RGBD Pipeline for Indoor Scene Reconstruction and Understanding

Maciej Stanisław Halber

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Professor Thomas A. Funkhouser

September 2019

Abstract

In this work, we consider the problem of reconstructing a 3D model from a sequence of color and depth frames. Generating such a model has many important applications, ranging from the entertainment industry to real estate. However, transforming the RGBD frames into high-quality 3D models is a challenging problem, especially if additional semantic information is required. In this document, we introduce three projects, which implement various stages of a robust RGBD processing pipeline.

First, we consider the challenges arising during the RGBD data capture process. While the depth cameras are providing dense, per-pixel depth measurements, there is a non-trivial error associated with the resulting data. We discuss the depth generation problem and propose an error reduction technique based on estimating an image-space undistortion field. We describe the capture process of the data required for the generation of such an undistortion field. We showcase how correcting the depth measurements improves the reconstruction quality.

Second, we address the problem of registering RGBD frames over a long video sequence into a globally consistent 3D model. We propose a "fine-to-coarse" global registration algorithm that leverages robust registrations at finer scales to seed detection and enforcement of geometrical constraints, modeled as planar structures, at coarser scales. To test global registration algorithms, we provide a benchmark with 10,401 manually-clicked point correspondences in 25 scenes from the SUN3D dataset. We find that our fine-to-coarse algorithm registers long RGBD sequences better than previous methods.

Last, we show how repeated scans of the same space can be used to establish associations between the different observations. Specifically, we consider a situation where 3D scans are acquired repeatedly at sparse time intervals. We develop an algorithm that analyzes these "rescans" and builds a temporal model of a scene with semantic instance information. The proposed algorithm operates inductively by using

a temporal model resulting from past observations to infer instance segmentation of a new scan. The temporal model is continuously updated to reflect the changes that occur in the scene over time, providing object associations across time. The algorithm outperforms alternate approaches based on state-of-the-art networks for semantic instance segmentation.

Acknowledgements

First, I would like to thank my advisor, Thomas Funkhouser, for his support and mentorship. I am incredibly grateful for being able to learn from him how to think and approach research problems and for many invaluable lessons he has taught me during my time at Princeton.

I would like to thank my other mentors — Szymon Rusinkiewicz, Adam Finkelstein, Angel Chang, Manolis Savva, and Kevin Kai Xu — for their invaluable advice, insight, and always being available for a chat or a discussion. They are an inspiration and set an example that I hope to follow in the future.

My work at Princeton could not be completed without the help of my incredibly talented and helpful collaborators — Ohad Fried, Sema Berkiten, Linguang Zhang, Yifei Shi, Angela Dai, Matthias Niessner, Justin Solomon, Hao Li, Elena Sizikova, Shuran Song, Yinda Zhang, and Andy Zeng — thank you all for your help and support. I also would like to thank members of PIXL group at Princeton, especially to Nora Willett, Xinyi Fan, Kyle Genova, Riley Simmons-Edler, Fisher Yu, and Siddhartha Chaudhuri for all the great discussions, both technical and otherwise.

I also wanted to extend my thanks to other researchers — Jianxiong Xiao, Songjoon Choi, Thomas Whelan, Chen Liu, and Benjamin Graham — for distributing their code, which were used for many comparisons presented in this work.

The work presented in this thesis has been graciously supported by Princeton Fellowship, support from Google, Intel, NVidia, Adobe, Pixar, and NSF grants IIS-1251217 and VEC 1539014/1539099.

Dla moich rodziców, Małgorzaty i Przemysława

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Indoor scene reconstruction is a general term that refers to many algorithms that pertain to the problem of transforming the observations of an indoor scene to a digital, three-dimensional representation. Many different devices can produce such observations — including traditional color cameras and a variety range measurement devices, ranging from commodity depth sensors to LiDAR solutions. While many variants exist, this document focuses on indoor scene reconstruction with RGBD cameras. These devices are the standard choice for this application, due to their small form-factor and a relatively low price. Indoor scene reconstruction with an RGBD camera is a process of transforming visual observations in the form of color and depth images into a three-dimensional mesh describing the geometric appearance of the scene (see figure 1.1). We use the word "scanning" to describe the process of obtaining such visual information. To "scan" an indoor scene, a user visits the environment and points the camera at its various elements. The result of this action is a video sequence. The goal for scene reconstruction is to recover the location and orientation of the camera for any frame in such a video sequence.

The ability to reconstruct visual observations into a digital representation has useful applications in many areas. First, the digital content creation pipelines can

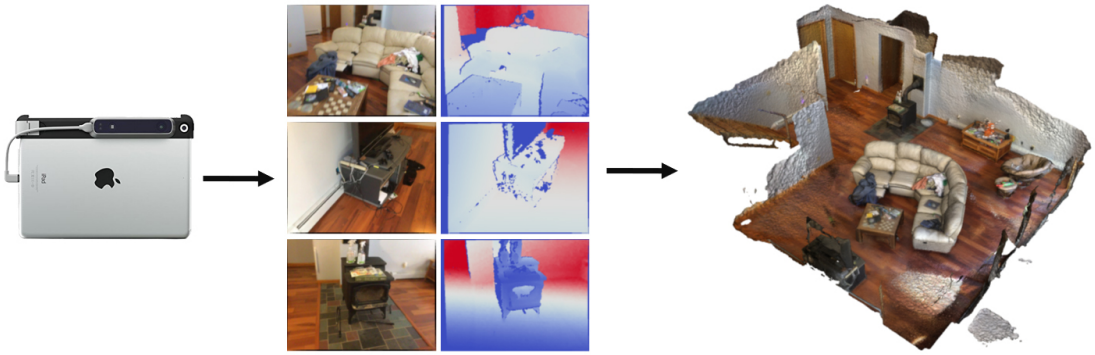


Figure 1.1: A visualization of the input and output data for methods considered in this document. From left: An RGBD camera produces a series of color and depth image pairs. Depth images provide an estimate of how far each pixel is from the camera. A video sequence of such images is converted into a three-dimensional mesh representing the scene.

benefit from such data. Scanning is an affordable way of producing digital assets and is now commonly used in the entertainment industry. Moreover, a digital version of an existing space allows for telepresence experiences — a user can be "projected" into some remote location and be able to see its contents. An area where such applications are particularly desirable is real-estate. Clients can view the reconstructions of properties on the market to help them make a more informed decision about the potential purchase. Similarly, the construction industry can use three-dimensional reconstructions to archive the progress made in a specific location. Lastly, indoor space reconstructions serve as an input to a whole range of scene understanding algorithms. These algorithms aim to estimate the contents of a reconstructed scene — they usually seek two types of information: *semantic* and *instance* information. The semantic information describes the category of scene geometry (i.e., is this scene element a part of a chair?). The instance information explains what parts of the scene constitute a single object instance (i.e., where is *my* chair?). A system capable of answering such questions is necessary to provide robots with a human-like capability to perceive and understand the world around them.

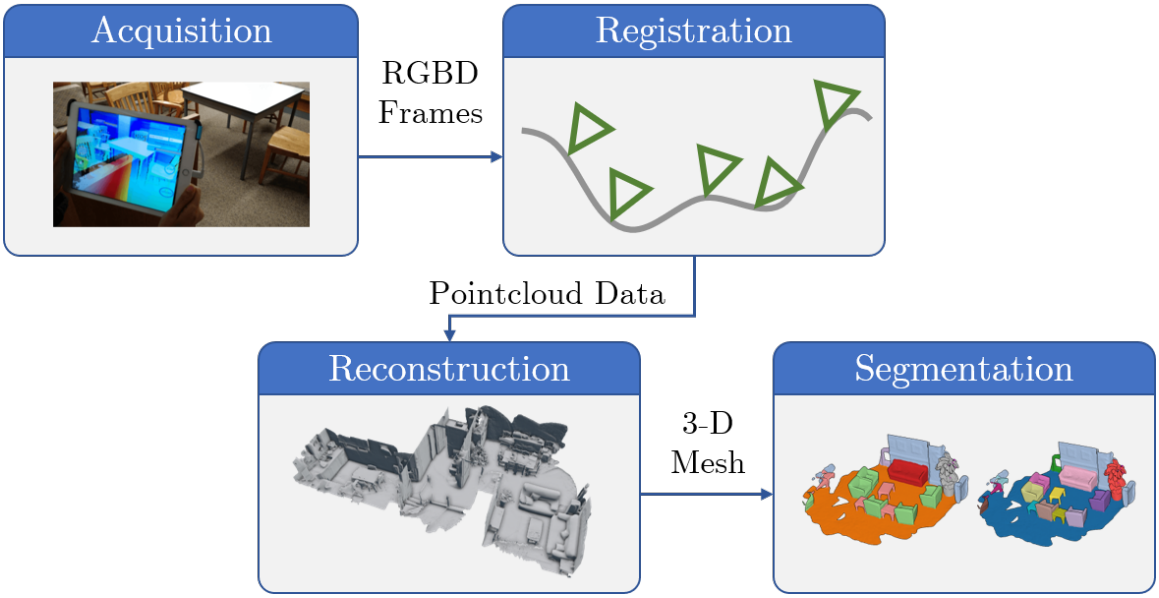


Figure 1.2: An RGBD pipeline visualization. We first capture color and depth frames in the camera registration process. Camera registration outputs location and orientation for each frame. Using camera pose information we produce a pointcloud representation of the scene. We estimate a surface representation in the mesh reconstruction process. We then use the mesh of the scene to produce semantic and instance segmentation.

Transforming the visual data to a digital representation which contains both geometric and semantic-instance information is a challenging problem. It decomposes into sub-tasks, with data flowing through from one task into another, defining a pipeline. Since we deal with color and depth data, we refer to this pipeline as an "RGBD processing pipeline". Even with an evident decomposition of the overall problem, the creation of such a pipeline is a non-trivial task. Each of the sub-problems is of considerable algorithmic complexity, with more difficulties arising when considered together. This thesis describes the algorithms required for improving many stages of an RGBD processing pipeline (see fig. 1.2). We present a complete system for producing semantic reconstructions of indoor spaces from color and depth observations.

We begin with the discussion of the capture process, focusing on understanding the properties of a commodity RGBD camera. Such capture devices are now a mainstay in the area of indoor reconstruction. This popularity is thanks to their relatively low

cost and high effectiveness in the target application. At the same time, the quality of the produced data may leave something to be desired. It is especially true in the context of indoor reconstruction, where the depth quality directly impacts the result of reconstruction algorithms. We analyze the specific issues with the data that a commodity RGBD camera produces, which inform a straightforward method for improving the quality of the returned depth measurements. Specifically, we present a calibration procedure that increases the effective range of depth. Calibration data can be obtained in minutes and does not require costly setup. To showcase the usefulness of the proposed method, we present qualitative results on a reconstruction task. We compare the quality of the reconstructions obtained using calibrated and un-calibrated data.

Second, we present an algorithm for camera registration. Camera registration is a problem of recovering the camera path from visual observations obtained during scanning. This process requires estimating the transformation that relates the consecutive frames from the input video sequence. A common issue that arises during this process is drift. Drift refers to a problem where the recovered camera locations are different for images observing the same physical space. We present a method that can battle these drift-related issues. Unlike the traditional approaches, our method achieves drift-free results thanks to the incorporation of the Manhattan-World priors into a novel optimization framework. Given an initial estimate of a camera path, our method is able to recover globally consistent trajectory, by alternating between constraint estimation and optimization. We compare our approach to existing methods by introducing a new dataset of point-to-point correspondences, allowing us to quantify the quality of reconstruction. Through this comparison, we show that the proposed method can deal with a wide range of video sequences, even in cases where other methods fail.

Lastly, we define a task of a segmentation transfer between observations of the same space. We assume that the user captures the scene observations at sparse time intervals — in the order of hours or days. In such a setting it is not possible to easily associate the data between multiple observations. However, there are still large amounts of redundancies that we can take advantage of. In our approach, we leverage the geometric similarity between the scene observations. We aim to estimate how to best fit objects from the previous observations to a novel one. This fitting procedure defines an inductive algorithm, which can be used each time one obtains a new observation of the scene. We show that our approach can produce high-quality semantic and instance segmentation. We compare our approach to baselines based on advanced deep-learning approaches.

- A low-cost calibration method to improve the effective range of a depth sensing device,
- A new iterative refinement strategy for global registration of large-scale RGBD scans,
- A new benchmark dataset for evaluating registration algorithms quantitatively,
- An inductive algorithm that infers the shapes, placements, and associations of objects from infrequent RGBD scans by utilizing data from past scans,
- A dataset of repeated scans of 13 scenes with 45 time-steps in total, along with ground-truth annotations for object instances and associations across time.

Chapter 2

Related Work

There has been a tremendous amount of work focusing on the digitization of objects and spaces in the past. In this section, we present related approaches and discuss them in the context of indoor scene reconstruction.

2.1 Acquisition

As mentioned in the introduction, this document's focus is RGBD cameras. More specifically, we describe methods that use RGBD data obtained by PrimeSense devices. These devices use a specific algorithm to obtain dense depth information — please refer to Chapter 3 for more details. However, the algorithm used by devices like Microsoft Kinect v1 or Occipital Structure Sensor to estimate the depth is not the only way of obtaining range information. In this section, we discuss a set of existing alternatives and consider their applicability to indoor scene reconstruction.

There exists a wide range of the range-sensing algorithms [Szeliski, 2010], many of which have been implemented as dedicated hardware. Given that in this work we focus on a particular, hardware implementation of a depth estimation algorithm, a sensible taxonomy to classify other approaches is to look at available hardware

(a) FARO Focus (b) Matterport Pro 2 (c) Next-Engine Scanner

Figure 2.1: Examples of various range measurement devices.

solutions, and how they compare to RGBD cameras. Different possible classification methodologies include scanning time, accuracy, allowed object size, etc.

Industrial Solutions. The first class of devices that we can discuss are Light Detection and Ranging (LiDAR) devices. The principle behind LiDAR devices is the time-of-flight calculation. These devices emit a light signal (using an LED or a laser) and record the reflected signal. The time difference between when the signal is emitted and received, coupled with the speed of light constant, allows one to calculate the distance. For more information on the principles behind such devices, please see [Adams, 2002]. An example of a hardware implementation that is commonly used in indoor scene reconstruction is the Focus product-line by FARO [FARO Technologies, 2019]. They use a laser as the light emitter, which coupled with a mirror surface rotating at high speeds allows these devices to capture 360° sweeps of the environment around themselves. The main advantage of this solution is its high accuracy and range. FARO Focus 350 (fig. 2.1a) boasts a range between $0.6m$ to $330m$, with distance accuracy up to a single millimeter. Additionally, these devices are well-suited to scanning both indoor and outdoor environments. Recent datasets [Park et al., 2017, Knapitsch et al., 2017] show high-quality indoor and outdoor reconstructions obtained with a FARO Focus scanner. While capable of generating high-quality results, the distinct disadvantage of a LiDAR device is its

price — as a professional tool, the cost is in tens of thousands of dollars. As such, LiDAR sensors cannot be adopted by an average consumer. Additionally, the coaxial range measurement devices, like FARO Focus, only scan a single point at a time — thus increasing the scanning time required to obtain a scan of the environment. Furthermore, one needs to carefully plan the locations of the scanner to obtain a complete reconstruction of the space. Using a hand-held RGBD camera, allows the user to move freely within the space, without a need for such pre-planning.

Another professional grade camera is the Matterport Pro 2 [Matterport, 2019] (fig. 2.1b). This camera is a proprietary design, based on three RGBD cameras. The Matterport Pro 2 is built with an indoor scanning application in mind. The main focus of these devices is the creation of high-resolution, HDR color imagery. The depth is obtained using the structured light algorithm (see Chapter 3). Because of this choice, the device has a limited range of depth measurements (up to $4.5m$). However, it needs to be noted that the returned depth maps are more accurate than those from a commodity RGBD sensor. While Matterport Pro 2 is cheaper than the FARO Scanner, its price is still high, preventing adoption by the general public.

Projection based devices. The next type of a device producing depth measurements is a laser-stripe scanner, like the Next-Engine system [NextEngine, 2019](fig. 2.1c). These devices use a laser-stripe projection to estimate the depth — given a known plane equation, and a recording of a laser stripe in the image, one can estimate the depth of points lying on the projected stripe [Curless and Levoy, 1996]. By sweeping the laser along the object's surface, one can obtain depth measurements. Given that the localization of a single laser stripe within the image is relatively straightforward, this method returns highly accurate depths. The limitations are two-fold. First, the scanning time — since only points along laser-stripe can be reconstructed at a single time, the scanning time can become quite long. Many systems [Brown et al., 2008, Fan et al., 2016] have been introduced to combat such

limitations. Secondly, only objects of limited size can be scanned — extending this procedure to large scale structures is a non-trivial process [Levoy et al., 2000].

The speed issues of a single laser-stripe camera were addressed in computer vision literature by projecting multiple stripes onto the surface with a projector. In this setting, the challenge is to differentiate between the multiple stripes visible in the image. A classic solution [Posdamer and Altschuler, 1982] is to project numerous stripe patterns in order to create a unique code for each one. This class of methods is referred to as temporal structured light and is related to the algorithm used by the PrimeSense cameras, which in turn use what is called spatial structured light. Many extensions have been proposed over the years to limit the amount images that need to be projected by temporal structured light methods [Hattori and Sato, 1996, Horn and Kiryati, 1997, Batlle et al., 1998, Li Zhang et al., 2002], or how to deal with slowly moving objects [Hall-Holt and Rusinkiewicz, 2001, Rusinkiewicz et al., 2002]. The commercial solutions for this type of algorithm include HP 3D Scan systems [HP, 2019]. While these methods can produce high-quality depths, just like the laser stripe methods described above, the main limitations come from the size objects that can be scanned using these systems — they are not well suited to scanning environments, but rather table-top objects. Additionally, due to the speed of acquisition, scanning of the environments with such devices is also limited, as one needs to place the camera down before taking the scan.

Alternative depth cameras. Apart from all the above solutions, there are also alternative depth cameras one could use instead of the PrimeSense devices. First, is the updated Microsoft Kinect v2 [Microsoft, 2019]. Microsoft has changed the design of the device completely, moving away from the spatial structured light and onto the time-of-flight (TOF) camera. The principle behind the TOF cameras is related to that of a LiDAR, however instead of producing just a single depth measurement at a time,

depth measurements are made for every surface element within the camera frustum. The underlying principle is, however, the same — a device emits a light signal, usually in the near infra-red spectrum, for a very short time. Then the light sensor, exposed to that spectrum only, gathers the light over a short time interval. The intensity stored at a pixel is proportional to the depth [Iddan and Yahav, 2001, Gokturk et al., 2004]. While conceptually simple, this approach has an issue of interference with an ambient light — these devices cannot work outside. Additional issues come from the fact that since the entire scene is illuminated at a time, the light reaching the camera might come from multiple reflections, affecting measurement accuracy. Despite these issues, the TOF camera design is quite attractive for indoor scanning. However, with Microsoft Kinect's v2 much more closed environment, these cameras have not seen much use from the reconstruction community.

Another class of commercially available depth cameras comes from Intel. The Intel RealSense [Intel, 2019] line of device uses what's called an active stereo algorithm. The depth measurement setup includes a pair of infrared cameras and an infrared projector. The infrared projector projects a pattern into the environment to help with the correspondence problem. Once the correspondence between pixels in two images has been established, the depth can be estimated from disparity, the difference in distances to the left edge of the picture from corresponding images [Hartley and Zisserman, 2004].

Lastly, one can use only color cameras to obtain depth. With two cameras, as mentioned in the previous paragraph, one can establish the pixel correspondence and recover the depth. In this case, there is no actively projected pattern. Hence we refer to this class of methods as *passive stereo*. The challenge comes from the correspondence estimation problem, where usually the correspondence can be estimated between the set of distinctive features only, producing sparse depth measurements. Techniques to provide a dense reconstruction from a set of color images exist (for

example work by [Furukawa and Ponce, 2010]) — a good overview of various alternatives can be found in the recent reconstruction benchmark [Knapitsch et al., 2017].

To conclude, while many alternatives exist, the PrimeSense camera offers the right combination of the cost, form-factor, and depth quality. As such, it is not surprising that they have become a standard acquisition device for indoor scanning applications.

2.2 Registration

In Chapter 4, we discuss a method for improving the quality of camera poses. We propose to use a straightforward way to estimate the initial set of poses to highlight the robustness of our approach. At the same time, it is essential to note that there do exist alternative tracking methods that are often more robust than the one we describe. The basic building block of most of the tracking methodologies is a registration procedure — given two observations of the same scene, we wish to estimate a set of parameters that relate them. In the majority of cases, this relationship is modeled as a rigid body transformation. Here we review various methods that aim to solve this problem.

Iterative Closest Point. A classic algorithm coming from the range-sensing community is the *Iterative Closest Point* (ICP). ICP algorithm's goal is to estimate the rigid transformation $[R|t]$ that aligns two surfaces $\mathcal{P}$ and $\mathcal{Q}$ to each other. The basic idea behind ICP is to first estimate a set of correspondences between $\mathcal{P}$ and $\mathcal{Q}$. As the ICP algorithm assumes that $\mathcal{P}$ and $\mathcal{Q}$ are approximately aligned, the way correspondences are generated is by finding closest points on both $\mathcal{P}$ and $\mathcal{Q}$ (hence the *Closest Point* part). With some estimate of correspondences, one can calculate the transformation minimizing the distance between the two sets of corresponding points. The process of the correspondence finding and distance minimization is repeated until convergence (hence the *Iterative* part). Many extensions to the classic ICP approach exist, as described in [Rusinkiewicz and Levoy, 2001] —

the main direction of improvement, explored by the prior work, is the definition of a more descriptive error function to estimate the transformation. The initial work proposed minimization of a point-to-point distance [Besl and McKay, 1992]. A commonly accepted extension was the point-to-plane distance [Chen and Medioni, 1991], which is shown to converge faster. Follow-up work [Mitra et al., 2004] proposed to approximate the squared distance between two surfaces and has shown how it leads to further improved convergence at the cost of additional data-structures. Recently, [Rusinkiewicz, 2019] proposed a novel symmetric distance function, showing yet again improved convergence compared to the classic techniques, but without any additional data-structure requirement. Another direction for improvement is deciding what points to use for the closest point search and transformation estimation. Work by [Rusinkiewicz and Levoy, 2001] proposes normal-space sampling that aims to sample points in such a way that the distribution of normals of the selected points is uniform. In follow-up work, [Gelfand et al., 2003] introduces sampling based on the analysis of the surface properties to select points that best limit the motion's degrees of freedom.

Global registration. While the ICP algorithms assume initial alignment between the input shapes to estimate the transformation, an approximately correct alignment might not be known. In such a situation, we need to turn to the global registration algorithms. This class of methods' goal is to estimate transformation with no assumptions made on the initial configuration. A traditional approach is to employ a Randomized Sample Consensus [Fischler and Bolles, 1981] algorithm, where two sets of n points are selected on each surface $\mathcal{P}$ and $\mathcal{Q}$ randomly and are used to estimate the transformation. The value of n depends on the model we are trying to estimate, and it needs to be high enough to constrain all degrees of freedom the model allows. For a rigid body transform $n = 3$. After estimating the model, we can quantify its quality. This process is repeated many times over, keeping track of the best model. It can be shown that after k trials, the probability of obtaining a good model is high.

Given enough time, the RANSAC approach will provide the correct answer. In situations when the time is the constraint, alternative solutions have been proposed — 4PCS [Aiger et al., 2008] and follow-up Super4PCS approaches [Mellado et al., 2014] propose to find sets of 4 coplanar points in both scans that are approximately congruent. Authors show how imposing such constraint on the sampled points leads to improved asymptotic performance compared to traditional randomized methods.

Another alternative is to use descriptors to encode points on the input surfaces. While standard ICP formulation uses the Euclidean distance between the locations of each point to determine the relationship between them, an alternative is to use some higher dimensional, transformation-invariant descriptor to perform the matching. A number of such descriptors were proposed, including spin images [Johnson and Hebert, 1999], integral descriptors [Gelfand et al., 2005], point-pair features [Drost et al., 2010], point feature histograms [Rusu et al., 2009]. More recently, methods for learning the descriptors, rather than hand-designing them, were proposed. Recently, [Zeng et al., 2016] proposes to use the reconstruction of 3D scenes to get corresponding observations of the same surface for training. A similar approach is presented in work by [Khoury et al., 2017] — methods differ on how the local geometry is encoded. The former method uses a regular grid, while the latter uses a spherical binning strategy. Additionally, methods that utilize the descriptors to estimate the transformations in a way that is robust to incorrect matches exist. For example, [Zhou et al., 2016] proposes an iterative registration procedure that can down-weight incorrect matches, allowing for effective registration even with imperfect descriptors.

