ improves the reconstruction of sparsely sampled regions, while removing detail from highly sampled regions due to the downsampling. Technically, τ should be chosen as the minimal sampling resolution, i.e., largest distance between neighbored points in the area of interest. In usual cases this parameter is selected in a range between 0.75 and 4 cm.

ε is used for the multi-scale line fitting and clustering of extracted wall directions. It indirectly relates to the number of cells generated in the space partitioning. High values of ε lead to reconstructed models with low levels of detail (Figure 4.9), in short computation times, especially for model extraction. A value of at least twice the range of scanner noise is appropriate.

The Hough transform is used to cluster the detected wall segments. The effect of the chosen resolution is depicted in Figure 2.4. As the clustering is also restricted by the choice of ε , a coarser resolution of the Hough accumulator is used as a starting point for parameterization: $2^\circ \cdot \tau$.

α is used to trade data fidelity for regularity in the energy minimization formulation. For densely scanned data, a low value of α enforces fidelity to the input data. For incomplete data a high value of α provides regularity by filling gaps but may over-simplify the model (Figure 4.9). Note that α is only used for the last step, i.e., labeling. An adjustment and extraction of a new result requires only few seconds of computation time.

Accuracy. We evaluate the accuracy of our method by comparing the reconstructed model to the synthetic multi-level dataset used as ground truth. Figure 4.7 illustrates that different wall directions are reconstructed accurately: the symmetric Hausdorff distance is 2.6 cm and the one-sided Hausdorff distance from the result to the ground truth is 2.3 cm (Figure 4.7).

Performances. Running times and parameters are provided in Table 4.1. Timings are measured on an Intel Core i7 920 with 16 GBs RAM. The time spent to estimate the normals is omitted. The most time consuming steps of the algorithm are the stochastic ray-casting to determine the data term and the multi-scale line fitting. A kD-Trees is used to speed up the calculation of the edge weights for the

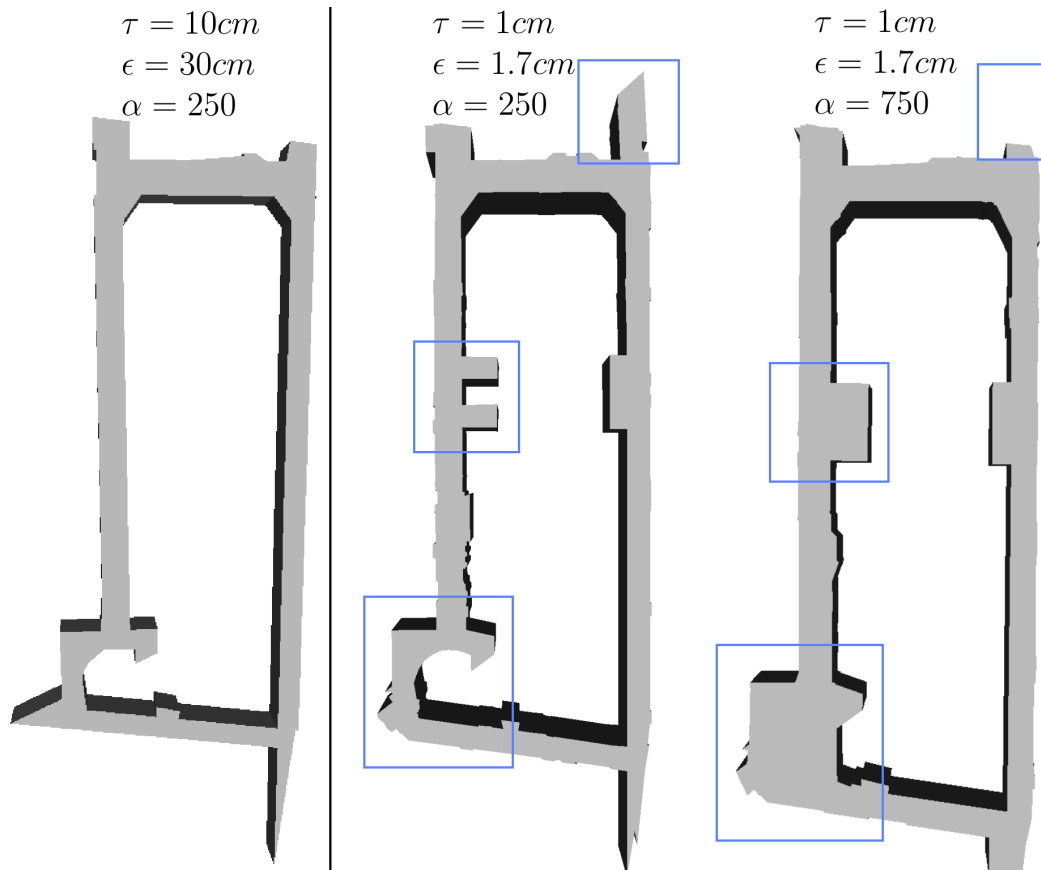


Figure 4.9: **Impact of parameters.** Left: Reconstruction of the Cory 5th floor with $\tau = 10\text{ cm}$ and $\epsilon = 30\text{ cm}$. Such low level of detail requires 7.2 s. Middle & right: Respectively 250 and 750 for α . The higher regularization $\alpha = 750$ leads to a lower level of detail through area minimization.

	Synthetic	<i>Cory 5th floor</i>	<i>Euler</i>	Kinect
#points	1,000,000	4,391,604	2,305,938	86,694
Vertical distribution - bin size	1 cm	8 cm	8 cm	x
τ	2 cm	1 cm	0.75 cm	1 cm
ε	2.5 cm	1.7 cm	1 cm	6 cm
Hough res: <i>angle · distance</i>	$1^\circ \cdot 0.25 \text{ cm}$	$1^\circ \cdot 0.8 \text{ cm}$	$1^\circ \cdot 0.8 \text{ cm}$	$3^\circ \cdot 5 \text{ cm}$
α	20	250	250	250
Spatial partitioning	63 s	192 s	36 s	6.9 s
Model extraction	58 s	206 s	66 s	15.2 s

Table 4.1: **Running times.** Chosen parameters and running times (single thread). Timings for model extraction include the ray-casting performed to compute the data term.

regularization term (see section *Regularization Term*).

Robustness. We first evaluate the robustness of our approach against sparse sampling with downsampled instances of the *Cory 5th floor* (50%, 20%, 10% and 5% of the original point cloud, generated by selecting every 2nd, 5th, 10th or 20th point of the original point cloud). The parameters set for reconstruction are adapted to each downsampled version. As the point density is lowered by downsampling the parameter τ used for adjusting the grid size during line extraction is increased. The tolerance ε used for clustering is also increased, as line fitting is less accurate on sparse point clouds. While small values for τ and ε allows for a detailed reconstruction of high resolution scans the method tends to generate more bumpy artifacts. Higher values of τ and ε may miss details, but provide lower computation times and more regular reconstructions. Timings and parameters for the lower resolution datasets are provided in Table 4.2.

The reconstructions from the downsampled instances are shown in Figure 4.10. The reconstruction of the 50% dataset provides the same amount of details as the original dataset. In the lower resolution datasets the connected rooms and sideways are inaccurately recovered or not at all, as they are even sparsely scanned in the original mesh. The recovery of details for the lower resolution datasets is depicted in the right column. Note that the doors and niches in the corridor are extended to the height of the ceiling.

Robustness to noise is evaluated by adding noise to the *Cory 5th floor* dataset. More specifically, each point of the input point cloud is displaced along a random

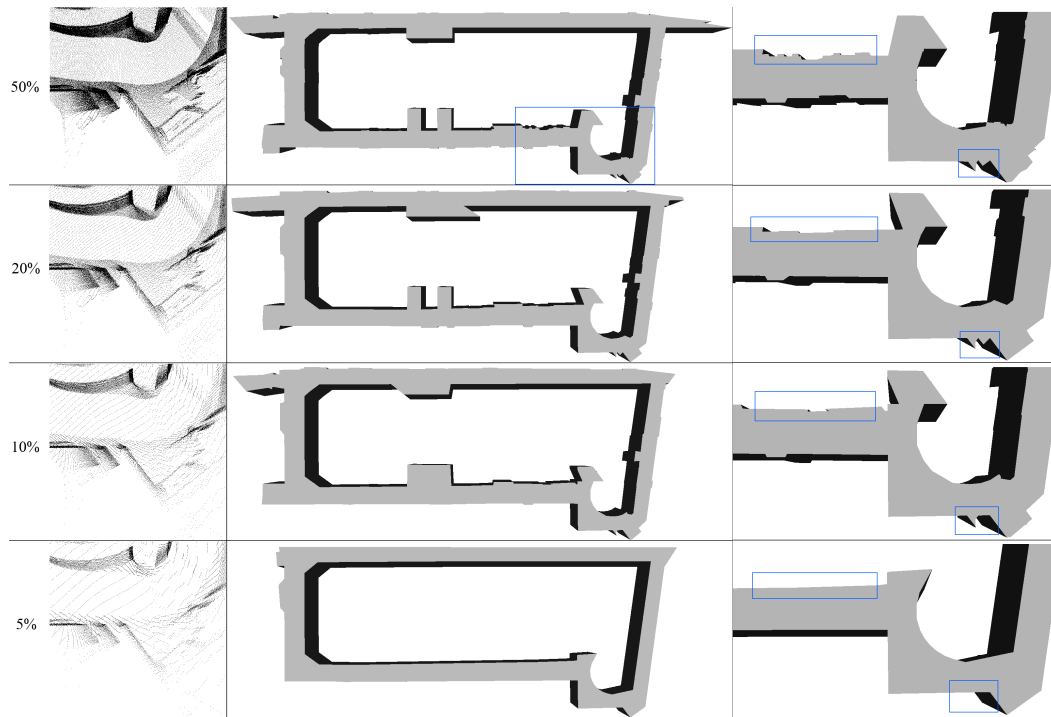


Figure 4.10: **Reconstruction at lower resolutions.** Each row of the image shows the reconstruction of one downsampled dataset (from top to bottom): 50%, 20%, 10% and 5%. Left column: Excerpt of the input data. Left-mid column: Overview of the reconstructed model. Right column: Close-ups showing the level of details recovered.