Structure-From-Motion. While the methods mentioned above look at the alignment of two 3D surfaces, Structure-From-Motion (SfM) considers itself with registering images taken from different viewpoints. Usual SfM pipeline consists of a feature extraction step, where distinctive points in two images are extracted (using ap-

proaches like [Lowe, 2004], or [Bay et al., 2008]). Then a matching step is performed, to create a set of correspondences between two sets of keypoints. Keypoints can then be used to estimate motion between two views, and the 3D positions of the keypoints can be estimated via triangulation [Hartley and Zisserman, 2004]. SfM methods often extend the problem to registering multiple views and estimating an optimal 3D structure, defining a bundle adjustment problem [Triggs et al., 2000]. Systems capable of producing sparse reconstructions of many well-known, real-world locations were presented [Snavely et al., 2006, Agarwal et al., 2011]. While able to produce impressive results, the fact that these methods operate on RGB data makes them not well suited to reconstructing indoor spaces. The success of these methods relies on finding distinctive points in two images, which might be difficult in texture-less, indoor spaces.

RGB Motion Estimation. SfM techniques described in the previous paragraph mostly look to register unorganized collections of photographs, with a possibly sparse set of observations, where the overlap between images might be low. Additionally, SfM techniques are generally computationally expensive and thus are performed offline. Simultaneous Localization And Mapping (SLAM) algorithms, on the other hand, are designed to perform under the real-time constraints.

The literature on SLAM is vast, and many variants have been proposed over the years. A systematic way to classify different methods was introduced in [Engel et al., 2016], based on four criteria. A class of a method is determined based on whether it *directly* optimizes the motion from the device measurements, or whether it performs *indirect* step of extracting intermediate representation to define an error function. Another pair of criteria concerns itself with whether the method uses all measurements reported (*dense*), or whether it attempts to extract a set of meaningful information first (*sparse*).

Sparse and Indirect methods are the most common. In such systems, a sparse set of observations is extracted in the form of keypoints (traditionally corners [Harris and Stephens, 1988]), and then matched to estimate the motions. These methods are indirect, as they optimize geometric error based on matched points, rather than comparing the measurements (pixel-values) directly. Examples of such methods include [Klein and Murray, 2007, Mur-Artal et al., 2015]. The main advantage of such methods is their robustness to various issues, like the sensor noise and even geometric errors caused by the rolling shutter. They, however, trade this robustness for not being able to use all information within the image and will fail if distinctive keypoints are not available.

Dense and Indirect methods are less common. The main idea behind them is to first extract indirect information in the form of, for example, a dense optical flow field between two images and then aim to minimize the error. Examples include [Valgaerts et al., 2012, Ranftl et al., 2016]. The limited popularity of these methods can be attributed to the fact that the optimization problem arising from a dense approach is harder to solve, as pointed out in [Engel et al., 2016].

Dense and Direct methods, like [Stühmer et al., 2010, Newcombe et al., 2011, Engel et al., 2014], aim to use all pixel information to optimize for the motion. These methods are direct, as they build the error function around the photometric response recorded by the camera. The advantage of these approaches is a more stable motion estimation that does not depend on the keypoint extraction step. However, these methods are more susceptible to issues caused by sudden illumination changes, sensor noise, or rolling shutter errors.

Sparse and Direct methods also propose to minimize the photometric error using only the sparse information. The motivation behind such an approach is noted by [Engel et al., 2016] — the dense methods lead to a harder optimization problem.

As such, [Engel et al., 2016] proposes only to extract sparse information so that the motion estimation procedure is more efficient.

To sum up, the choice of specific method class largely depends on the modality of input data. The benefits of indirect methods include robustness to certain types of noise and distortion, like sudden illumination changes or rolling shutter errors. As such, these methods can be applied to images from consumer grade color cameras. This robustness trades for the method's performance — direct methods generally lead to more robust trajectory estimation but require photometrically calibrated cameras, with a global shutter.

Regardless of the method chosen, most methods that rely only on the RGB information will have issues with estimating trajectory in indoor environments, where texture-less surfaces are common (although, this issue has been recently addressed by using a wide-lens cameras [Caruso et al., 2015]). Even when a technique is able to recover the correct camera motion, the resulting reconstruction will be sparse, preventing its use for visualization applications. Section 4.2 discusses SLAM systems that recover camera trajectory from the depth measurements.

2.3 Understanding

Once the camera poses are estimated, and the scene geometry is reconstructed, the obvious question to answer is what are the contents of the reconstructed environment. Given the information of the scene in the form of 3D geometry, one would like to understand what objects are within the scene. As such, the problem of scene understanding is analogous to the traditional 2D image understanding, where standard tasks include image classification [Krizhevsky et al., 2012], image captioning [Johnson et al., 2016] and image segmentation [Long et al., 2015].

For 3D scene understanding, the standard tasks include semantic and instance segmentation. The former task asks to classify the scene's geometry into a set of discrete groups, like a chair, bed, or sofa. Instance segmentation asks to enumerate all objects within the scene and to specify their boundaries.

Much progress has been made in the visual understanding of the color images, with widely accepted approaches emerging [He et al., 2015, He et al., 2017]. The massive success of these approaches inspires the use of deep-learning in for 3D data. However, since large, labeled datasets of 3D data, required for deep-learning, have become available only relatively recently [Chang et al., 2017, Dai et al., 2017, Hua et al., 2016], there is not a generally accepted way to learn using three-dimensional data. Below we present some of the ways the previous work has proposed. The main differencing factor between them is the underlying representation of the scene geometry:

Volume-Based. The most straightforward way to approach learning using 3D data is to use volume-based representation, like the initial baseline approaches released with the Scannet dataset [Dai et al., 2017]. In the volume-based approaches the scene geometry is represented using a regular grid, often storing truncated signed distance to the surface [Zeng et al., 2016]. The advantage of the regular grid representation is the fact that one can use standard convolutions to train the network, where simply an additional dimension is involved. The main drawback is that such a representation is very memory intensive, preventing the representation of fine details, that might have existed in an input surface. Attempts have been made to address this issue — [Riegler et al., 2017] proposes to use hierarchical, octree representation of the surface, allowing for much more compact storage. More recently,

[Graham et al., 2018], shows how to directly perform convolutions on sparse data, leading to much-improved results.

Mixed Representation. There also exist approaches that aim to bring in the high-frequency information that might be lost when converting to a volumetric grid, by employing joint representations. [Dai and Nießner, 2018] uses a joint 3D-2D network, where both the color images and the voxel representation of the 3D geometry are used. Authors propose to first extract the 2D features from the color images and then backproject them onto the voxel grid. This procedure allows authors to perform convolutions on two voxel grids, one representing the geometry and the other representing the features extracted from the color images. They show how this approach leads to a significant improvement in performance on the semantic segmentation task.

Similarly, [Narita et al., 2019] presents a semantic mapping system that uses both the geometry and the color images. Unlike the previous system, this system performs in an online fashion. The authors show how to fuse the results from a 2D segmentation approach (using the system by [He et al., 2017]) onto the 3D geometry, and how to track and associate label instances over time. As a result, they are able to obtain high-quality instance segmentation of the scene, while also building the geometric representation at the same time.

Point-based. Yet another way to reason about the scene is to use point-based representation, where points are sampled from the input surface. The initial work of [Qi et al., 2017a] has introduced the Pointnet architecture, capable of learning on unstructured sets of points. The network tries to approximate a symmetric function f so that its results are independent of the ordering of the input points. Authors show how such an architecture can be used for tasks like classification and semantic segmentation. However, due to the use of a single max pooling operation to aggregate the whole point set, the Pointnet is unable to represent the local structures and in-

tricate local details. In the follow-up work [Qi et al., 2017b], they extend the original architecture by first sampling and grouping local regions and then using Pointnet to encode these regions into feature vectors. This way, they are able to capture more information about the local details.

Surface based. Lastly, there are also approaches that propose to represent the scene geometry just as a surface and perform learning directly on it. The approach of [Tatarchenko et al., 2018] defines convolutions on tangent planes of the input surface, allowing for learning on top of surface-based, geometric information. Recent work by [Huang et al., 2018] is showing how to learn from color information stored on the surface.

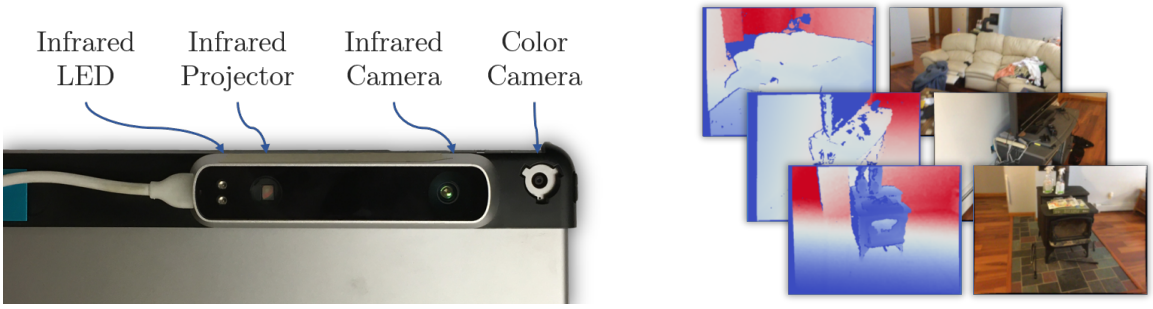
In addition to the approaches mentioned above, the field of scene understanding has also defined a set of novel tasks. These include shape completion [Dai et al., 2016b], semantic scene completion [Song et al., 2017, Dai et al., 2018] where the goal is to estimate the semantics and geometry of the scene jointly. Another new task is the scan-to-CAD alignment [Avetisyan et al., 2019]. Given a scene reconstruction, the goal is to find CAD models (from the Shapenet dataset [Chang et al., 2015]) that best fit the scene geometry and appearance. System by [Avetisyan et al., 2019] introduces an entire dataset for this task, based on the Scannet dataset [Dai et al., 2017], and shows a preliminary solution, alongside a benchamrk.

Chapter 3

Camera Calibration

3.1 Introduction

As discussed in section 2.1, a variety of approaches and devices can be used to obtain the depth information. Here, we limit the discussion to a particular class of cameras, which we refer to as the PrimeSense cameras. The name is due to a company holding a patent on the depth conversion algorithm [Freedman et al., 2007] implemented in these cameras. The devices in the PrimeSense-class include Microsoft Kinect, Asus Xtion Pro, and Occipital Structure Sensor. These cameras estimate the depth information using an approach based on a spatial structured light, where a known pattern is projected onto the environment in the infrared spectrum. First, the image of the projected pattern is recorded back using the infrared camera. Then, by comparing the appearance of the captured pattern with the reference pattern, the algorithm estimates the depth at a pixel location. As such, PrimeSense cameras can be considered as an implementation of an active stereo system, where the infrared projector replaces one of the cameras. However, the exact details of the depth estimation algorithm are not known, as it is a subject of the aforementioned, proprietary patent [Freedman et al., 2007].



(a) Occipital Structure Sensor on an Apple iPad (b) RGBD images

Figure 3.1: Commodity RGBD camera and an example output.

In terms of the output, the PrimeSense devices produce "RGBD images". An RGBD image consists of a three-channel image $\mathcal{I}$ and single-channel image $\mathcal{D}$. Image $\mathcal{I}$ stores the color information, and image $\mathcal{D}$ stores the information about the distance between the observed scene and the sensor (see figure 3.1). Unlike the RGB-only approaches, the depth cameras can produce depth estimates even for texture-less regions. Given that many indoor spaces are often texture-less, RGBD cameras have become widely popular in the field of indoor scene reconstruction. At the same time, it needs to be noted that the initial application for commercial RGBD cameras was human body tracking [Shotton et al., 2011]. Not surprisingly, the body-tracking application has different constraints than the indoor reconstruction task. The above observations motivate the investigation of the PrimeSense camera properties to understand how to use them for indoor reconstruction effectively.

The methods presented in this section were developed for the Scannet project [Dai et al., 2017]. Scannet consists of 1513 labeled reconstructions of indoor scenes, and the data has been captured using the Occipital Structure Sensor camera. As mentioned previously (Chapter 1), the quality of the depth data affects the reconstruction quality. This dependence is caused by the fact that most of the tracking algorithms optimize for the camera poses by minimizing residuals based on the depth data. Hence for the Scannet project, the correct calibration of the acquisition devices is critical. To provide users of the datasets with the high-quality data, we analyze

issues present in the data produced by the Occipital Structure Sensors. As a result, we introduce two calibration procedures.

First, we show a way to address the problem of alignment between depth and color. As seen in figure 3.1, the depth and color images are taken from different locations. As such, the images are observing slightly different parts of the scene. Simply overlaying depth and color information leads to significant visual errors. Additionally, if the registration algorithm depends on the correct alignment between depth and color, the simple approach fails. We provide a description of a method producing an alignment between the color and depth images.

Second, we describe the depth undistortion strategy to solve the issue of a low-distortion warp present in the depth images. This warp refers to the phenomena of incorrect depth estimates returned by the device. We show that the errors are dependent on both the pixel location and the estimate of the depth. We aim to calibrate the measurements returned by a depth camera to minimize these errors. Concretely, we estimate an undistortion function from pairs of observed and ground-truth depths that minimizes the distance between the two. We follow an existing method for producing such undistortion function. Our contribution lies in the design of a novel, low-cost method for acquiring the necessary calibration data. Lastly, we investigate how the depth calibration affects the quality of camera registration algorithms.

Overall, we make the following contributions:

- An automatic approach to estimate intrinsic and extrinsic properties of the color and depth cameras
- A low-cost solution for capturing calibration sequence required for estimating the depth undistortion function

3.2 Related Work

Depth camera calibration has not received much of research interest. Most of the methods which use RGBD data simply use the images after the stock calibration is applied. In this section, we are going to give a brief overview of how a PrimeSense camera generates a depth image. We also discuss a number of works that look at estimating the parameters in the PrimeSense cameras.

3.2.1 Depth Estimation

The way PrimeSense cameras work is a subject of a proprietary patent, currently held by PrimeSense and Apple [Freedman et al., 2007]. Prior work has, however, looked into trying to understand the principles behind how these devices operate. Work by [Martinez and Stiefelhagen, 2013] and [Martinez and Stiefelhagen, 2014] gives a good overview of that process. We provide only a summary here.

To estimate depth information PrimeSense devices use a structured light approach, where a known pattern is projected onto the environment. To achieve real-time performance, these cameras use a spatial structured light, where a known speckle pattern is projected. Local blocks within this speckle pattern are unique. To enable the capture of the regular color images and to lower the interference with the visible light, the speckle pattern is projected in the infrared (IR) spectrum.

An associated IR camera captures the image of the environment with the projected pattern. The image of the environment and the expected appearance of the projected pattern constitute a stereo image pair. As such, the PrimeSense device can be considered a stereo setup, where a projector replaces one of the cameras.

With the two images of the projected pattern and a known relationship between the IR camera and the IR projector, the depth can be estimated from the disparity between corresponding pixels. To compute correspondence between the pixels

in the two images PrimeSense devices are most likely using a block matching algorithm [Martinez and Stiefelhagen, 2013] — which can take advantage of the uniqueness of the local blocks within the projected pattern. With the pixel-to-pixel correspondence established, the disparity values (difference of the distances to the left side of the image) are trivially computed. These values can be used to calculate depth, using a well-known formula $z = \frac{fb}{d}$, where z, f, b, d are, respectively, the depth, focal length, baseline between cameras and the disparity value [Szeliski, 2010].

We note that we do not consider the calibration of the infrared projector and the infrared camera pair. While calibrating these could improve the depth-quality, it would prevent us from using the hardware implementation of the depth estimation algorithm.

It is important also to note that several systems attempting to improve the original idea exist, like [Fanello et al., 2016] or [Fanello et al., 2017]. Regrettably, these work require a specific hardware setup, and the algorithm implementations are not publicly available.

3.2.2 Camera Calibration

Several works [Nguyen et al., 2012, Andersen et al., 2012] have looked into identifying and modeling common issues that the PrimeSense devices exhibit.

Early work by [Smisek et al., 2011] discusses a system for Microsoft Kinect sensor calibration. They introduce a framework for estimating intrinsic and extrinsic parameters for both depth and color cameras. The set of estimated parameters is very similar to our proposed model. Smisek's work is also the first one to provide a discussion regarding the image-space distortion effect. When capturing a depth map of a flat wall and comparing the returned depth measurements to a known distance to the plane, authors observe a spatial distortion pattern. To correct it, they propose to compute the per-pixel residual mean and apply it to the depth images. However,

since this work performs experiments on small depth ranges only, no relationship between the image space effect and distance to the camera is described.

Follow-up by [Herrera C. et al., 2012] introduces a novel calibration method for the Primesense-class cameras. The main difference between this work and [Smisek et al., 2011] is the modification of how to undistort the depth values returned by the device. Authors introduce a way to estimate spatial distortion pattern, that varies in image space. Additionally, the effect of this pattern is modulated by a function of observed depth. The authors report how the proposed calibration decreases the error in the measured depth by comparing the depth measurements to a known 3D plane.

Work by [Nguyen et al., 2012] analyses the noise properties of a depth sensing device, with a goal of improving tracking quality in the KinectFusion system [Newcombe et al., 2011]. In addition to identifying the dependence of the noise on the distance from the camera (what the authors call axial noise), this work also classifies the second type of noise, called the lateral noise. The lateral noise does not depend on the distance but instead occurs when the sensor is observing the surface at a grazing angle. As such, the authors model the lateral noise as proportional to the angle a viewing ray makes with the observed surface. The authors propose to model the axial noise with a quadratic function, while the lateral noise is modeled as a linear one. The proposed noise model is used to drive an algorithm to denoise the depth maps. The authors demonstrate improved tracking results when using the denoised depth images. This work does not provide any discussion on the image space effects.

Work by [Teichman et al., 2013] looks at the problem of calibration from the perspective of the indoor reconstruction. Th authors provide clear evidence of how such a process improves the registration quality. The authors model the relationship between the error, the image location, and the measured depth using a look-up table. Each

cell of this table stores an "undistortion value". A corrected depth is a product of an observed depth and the related "undistortion value". We provide a detailed discussion of this approach in the following section 3.5. Follow-up work by [Di Cicco et al., 2015] uses an undistortion approach similar to that introduced in [Teichman et al., 2013]. The significant difference is how both works obtain values required for estimating values stored in the look-up table.

3.3 Approach

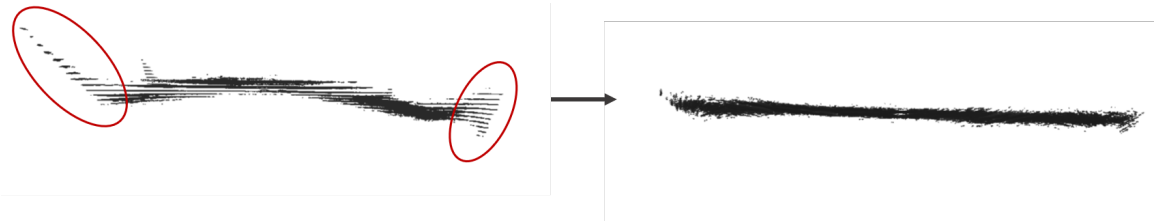


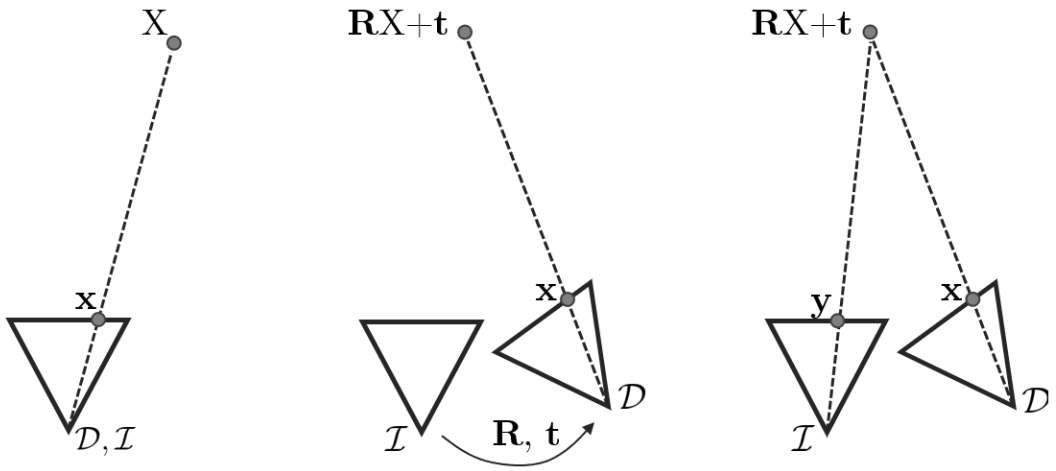
Figure 3.2: Visualization of the application of the calibration procedure described in this section on the depth map produced by a PrimeSense device. **Left:** A depth image of a flat wall taken with a Structure Sensor at four meters away. **Right:** Same depth image after applying the proposed calibration method.

In this section, we present two techniques used to prepare data from a PrimeSense-class device to be used in the indoor scanning applications. First, we perform the depth-to-color alignment to provide correspondences between the pixels in color and depth images. Second, we perform the depth undistortion to improve the quality of the measured depth by removing undesired distortion — as seen in figure 3.2.

3.4 Depth-to-Color Alignment

3.4.1 Model

First, we aim to find an alignment between the depth image $\mathcal{D}$ and the color image $\mathcal{I}$. Specifically, we seek to find a mapping between a pixel coordinate $\mathbf{p}$ in $\mathcal{D}$ and a



(a) (b) (c)

Figure 3.3: (a) Function B backprojects image coordinate $\mathbf{x}$ to X. (b) Function T moves point X to $X' = \mathbf{R}X + \mathbf{t}$. (c) Function P projects point X' onto coordinate $\mathbf{y}$.

pixel coordinate $\mathbf{q}$ in $\mathcal{I}$. The relationship between the two can be described using the following equation:

$$\mathbf{q} = \Phi(\mathbf{x}, \Theta) = P(T(B(\mathbf{p}, \theta_B), \theta_T), \theta_P) \tag{3.1}$$

The transformation Φ with parameters $\Theta = \{\theta_B, \theta_T, \theta_P\}$ from $\mathbf{p}$ to $\mathbf{q}$ is a combination of three functions. Figure 3.1 shows a visual explanation of the transformations performed by each of the functions. We define each one of them in turn.

Backprojection $X = B(\mathbf{x}, \theta_B)$ takes in a two-dimensional coordinate $\mathbf{x}$ and returns a three-dimensional point X in the camera space. Backprojection function allows us to specify a camera-space 3D position for a pixel $\mathbf{x}$ with an associated depth. The set of parameters θ_B is $\{\mathcal{D}, K_{\mathcal{D}}\}$

$$B(\mathbf{x}, \theta_B) = \mathcal{D}(\mathbf{x})\mathbf{K}_{\mathcal{D}}^{-1}\dot{\mathbf{x}} \tag{3.2}$$

Operation $\mathcal{D}(\mathbf{x})$ returns the depth value stored in the depth image $\mathcal{D}$ at a coordinate $\mathbf{x}$. $\dot{\mathbf{x}}$ specifies a homogenization of a coordinate $\mathbf{x}$. That is, if $\mathbf{x} = [u\ v]^\intercal$,

then $\dot{\mathbf{x}} = [u\ v\ 1]^\intercal$. $\mathbf{K}_\mathcal{D}$ is an intrinsic matrix associated with the depth camera that produced image $\mathcal{D}$. When modeling the properties of PrimeSense-class cameras we have found that pixels are approximately square, and as such we omit the skew factor γ and model the any intrinsic matrix $\mathbf{K}_\mathcal{X}$ as:

$$\mathbf{K}_\mathcal{X} = \begin{bmatrix} f_x^\mathcal{X} & 0 & c_x^\mathcal{X} \\ 0 & f_y^\mathcal{X} & c_y^\mathcal{X} \\ 0 & 0 & 1 \end{bmatrix}$$

Rigid Body Transformation $X' = T(X, \theta_T)$. Transformation function T takes in a three-dimensional point X and returns a transformed point X'. This function allows us to move between the different camera spaces. Goal of T in this formulation is to move from the depth camera space to the color camera space (see fig. 3.3b). The parameters θ_T are the rotation matrix $\mathbf{R}$ and the translation vector t, which describe geometric relationships between cameras that generated $\mathcal{D}$ and $\mathcal{I}$. We define T as:

$$T(X, \theta_T) = \mathbf{R}X + \mathbf{t} \tag{3.3}$$

Projection $\mathbf{y} = P(X, \theta_P)$. Projection function P takes in a three-dimensional point X and computes a two-dimensional coordinate location in the associated camera plane. This allows us to finalize the function Φ (eqn. 3.1). In the case of P, the parameter set θ_P consists of K_I, the intrinsic matrix of the color camera. We define P as:

$$P(X, \theta_P) = \pi(K_\mathcal{I} X) \tag{3.4}$$

The function π is the perspective divide transform. If $X = [x\ y\ z]^\intercal$, then $\pi(X) = [x/z\ y/z]^\intercal$.