	50%	20%	10%	5%
#points	4.391.604	1.756.642	878.321	439.161
τ	1 cm	2 cm	2 cm	4 cm
ε	1.7 cm	3cm	3.5 cm	5 cm
Spatial partitioning	192 s	70 s	58,8 s	16.8 s
Model extraction	206 s	68,4 s	58,5 s	12.1 s

Table 4.2: **Parameters and running times of lower resolution datasets.** The bin size for the vertical distribution (0.1), $\alpha = 250$ and the resolution for the Hough Accumulator $1^\circ \cdot 0.8cm$ do not vary.

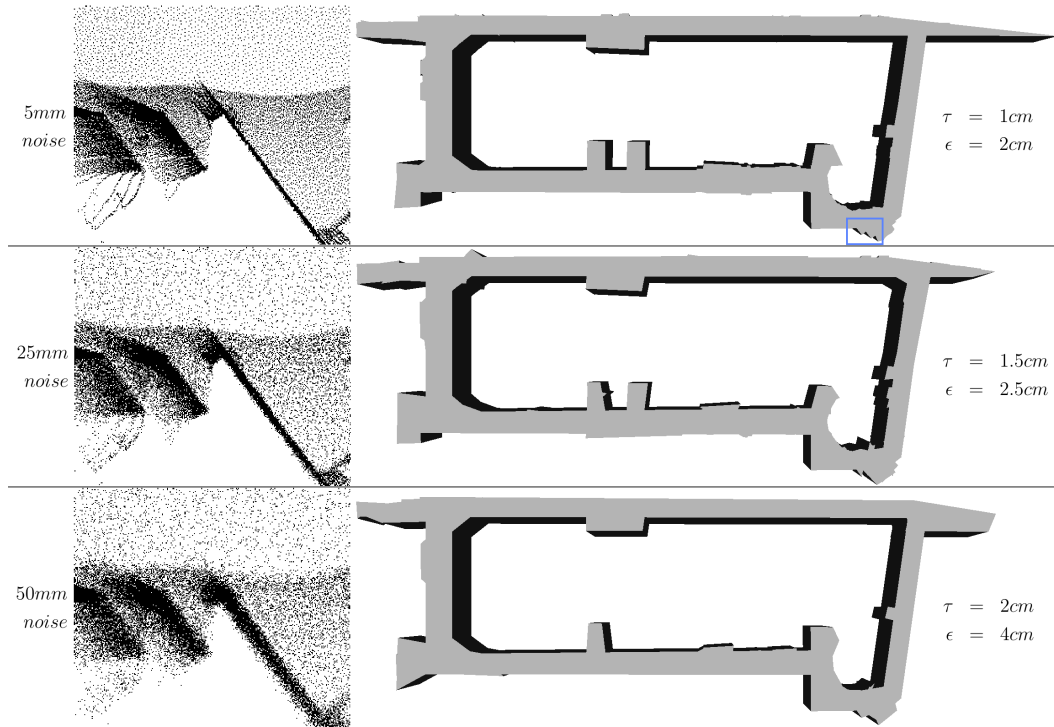


Figure 4.11: **Robustness to noise.** Reconstruction of the *Cory 5th floor* with added noise. Left: Closeups on the outlined rectangle. Right: Reconstructions and corresponding parameters.

vector, with maximum length set to 5 mm, 25 mm and 50 mm. The noise is thus added to the noise of the measurement device, estimated between 3 and 5 mm for common laser scanners. Figure 4.11 illustrates that noise impacts the level of detail of the reconstruction without altering the coarse structures. A small amount of noise (5mm, the usual range of laser scanners) has minor impact on the small details. With higher amount of noise less details are reconstructed but the coarse structure of the architecture is correctly recovered, except for sparsely scanned areas such as the outer rooms and the two hallways. As the accuracy of the line fitting process is impacted by noise, the parameters τ and ϵ are adjusted to the level of noise. The bin size for the vertical distribution (0.1), $\alpha = 250$ and the resolution for the Hough Accumulator $1^\circ \cdot 0.8 \text{ cm}$ are constant.

Robustness to outliers is evaluated by adding random points uniformly distributed within the bounding box of the *Cory 5th floor* dataset: respectively 1% and 5% outliers. The reconstruction of those modified datasets are shown in Figure 4.12. The corridor is reconstructed correctly despite the presence of outliers in both

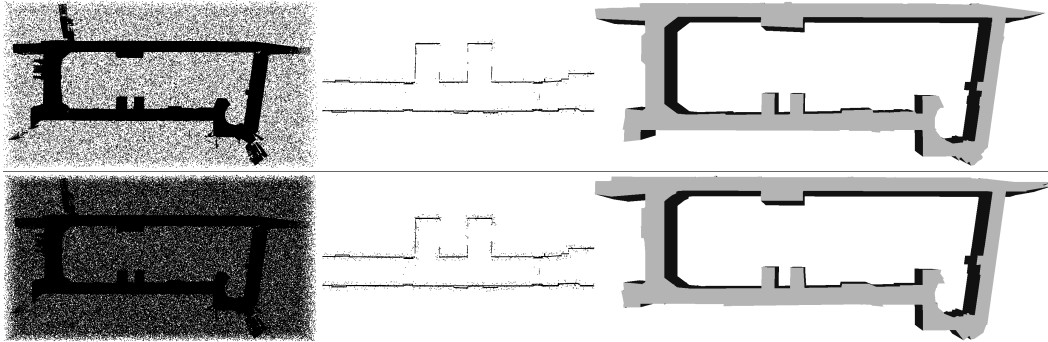


Figure 4.12: **Robustness to outliers.** Reconstruction of the *Cory 5th floor* with added outliers. Top: 1% outliers. Bottom: 5% outliers. Left: Point cloud with outliers. Middle: Wall-slice generated by horizontal slicing with only some added points close to the walls remaining. Right: Reconstructed model of the indoor space.

datasets. However, the amount of recovered details is lowered and the circular area exhibits minor geometric inaccuracies. The horizontal slicing step is effective at removing outliers. A major part of the outliers is filtered by their normal during the horizontal slicing step described in Section 3.1. Some outliers close to the original points remain and lower the accuracy of the reconstruction, mostly in sparsely scanned areas or small details.

Limitations. Our algorithm is designed to handle arbitrary vertical wall directions. For cases of partially scanned rooms, where some walls are not sampled at all, our method relies on other wall directions detected in the scene or on the borders of partially scanned parts. These cases may result in improper data consolidation, while algorithms strictly restricted to Manhattan-world scenes often yield more plausible reconstructions. Note however that other approaches fail on these cases [JHS09, XF12].

Global clustering in a Hough transform space performs a regularization as the Hough Accumulator determines the main wall directions and aligns the extracted lines to these directions. We notice, however, that such regularization may hamper the reconstruction of fine details (Figure 4.5) and requires parameter adjustments. The latter trial-and-error process is time consuming and suggests to investigate automatic parameter selection.

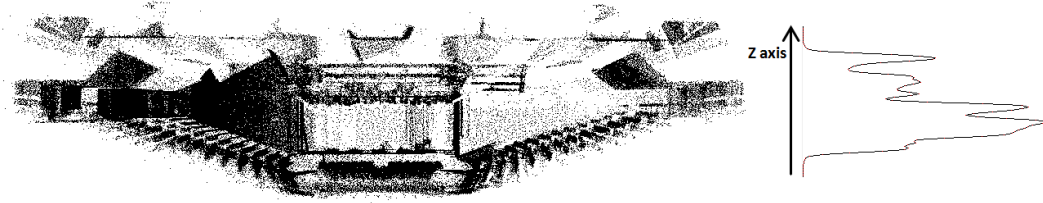


Figure 4.13: **Failure case with stepped floors.** Side-view of the auditorium at the Delft University of Technology. The vertical distribution is shown right. Due to the clutter the lowest horizontal part of the floor is not distinguishable from the stepped floor and clutter.

The algorithmic complexity of the ray casting is quadratic in the number of cells of the space decomposition, this number being itself related to the level of detail adjusted through a tolerant or selective line clustering. The computation times thus increase rapidly with the level of detail sought after.

Locating the splitting heights for the point cloud (Section 3.1) depends on the amount of points detected on the horizontal structures. For large scenes such horizontal structures require a very large number of sample points. The door lintels in the *Cory 5th floor*, for instance, are sparsely sampled and therefore the height of doors is extended to the ceiling (Figure 4.5, lower right).

Finally, some buildings contain non-vertical walls, non-horizontal floors and ceilings, stepped floors and ceilings, and even non-planar structures. One of such examples with stepped floors and non-horizontal ceilings is the auditorium of the Delft University of Technology (Figure 4.13). Our approach fails on such cases with stepped floors and large amount of clutter.

4.5 Summary

We proposed a new method for indoor scene reconstruction. Through detecting the permanent structures in a cluttered scene we reconstruct a model of the indoor space with satisfactory tradeoff between accuracy and low complexity. We label the cells of a 3D space partitioning in order to reconstruct a watertight model consolidating missing data by minimizing an energy formulation via min-cut. Our experiments show that our method is moderately robust to noise and outliers, and generates satisfactory results from data measured with Kinect sensors.

Conclusion

In this thesis we investigated novel methods for indoor scene modeling from measured point data. Acquiring indoor scenes yields dense point clouds of sampled locations from the surrounding objects and structures. We explored the geometric modeling and semantization of indoor spaces from planar shapes. Abstracting the input data by planar shapes is our means to lower the complexity and to gain robustness to noise and outliers.