Together, functions B, T, P allow us to model the relationship Φ between coordinates $\mathbf{p}$ and $\mathbf{q}$. However, note that the above derivations assume a pinhole camera model for both devices generating $\mathcal{D}$ and $\mathcal{I}$. In our experiments with the Structure Sensor camera, we have found that both cameras have moderate levels of barrel distortion. As such to apply the above equations, we need to resample the images $\mathcal{D}$ and $\mathcal{I}$ to provide images as if taken by pinhole-cameras. We model the barrel distortion with two coefficients k_1, k_2 as:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} x_u(1 + k_1 r^2 + k_2 r^4) \\ y_u(1 + k_1 r^2 + k_2 r^4) \end{bmatrix}$$

where x_u, y_u are the undistorted coordinates, x_d, y_d are the distorted coordinates, and $r = x_u^2 + y_u^2$. Coordinates above are specified as normalized image coordinates. Normalized image coordinates are calculated from pixel locations by subtracting the projection center and dividing by the focal length [MathWorks, 2019a].

Overall, the set of parameters that we wish to find to model the depth-color camera pair for PrimeSense devices is given in table 3.1:

Parameters	Description
$f_x^{\mathcal{I}}, f_y^{\mathcal{I}}, c_x^{\mathcal{I}}, c_y^{\mathcal{I}}$	Intrinsic parameters fort the color camera ($\mathbf{K}_{\mathcal{I}}$)
$f_x^{\mathcal{D}}, f_y^{\mathcal{D}}, c_x^{\mathcal{D}}, c_y^{\mathcal{D}}$	Intrinsic parameters fort the depth camera ($\mathbf{K}_{\mathcal{D}}$)
$k_1^{\mathcal{I}}, k_2^{\mathcal{I}}$	Barell distortion coefficient for the color camera
$k_1^{\mathcal{I}}, k_2^{\mathcal{I}}$	Barell distortion coefficient for the depth camera
$\mathbf{R}$	Rotation between depth and color cameras
$\mathbf{t}$	Translation between depth and color cameras

Table 3.1: Parameters Θ modelling Structure Sensor and iPad camera setup.

3.4.2 Optimization

The relationship Φ (eqn. 3.1) describes a stereo camera rig. Assuming that the geometry of depth camera and infrared camera are related [Smisek et al., 2011], we can obtain the required set of parameters Θ by capturing a set of image pairs of calibration grid target from both color and infrared cameras. Once such a set of images is obtained, optimizing for Θ is done by using a standard stereo-calibration method. An example of required image pairs can be seen in figure 3.4. In this work, we use Matlab's implementation [MathWorks, 2019b] of the strategy described in [Zhang, 2000].

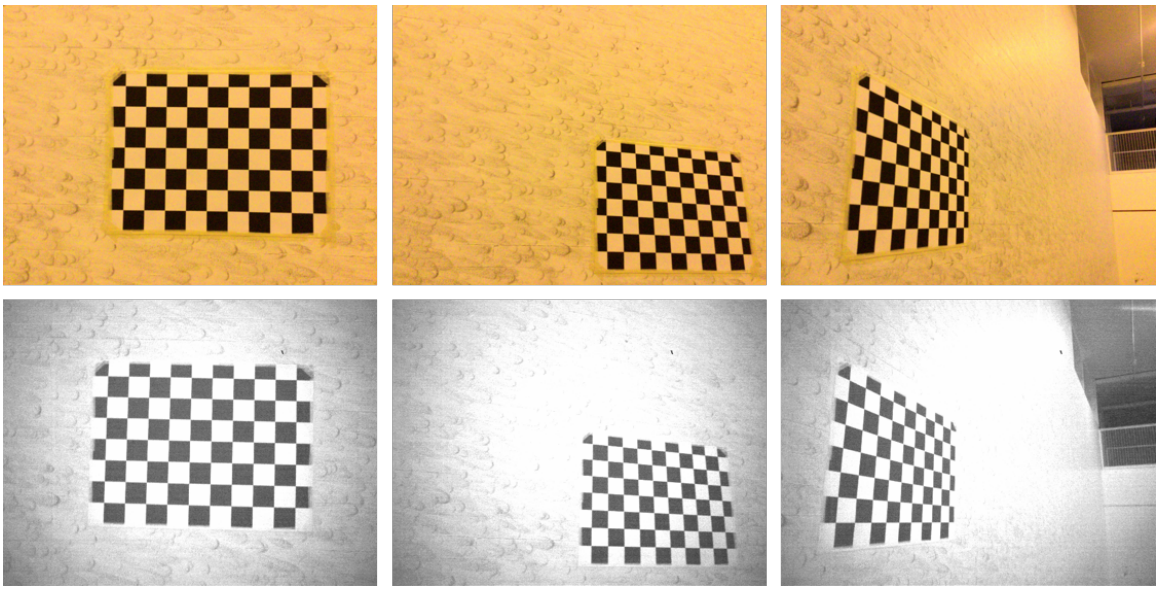


Figure 3.4: Set of image pairs of a calibration grid, taken from color and infrared cameras

3.5 Depth Undistortion

3.5.1 Model

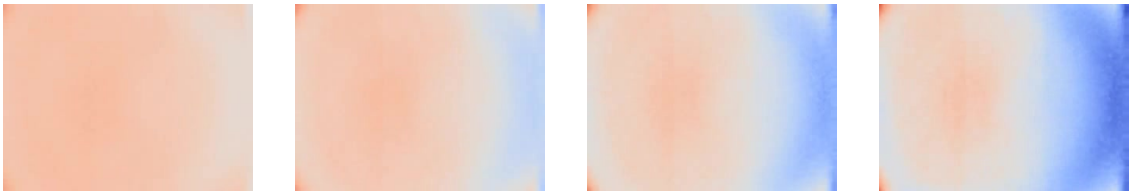
The intrinsic parameters in the previous section do not model the entire behavior of a PrimeSense camera. When analyzing depth maps produced by the Occipital

Structure Sensor, one can notice significant distortion effects, especially near the image corners 3.2. As mentioned before, the PrimeSense cameras generate depths based on a proprietary algorithm, so the exact source of these errors is not known. The previous work [Di Cicco et al., 2015] points out that some of the errors come from the non-perfect calibration of the infrared camera and the projector, which together also form a stereo-camera pair. Additionally, it is supposed that the errors near image corners might be a result of particularities of the block-matching algorithm performed by the PrimeSense PS1080 chip [Martinez and Stiefelhagen, 2013]. While it could be possible to recalibrate the infrared projector and the infrared camera pair, as well as to investigate other ways to estimate the depth, the cost of it would be losing real-time depth estimation that these devices offer.

As a practical solution, prior work [Teichman et al., 2013] proposes a non-parametric way to deal with the depth distortion issues. One can observe that the distortion is dependent on both the magnitude of the estimated depth and the location within image plane. As such, the undistortion can be achieved by estimating a function $U(u, v, \hat{d})$, such that the ground truth depth d is calculated as a product of the measured depth $\hat{d}$ and the value $U(u, v, \hat{d})$.

$$d = U(u, v, \hat{d})\hat{d}$$

Additionally, we know empirically that the distortion function is slowly varying. This motivates the use of a look-up table $\mathbf{U}$ to model $U(u, v, \hat{d})$. Prior work [Teichman et al., 2013] explains this approach by assuming that the observed depth $\hat{d}$ is generated as a product of an unknown multiplier w and the ground truth depth d, under Gaussian noise. That formulation leads to finding a multiplier that is acting on the observed depth d. We adopt a similar strategy but aim to directly find the multiplier m that affects the observed depth $\hat{d}$. As such, for each cell k of $\mathbf{U}$ we seek



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

Figure 3.5: Depth slices of **U**

a multiplier m_k which minimizes the distance between the desired depth d and the product $m\hat{d}$.

$$\arg\min_{m_k} \sum_{i}^{N} (d_i - m_k \hat{d}_i)^2 \tag{3.5}$$

Since the error is quadratic, we can easily find that the minimum at

$$m_k = \frac{\sum_i \hat{d}_i d_i}{\sum_i d_i}$$

where we sum over all $u, v, \hat{d}$ that fall into the specific cell k of U. In our implementation, the size of table **U** is $320x240x15$, with a maximum distance of $d_{max} = 6m$. Given triplet $u, v, \hat{d}$, the index k is calculated as:

$$k = \left\lceil s_d \mathbf{U}_x \mathbf{U}_y \hat{d} \right\rceil + s_y \mathbf{U}_x v + s_x u$$

$\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_d$ return width, height and depth of the table U, s_x, s_y, s_d are scalar factors which convert each of the $u, v, \hat{d}$ to range accepted by the table. For example $s_d = \frac{\mathbf{U}_d}{d_{max}}$. Figure 3.5 shows a visualization of depth slices of **U**.

3.5.2 Optimization

Optimizing for parameters m_i is straightforward — the only issue is the generation of the training data, in the form of depth pairs $\{d_i, \hat{d}_i\}$. Training data constitute pairs of observed depth $\hat{d}_i$ and ground truth depth d_i. The procedure for the collection of training pairs is our main contribution. We first describe how previous work decides

(a) (b)

Figure 3.6: Visualization of flat wall at different depths. (a) Backprojected depth image of a flat wall at two meters away. A distortion is still present, even at such moderate depths. (b) Backprojected depth image of a flat wall at 5 meters away. The depth quantization artifacts are significant.

to solve the training data acquisition problem. We follow with providing our improved solution.

In work of [Teichman et al., 2013], the authors propose to capture the calibration maps. To generate such maps, the authors first estimate the camera trajectory using a SLAM system. Then a 3D model is generated by using the depth provided by the depth sensor. However, the reconstruction algorithm only uses depths smaller than $2m$. The rendering of the reconstructed model from the point of view of each camera provides ground truth depths. Combined with the values returned by the sensor, authors obtain the training pairs. Apparent issues with this approach include the necessity of creating the calibration map, introducing the dependence of calibration on the SLAM system performance. Moreover, it is unclear whether the renderings of the created map cover the entire view frustum. Without this, some training data might be missing. Lastly, this work assumes depths that are less than two meters are correct, which we find to be generally not true (see figure 3.6a).

Follow-up by [Di Cicco et al., 2015] simplifies the training data acquisition, by proposing to capture a sequence of images containing a large, flat surface. They estimate the equation of the observed plane using the depths reported in the center of an image. The plane equation is then used to generate the ground truth depths by intersecting a ray from camera origin through a pixel $\mathbf{x} = [u\ v]^\intercal$ with the estimated

plane. While alleviating the issue of cumbersome calibration map creation, this approach relies on the assumption that depths within the image center are reliable. In our experiments, this is not true, especially at larger distances (see figure 3.6b).

We propose to capture the data in a way similar to the approach presented in [Di Cicco et al., 2015]. We capture a set of n color and depth image pairs of a large planar surface. However, instead of utilizing the depth maps to estimate a plane equation of the observed plane, we instead include the calibration grid, which is observed by the color camera. With multiple color images, we can estimate the 3D locations of grid points. Locations of calibration grid corners can be estimated by first detecting the corners in color images. Since the calibration grid structure is known, we have implicit correspondence between m calibration grid corners $g_i; i \in \{0, m\}$. We also note that we know the camera intrinsic parameters $\mathbf{K}_\mathcal{I}$ from the previous step. As such, we wish to recover the world positions of the calibration grid G_i and the transformations of each of the color cameras $\mathbf{R}_i$, $\mathbf{t}_i$. We thus arrive at a standard structure-from-motion formulation [Snavely et al., 2006] of a reprojection error:

$$\arg\min_{\mathbf{R}_i, \mathbf{t}_i} \sum_{j}^{n} \sum_{i}^{m} ||g_{ij} - P(\mathbf{R}_j G_i + \mathbf{t}_j, \theta_P^i)||^2 \tag{3.6}$$

Once we estimate the camera locations and grid positions, we can use the rigid body transformation $[\mathbf{R}, \mathbf{t}]$ to transform points G_i into the depth camera space and estimate the plane equation. We follow to compute the ground truth depths by intersecting the ray through a pixel $\mathbf{x} = [u\ v]^\intercal$ with the plane $\mathbf{p}$. As a result, we do not depend on the depth camera in any way to generate the training pairs. At the same time proposed approach only requires a laser printed calibration grid, a large planar surface and a short scanning time.

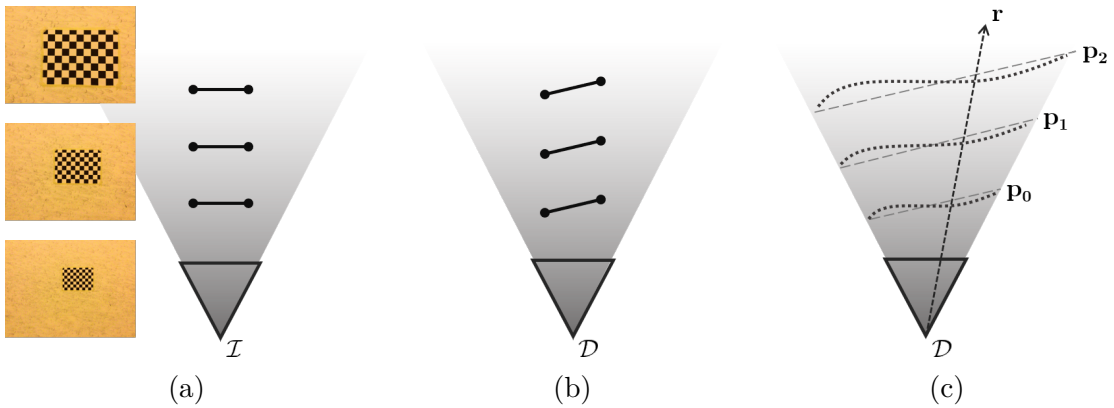


Figure 3.7: Process of generating the training pairs $\{d_i, \hat{d}_i\}$. (a) We first estimate positions of calibration grid corners in the color camera space. (b) We transform the grid corners locations into the depth camera space, and detect planes $\mathbf{p}_j$. (c) We generate the pairs $\{d_i, \hat{d}_i\}$ by intersecting plane $\mathbf{p}_j$ with ray $\mathbf{r}$ through pixel $\mathbf{x}$. d_i is given as distance from the intersection point, and $\hat{d}_i = \mathcal{D}(\mathbf{x})$.

3.6 Results

We present results for both stages of the calibration pipeline. First, we showcase a simple before-and-after comparison for the depth-to-color alignment — see figure. 3.8. Second, figure 3.9 showcases results of the depth undistortion procedure, visualizing the backprojected depth maps of a flat wall at different distances. One can see how the distortion effect becomes more pronounced with larger depths. At the same time, the proposed undistortion procedure is able to produce results that have less variance and are more faithful to the underlying geometry. Additionally, we also present error reduction statistics in table 3.2. For each depth range, we calculate distance from the ground-truth plane (see sec. 3.5) and the values in the depth maps, both raw and undistorted. For small distances the error reduction is negligible, however as soon as the depth increases to above $1m$, the error reduction becomes significant — between 30% to 70%.

Lastly, we also showcase how reducing the depth error affects the 3D reconstruction. We reconstruct RGBD sequences using both calibrated and non-calibrated depth data. To prevent confounding the effects of the depth-to-color calibration with a ro-

Figure 3.8: Before-and-after comparison of the estimated depth-to-color alignment. In the overlay images, It is easy to see that without calibration the objects in both images do not align correctly.

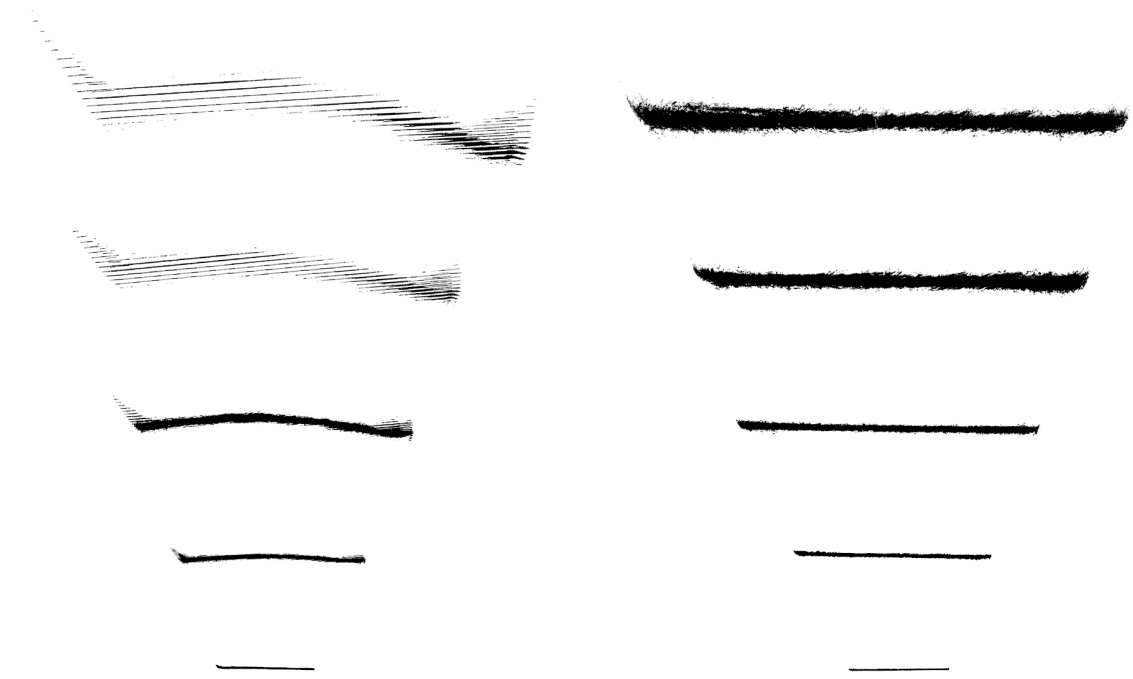


Figure 3.9: Visualization of depth maps of a flat wall, captured at different distances (one to five meter range) **Left:** Raw depth maps. **Right:** Undistorted depth maps.

Dist. Range	Raw	Calibrated
$0 <= d < 1$	0.020239 (0.003663)	0.018943 (0.004324)
$1 <= d < 2$	0.031677 (0.013251)	0.019190 (0.008169)
$2 <= d < 3$	0.055832 (0.043240)	0.021660 (0.014711)
$3 <= d < 4$	0.096119 (0.085278)	0.032648 (0.024181)
$4 <= d < 5$	0.162745 (0.151273)	0.043619 (0.034148)
$5 <= d$	0.208792 (0.181228)	0.065345 (0.075011)

Table 3.2: Statistics (mean and standard deviation) of distance from depth map to ground truth plane.

bust reconstruction algorithm, we use only a simple tracking algorithm (for details, see section 4.4.1). Note, however, that we do apply the depth-to-color calibration in both cases so that the tracking algorithm has access to the correct correspondence between the color and depth images. The results can be seen in figure 3.10. One can see that with the depth undistortion procedure, even a basic tracking algorithm can produce good camera poses.

3.7 Conclusion

In this section, we have discussed the ways to calibrate the PrimeSense-class depth cameras, improving their applicability to the indoor scene reconstruction applications. The proposed approaches can successfully estimate the correspondence between color and depth images, as well as rectify the issues present in the acquired depth images.

Future work directions include a more in-depth analysis of the depth estimation algorithms performed on the PrimeSense PS1080 chip. While the proposed non-parametric undistortion seems to work well in practice, it does not model the source of the errors. Such an analysis could allow for the introduction of algorithms that produce higher quality depth images.

Additionally, the proposed lookup table approach confounds the error coming from the inherent measurement noise, and the low-frequency distortion. An alternative

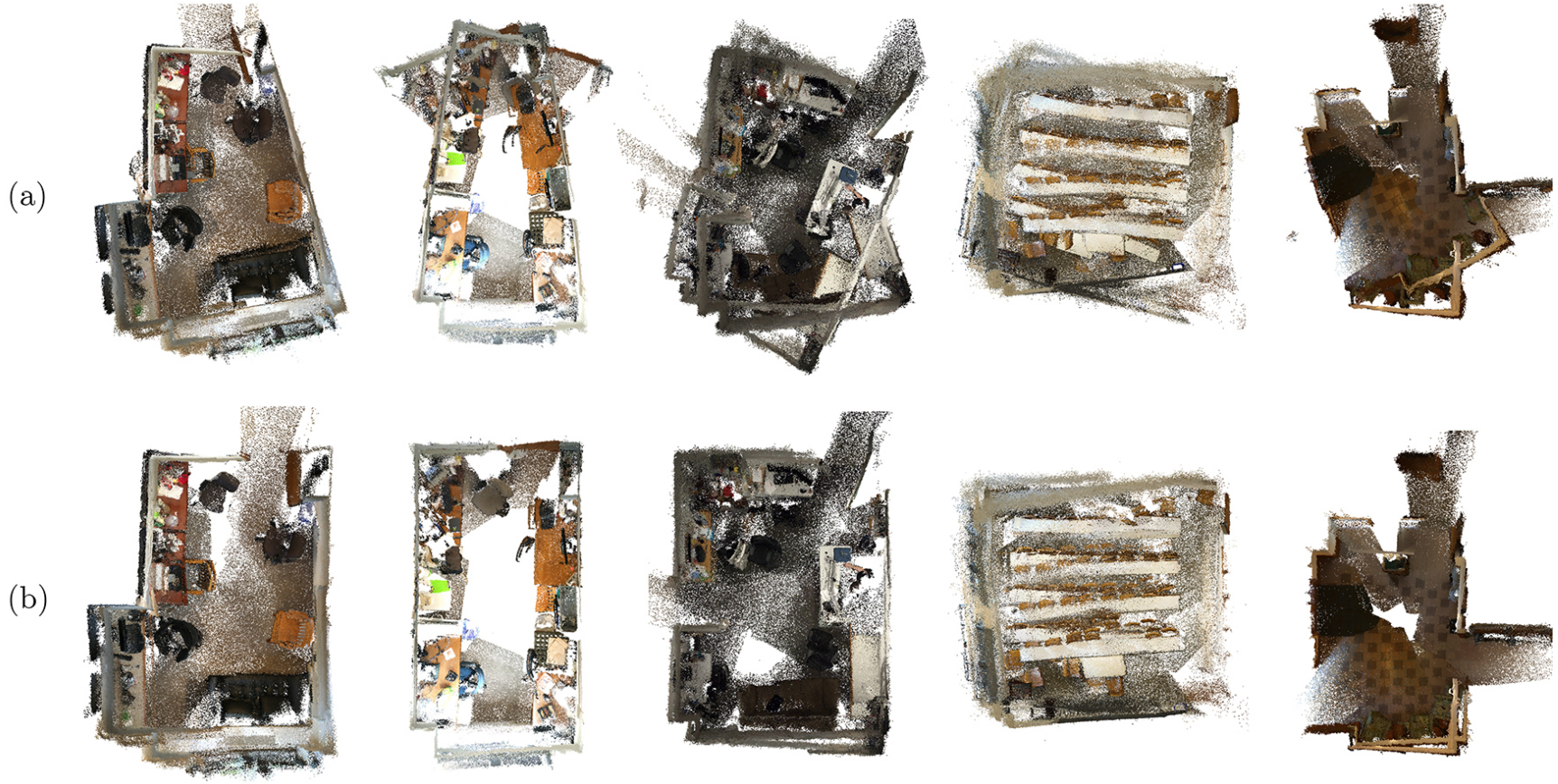


Figure 3.10: Reconstructions using uncalibrated(a) and calibrated(b) depth images.

approach could decompose these two sources of error, to first undistort the captured depth map and then apply a robust filtering method using an estimated noise model. Such an approach could lead to an even higher quality of the obtained depth maps.