Contributions

The directions explored during this PhD thesis has led to the following contributions:

- **Pipeline for indoor scene reconstruction:** We proposed a fully automated pipeline for geometric modeling of indoor scenes from acquired point data. We used statistical methods to process the input data containing a multi-level building without a priori knowledge about the levels. We performed a multi-scale, feature-preserving approach for detecting wall segments, followed by global clustering in a Hough transform space in order to extract a detailed linear approximation of the wall geometry.
- **Shape detection with regularization:** We introduced a planar shape detection method that both detects and reinforces common relationships within man-made objects to achieve high accuracy and robustness against defect-laden data. The abstraction of complex point data by primitive shapes yields a complexity reduction and leads to practical algorithms. Primitive shapes are commonly used as building blocks for other methods in indoor and urban modeling, such as surface reconstruction. High fidelity to the input point data is crucial as any introduced error is propagated to the final result. We designed the method for implementation on GPU to achieve fast processing suitable to the preprocessing of millions of points within seconds.

- **Object classification via planar shapes:** We introduced an object classification method based on global features extracted from a set of planar shapes detected from point data. Extracting global features from the relationships between shapes at multiple scales helps distinguishing between typical indoor objects such as different types of furniture and tabletop items. Compared to point-based features these global features better generalize from object instances to object classes. This approach offers several added values in terms of robustness, orientation and scale invariance and computational complexity.
- **Watertight surface reconstruction:** We proposed a primitive driven space partitioning to ensure a good alignment with permanent structures. A watertight geometric model is generated through labeling the cells of a space partitioning, with satisfactory tradeoff between accuracy and low complexity. The labeling of the space partitioning was stated as an energy formulation and solved via min-cut to gain robustness to missing data and false detected primitives. Although such approach was not entirely novel and already used in urban modeling, we introduced the use of graph-cut in indoor modeling by introducing graph edge weights. A stochastic occupancy estimation allows to be independent from the scanning origins and thus our method is more general than most previous methods.

Limitations

The assumptions and technical choices made for the methods contributed in this thesis also have the following limitations:

- **Defect-laden data:** Our methods were mostly tested and developed with LIDAR data, and only small scenes acquired via Kinect or multi-view stereo were reconstructed. Although the methods are robust to defect-laden data to some degree, large scenes from defect-laden data are difficult to process. The abstraction of the input data by primitives provides robustness against noise and outliers. Robustness to a moderate amount of missing data is gained through the stochastic occupancy estimation and a global energy formulation. For highly occluded scenes the results deteriorate. Our shape detection method is able to reconstruct the scene recorded with

a Kinect sensor by applying a strong regularization. However, due to the strong regularization only shapes aligned with the permanent structures are detected. This results in a lower coverage than other methods that, however, fail at recovering the permanent structures with correct alignment.

- **Scalability:** Using a space decomposition for reconstruction yields a watertight volume. However, the complexity of the space decomposition does not scale well with large scenes. A high number of splitting planes results in a high fragmentation of space. This leads to a high computational complexity and has negative impact on the reconstruction result. A high fragmentation of the space decomposition is departing from the initial idea of using the decomposition to restrict the solution space to feasible geometric models. In our method we choose to extend only wall segments of a certain length to planes, and restrict small segments to partition the space only locally. This helps limiting the complexity, but is a heuristic.

Our method for shape detection and regularization performs region growing and thus requires neighborhood linkage between points. We precompute all k-nearest neighbors and store the indices of the neighbored points for each point. The computation of the k-nearest neighbors takes a major part of the total processing time. However, the memory consumption due to the stored indices limits the maximum size of input data. Depending on chosen parameters our method can process large but not massive scenes, i.e. around 8M points per GB of GPU memory.

- **Planarity assumption:** Assuming planarity for permanent structures greatly simplifies the reconstruction process. For geometric modeling of indoor spaces we assume that large horizontal or vertical planar shapes belong to structures. The minor amount of false detected shapes, such as doors or closets, does not hamper the labeling process. However, curved walls can only be correctly modeled if they are vertical and exhibit a low curvature. Although curved or freeform structures are rare in residential homes they are getting more common in public or commercial buildings.

Perspective

The state of the art in geometric modeling of indoor scenes has evolved in the last years, in terms of details and size of the data sets. While there are some methods [BdLGM14, ZK13] for detailed modeling of small scenes and some methods [MMV⁺14] for modeling large scenes with a lower amount of details, there is still no method for robust and detailed large scale reconstruction. Albeit more research is required, the current methods yield to for practical applications.

Collaborative semantization and mapping. In recent years a few new affordable technologies have arisen for real-time acquisition of depth images. Based on such technologies a range of products have been announced that will allow consumers to record depth and color information. Augmented reality, one of the technologies that strongly benefit from real-time depth images, requires an understanding of the surrounding environment for manipulating it. Having many consumers capturing depth and color data by, for instance, wearing augmented reality glasses, opens a wide range of possibilities for using community data.

Augmented reality requires a semantization of the scene for interacting with it. Providing a comprehensive object classifier is a notoriously difficult and labor-intensive task as it requires consistent training. Allowing for a continuous training by involving the user provides a large amount of training data while adapting to the needs of users. Observing the user wearing glasses while manipulating objects would allow for analyzing the interactions to infer the use of objects.

A second direction to explore is collaborative mapping. Similar to the use of community data in image processing [AFS⁺11], combining the data collected by many users would allow to create larger and widespread models. However, this is an enduring challenge. Contrary to image based community data, recorded depth and image data are huge and cannot be easily shared. Generating a local geometric model allows for compact representation. The methodological challenge is to merge several local geometric models into a global model. Acquired at different points in time the featured scene might have changed.