Chapter 4

Fine-to-Coarse Registration

4.1 Introduction

In Chapter 3, we explore ways to prepare an output of a commodity RGBD camera for use in an indoor scanning application. However, the task of camera tracking remains challenging, even with the improved data. In some settings, we might also not be able to apply the calibration techniques presented in the previous chapter. As such, we explore methods to robustly estimate the camera poses, even when the input depth data is uncalibrated. While the camera poses can be tracked confidently over short distances [Newcombe et al., 2011], local tracking methods can still fail in feature-less regions. Moreover, even small errors in camera tracking can lead to significant drift over long trajectories [Whelan et al., 2012, Whelan et al., 2015a, Nießner et al., 2013] (left side of the figure 4.1). One can address the drift-related issues with global optimizations based on the detected loop closures [Choi et al., 2015, Henry et al., 2010, Whelan et al., 2015b]. However, finding the loop closures is difficult in the large real-world scans with multiple rooms or repeated structures. In our experience, even state-of-the-art global registration meth-

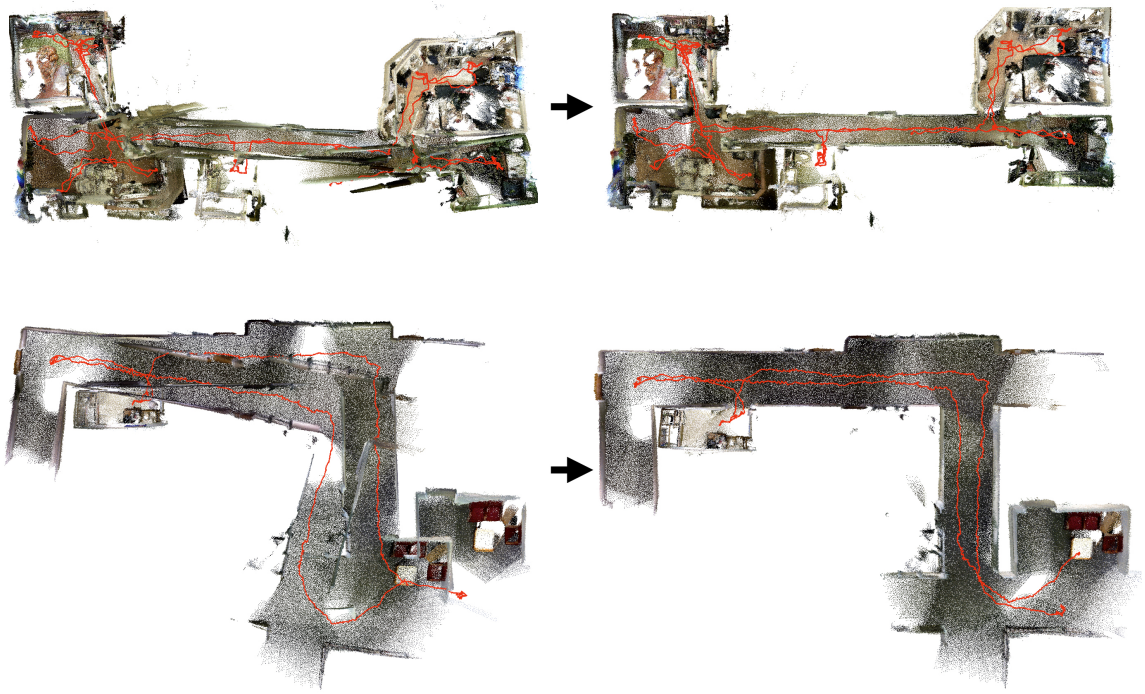


Figure 4.1: Given an initial registration (left), fine-to-coarse registration algorithm iteratively detects and enforces planar structures and feature correspondences at increasing scales. This way, it discovers long-range constraints important for a globally consistent registration – e.g., note how opposing walls are parallel even across different rooms in our results on the right.

ods [Choi et al., 2015] produce warped surfaces and improbable structures in these cases.

Prior work proposes a variety of global refinement methods to generate geometrically plausible output. Works by [Li et al., 2011], [Ma et al., 2016], and [Monszpart et al., 2015] all propose energy functions based on structural models fitted to the input data. [Brown and Rusinkiewicz, 2007], as well as [Yu et al., 2015], aim to align scans based on the closest point correspondences. However, these methods only succeed when the alignments provided as input are nearly correct. Otherwise, they may detect and amplify erroneous constraints found in the misaligned inputs.

We introduce a new "fine-to-coarse" global registration algorithm that refines initial estimates of camera poses by iteratively detecting and enforcing geometric constraints. The algorithm creates constraints only within local subsets of the trajectory and increases the extent of these subsets as the iterations progress. Specifically, during each iteration, closest point and geometric constraints (parallelism, perpendicularity) are detected and enforced only within "windows" of neighboring RGBD frames. Windows start small, such that relative initial alignments are likely to be correct. As the algorithm proceeds, windows gradually increase in size, enabling the detection of longer-range correspondences and large-scale geometric structures. The detection of long-range constraints is possible as the algorithm can leverage the improved trajectory provided by previous iterations. This process continues until a single window includes the complete scan, and a global refinement can be done robustly.

The advantage of this "fine-to-coarse" approach is that the closest point correspondences and planar structures are detected in each iteration only at the scales at which previous iterations have already aligned the scans. Enforcing these constraints in one iteration improves the registration for the next one. For example, in figure 4.2, note how the geometric constraints between walls become easier to detect in each iteration (left to right). Enforcement of those constraints gradually rectifies the reconstruction. As the algorithm continues to consider longer and longer subsets of the trajectory, the alignment is almost perfect, making it trivial to detect very large-scale structures and long-range constraints (e.g., parallel walls in different rooms), which are crucial for correct global registration.

To evaluate this algorithm and enable comparisons to the previous work, we have created a new registration benchmark based on the SUN3D dataset [Xiao et al., 2013a]. It contains 10,401 manually-clicked point correspondences in RGBD scans containing 149,011 frames in 25 scenes, many of which span multiple rooms. During experiments with this new benchmark, we find that our fine-to-coarse

algorithm produces more accurate global registrations and handles more difficult inputs than previous approaches.

In summary, our contributions are:

- We propose a new fine-to-coarse, iterative refinement strategy for global registration of large-scale RGBD scans.
- We introduce a new benchmark dataset for evaluating global registration algorithms quantitatively on real RGBD scans.

4.2 Related Work

There exists a long track of research on registration of RGBD images in both computer graphics and computer vision, as well as in augmented reality, robotics, and other fields [Stotko, 2016]. In the following sections, we describe work that is closely related to ours, both from the tracking, as well as model-fitting perspectives.

4.2.1 Real-time reconstruction

Most prior work has focused on real-time registration motivated by SLAM applications in robotics and augmented reality [Stotko, 2016]. Early systems use ICP [Besl and McKay, 1992] to estimate pairwise alignments of adjacent video frames and feature matching techniques [Angeli et al., 2008] to detect and align loop closures. More recent methods align frames to a scene model, represented as a point cloud [Henry et al., 2010, Keller et al., 2013a, Whelan et al., 2015b] or an implicit function [Chen et al., 2013, Dai et al., 2016a, Kahler et al., 2015, Newcombe et al., 2011, Wang et al., 2016, Whelan et al., 2012, Whelan et al., 2014]. With these methods, small local alignment errors can accumulate to form gross inconsistencies at large scales [Klingensmith et al., 2015, Nießner et al., 2013].

4.2.2 Offline global registration

To rectify misalignments in online camera pose estimates, one can utilize offline or asynchronously executed global registration procedures. A typical formulation is to compute a pose graph with edges representing pairwise transformations between frames and then optimize an objective function penalizing deviations from these pairwise alignments [Grisetti et al., 2010, Henry et al., 2010, Zhou and Koltun, 2013, Zhou and Koltun, 2014]. A significant challenge in these approaches is to identify which pairs of input frames constitute a loop closures. Previous methods have searched for similar images with Bag-of-Words models [Angeli et al., 2008], randomized fern encodings [Whelan et al., 2015b], convolutional neural networks [Chen et al., 2014], and other methods. [Choi et al., 2015] recently proposed a method that uses indicator variables to identify actual loop closures during global optimization using a least-squares formulation. In our experiments, their algorithm is successful on scans of small environments, but not for ones with multiple rooms, large-scale structures, or many repeated elements.

4.2.3 Hierarchical graph optimization

Prior methods also propose to fuse subgraphs of a pose graph hierarchically to improve optimization robustness and efficiency [Choi et al., 2015, Estrada et al., 2005, Frese et al., 2005, Ratter and Sammut, 2015, Tang and Feng, 2015]. Some of the ideas motivating these methods are related to ours. However, they detect all potential loop closures before the optimization starts. In contrast, we detect new constraints (planar relationships and feature correspondences) in the inner loop of an iterative refinement, which enables the gradual discovery of large-scale structures and long-range constraints as the registration improves.

4.2.4 Iterative refinement

Other methods have used Iterative Closest Point [Besl and McKay, 1992] to compute global registration [Brown and Rusinkiewicz, 2007, Yu et al., 2015, Pulli, 1999]. The advantage of this approach is that the dense correspondences (including loop closures) are found only with local queries for closest points based on prior alignments, rather than with global search that considers all pairs of frames. However, ICP generally requires a good initial alignment and thus is rarely used for global RGBD registration, except as fine-scale refinement in the last step [Choi et al., 2015]. Our "fine-to-coarse" strategy addresses that specific limitation.

4.2.5 Planar feature detection

Several methods, like [Bartoli and Sturm, 2003, Weingarten and Siegwart, 2006, Nguyen et al., 2007, Pathak et al., 2010, Salas-Moreno et al., 2014, Dou et al., 2012, Trevor et al., 2012, Elghor et al., 2015, Taguchi et al., 2013, Ma et al., 2016], detect correspondences between planar features. They motivate this choice by noting that planar features are easier to extract and can be matched in noisy, partial scans. [Ma et al., 2016] and [Salas-Moreno et al., 2014] go further by associating planar features with global planes estimated with an E-M algorithm. [Zhang et al., 2015] use detected planes and relationships between them to update a volumetric model. We build upon this body of work by introducing a hierarchical structural model of plane detections built fine-to-coarse in the inner loop of an iterative global registration algorithm.

4.2.6 Extracting structural models

Other methods have been proposed for beautifying 3D reconstructions as a post-process [Li et al., 2011, Monszpart et al., 2015, Nguyen et al., 2015]. However, they

only work when scans are already almost perfectly aligned — they cannot detect large-scale structures and relationships to rectify when scans are severely misaligned. We draw upon many ideas in these papers, including the RAP (regular arrangement of planes) [Monszpart et al., 2015] representation encoding local planar regions and global inter-plane relationships. However, we extend them to the more common case in which RGBD images are not registered in advance.

4.3 Approach

We describe a global registration algorithm that leverages the detection and enforcement of nearly-satisfied constraints in the inner loop of an iterative refinement procedure. The algorithm starts with initial, imperfect registration and then follows the general E-M strategy of alternating between a discrete E-step (detecting a set of viable constraints) and a continuous M-step (solving for the camera poses that best satisfy the detected constraints).

Though the method is general, we consider two major types of constraints in this work: feature correspondences and planar structure relationships. During each iteration of the algorithm, constraints are created based on correspondences between closest compatible features (like in ICP) and based on geometric relationships between detected planar structures (parallelism, orthogonality). The constraints are integrated into a global optimization that refines camera poses before proceeding to the next iteration.

The key new idea is that the detection of constraints occurs in every iteration within sliding windows that grow gradually as the algorithm proceeds. In the early iterations, only a small number of neighboring RGBD frames are within each window. Since the relative camera poses of the initial alignment should be nearly correct for such neighboring frames, it is possible to detect geometric constraints and closest

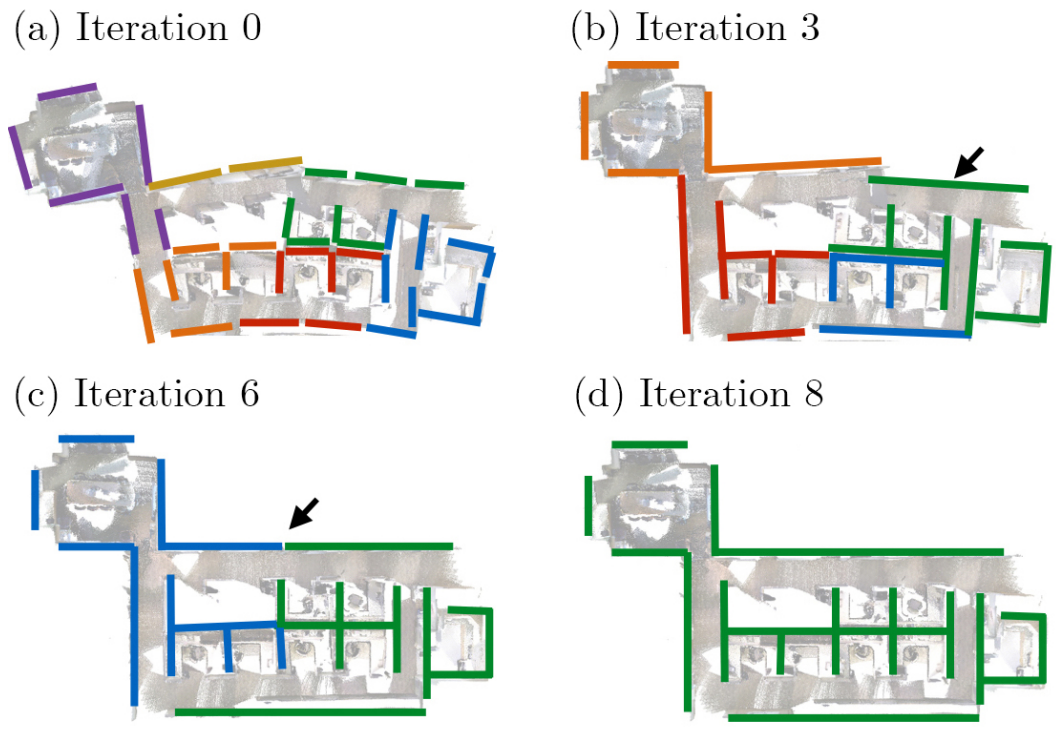


Figure 4.2: Schematic view of fine-to-coarse registration. Starting with initial alignment T_0 shown on the left, our algorithm detects and enforces structures in local regions (color-coded) in the first few iterations. As the algorithm progresses, the trajectory is refined, allowing for the detection of larger geometric structures. By iteration 6, we have properly aligned the wall marked by the arrow, without using explicit loop closures.

point correspondences robustly, even if the global reconstruction is grossly inaccurate (Figure 4.2a). As the iterations of the algorithm proceed, the window size is increased. As a result, the detection and enforcement of larger-scale and longer-range planar structures and correspondence constraints is possible (Figure 4.2c). Since previous iterations have optimized the camera trajectory based on constraints discovered within smaller windows, we can expect the current trajectory estimate to be nearly correct within each window and use it to discover planar structures and feature correspondences. Ultimately, in the later iteration, the final window contains all the input data, and the algorithm performs global optimization of large-scale structures

and correspondences spanning the entire trajectory in one large joint optimization (Figure 4.2d).

This approach has two essential differences from the previous work. First, it avoids a global search for pairwise loop closures — they are instead found incrementally as the registration becomes nearly aligned. Second, it discovers and enforces large-scale geometric constraints (like planar structure relationships) even though they might not be evident in the initial alignment (e.g., the parallel relationship between the leftmost and rightmost walls in the example of figure 4.2 would be difficult to infer in iteration 0, but is simple to detect in Iteration 6). As a result, our method achieves significantly better registration results for large-scale scans compared to previous methods (see section 5.4).

4.4 Algorithm

The input to our system is a set of n RGBD images I acquired with a consumer level RGBD camera. The output is a set of camera poses T, where $T[k]$ represents the position and orientation of the camera for $I[k]$.

Pseudocode 1 showcases the stages of the proposed method. During a preprocessing phase, we first extract features F and base planar regions B from all images in I. Second, we estimate a set of local, pairwise alignment transformations L, and concatenate these local transformations to form an initial guess for global transformations T_0. Then, in each iteration i, we refine the transformations T_i by including two major sets of constraints. First, we detect feature correspondence constraints C_i by closest point queries. Second, we detect structural model constraints S_i. S_i are based on the detected clusters of coplanar base planar regions P_i and geometric constraints (H_i and G_i) between them. Both sets of constraints are detected only between pairs of RGBD frames whose distance along the trajectory is less than window size l_i. With

Algorithm 1: Fine-to-coarse refinement

Input : Images I, window length l_0, n_iter
Output: Camera transformations T

F = ExtractFeatures(I)
B = CreateBaseProxies(I)
L = AlignAdjacentImages(I)
T_0 = ConcatenateTransformations(L)
for $i \leftarrow 0$ **to** n_iter **do**
    $\{P_i, H_i\}$ = ClusterCoplanarProxies(B_i, l_i)
    G_i = DetectGeometricConstraints(P_i)
    C_i = CreateCorrespConstraints(F_i, l_i)
    $S_i = \{P_i, G_i, H_i\}$
    T_{i+1}=Solve $argmin_T E(T_i, S_i, C_i)$
    $l_{i+1} = 2l_i$
end

the constraints detected, we optimize the global transformations for the next iteration T_{i+1} by minimizing an error function encoding penalties for the detected constraints. We finish by increasing the size of the window by a factor g. The window size for the next iteration is then give as $l_{i+1} = gl_i$. The following subsections describe the core ideas for each of these steps.

4.4.1 Preprocessing

Extracting Features. The first step of preprocessing is to extract a dense set of features F from input RGBD images I. Our goal in this step is to construct a set of well-spaced and repeatable features that can be matched robustly when searching for correspondences. We have experimented with a number of feature types, including SIFT and Harris corners in both color and depth images. However, we have ultimately found planar patches [Bartoli and Sturm, 2003, Dou et al., 2012, Dzitsiuk et al., 2016, Elghor et al., 2015, Ma et al., 2016, Nguyen et al., 2007, Pathak et al., 2010, Salas-Moreno et al., 2014, Taguchi et al., 2013, Trevor et al., 2012, Weingarten and Siegwart, 2006] and linear edges along creases and contours in the

depth images [Zhou and Koltun, 2015] to be most robust. Features are detected per pixel, for every 5^{th} frame, and then subsampled using the Poisson Dart Algorithm, with a minimum spacing between features equal to $0.05m$. We classify detected features into four types:

1. **Plane** All parts of the image that are locally planar, i.e. can be represented as a position and a normal. We construct a planar surface feature for each pixel in coplanar clusters of size ≥ 100.

2. **Silhouette** We have found it useful to mark depth discontinuities with a separate feature type. These are represented as a point, normal and a direction along the silhouette. Normal and direction are estimated with PCA on neighboring silhouette pixels. The same approach is used to estimate direction for **Ridge** and **Valley** features.

3. **Ridge** If two large planes are intersecting and the angle between their normals is larger than π we mark the intersection as a ridge feature.

4. **Valley** - If two large planes are intersecting and the angle between their normals is smaller than π we mark them as a valley feature. Both valley and ridge features are represented as a position, normal (average of the normals of neighboring planes) and direction along the intersection.

Once set F is created, we define a feature from image $I[k]$ at iteration i as

$$F_i[k][j] = \{T_i[k](p_j), T_i[k](\vec{n_j}), T_i[k](\vec{d_j})\}$$

where p_j, $\vec{n_j}$ and $\vec{d_j}$ respectively denote the feature's position, normal (for planar patches) and direction (for linear edges) in the camera space.

Creating Base Planar Proxies. The next step is to extract base planar regions (which we refer to as proxies) B from input images I. Our goal is to create base

proxies that can form the basis for the geometric constraints introduced later during the fine-to-coarse refinement. To do this, we use a method based on the agglomerative hierarchical clustering, where clusters of nearly coplanar features are repeatedly merged based on the compatibilities of their positions and normals. Specifically, we perform the following steps:

- Apply a bilateral filter to reduce noise and quantization effects ($\sigma_{xy} = 3px$, $\sigma_{depth} = 5cm$).
- Mark pixels as boundaries if their depths differ from any of their neighbors by more than 10%.
- Compute connected components by partitioning the image on boundaries
- Estimate normals for pixels using RANSAC on neighborhoods of radius $r = 10cm$ within the same connected component.
- Compute sets of coplanar pixels using hierarchical clustering.
- Refine clusters with a RANSAC algorithm to reassign pixels to their largest compatible cluster.
- For each cluster insert proxy to B. Assign the centroid, normal, and radius for the proxy based on PCA of the associated pixels.

Once B is created, we define a proxy from image $I[k]$ at iteration i as

$$B_i[k][j] = \{T_i[k](p_j), T_i[k](\vec{n_j})\}$$

where p_j, is the centroid of inlier features, and $\vec{n}_j$ is the fitted normal.

Aligning Adjacent Images. The final step of preprocessing is to estimate a set of local alignment transformations L for input images I. Our goal in this step is to

create local alignment transformations that can be used later in the optimization to preserve the local shape of the estimated camera trajectory. To accomplish this goal, we use a pairwise image alignment approach based on [Xiao et al., 2013a]: we detect SIFT features in images $\{I[k-1], I[k]\}$, prune out ones without valid (missing or high) depth values and then use RANSAC on backprojected SIFT keypoints to search for the rigid transformation $L[k]$ aligning as many of these keypoints as possible. We form the initial camera-to-world transformations T_0 by simply concatenating the estimated local transformations L ($T_0[0] = I_{4\times4}$; $T_0[k] = L[k-1]T_0[k-1]$; $k \in [1, n]$). This process gives us an initial set of transformations that are locally accurate, but not globally consistent.

Initializing Transformations. We then concatenate the transformations to create an initial set of poses for our optimization T_0.

$$(T_0[0] = I; T_0[j] = L[j]T0[j-1]; j \in [1; k])$$

4.4.2 Fine-to-Coarse Refinement

After preprocessing the images I, the algorithm iteratively detects constraints within windows of increasing sizes and solves for all camera transformations T based on those constraints. The input to each iteration i is a window size l_i ($l_0 = 3m$) and a set of transformations T_i from the previous iteration. The output is a set of new camera transformations T_{i+1}.

Creating Coplanarity Constraints. We model coplanarity constraints by clustering the transformed base proxies B_i into representative cluster proxies $P_i[j] = \{p_j, \vec{n_j}\}$. Clustering is achieved using agglomerative hierarchical clustering algorithm, similar to the one used for base proxies extraction. However, in this step, rather than clustering features within each individual image, we cluster base proxies from dif-

ferent images whose distance along the estimated trajectory is less than the current window size l_i. To perform the hierarchical clustering we need to specify a way to quantify a similarity between two proxies. We express this similarity as a product of pairwise point-to-plane distance factors f_{pp} and normal deviation factors f_{nd}. For pair of proxies B_a, B_b:

$$S_{ab} = f_{pp}(B_a, B_b) f_{pp}(B_b, B_a) f_{nd}(B_a, B_b) \tag{4.1}$$

Factors are expressed with respect to set maximum thresholds. In our implementation the max pairwise distance was set to $t_d = 30cm$ and the maximum normal deviation to $t_a = \frac{\pi}{8}$

$$f_{pp}(B_a, B_b) = \begin{cases} 1.0 - \frac{d(p_a, B_b)}{t_d} & \text{if } d(p_a, B_b) \geq t_d \\ 0.0 & \text{if } d(p_a, B_b) < t_d \end{cases} \tag{4.2}$$

$$f_{nd}(B_a, B_b) = \begin{cases} 1.0 - \frac{\angle(\vec{n_a}, \vec{n_b})}{t_a} & \text{if } \angle(\vec{n_a}, \vec{n_b}) \geq t_a \\ 0.0 & \text{if } \angle(\vec{n_a}, \vec{n_b}) < t_a \end{cases} \tag{4.3}$$

The result of this procedure constitutes a set of cluster proxies P. Once P is generated we proceed to insert two types of constraints into H_i, a set of feature-to-proxy constraints joining the frame features F_i to B_i, and a set of proxy-to-proxy constraints joining members of B_i to their representative cluster proxy in P_i. The constraint hierarchy implied by H_i is depicted for a single-room example in figure 4.3. Note that the structure is shown for a late iteration, and thus the planar structures in green span entire walls. In contrast to previous methods based on alignment to planes [Ma et al., 2016, Salas-Moreno et al., 2014, Zhang et al., 2015], it is possible for us to detect these large planar structures because previous iterations already aligned overlapping subsets of the walls.

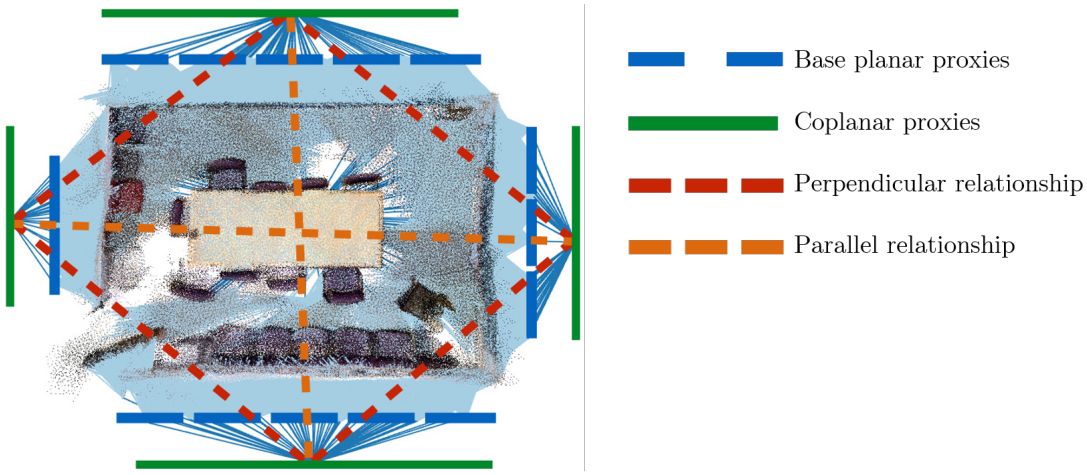


Figure 4.3: Exploded view of our structural model for one of the SUN3D scenes. Geometric properties like parallelism (dashed orange) and orthogonality (dashed red) are created between parent proxies (green). Cluster proxies P_i are connected to the scan features (point-cloud) through base proxies B_i via coplanarity constraints (blue and light blue, respectively).

Creating Geometric Relationship Constraints. Following grouping of the coplanar proxies, we build a set of constraints G_i representing geometric relationships between neighboring cluster proxies from the set P_i. Our goal is to detect salient relationships between planar structures that can help guide the optimization towards the correct registration. We detect three main types of relationships: parallel, antiparallel, and orthogonal. The distinction between the *parallel* and *antiparallel* is that *parallel* relationship requires corresponding vectors to be the same, while *antiparallel* requires them to be opposite.

We create typed and weighted planar relationships for every pair of cluster proxies $\{P_i[a], P_i[b]\}$) such that the distance along a trajectory between the inlier images is less that $2l_i$. The type of the structural relationship g_{ab} and its weight w_{ab} are based on the angle between the normals, $\theta = acos(\vec{n}_a \cdot \vec{n}_b)$.

For parallel relationships the weight is defined as $w_{ab} = \exp(-\theta^2/2\sigma_\theta^2)$, for orthogonal $w_{ab} = \exp(-(\theta - \frac{\pi}{2})^2/2\sigma_\theta^2)$, and for antiparallel $w_{ab} = \exp(-(\theta - \pi)^2/2\sigma_\theta^2)$.

These weights are chosen so as to guide the registration when constraints are nearly met, but have little influence when they are not. For our experiments we have chosen $\sigma = 7.5°$.

Creating Feature Correspondence Constraints. Next, we build a set of correspondence constraints C_i between features detected in images within the same sliding window. Following the general strategy of ICP, we construct correspondences between the closest compatible features. We determine the feature compatibility by a maximum distance and maximum normal angle deviation threshold, as well as a feature type check (i.e., planar features only match to planar features).

Since we expect the alignment of the images within the same window to improve as their poses are optimized, we set the maximum distance and angle thresholds for rejecting outliers dynamically for every pair of images based on their pairwise distance along the trajectory. The first time any two images are considered for correspondence detection (pairwise distance is $0.5l_i$) the thresholds are quite large: $0.5m$ and 45°. Conversely, we expect close-by images to be already aligned well, thus the thresholds fall-off with the square root of decreasing pairwise distance, down to $0.2m$ and 20° for adjacent frames.

Finally, for performance reasons, we subsample the set of correspondences created such that the total number of them is equal to $|C_i| = 25n$.

4.4.3 Optimization

The final step for each iteration i is to optimize the camera transformations T_i and transformations of proxies P_i to minimize an error function encoding the detected constraints.

Our error function is a weighted sum of terms penalizing deformations of structural relationships (E_H, E_G), distances between corresponding features (E_C), mis-

alignments of the local transformations (E_L), and large changes in transformations (E_I).

$$\begin{aligned} E(T_i, S_i, C_i) = w_H E_H(H_i) + w_g E_G(G_i) \\ + w_C E_C(T_i, C_i) + w_L E_L(T_i) + w_I E_I(T_i, P_i) \end{aligned} \tag{4.4}$$

Throughout the iterations weights w_H, w_G, w_C, w_L, w_I are varied linearly from an initial set of $\{1500, 1500, 1500, 1000, 1\}$ to a final one of $\{1000, 1000, 1000, 1000, 1\}$.

Structural Error. E_H and E_G are designed to enforce the constraints implied by the structural model S_i. E_H enforces coplanarity between proxies and their inlier features in depth images. Note that H_i contains both feature-to-proxy and proxy-to-proxy constraints. If we use $Q_a = \{q_a, \vec{n_a}\}$ to represent the transformed plane of a feature or proxy, we can write each error term including all these constraints as:

$$E_H(H_i) = \sum_{j=1}^{|H_i|} (E_{cp}^{\rightarrow}(Q_a, Q_b) + E_{cp}^{\rightarrow}(Q_b, Q_a))$$

where $E_{cp}^{\rightarrow}(Q_a, Q_b) = \sum_{s=1}^{s_{max}} ((q_a - q_b) \cdot \vec{n_b})^2$ measures the deviation of two planar structures from coplanarity. For feature-to-proxy relationships, q_a and q_b are positions of inlier features. For proxy-to-proxy constraints, each q_s is either sampled from the boundary of a 0.5 meter radius disk around p_a, or is at the same location as p_a (in our experiments $s_{max} >= 5$).

The error in geometric relationships $E_G(G_i)$ between proxies $P_i[j]$ and $P_i[k]$ is:

$$E_G(G_i) = \sum_{j=1}^{|G|} \begin{cases} w_{jk}(1 - (\vec{n}_j \cdot \vec{n}_k))^2 & parallel \\ w_{jk}(1 - (\vec{n}_j \cdot (-\vec{n}_k))^2 & antiparallel \\ w_{jk}(\vec{n}_j \cdot \vec{n}_k)^2 & orthogonal \end{cases} \tag{4.5}$$

Feature Correspondence Error. E_C is designed to encourage alignment of detected correspondences between transformed features $F_i[s][a]$, $F_i[r][b]$:

$$E_C(T_i, C_i) = \sum_{j=1}^{|C_i|} \begin{cases} ((p'_b - p'_a) \times \vec{d'_a})^2 & edges \\ ((p'_b - p'_a) \cdot \vec{n'_a})^2 & planes \end{cases} \tag{4.6}$$

where p'_a, n'_a, d'_a and p'_b denote feature attributes transformed using respective transformations $T_i[s], T_i[r]$.

Local Alignment Error. E_L is designed to encourage pairwise transformations between adjacent frames to match the ones computed during preprocessing:

$$E_L(T_i) = \sum_{j=0}^{n-1} \sum_{k=0}^{k_{max}} E_t(T_0[j + 2^k]^{-1}(T_0[j]), T_i[j + 2^k]^{-1}(T_i[j])) \tag{4.7}$$

where $k_{max} = 16$ and E_t measures the misalignment of transformation $T[j]$ to another $T[k]$. We compute E_t by summing the squared distances between points p_s ($s \in [1, 8]$) sampled uniformly on a 1 meter radius sphere when they are transformed by $T[j]$ versus $T[k]$:

$$E_t(T[j], T[k]) = \sum_{s=1}^{s_{max}} (T[j](p_s) - T[k](p_s))^2 \tag{4.8}$$

Inertia Error. E_I is added to provide stability for the optimization and prevent the system of equations from being under-constrained. Here we denote transformation of proxy $P_i[j]$ as $T^{P_i}[j]$.

$$E_I(T_i, P_i) = \sum_{j=1}^{|I|} (\Delta T_i[j])^2 + \sum_{j=1}^{|P_i|} (\Gamma T^{P_i}[j])^2 \tag{4.9}$$

ΔA represents the sum of squared differences between Euler angle rotations and translations for A from one iteration to the next. ΓA is identical to ΔA when the previous transformation is an identity.

4.5 Experimental Results

We performed a series of experiments designed to test the performance of the proposed method with comparisons to previous methods and ablation studies.

4.5.1 Benchmark Dataset

RGBD scans of indoor scenes with ground truth alignments are scarce. Most existing dataset contain only part of a room [Anand et al., 2012, Dai et al., 2016a, Handa et al., 2014, Lai et al., 2014, Mattausch et al., 2014, Silberman et al., 2012, Sturm et al., 2012], contain less than ten test examples [Mattausch et al., 2014, Xiao et al., 2013a], or are based on synthetic data [Handa et al., 2014]. As a result, the research community has compared registration results on small, clean datasets, not representative of the large real-world scans required for most applications.

To address this issue, we introduce a new registration benchmark based on the SUN3D dataset [Xiao et al., 2013a]. SUN3D contains a large set RGBD videos captured with an ASUS Xtion Pro sensor attached to a hand-held laptop in a variety of spaces (apartments, hotel rooms, classrooms, etc.). Each scan contains $10^3 - 10^4$ images, often covering multiple rooms. Previously, only eight of the scenes were released with full annotations and pose correction. Because of the lack of ground truth poses, these have not been used for quantitative evaluation of registration algorithms.

Moreover, we note that the SUN3D data is available only with a factory calibration applied to the input images. As a result, it is not possible to use techniques presented in Chapter 3 to improve the data quality. Here we demonstrate that the proposed

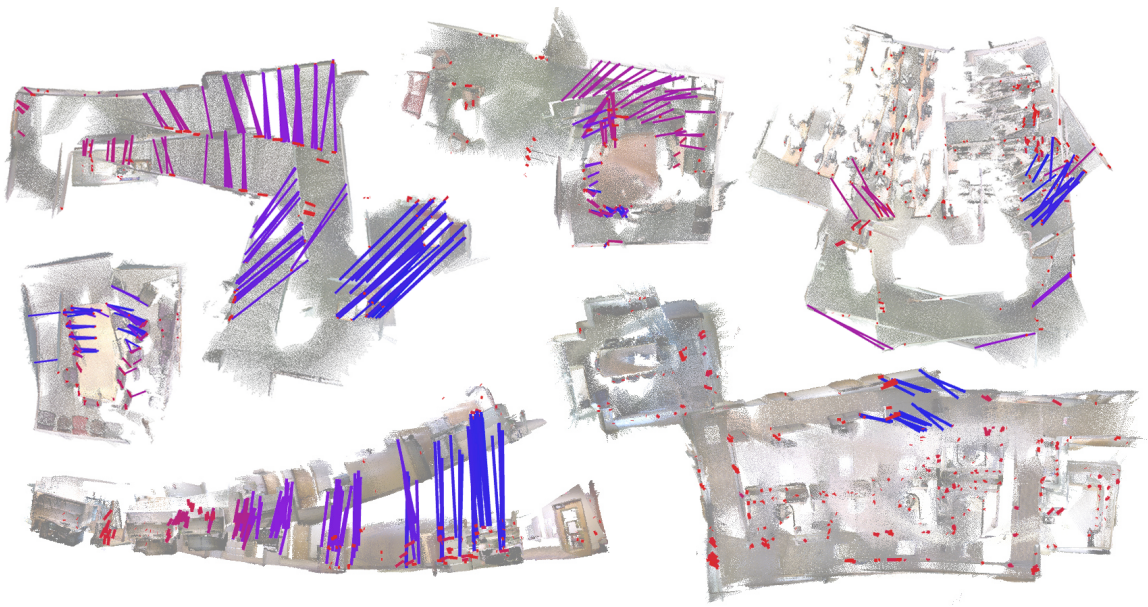


Figure 4.4: Ground truth correspondences for 6 out of 25 scenes in our benchmark. The visualization shows lines between manually-clicked corresponding points after alignment with T_0, the initialization for our method. Color indicates the frame distance — blue denotes loop closure pairs, while red denotes local pairs.

method is robust enough to deal with imperfect data as well. While we are able to recover high-quality camera poses despite issues with depth images, calibration of the capture device is generally advised as it can improve the visual fidelity of the final reconstruction.

We provide a dataset of ground-truth point correspondences for 25 of the largest scenes in SUN3D. It contains 10,401 point correspondences with pixel-level accuracy. These ground-truth correspondences are mostly in pairs of overlapping frames forming loop closures, but they also appear in pairs of nearby frames spread evenly throughout the sequence, as shown in figure 4.4. The average number of correspondences per scan is 416, with a minimum of 239 and a maximum of 714.

We have designed a user interface to collect the set of ground truth correspondences and to evaluate the quality of the collected data. In our interface, the user is presented with view of image pairs (see figure 4.6a). The user can freely scroll through the sequence to select corresponding pairs. Once a pair of images is marked,

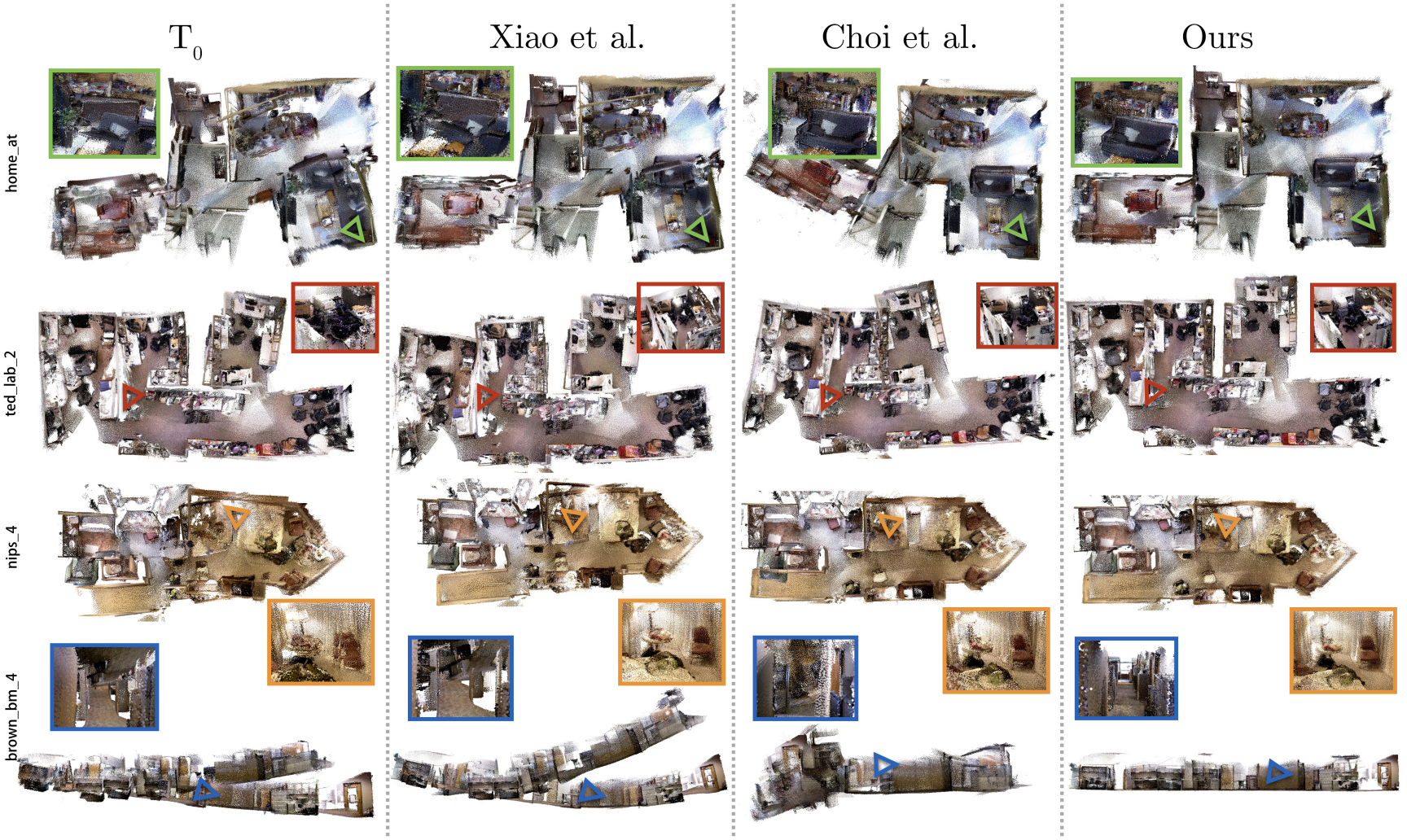
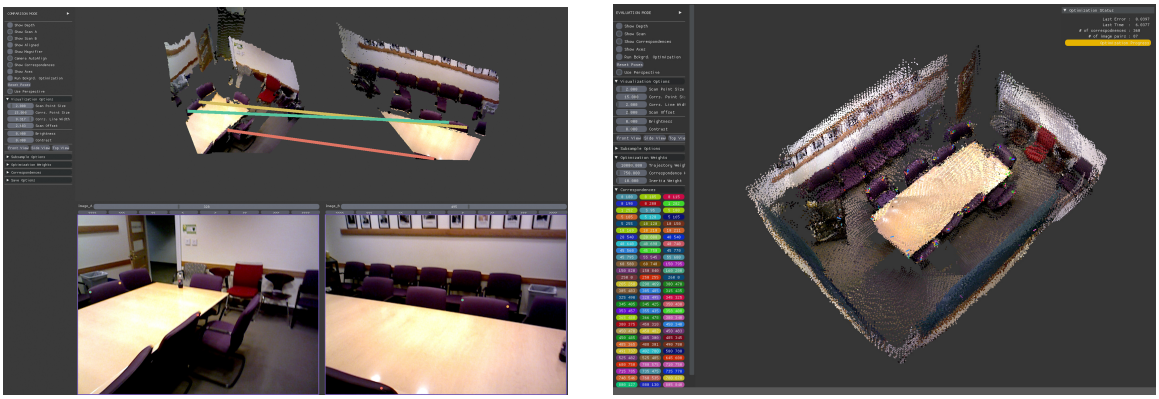


Figure 4.5: Qualitative comparison of global registration results for a subset of SUN3D scenes. The rightmost column shows our results. The leftmost column shows the solution used to initialize our algorithm (T_0). The middle two columns show results produced with prior work [Choi et al., 2015, Xiao et al., 2013a]. In insets, we show close-ups of particular regions. In the first two rows, our method can recover the correct arrangement of captured multi-room environments, while previous work produces improbable structures, like intersecting rooms. The third row shows a sequence with non-Manhattan walls, which we can register correctly. Our method is also able to correctly align a challenging corridor sequence in the fourth row, where for Xiao et al., the visual place recognition has failed. Due to a lot of geometric self similarities, Choi et al. is unable to recover proper geometry.



(a) (b)

Figure 4.6: (a) Interface for collecting ground truth correspondences. Bottom panel allow users to select image pairs for which they wish to create correspondences between. (b) Interface showing resulting reconstruction. On the side panel you can see colored button relating to pairs of images marked by the user.

the user can begin clicking points. Anytime a point is clicked the 3D view is also updated, which gives the user a sense where exactly a point lands in 3D space. As we mentioned, the proposed interface also has an evaluation mode, which runs an iterative reconstruction algorithm with pairwise transformations and ground truth correspondences as constraints (see figure 4.6b). We found this to be an easy way to validate that the clicked correspondences are correct, as an erroneous ground correspondence will lead to big errors in the resulting reconstruction and can be easily fixed by the user.

We use these ground truth correspondences to evaluate and compare RGBD registration algorithms by computing their root mean squared error (RMSE). To quantify a lower bound on the RMSE in this test, we have aligned the ground truth correspondences for all scenes with no other constraints and report the errors in the left column of Table 4.1. Note that these lower-bounds are non-zero, even though clicked correspondences are pixel-accurate. This error is due to the extreme noise in the uncalibrated SUN3D depth maps.

	Ground Truth	Ours	T_0	Xiao et al.	Choi et al.
Average	0.031	0.073	0.519	0.425	0.999
Standard Deviation	0.006	0.023	0.394	0.493	1.464
Median	0.031	0.065	0.410	0.214	0.247
Minimum	0.019	0.040	0.118	0.078	0.047
Maximum	0.045	0.139	1.560	2.001	5.901

Table 4.1: Comparison of RMSE statistics in meters with different registration methods for the 25 scenes in our SUN3D benchmark.

4.5.2 Evaluation

We evaluate our method in comparison to two prior methods for offline registration: Xiao et al.'s Sun3DSfm[Xiao et al., 2013b] and Choi et al.'s Robust Reconstruction of Indoor Scenes [Choi et al., 2015] (see figure 4.5). The first method by Xiao et al. uses a similar method for tracking, but also predicts loop closures via visual place recognition with a BoW approach and performs a global bundle adjustment to optimize for camera poses. The second method by Choi et al. fuses consecutive groups of 50 frames into fragments, aligns all pairs of fragments with a variant of RANSAC, selects pairs as potential loop closures, and then solves a least squares system of nonlinear equations that simultaneously solve for camera poses and loop closure weights. At the time, the method by Choi et al. was the state-of-the-art for offline global registration amongst ones with code available, even though it only uses the depth information.

Table 4.1 and figure 4.7 show quantitative results for the comparison evaluated on our new SUN3D benchmark. Table 4.1 compares overall statistics of RMSEs for each algorithm, while the figure 4.7 shows the distributions of RMSEs. It can be seen in both of these results that our reconstruction algorithm aligns the ground truth correspondences better than either of the other two methods: our median error

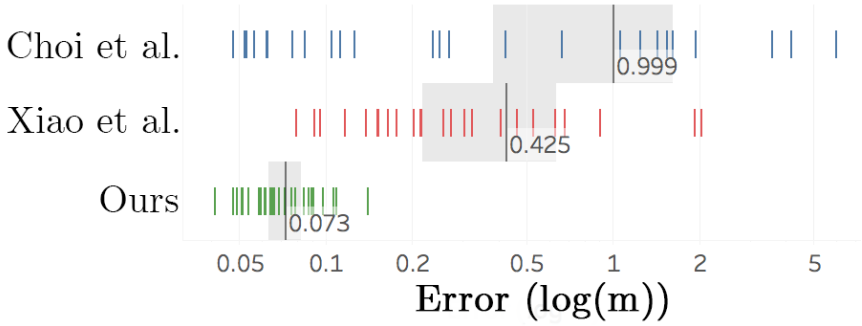


Figure 4.7: **Quantitative comparison.** Every vertical bar in each row represents the RMSE achieved for one of the 25 SUN3D scenes with the algorithm listed on the left. The vertical gray bar shows the average RMSE for each method, and the shaded gray regions represent one standard deviation.