Mixing modalities The current methodologies mainly focus on processing a certain amount of homogeneous point data, i.e., data acquired by the same device. However, most acquisition technologies either record color information directly, such as RGB-D cameras or kinect, or as an additional option, such as LIDAR scanners.

Multi-view stereo is entirely color-based and estimates depth from keypoints visible from different view points. However, this color information is rarely used, albeit many works on image processing aim at recovering the scene geometry from single images [FMMR10, HHF09, WGK10]. Although one might think that the gain on geometric information is lower, the color information helps segmenting the point data and provides semantic information that not only helps distinguishing clutter from structure, but also enhances the classification of objects.

Joined facade reconstruction. Reconstructing the indoor scene is often challenged by occlusions and hidden geometry. Providing a plausible geometric model for small parts of structure hidden by furniture, can often be provided by prolonging adjacent structures. This problem gets increasingly more difficult as the hidden area gets larger, especially if no wall segments can be extended to bound the area. An outer bound can be established by joining an indoor and outdoor reconstruction. Acquiring the interior of a building might require several hours. The complicated geometry and high occlusion requires many scanning positions to cover the scene. Acquiring the facade of a building, however, is often done with a handful of scans as the occlusion is rather low outside. Instead of extending wall segments or following known directions, the problem turns into room layout partitioning as the occupied space is known. Detected windows in the facade provide further information about the actual floor plan.

Repetition. In Chapter 2 we considered certain relationships between planar shapes, i.e. parallelism, orthogonality and coplanarity, for detection. In actual architecture there are more regularities as equal spacing, i.e., wall thickness, corridor width, etc. and repeating elements. Detecting the wall thickness and other regularities like door or windows sizes helps to provide a coherent model and plausible reconstructions in hidden parts. In urban modeling it is common to use grammars for describing the repetitive pattern of facades. An application of this grammar to the indoor domain seems difficult. However, detecting repeating patterns within the input data allows to combine those parts to complete occlusions and enrich the level of detail by gaining a higher resolution.

Bibliography

- [AFS⁺11] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, October 2011. (Cited on page 96.)
- [AH11] Antonio Adan and Daniel Huber. 3d reconstruction of interior wall surfaces under occlusion and clutter. In *3DIMPVT*, 2011. (Cited on pages 17 and 21.)
- [Ale12] Luis A Alexandre. 3d descriptors for object and category recognition: a comparative evaluation. In *Workshop on Color-Depth Camera Fusion in Robotics at IROS*, 2012. (Cited on pages 14 and 22.)
- [AMT⁺12] A. Aldoma, Zoltan-Csaba Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R.B. Rusu, S. Gedikli, and M. Vincze. Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. *Robotics Automation Magazine, IEEE*, 19(3):80–91, Sept 2012. (Cited on page 14.)
- [AS98] Pankaj K. Agarwal and Micha Sharir. Arrangements and their applications. In *Handbook of Computational Geometry*, pages 49–119. Elsevier Science Publishers B.V. North-Holland, 1998. (Cited on page 75.)
- [BB09] Angela Budroni and Jan Boehm. Toward automatic reconstruction of interiors from laser data. In *3D ARCH*, 2009. (Cited on page 19.)
- [BB10] Angela Budroni and Jan Boehm. Automated 3D Reconstruction of Interiors from Point Clouds. *IJCV*, 8:55–73, 2010. (Cited on page 19.)
- [BdLGM14] Alexandre Boulch, Martin de La Gorce, and Renaud Marlet. Piecewise-planar 3d reconstruction with edge and corner regularization. *SGP*, 2014. (Cited on pages 17, 18, 21, 23, 73 and 96.)
- [BELN11] Dorit Borrmann, Jan Elseberg, Kai Lingemann, and Andreas Nuechter. The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2), 2011. (Cited on page 12.)

- [ben04] The princeton shape benchmark. In *SMI*, 2004. (Cited on pages 66 and 68.)
- [BLN⁺13] Matthew Berger, Joshua A. Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T. Silva. A benchmark for surface reconstruction. *SIGGRAPH*, 2013. (Cited on page 43.)
- [BM12] Alexandre Boulch and Renaud Marlet. Fast and robust normal estimation for point clouds with sharp features. *SGP*, 2012. (Cited on pages 3 and 12.)
- [Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. (Cited on pages 65 and 66.)
- [BSMW14] Ben Bellekens, Vincent Spruyt, and Rafael Berkvens Maarten Weyn. A survey of rigid 3d pointcloud registration algorithms. In *AMBIENT*, 2014. (Cited on page 4.)
- [BTS⁺14] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Joshua A. Levine, Andrei Sharf, and Claudio Silva. State of the art in surface reconstruction from point clouds. *Eurographics*, 2014. (Cited on page 16.)
- [BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, November 2001. (Cited on pages 75 and 80.)
- [CGAL15] CGAL. Computational Geometry Algorithms Library, 2015. <http://www.cgal.org>. (Cited on pages 66 and 80.)
- [Che95] Yizong Cheng. Mean shift, mode seeking, and clustering. *PAMI*, 17(8):790–799, 1995. (Cited on pages 12 and 57.)
- [CLB04] Chao Chen, Andy Liaw, and Leo Breiman. Using Random Forest to Learn Imbalanced Data. Technical report, University of Berkeley, 2004. (Cited on page 66.)
- [CLP10] A.-L. Chauve, P. Labatut, and J.-P. Pons. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In *CVPR*, 2010. (Cited on page 22.)