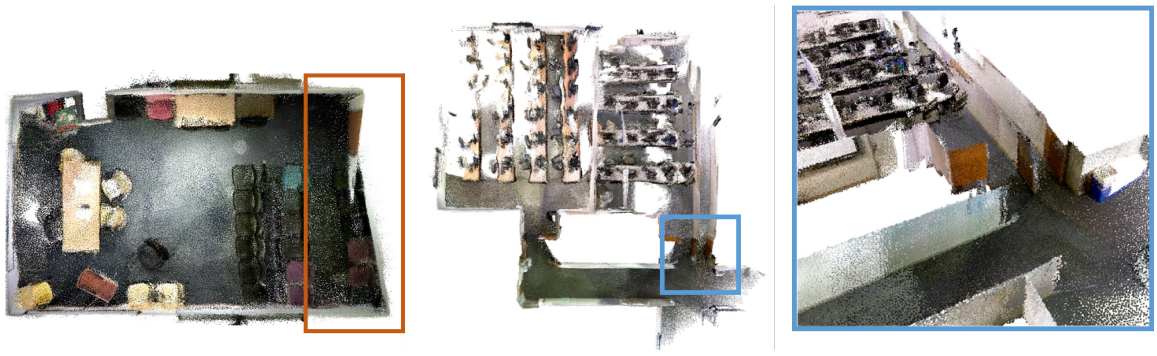


Figure 4.8: Failure cases of our method. Left: the real world room is a trapezoid. Our structural model introduces error, attempting to create a rectangular room. Right: Sliding along the corridor (in blue) causes failure in detecting loop closures. Note the misaligned wall on the right marked by an arrow.

is 0.065m in comparison to 0.214m for Xiao et al. and 0.247m for Choi et al. In case-by-case comparisons, our method has the lowest error in 21 of 25 scenes.

Failure Cases. Our method does not always succeed. For example, it can fail when multiple rooms are connected via featureless straight corridors and when rooms that are nearly (but not exactly) rectangular (figure 4.8). Failures of the second type are rare – since the weight of enforcing parallelism and orthogonality constraints is low for pairs of planes at off-angles, we are able to reconstruct most scenes with non-Manhattan geometry correctly (as in the third row of figure 4.5).

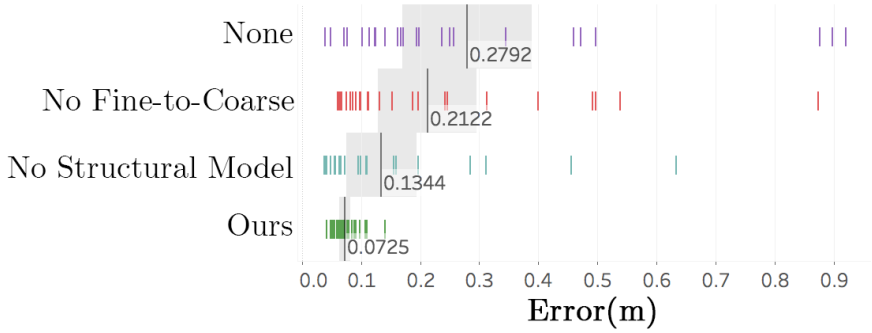


Figure 4.9: **Ablation studies.** Distributions of errors in the SUN3D benchmark for various alternatives of our algorithm. One can see that both strucural model and fine-to-coarse optimization are required for best performance. Disabling either of them leads to diminished performance.

4.5.3 Ablation Studies

To investigate the value of our proposed a) fine-to-coarse iteration strategy and b) structural model, we performed comparisons of our method with all combinations of these methods enabled or disabled. The results in Figure 4.9 and 4.10 show that both provide critical improvements to the results. In particular, it is interesting to note that both the structural model and fine-to-coarse iteration strategy improve over the basic refinement. However, we are able to achieve significantly better results only when both are used. This result highlights the value of aligning local structures before searching for constraints at larger scales.

Additionally, we would like to develop an understanding of how well does the fine-to-coarse algorithm aligns images at different ranges. To this end, we computed histograms of L_2 distances versus frame index differences between pairs of frames linked by ground-truth correspondences. Figure 4.11 shows a comparison of these histograms for the registrations at the start of our algorithm (blue) and at the end (orange). It is interesting to note that our algorithm not only reduces the distances between ground-truth correspondences forming long-range loop closures (the right side of the plot) but also over short ranges. This result demonstrates that the extracted structural model helps to fix not only global alignments but also local ones.

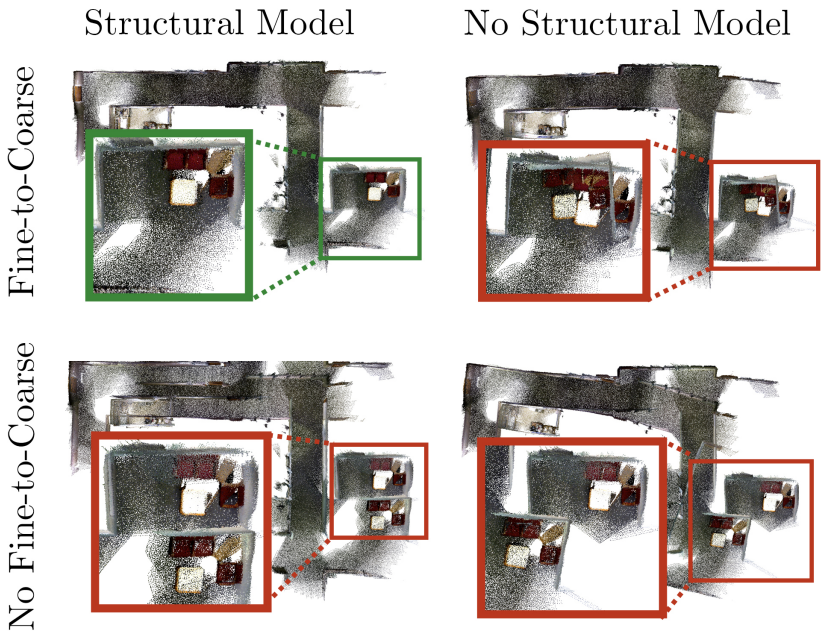


Figure 4.10: Qualitative examples of our ablation studies. Only our full method, using both fine-to-coarse strategy and structural model is able to align the region with red chairs correctly (see zoom-in)

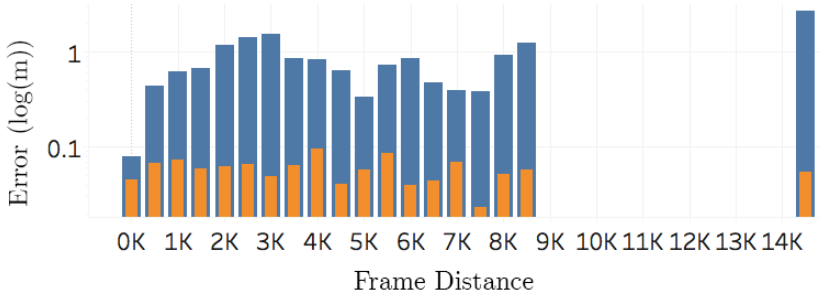


Figure 4.11: Investigating fine-to-coarse iteration. Each bin gathers correspondences that are specific numbers of frames away from each other in the RGBD video. Blue bars show the correspondence errors using initial pairwise transformations (T_0), while orange bars show errors after applying our method (on a log scale). Note that errors decrease for both long-range loop closures and nearby frames.

4.5.4 Additional Results

For the Scannet dataset, we compare our reconstructions with the reconstructions provided with the dataset. We reconstruct the entirety of the Scannet dataset. As the Fine-to-Coarse reconstruction has been designed for the reconstruction of large scenes,

we show a selection of the largest Scannet scenes. Scannet contains 12 reconstructions with the total surface area greater than $145m^2$ (less than 1% of the dataset), eight of which are unique scenes. We use the surface area metric to identify the most challenging scenes in the Scannet dataset and showcase the results in figure 4.12. Overall, our algorithm can produce results that are either better or comparable to those produced with BundleFusion [Dai et al., 2016a].

For the SceneNN dataset, we showcase a selection of our reconstructions among the 100 scenes available in this dataset. Samples were selected randomly, without any specific metric.

4.6 Conclusions

This chapter describes a method for global registration of RGBD scans captured with a hand-held camera in a typical indoor environment. The key idea is a fine-to-coarse scheme that detects and enforces constraints (geometric relationships and feature correspondences) within windows of gradually increasing scales in an iterative algorithm. We demonstrate the benefits of the proposed approach in experiments with a new benchmark for RGBD registration, which contains 10,401 manually specified correspondences across 25 SUN3D scenes.

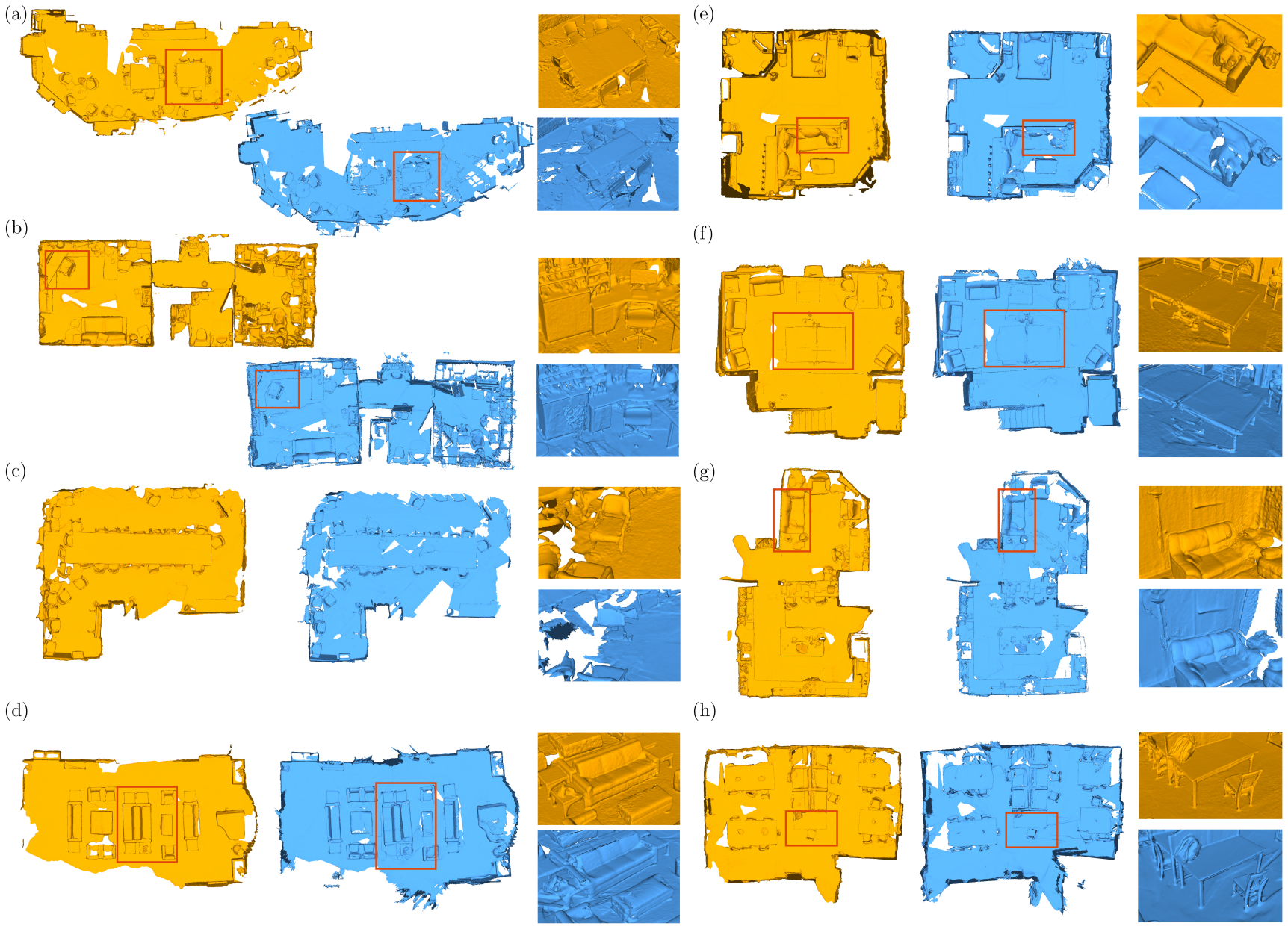


Figure 4.12: Qualitative comparisons on the largest scenes in the Scannet Dataset. **Orange** — Fine-to-Coarse. **Blue** — BundleFusion. Overall results are comparable (examples b,g). However, notice how our method is often able to retain a better global shape (examples a,b), as well as proper representation of local objects like the furniture (examples a,c,d,e,f)

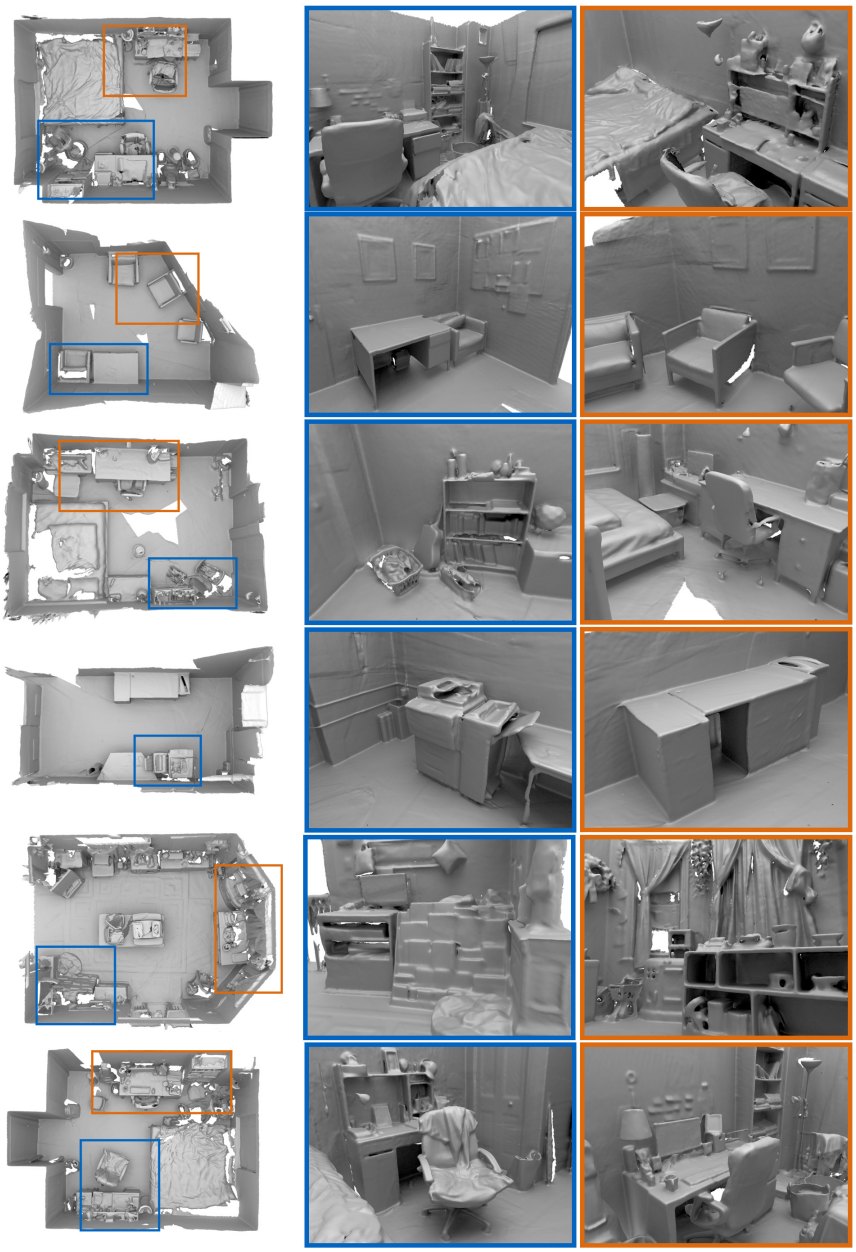


Figure 4.13: Qualitative results on the SceneNN [Hua et al., 2016]. On the right hand side we show birds-eye view of the scene with regions marked, locating zoom-in views. Note that our alignment is able to produce reconstructions that both preserve the overall structure of the room, and low-scale geometric details.

Chapter 5

Inductive segmentation transfer

5.1 Introduction

PREVIOUS chapters focus on methods to improve various stages of reconstruction pipeline. As demonstrated in Chapter 4, the fine-to-coarse method offers state-of-the-art reconstruction results. In this part, we turn our attention to processing of the 3D data that can be obtained using such reconstruction algorithms. Specifically, we seek to understand the scene contents — we wish to be able to know which parts of the scene make a single object, as well as know what type of object it is.

Additionally, as depth capturing devices become smaller and more affordable, and as they operate in everyday applications (AR/VR, home robotics, autonomous navigation), it is plausible to expect that 3D scans of many environments will be acquired daily. We can expect that numerous 3D reconstructions of many spaces, visited at different times and captured from different viewpoints, will be available in the future, just like photographs are today.

With the above observations in mind, in this chapter we investigate how repeated, infrequent scans captured with hand-held RGBD cameras can be used to build a

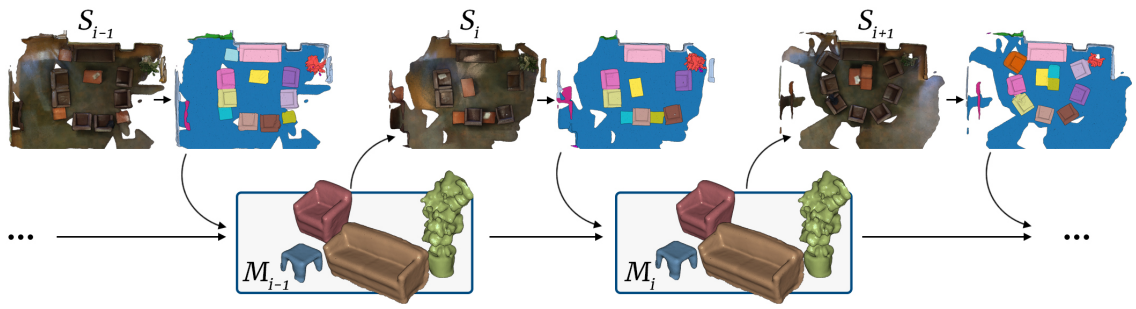


Figure 5.1: The proposed method estimates a persistent, temporally-aware scene model M_i from a series of scene observations S_i, captured at sparse time intervals. M_{i-1} is used to estimate an arrangement of objects in a new observation S_i. The estimated arrangement is used to estimate the instance segmentation of S_i, which is then used to update the model M_i.

spatio-temporal model of an interior environment, complete with object instance semantics and associations across time. Creating such a spatio-temporal model is challenging. First, each RGBD scan captures the environment from different viewpoints, possibly with noisy data. Second, scans separated by long time intervals (once per day, every Tuesday, etc.) can have significant differences due to object motion, entry, or removal. Thus simple algorithms that perform object detection individually for each scan and cluster object detections and poses in space-time are ill-suited for this the problem. Moreover, the proposed task requires multiple reconstructions per scene. Large training sets with this property are not yet available. As such, it is not practical to train a neural network to solve it.

We propose an inductive algorithm that infers information about new RGBD capture of a scene S_i from a temporal model M_{i-1} obtained from previous observations of S (see figure 5.1). The input to the algorithm is the model M_{i-1}, representing all previous scans and a novel scene scan S_i. The output is an updated model M_i that describes the set of objects $\mathcal{O}$ appearing in the scene and an arrangement $\mathcal{A}$ of those objects at each time step, including the most recent. At every iteration, our algorithm optimizes for the arrangement A_i of objects in S_i, and then uses A_i to infer

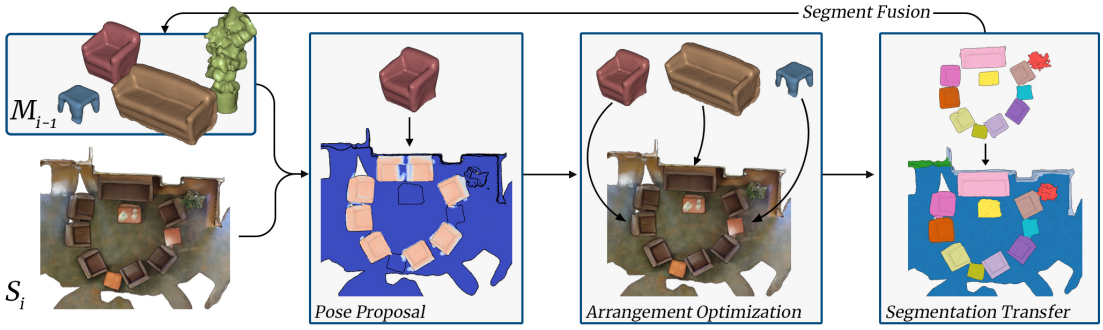


Figure 5.2: A single inductive step of the proposed method. Given a novel scene observation S_i and a model from the past M_{i-1}, our goal is to create an updated model M_i. We first perform *Pose Proposal*, where we search for a set of potential locations for each object in M_{i-1} . Then, we perform *Arrangement Optimization*, where we search for the selection and arrangement of objects to minimize an objective function. Next, we perform *Segmentation Transfer*, in which S_i is annotated with semantic instance labels from M_{i-1}. Finally, geometry from segments in S_i is fused with M_{i-1} to create an updated model M_i.

the semantic instance segmentation of S_i. Segmentation of S_i is then used to update object set $\mathcal{O}$ (see figure 5.2).

To evaluate our algorithm, we present a novel benchmark dataset that contains temporally consistent ground-truth semantic instance labels, describing object associations across time within each scene. Experiments with this benchmark suggest that our proposed optimization strategy is superior to alternative approaches based on deep learning for semantic and instance segmentation tasks.

Overall, our contributions are three-fold:

- A system for building a spatio-temporal model for an indoor environment from infrequent scans acquired with hand-held RGBD cameras,
- An inductive algorithm that jointly infers the shapes, placements, and associations of objects from infrequent RGBD scans by utilizing data from past scans,

- A benchmark dataset with rescans of 13 scenes acquired at 45 time-steps in total, along with ground-truth annotations for object instances and associations across time.

5.2 Related Work

While the semantic scene understanding is now an area of active research [Graham et al., 2018, Liu and Furukawa, 2019], relatively few prior works in computer vision have considered instance segmentation from a temporal perspective.

5.2.1 Temporal Modelling

Most work in computer vision on RGBD scanning of dynamic scenes has focused on tracking [Song and Xiao, 2013] and reconstruction [Newcombe et al., 2015]. For example, [Newcombe et al., 2015] showcase a system where multiple observations of a deforming, non-rigid object (e.g. a human) are fused into a single consistent reconstruction. Associations across time are made by estimating a deformation graph that maps the surface observation to a canonical model. [Yan et al., 2014] present a similar system for articulated, rather than non-rigid shapes. Parts of articulated shapes are tracked, as they deform over time, and a temporally aware model is constructed. These methods differ from ours, as they require observation of motions as they occur.

For sparse temporal observations, early work in robotics focuses on the analysis of 2D maps created from 1D laser range sensors [Anguelov et al., 2002, Biswas et al., 2002, Gallagher et al., 2009]. For example, Biswas [Biswas et al., 2002] used 1D laser data to detect objects within a scene and associate them across time. However, their method relies upon 2D algorithms and assumes that object instances cannot overlap across time, which makes it inapplicable in our setting.

More recent work in robotics aims at life-long scene understanding using data captured with actively controlled sensors [Fäulhammer et al., 2017, Krajník et al., 2017, Santos et al., 2017, Young et al., 2017]. For example, several algorithms proposed in the STRANDS project [Hawes et al., 2017] process the scenes observed from a repeated set of views [Ambruş et al., 2014, Bore et al., 2017, Schulz and Burgard, 2001]. Others focus on controlling camera trajectories to acquire the best views for object modeling [Ekekrantz et al., 2016, Fäulhammer et al., 2017]. These problems are different than ours, as we focus on analyzing previously acquired RGBD data captured without a specifically tailored robotic platform and active control.

Other work has focused on the automatic clustering of 3D points into clusters across space and time [Finman et al., 2014, Herbst et al., 2014]. For example, [Herbst et al., 2014] jointly segments multiple RGBD scans with a joint MRF formulation. [Finman et al., 2014] detect clusters of points from pairwise scene differencing and associates new detections with previous observations. Although similar in spirit to our formulation, these methods operate only on clusters of points, without semantics, and thus are not suited for applications that require a semantic understanding of how objects move across space-time.

5.2.2 Change Detection

Some work in computer vision has focused on change detection and segmentation of dynamic objects in RGBD scans [Ambruş et al., 2014, Fehr et al., 2017, Lee and C. Fowlkes, 2017, Wang et al., 2003]. Work by [Ambruş et al., 2014] showcase a system where a scene is visited on multiple occasions by a robot. The surface observations are captured from the same location each time, and by analysis of the frame data authors are able to classify scene element as static or dynamic. For example, [Fehr et al., 2017] can be considered an extension of the system by

Ambrus. In [Fehr et al., 2017] the aim is also the separation of the scene into static and dynamic parts. The proposed algorithm is able to deal with sequences captured using a hand-held RGBD camera, rather than a robotic platform.

Our system performs prediction of instance semantic segmentation for the entire input scan, rather than binary classification. Despite this difference, the ability to determine static and dynamic parts of the scene is highly beneficial for some applications. [Wang and Thorpe, 2002] performs change detection to determine dynamic objects so that they can be removed from a SLAM optimization. This leads to a more robust solution that is not corrupted by measurements from dynamic objects. More recently, [Lee and C. Fowlkes, 2017] propose a probabilistic model to isolate temporally varying surface patches to improve camera localization. While operating on RGBD captures from hand-held devices, these methods do not produce instance-level semantic segmentation, nor do they generate associations between objects across time.

5.2.3 Alternate Domains

Finally, many projects have considered temporal modeling of environments in specific application domains. For example, several systems in civil engineering track changes to a Building Information Model (BIM) by alignment to 3D scans acquired at sparse temporal intervals [Golparvar-Fard et al., 2012, Karsch et al., 2014, Rebolj et al., 2017, Tuttas et al., 2017]. They generally start with a specific building design model [Han and Golparvar-Fard, 2015], construction schedule [Turkan et al., 2012], and/or object-level CAD models [Bosche et al., 2009], and thus are not as general as our approach. The Scene Chronology project [Matzen and Snavely, 2014], as well as 4D Cities project [Schindler et al., 2007], and system by [Martin-Brualla et al., 2015] all build temporal models of cities from image

collections – however, they do not recover a full 3D model with temporal associations of object instances as we do.

5.3 Algorithm

5.3.1 Scene Representation

Our system represents a scene at time t_i with a temporal model M_i comprising a tuple $\{\mathcal{O}, \mathcal{A}\}$, where $\mathcal{O} = \{o_0, \ldots, o_n\}$ is a list of n object instances that have appeared within this or any prior observation S_j for $j \in [0, i]$, and $\mathcal{A} = \{A_0, \ldots, A_i\}$ is a list of object arrangements estimated for each observation S_j. Each object instance o_k is represented by $\{u_k, G_k, c_k\}$, where u_k is unique instance id, G_k is the object's geometry, and c_k is the semantic class. Each arrangement A_i is a list of poses $\{a_i^0, \ldots, a_i^m\}$, where $a_i^j = \{u_j, \mathbf{T}_j, s_j\}$. u_j is the unique id of j-th object and function $\Omega(u_j)$ returns index k to $\mathcal{O}$. $\mathbf{T}_j$ is a transformation that moves geometry G_k into correct location within the scene S_i. Lastly s_k is a matching score quantifying how well $\mathbf{T}_j G_k$ matches the geometry of S_i.

5.3.2 Overview

Our algorithm updates the temporal model in an inductive fashion. Given the previous model M_{i-1} and a new scan S_i, we predict a new model M_i (see fig. 5.2) by executing four consecutive steps. The first proposes potential poses for objects in $\mathcal{O}$ (see section 5.3.3). The second performs a combinatorial optimization to find the arrangement A_i that maximizes a new objective function jointly accounting for geometric fit and temporal coherence (see section 5.3.4). The third step uses $\mathcal{O}$ and A_i to infer an instance-level semantic segmentation of S_i. The fourth step updates the geometry G_k of each object $\in A_i$ by aggregating its respective segment from S_i. The following four subsections offer details on how each of these steps is implemented.

5.3.3 Object Pose Proposal

The first step of our pipeline is to find a discrete set of potential placements for each object $o_k \in \mathcal{O}$. The input to this stage is a set of objects $\mathcal{O}$ and a scan S_i. The output is a set $\mathcal{P}$ of scored pose lists $P_k = \{p_k^0, \ldots, p_k^x\}$ for each object o_k. A scored pose p_k^l is defined as a tuple $\{\mathbf{T}_k^l, s_k^l\}$, where $\mathbf{T}_k^l$ is the proposed rigid body transformation and s_k^l is the geometric matching score, describing how well pose $\mathbf{T}_k^l$ aligns G_k to the geometry of S_i.

Finding transformations that align surfaces A and B is a longstanding problem in computer graphics and vision [Rusinkiewicz and Levoy, 2001]. In our setting, we wish to find a set of poses for the surface A with good alignment with surface B, where $A = o_k$ and $B = S_i$. Prior work usually attempts to solve similar problems by employing feature-based methods. Such methods sub-sample the two surfaces to obtain a set of meaningful keypoints and then match them to produce a plausible pose (e.g., using Point-Pair Feature matching[Drost et al., 2010]). However, as it is noted in other domains, keypoints may limit the amount of information a method considers, with dense matching methods leading to fewer failures [Engel et al., 2014].

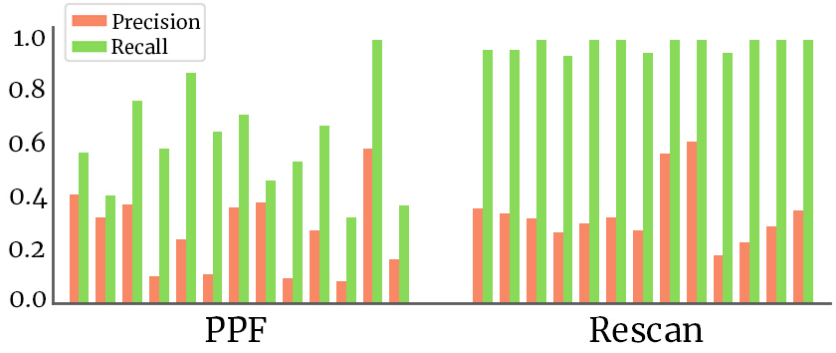


Figure 5.3: Comparison of the precision/recall scores obtained for all scenes in our database, comparing PPF matching [Birdal and Ilic, 2015] to our method. In our experiments, a pose of an object o_k is considered a true positive if the distance between object centers is less than $0.2m$ and object's classes agree.

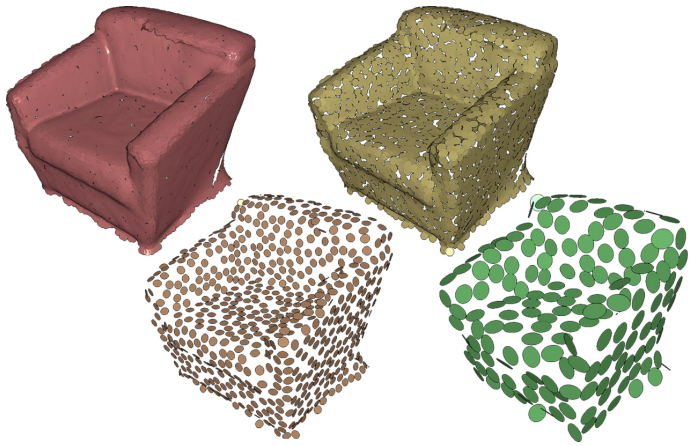


Figure 5.4: The multi-resolution pointcloud representation used in our method. Surfels are scaled up to represent the surface at coarser levels better.

Following this intuition, we propose a dense matching procedure, where we slide each of the objects o_k across the scene, perform an ICP optimization at each of the discrete locations and compute a matching score based on the traditional point-to-plane distance metric [lim Low, 2004].

The use of this approach might seem counter-intuitive, as a naive implementation of such grid-search would lead to a prohibitive run-time performance. We argue that such an approach can be made acceptably fast while leading to much better recovery of correct poses. To speed-up the run-time performance of our method, we make use of the multi-resolution approach. We compute a four-level hierarchy for the input point cloud (the geometries G_k), with minimum distance between any two points at a level equal to $\{0.01m, 0.02m, 0.04m, 0.08m\}$ respectively (see figure 5.4). To compute this representation, we follow an algorithm described in [Corsini et al., 2012]. Multi-resolution representation allows us to perform the dense search only on the coarsest level of the hierarchy, and return a subset of poses with sufficiently high scores to be verified at higher levels, leading to significant performance gains. Additionally, we make a simplifying assumption that objects in our scenes move on the ground plane and rotate around the gravity direction.

With the approach described above, we can produce a set $\mathcal{P}$ of pose lists P_k for each object o_k in $\mathcal{O}$. The advantage of this dense grid-search method is that it produces sets of poses that contain most of the true candidate locations, even if the local geometry of S_i might be different from G_k due to reconstruction errors. We showcase the comparison to keypoint based methods [Drost et al., 2010, Birdal and Ilic, 2015] in figure 5.3.

5.3.4 Arrangement Optimization

In the second step, our algorithm selects a subset of poses from the previous step to form an object arrangement. The input is a set of objects $\mathcal{O}$, a set of pose lists $\mathcal{P} = \{P_0, \ldots, P_k\}$ for each object o_k, and the scan S_i. The output is an arrangement A_i that describes a global configuration of objects which maximizes the objective.

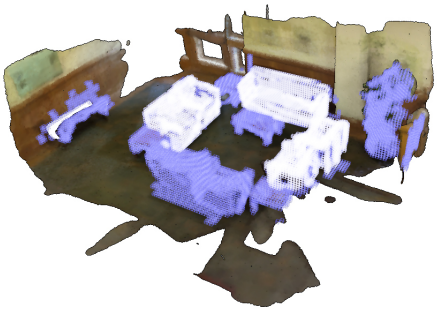
This problem statement leads to discrete, combinatorial optimization. The first reason for choosing this approach is that the number of objects within the scene S_i is not known a priori. A combinatorial approach allows us to propose arrangements A_i of variable lengths, that adapt to the contents of S_i. The second reason is that finding the optimum requires global optimization – the placement of one object can significantly affect the placement of another. Additionally, deep learning is hard to apply in this instance due to the lack of training data, as well as the non-linearity of the proposed objective function.

Objective Function

To quantify the quality of the candidate arrangement A_i', we use the objective function that is a linear combination of the following four terms:

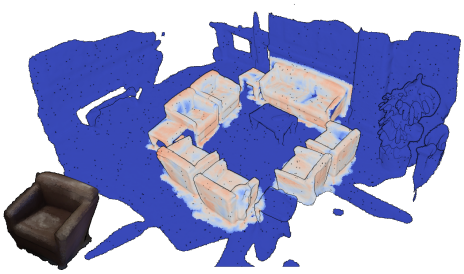
$$
\begin{aligned}
O(S_i, A_i', \mathcal{A}) &= w_c O_c(S_i, A_i') && \text{Coverage Term} \\
&+ w_g O_g(S_i, A_i') && \text{Geometry Term} \\
&+ w_i O_r(A_i') && \text{Intersection Term} \\
&+ w_h O_h(A_i', \mathcal{A}) && \text{Hysteresis Term}
\end{aligned}
$$

Each term O_x produces a scalar value $\in [0, 1]$ that describes the quality of A_i' w.r.t. that specific term. We scale each term by the respective weights $\vec{w} = \{2.0, 0.3, 1.0, 1.8\}$ to express the relative importance of each term.



The Coverage term measures the percentage of the scene that is covered by objects in A_i'. The intuition behind this term is that every part of the scene should ideally be explained by some object in A_i'. $O_c(S_i, A_i')$ takes as input a scene S_i and the candidate arrangement A_i'. To compute $O_c(S_i, A_i')$ we voxelize both the scene S_i and the objects in A_i', resulting in two 3D grids V_S and V_A. The $O_c(S_i, A_i')$ is calculated as the number of cells that are equal in both grids, over the number of cells in V_S - $O_c(S_i, A_i') = \frac{|V_s(j) \wedge V_A(j)|}{|V_s(j)|}$. For this formula to be accurate, we need to ensure that we only voxelize the dynamic parts of the scene S_i. As such, we deactivate any cells in V_S that belong to the static parts of the scene, like walls and floor, which can easily be detected with a method like RANSAC [Fischler and Bolles, 1981]. The inset figure above showcases a visualization of both grids V_S (blue cells) and V_A (white cells). As seen there, the V_S covers the non-static parts of the scene only, leading to O_c being a good estimate of the coverage.

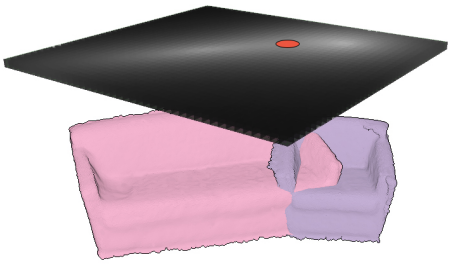
The Geometry term is a measure of the geometric agreement between the scene S_i and objects in the candidate arrangement A'_i. We include this term to guide the objective function to select objects that best match the geometry of the scene at a specific location. This value is computed as an average of scores s^l_k from the procedure described in section 5.3.3:



$$O_g(S_i, A'_i) = \frac{\sum_k g(a^j_i)}{|A'_i|}$$

$g(a^j_i)$ returns the geometric score fit for placement of object o_j. The inset figure shows a visualization of plausible locations for a chair object. Heatmap is generated by counting the number of valid poses for specific surfel S_i. First, we map the chair to a location using every proposed pose. Then a query is made from the scene S_i to see if any surfel of the chair's geometry is within a threshold t.

The Intersection term aims to estimate how much a pair of objects in the arrangement A'_i interpenetrate. Intuitively, such interpenetration would mean that two objects occupy the same physical location, which implies an impossible configuration. In our approach, we compute a rough approximation of this term. First, we compute a covariance matrix $\mathbf{\Sigma}_k$ of each G_k. Covariances for each object allow us to compute a symmetric Mahalanobis distance SD_M between objects to approximately quantify how close they are to each other:



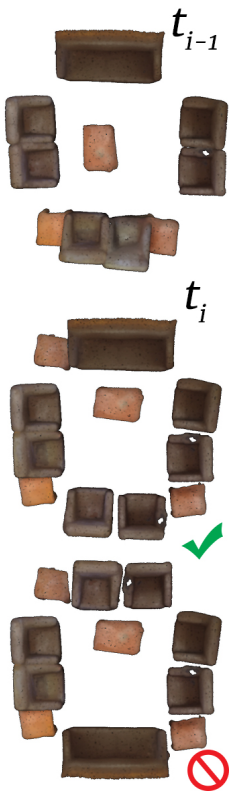
$$SD_M(O_r, o_j) = 0.5(D_M(m_{ij}, \mathbf{T}_i c_i, \Sigma_i) + D_M(m_{ij}, \mathbf{T}_j c_j, \Sigma_j))$$

$\mathbf{T}_i c_i$,$\mathbf{T}_j c_j$ are the transformed centroids of G_i, G_k, the midpoint between them is m_{ij}, and function D_M is the Mahalanobis distance. With SD_M computed for all pairs of objects o_k, the value $O_r(A'_i)$ is:

$$O_r(A'_i) = 1 - |||[\exp(\frac{-SD_M^2(o_0, o_1)}{2\sigma^2}), \ldots, \exp(\frac{-SD_M^2(o_{n-1}, o_n)}{2\sigma^2})]||_\infty$$

The rationale behind the use of the infinity norm is to generate a high penalty if just a single pair of objects exhibit a low score interpenetration. The inset figure showcases a visualization of SD_M for two intersecting objects. The point at which we evaluate the SD_M is marked with red, showcasing high values in regions where either or both objects are present, and low values in the free space. It is also clear that the value of SD_M would be higher if the objects interpenetrated even more.

The Hysteresis term informs how well the current arrangement estimate A'_i resembles a previously observed arrangements from the set $\mathcal{A}$. In addition, it expresses our preference for a minimal relative motion. Each object in A'_i is assigned a score v_k, with the value based on whether u_k is a new instance, or has been observed in the past. In the former case, the score assigned is a constant factor $v_k = h$, with $h = 0.4$. In the latter case, the object's o_k score is computed as:



$$v_k = h + (1 - h)exp(\frac{-||T(c_k, i) - T(c_k, j)||_2}{2\sigma_k^2})$$

$T(c_l, j)$ is a function that applies the appropriate transformation to centroid c_l at time t_j. Since smaller, lighter objects are more likely to be moved, we make the value of σ_k dependent on the volume of an object o_k. σ_k is computed as $\sigma(o_k) = aexp(-bV(o_k)) + c$. $V(o_k)$ returns

the approximate volume of object o_k based on its oriented bounding box. Parameters a, b, c were fit so that the resulting $\sigma(o_k)$ is inversely proportional to $V(o_k)$. This way larger objects obtain smaller σ_k, leading to a smaller *Hysteresis Term* value when object moves significantly. Smaller objects obtain larger σ_k, allowing them to move more freely within the scene with less penalty. With the above formulation using existing objects will always be preferred, unless they have undergone a significant transformation. In such a case, we would like O_h to express that novel object appearances have a similar probability. The value of $O_h(A_i, \mathcal{A})$ is computed as an average of the above scores. The inset figure above illustrates an arrangement at t_{i-1} and two possible arrangement estimates at t_i. The form of $O_h(A'_i, \mathcal{A})$ encourages the selection of middle arrangement as it does not contain significant motion the sofa and chairs.

Optimization

To find arrangement $A_i = \arg\max_{A'_i} O(S_i, A'_i, \mathcal{A})$, we employ a combination of greedy initialization and simulated annealing. We begin by greedily selecting an object o_k at a pose p^l_k which improves objective the most. This process of addition is continued until the objective function starts decreasing. After this stage, we perform simulated annealing optimization. We run the simulated annealing for 25k iterations, using a linear cooling schedule with a random restarts (0.5% probability to return to the best scoring state). To explore the search space, we use the following actions with a randomly selected object o_k:

- **Add Object** - We add o_k at a random pose p^l_k to A'_i.
- **Remove Object** - We remove o_k from A'_i.
- **Move Object** - We select o_k from A'_i and assign it new pose p^m_k.
- **Swap Objects** - We swap the location of o_k and o_l, another randomly selected object of the same semantic class.

5.3.5 Segmentation Transfer

The third step of the algorithm transfers the semantic and instance labels from A_i to scan S_i (fig. 5.5). The estimated arrangement from the previous step can be used to perform segmentation transfer, as we have semantic class c_k and instance id u_k associated with each object in $\mathcal{O}$. Using the estimated pose p_k^l for each of the objects o_k in A_i, we transform its geometry G_k to align with S_i. We then perform the nearest neighbor lookup (with a maximum threshold $d = 5cm$ to account for outliers) and use the associations to copy both the instance and semantic labels from objects in A_i to S_i. Since there is no guarantee that all points in S_i have a neighbor within the threshold d, we follow-up the lookup with label smoothing based on multi-label graph-cut [Delong et al., 2012].

5.3.6 Geometry Fusion

The final step of the algorithm is to update the object geometries G_k for objects in $\mathcal{O}$. To do so for each object $o_k \in A_i$, we extract the sub point clouds from S_i that were assigned instance label u_k in the previous step, and then we concatenate them

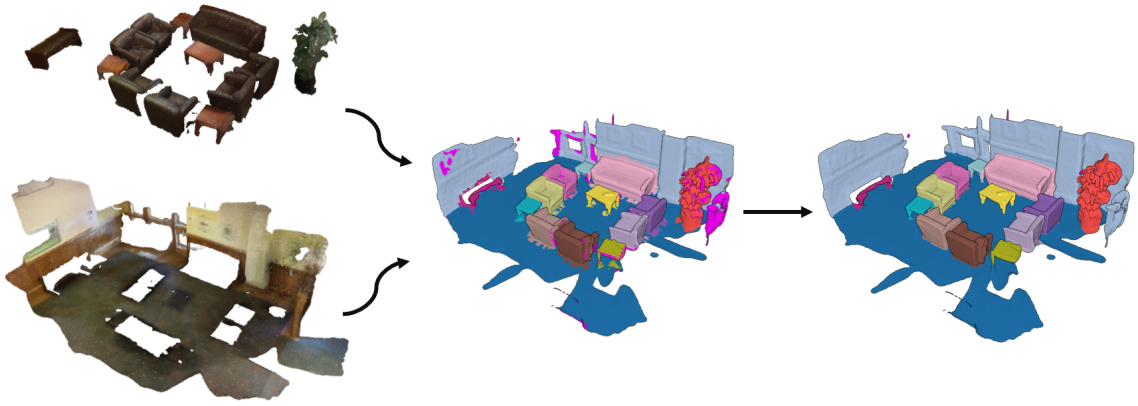


Figure 5.5: Segmentation transfer from the estimated arrangement to a new observation. Given the arrangement and the static objects in the database, we perform nearest neighbor queries to copy instance and semantic labels from the arrangement to the novel observations. We then perform GraphCut [Boykov et al., 2001] optimization to smooth out the copied labels.

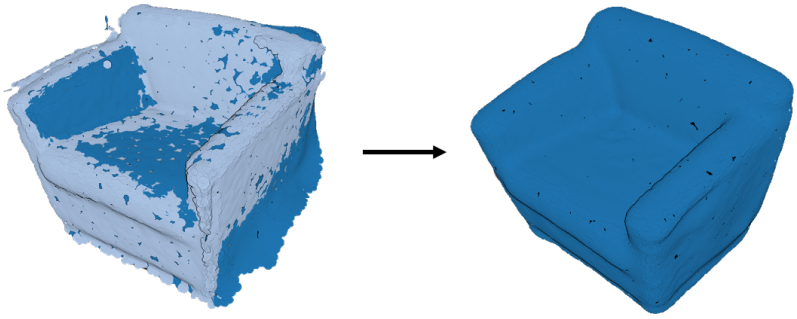


Figure 5.6: A visualization of a geometry merging using Poisson Surface Reconstruction [Kazhdan and Hoppe, 2013]. A two pointclouds on the left are merged into a single surface, which we sample to obtain the pointcloud on the right.

with G_k to generate new point cloud G'_k. In the idealized case, the two surfaces would be identical, as they represent the same object. However, due to partial observation, reconstruction, and alignment errors, we cannot expect that in practice. As such, we solve for a mean surface $\tilde{G}_k$ that minimizes the distance to all points in the G'_k, using Poisson Surface Reconstruction [Kazhdan and Hoppe, 2013]. After this process, we uniformly sample points on the resulting surface $\tilde{G}_k$ to get a new estimate of G_k that can be used for matching when a new scene S_{i+1} needs to be processed. Figure 5.6 showcases the result of this process for a simple chair model.

5.4 Evaluation

Evaluation of the proposed algorithm is not straightforward, as there is little to no prior work directly addressing the instance segmentation transfer between 3D scans.

Dataset: To evaluate the proposed approach, we have created a dataset of temporally varying scenes. Our dataset contains 13 distinct scenes, with a total of 45 separate reconstructions. Each scene contains three to five scans, where objects within each capture were moved to simulate changes occurring across long time periods. The captured spaces are relatively large, with an average approximate area of $67.58m^2$. Overall, the dataset contains 330 unique instances of moving objects (see figure 5.8b).

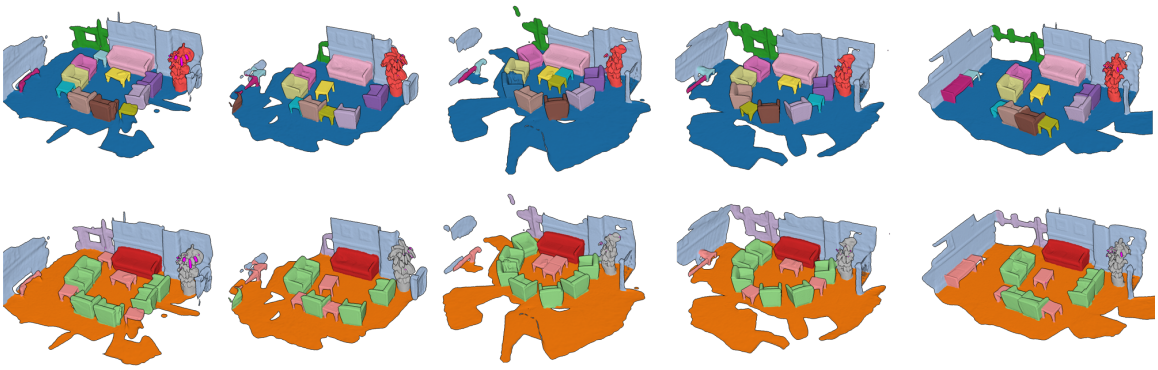
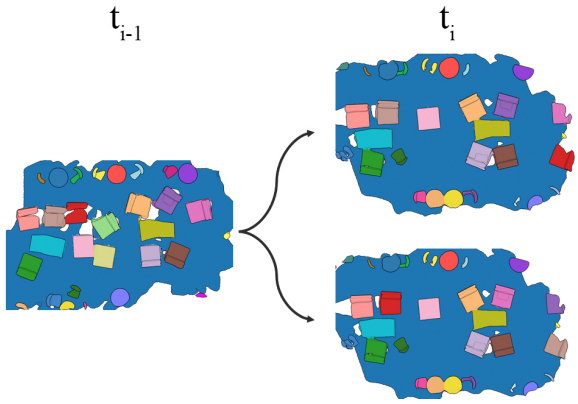


Figure 5.7: Visualizeation of the instance and semantic segmentation from the Rescan dataset. Notice how objects retain their identity over time (top row).