- [Dav05] E. R. Davies. *Computer and Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers Inc., 2005. (Cited on pages 12 and 29.)
- [dWudLM15] Akademie der Wissenschaften und der Literatur | Mainz. Inschriften im Bezugssystem des Raumes. <http://www.spatialhumanities.de/ibr/technologie/terrestrisches-laserscanning.html>, 2015. [Online; accessed 04-Feb-2015]. (Cited on page 5.)
- [EKS] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559. (Cited on page 18.)
- [FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. (Cited on page 10.)
- [FCODS08] Hongbo Fu, Daniel Cohen-Or, Gideon Dror, and Alla Sheffer. Upright orientation of man-made objects. *SIGGRAPH*, 2008. (Cited on page 22.)
- [FCSS09a] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Manhattan-world stereo. In *CVPR*, 2009. (Cited on pages 20 and 21.)
- [FCSS09b] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Reconstructing building interiors from images. In *ICCV*, 2009. (Cited on page 21.)
- [FMMR10] Alex Flint, Christopher Mei, David Murray, and Ian Reid. A dynamic programming approach to reconstructing building interiors. In *ECCV*, 2010. (Cited on page 97.)
- [GH97] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH*, 1997. (Cited on page 17.)
- [GKF09] Aleksey Golovinskiy, Vladimir G. Kim, and Thomas Funkhouser. Shape-based recognition of 3D point clouds in urban environments. In *ICCV*, 2009. (Cited on page 15.)

- [GMRFM14] Tawsif Gokhool, Maxime Meilland, Patrick Rives, and Eduardo Fernández-Moral. A dense map building approach from spherical rgb-d images. In *VISAPP*, 2014. (Cited on page 47.)
- [HB12] Dirk Holz and Sven Behnke. Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. In *IAS 12*, volume 194, pages 61–73, 2012. (Cited on page 12.)
- [HDD⁺92] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH*, 1992. (Cited on page 3.)
- [HHF09] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. (Cited on page 97.)
- [Hou62] P. V. C. Hough. A method and means for recognizing complex patterns, 1962. U.S. Patent No. 3,069,654. (Cited on page 12.)
- [HWG⁺13] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang. Edge-aware point set resampling. *SIGGRAPH*, 2013. (Cited on pages 28 and 29.)
- [IKH⁺11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinect-fusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *UIST*, 2011. (Cited on pages 6, 20 and 73.)
- [JHS09] Philipp Jenke, Benjamin Huhle, and Wolfgang Straßer. Statistical reconstruction of indoor scenes. In *WSCG*, 2009. (Cited on pages 19, 21 and 90.)
- [Joh97] Andrew Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997. (Cited on page 14.)
- [Kar12] Tero Karras. Maximizing parallelism in the construction of bvhs, octrees, and k-d trees. In *Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics*, pages 33–37, 2012. (Cited on page 41.)

- [KE12] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012. (Cited on pages 6 and 84.)
- [KMYG12] Young Min Kim, Niloy J. Mitra, Dong-Ming Yan, and Leonidas Guibas. Acquiring 3D indoor environments with variability and repetition. *SIGGRAPH*, 2012. (Cited on pages 14 and 22.)
- [KZ04] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts. *PAMI*, 26:65–81, 2004. (Cited on page 78.)
- [LA13] Florent Lafarge and Pierre Alliez. Surface reconstruction through point set structuring. In *Eurographics*, 2013. (Cited on page 78.)
- [LM] Florent Lafarge and Clement Mallet. Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. *IJCV*, 99(1):69–85. (Cited on page 45.)
- [Low99] D.G. Lowe. Object recognition from local scale-invariant features. 2:1150–1157, 1999. (Cited on pages 14 and 63.)
- [Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. (Cited on page 14.)
- [LWC⁺11] Yangyan Li, Xiaokun Wu, Yiorgos Chrysanthou, Andrei Sharf, Daniel Cohen-Or, and Niloy J. Mitra. Globfit: Consistently fitting primitives by discovering global relations. *SIGGRAPH*, 2011. (Cited on pages 11, 22 and 45.)
- [MG11] Duane Merrill and Andrew Grimshaw. High performance and scalable radix sorting: A case study of implementing dynamic parallelism for GPU computing. *Parallel Processing Letters*, 21(02):245–272, 2011. (Cited on page 41.)
- [MMJ⁺13] Claudio Mura, Oliver Mattausch, Alberto Villanueva Jaspe, Enrico Gobbetti, and Renato Pajarola. Robust reconstruction of interior building structures with multiple rooms under clutter and occlusions. In *Proceedings IEEE Conference on Computer-Aided Design and Computer Graphics*, pages 52–59, 2013. (Cited on pages 19 and 21.)

- [MMV⁺14] Claudio Mura, Oliver Mattausch, Alberto Jaspe Villanueva, Enrico Gobbetti, and Renato Pajarola. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Computers & Graphics*, 44:20 – 32, 2014. (Cited on pages 19, 21, 23, 31, 73 and 96.)
- [MPM⁺14] Oliver Mattausch, Daniele Panozzo, Claudio Mura, Olga Sorkine-Hornung, and Renato Pajarola. Object detection and classification from large-scale cluttered indoor scans. *Eurographics*, 2014. (Cited on pages 15, 22 and 61.)
- [MWA⁺13] Przemyslaw Musialski, Peter Wonka, Daniel G. Aliaga, Michael Wimmer, Luc van Gool, and Werner Purgathofer. A Survey of Urban Reconstruction. *Eurographics*, 2013. (Cited on page 21.)
- [NIH⁺11] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. (Cited on pages 6, 20 and 73.)
- [NXS12] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *SIGGRAPH Asia*, 2012. (Cited on pages 14 and 22.)
- [OLA14] Sven Oesau, Florent Lafarge, and Pierre Alliez. Indoor Scene Reconstruction using Feature Sensitive Primitive Extraction and Graph-cut. *ISPRS*, 90:68–82, 2014. (Cited on pages 23 and 31.)
- [OVW⁺14] Sebastian Ochmann, Richard Vock, Raoul Wessel, Martin Tamke, and Reinhard Klein. Automatic generation of structural building descriptions from 3d point cloud scans. In *GRAPP*, January 2014. (Cited on page 13.)
- [OXAH10] Brian E Okorn, Xuehan Xiong, Burcu Akinci, and Daniel Huber. Toward automated modeling of floor plans. In *3D DVPT*, 2010. (Cited on pages 16, 17, 31 and 56.)
- [PCYS12] Trung-Thanh Pham, Tat-Jun Chin, Jin Yu, and David Suter. The

- random cluster model for robust geometric fitting. In *CVPR*, 2012. (Cited on page 12.)
- [PF01] Valerio Pascucci and Randall J. Frank. Global static indexing for real-time exploration of very large regular grids. In *ICS*, 2001. (Cited on page 41.)
- [QZN14] Rongqi Qiu, Qian-Yi Zhou, and Ulrich Neumann. Pipe-run extraction and reconstruction from point clouds. In *ECCV*, 2014. (Cited on page 12.)
- [RBB09] R.B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, May 2009. (Cited on page 14.)
- [RMBB08] R.B. Rusu, Z.C. Marton, N. Blodow, and M. Beetz. Learning informative point classes for the acquisition of object model maps. In *ICARCV*, Dec 2008. (Cited on page 14.)
- [RvDHFV06] T Rabbani, F van Den Heuvel, and G Vosselman. Segmentation of point clouds using smoothness constraint. *ISPRS*, 36(5):248–253, 2006. (Cited on page 12.)
- [SB97] Stephen M. Smith and J. Michael Brady. Susan - a new approach to low level image processing. *IJCV*, 23(1):45–78, 1997. (Cited on page 28.)
- [SHFH] Chao-Hui Shen, Shi-Sheng Huang, Hongbo Fu, and Shi-Min Hu. Siggraph asia. (Cited on page 10.)
- [SHKF12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. (Cited on page 61.)
- [SWK07] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. *CGF*, 26(2):214–226, 2007. (Cited on pages 11, 13, 45, 62 and 66.)
- [SZ12] Victor Sanchez and Avidesh Zakhor. Planar 3D modeling of building interiors from point cloud data. In *ICIP*, 2012. (Cited on pages 18 and 21.)

- [TCZ14] E. Turner, P. Cheng, and A. Zakhor. Fast, automated, scalable generation of textured 3d models of indoor environments. *J-STSP*, (99):409–421, 2014. (Cited on page 20.)
- [TM14] L. Teran and P. Mordohai. 3d interest point detection via discriminative learning. In *ECCV*, 2014. (Cited on page 14.)
- [TZ14] Eric Turner and Avidesh Zakhor. Floor plan generation and room labeling of indoor environments from laser range data. In *GRAPP*, 2014. (Cited on pages 16 and 17.)
- [VKLP09] H-H. Vu, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *CVPR*, 2009. (Cited on pages 7 and 47.)
- [VKR08] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008. (Cited on page 77.)
- [WKG10] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, 2010. (Cited on page 97.)
- [XF12] Jianxiong Xiao and Yasutaka Furukawa. Reconstructing the world’s museums. In *ECCV*, 2012. (Cited on pages 19, 21, 23, 73, 77 and 90.)
- [XRT12] Jianxiong Xiao, Bryan C. Russell, and Antonio Torralba. Localizing 3d cuboids in single-view images. In *NIPS*, 2012. (Cited on page 10.)
- [ZK13] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *SIGGRAPH*, 2013. (Cited on pages 23 and 96.)
- [ZN12] Qian-Yi Zhou and Ulrich Neumann. 2.5 d building modeling by discovering global regularities. In *CVPR*, 2012. (Cited on page 22.)