(a)

Class	Total Count	Unique Count
chair	576	184
other	199	72
table	145	40
desk	46	16
sofa	38	12
shelves	14	5
bookshelf	3	1
Sum	1021	330

(b)

Figure 5.8: (a) Given an arrangement of objects at t_{i-1}, it is often the case that multiple arrangements at t_i make sense. Rescan dataset provides permutations of object id assignment to account for such cases. (b) *Total count* describes the number of objects in all of the scenes in our dataset. *Unique count* specifies the number of unique instances that have appeared across time.

The RGBD sequences were captured using the Structure Occipital Sensor — please see Chapter 3 for details. The 3D reconstructions were obtained using an algorithm described in chapter 4. Visualization of the entire dataset can be seen in table 5.1.

Along with the captured data, we also provide the manually-curated semantic category and instance labels for every object in every scene (see figure 5.7). The instance labels are temporally stable, providing associations between object instances across time, which we can use to evaluate our algorithms. Since the motion of the objects is

not observed, there might be many plausible permutations of object associations over time. To address this ambiguity, we provide permutations of instance assignments for each scene to account for cases where objects' motion is ambiguous and multiple arrangements can be considered correct 5.8a.

Metrics: We evaluate our approach using three metrics. The first is the *Semantic Label* metric that measures the correctness of class labels – it is implemented in the same way as the semantic segmentation task in the ScanNet Benchmark [Dai et al., 2017] and is reported as mean class IoU. The second is the *Semantic Instance* metric that measures the correctness of the object instance separations – it again comes from the ScanNet Benchmark [Dai et al., 2017] and is reported as mean Average Precision (IoU=0.5). Third, we propose a novel *Instance Transfer* metric, which specifically requires an agreement of instance indices across time. This metric is reported as mean IoU, where we count the number of points in both ground truth and prediction that share equivalent instance id. The *Instance Transfer* metric is much more challenging, as it requires associating objects with specific instance ids in different scans.

Baseline: Given the success of the recent deep models for the scene understanding (as shown on the leaderboard of [Dai et al., 2017]), it is interesting to compare the results of our algorithm to the best available method based on deep networks. One of the best available methods for 3D instance segmentation is MASC [Liu and Furukawa, 2019], which is based on semantic segmentation with SparseConvNet [Graham et al., 2018]. To test these methods on our tasks, we pre-trained the SparseConvNet and MASC models on ScanNet's training set. We also performed fine-tuning of MASC on the ground-truth category labels for the first scan of each scene S_0. This fine-tuned model provides instance segmentation, which can be combined with the Hungarian method [Kuhn, 1955] to estimate instance associations across time.

t_0	t_1	t_2	t_3	t_4
				

Table 5.1: Rescan Dataset

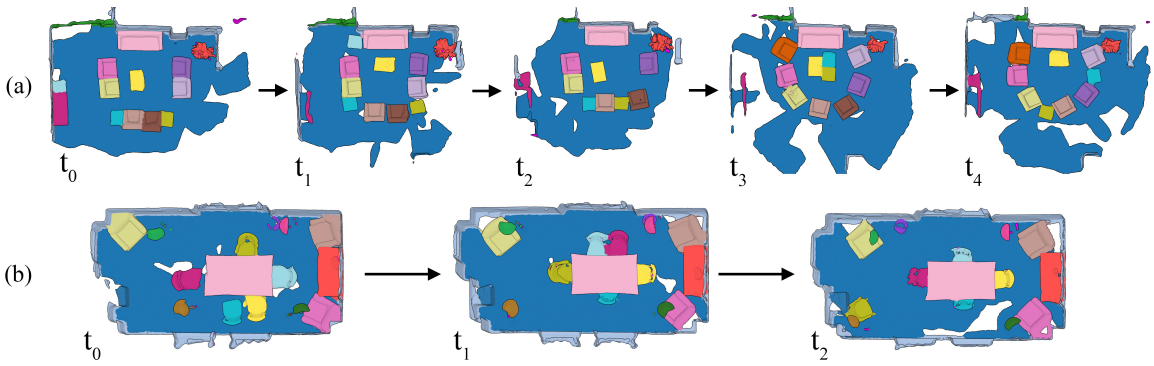


Figure 5.9: Inductive instance segmentation results. Given a segmentation at time t_0, our method is able to iteratively transfer instance labels to future times, even when the number of the objects in the scene changes.

This sequence of steps provides a solid baseline combining state-of-the-art methods for the instance segmentation with an established algorithm for assignment.

5.4.1 Quantitative Results

Method	Semantic Label	Semantic Instance	Instance Transfer
SparseConvNet	0.203	-	-
MASC	0.310	0.291	0.175
MASC (fine-tuned)	0.737	0.562	0.345
Rescan	0.859	0.837	0.650

Table 5.2: Comparison of our method to SparseConvNet [Graham et al., 2018] and MASC [Liu and Furukawa, 2019]. SparseConvNet does not produce instance labels, and hence we omit reporting on the *Semantic Instance* and *Instance Transfer* task, and only fine-tune MASC.

Evaluation and comparison: Since we solve an inductive task (predict the answer at time t, when given an answer at time $t-1$), it is not obvious how to initialize the system during experimentation. Since we aim to evaluate the inductive step alone, we chose to initialize time t_0 with a correct instance segmentation in our experiments. That choice avoids confounding problems with de novo instance segmentation at t_0 with the primary objective of the experiment. We have each algorithm in the

Method	Semantic Label	Semantic Instance	Instance Transfer
No Coverage Term	0.061	0.058	0.048
No Geometry Term	0.853	0.825	0.617
No Intersection Term	0.859	0.781	0.584
No Hysteresis Term	0.870	0.818	0.226
Full Method	0.859	0.837	0.650

Table 5.3: The influence of objective function terms on each of the proposed tasks.

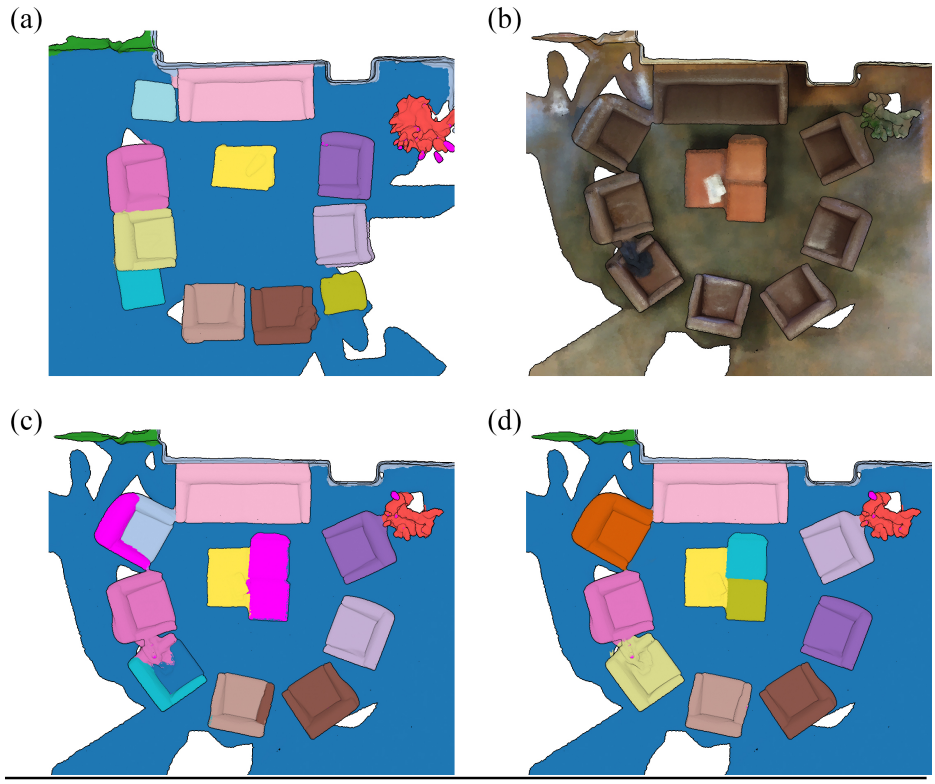
experiment transfer the instance segmentation from t_0 to t_1, then transfer the result to t_2, and so on.

We ran this experiment for our method in direct comparison to the baseline. Results for all three evaluation metrics are shown in Table 5.2. They show that our algorithm significantly outperforms competing methods. As expected, we see that the deep networks trained on the ScanNet training set [Dai et al., 2017] do not perform very well on our data without fine-tuning. After fine-tuning on the ground-truth labels in S_0, they do much better (of course, because S_0 is just like the test data at S_1 ... S_k). However, instance segmentation on later time steps still performs worse than our algorithm, and instance transfers across time are poor. We attribute the difference to the fact that our method is instance-centric, where the segmentation is inferred from the estimated arrangement of objects. This is in stark opposition to methods like MASC, where the instances are inferred from a semantic segmentation.

Ablation studies: Second, we present the results of ablation studies that showcase the influence of various terms in our objective function on the results in a specific task. As seen in table 5.3, by far the most important term of our proposed objective is the *Coverage Term*. Without it, the objective function is discouraged from adding more objects. The optimization simply finishes with a single object added to the scene

— as adding any more would lead to a decrease in other terms. The second most important term, especially for the *Instance Transfer* task, is the *Hysteresis Term*. It is intuitive that lacking this term, the objective function is not encouraged to find an arrangement that will be consistent with previous object configurations. We note that when omitting this term, the semantic segmentation task achieves a slightly better result. The reason is that to prevent the addition of superfluous objects the novel objects are assigned a relatively low score (see sec. 5.3.4). Without the *Hysteresis Term*, the proposed objective is free to insert additional objects. However, their configuration is often not correct, leading to lower scores for the other two tasks. This result suggests that there exists a better formulation of the hysteresis function — an interesting direction for future research. The presence of the *Intersection Term* important for the *Semantic Instance* and *Instance Transfer* tasks. Intuitively, the semantic segmentation score is unaffected as it is often the case that intersecting objects share the semantic class. The *Geometry Term* has the least influence on the results. This is not surprising, as the poses that survived the pose proposal stage (see sec. 5.3.3) were high scoring ones.

Effect of pose proposal alternatives: Third, we analyze the importance of the pose proposal step on the results. In the limited movement variant of our method, we do not perform the dense search for arbitrary poses for each object. Instead, we only allow for movement within a 20cm radius around the position of the object in the previous arrangements. From results in fig. 5.10 we can see that the limited movement leads to a significant decrease in performance. For the instance segmentation tasks, results are on average similar. However, both methods (full movement and limited movement) have different modes of failure. The full movement might produce incorrect permutation of chairs around the table, when *Coverage Term* outweighs the *Hysteresis Term*. The limited movement does not have that issue, as no additional



Method	Semantic Label	Semantic Instance	Temporal Instance
Limited Movement	0.747	0.675	0.623
Full Movement	0.859	0.837	0.650

Figure 5.10: A comparison of full vs. limited movement schemes. The limited movement variant leads to a significant drop in performance. (a) The source scene with instance segmentation. (b) The target scene visualization. (c) The result of our method with limited movement. (d) The result of the full version of the presented method.

poses for such chairs are produced. At the same time, it misses objects that have moved significantly.

5.4.2 Qualitative Results

Inductive segmentation transfer: We showcase qualitative results for the *Instance Transfer* task using our method in figure 5.9. Again, in this task we use the ground-truth segmentation provided by the user at t_0 and transfer it to all other

observations sequentially. The results of such segmentation transfer offer stable and well-localized instances. Even over multiple time-steps, our method can keep track of objects identities, providing us with information on their location and motion. Additionally, thanks to the fact that the objective function prefers minimal change, we are able to deal with challenging configurations. For example in 5.9a our method is able to correctly recover three coffee tables at time t_3, despite their proximity and visual similarity.

Semantic segmentation: Figures 5.11 and reffig:understanding:instanceresults showcases qualitative comparisons between our method and DNN-based methods [Liu and Furukawa, 2019, Graham et al., 2018]. Without fine-tuning, the segmentation issues are obvious. Learned methods confuse labels like *sofa* and *chair*, which explains low scores in table 5.2. Fine-tuning helps reduce these effects — however, we also see some overfitting errors. Our method is able to recover high-quality semantic segmentation, where due to the fact that our approach is instance-centric, a single instance can not have more than a single semantic class. Our method's success is however dependent on the overlap between current and previous observations of S. When lots of novel objects appear, the *Hysteresis Term* might discouraging the addition of all of them, as it aims to produce arrangement similar to previously observed ones (fig. 5.11a).

Model completion results: Our method for aggregating the observations of moving objects from multiple time steps allows it to produce more complete surface reconstructions than would be possible otherwise. Many other systems explicitly remove moving objects before creating a surface model (to avoid ghosting) [Keller et al., 2013b]. Our approach uses the estimated object segmentations and transformations to aggregate points associated with each object o_k to form a G_k that is generally more complete than could be obtained from any one scan. Composing

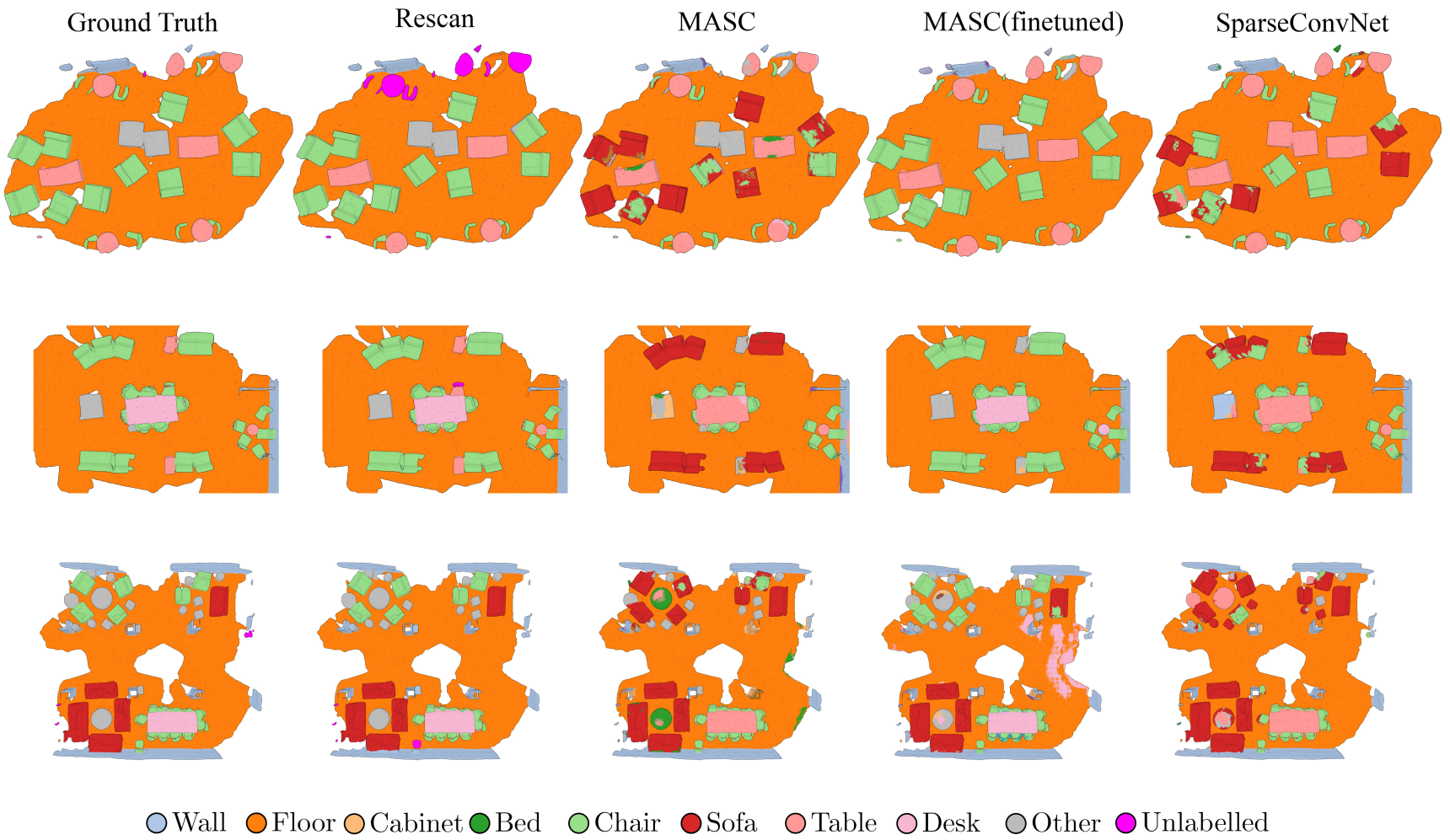


Figure 5.11: Qualitative comparison between different methods on the semantic segmentation task. The proposed method can provide high-quality semantic labels as a result of instance segmentation transfer. Compared to competing methods, ours is able to produce better per object labels and does not confuse object classes.

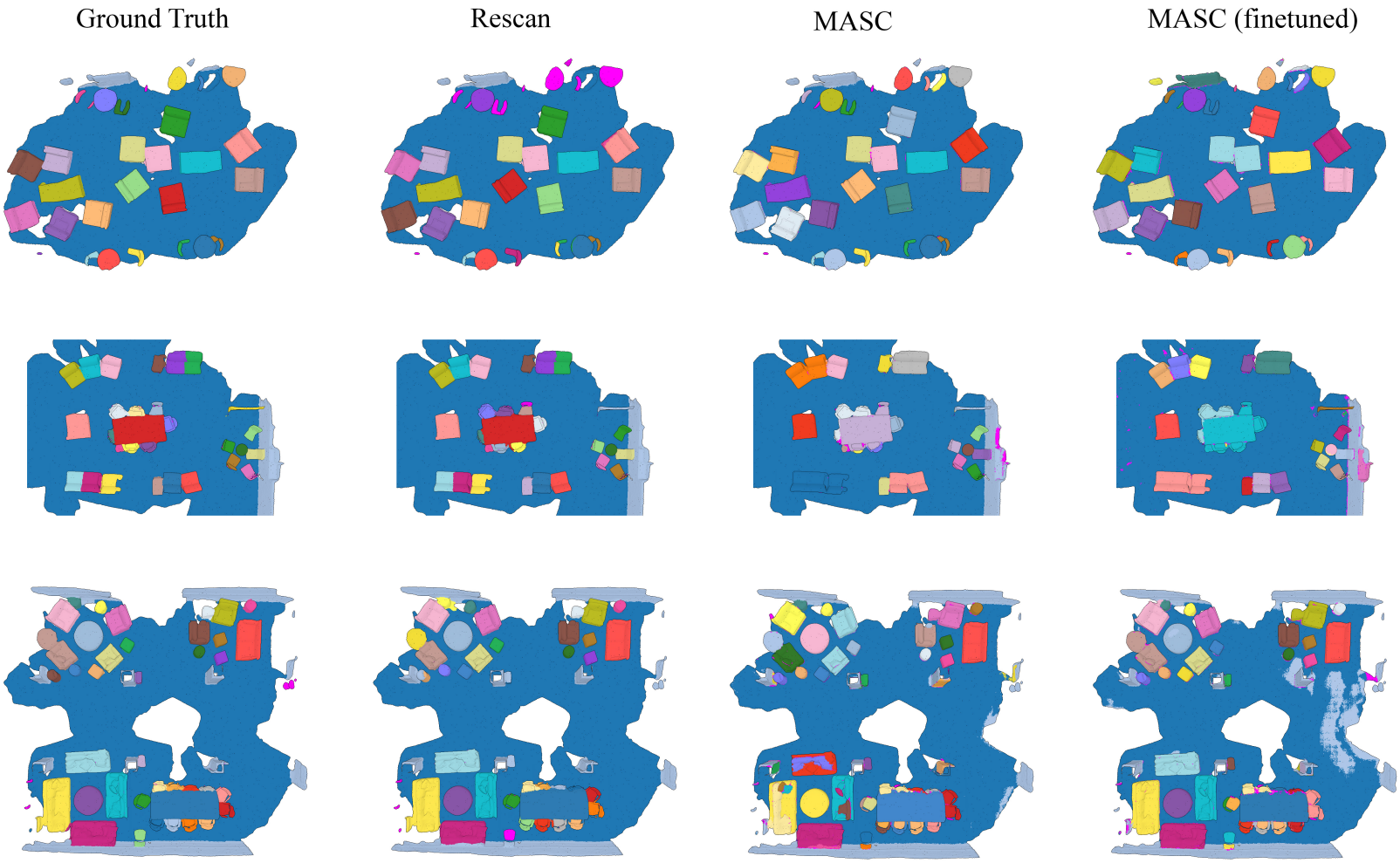


Figure 5.12: Qualitative comparison of different methods on the instance segmentation task. Our approach is able to discern the object boundaries much better. Compared to other methods, ours is able to deal with challenging cases, like the sofa's in the middle row, and provide the correct number of instances.

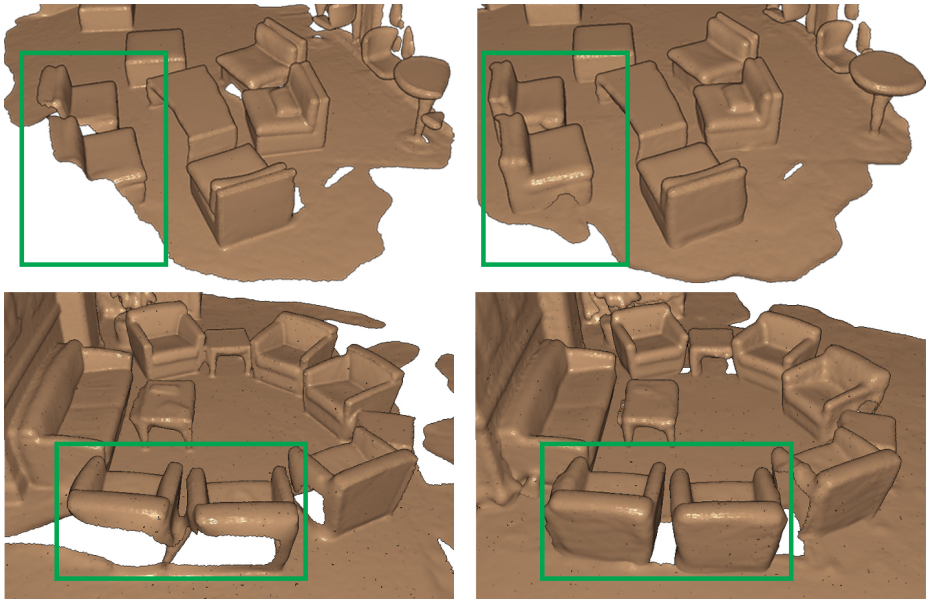


Figure 5.13: Model completion results. The left column shows two scans of a scene with moving objects. The right column shows our reconstruction of the scene using objects and locations from the temporal model M.

the aggregated G_k using transformations T_k in each object arrangement A_i provides a model completion result (fig. 5.13).

Failures: Our method's performance is competitive with the learned based approaches. Still, there are cases where our method fails. At the same time, it is important to note that because the proposed method is geometry driven, most failures are easy to interpret. As such we identify two major types of failures — *external* and *internal*.

External failure mode describes any failure that is due to the appearance of a novel object, that does not exist in the temporal model M_{i-1}. When a novel object outside $\mathcal{O}$ appears, our method either mislabels it or fails to provide any label. The first case happens if the novel object's geometry is similar to some object $\in M_{i-1}$, so that *Pose Proposal* stage is able to generate potential locations. The second case is encountered when the novel object is completely different than any of the objects in $\mathcal{O}$. In the Rescan dataset, we encounter a single case where such a situation happens.

Figure 5.14: Failure modes of the proposed method. (a) Partial scanning of furniture prevents the *Pose proposal* stage from generating plausible poses. (b) Small objects contribute little to the *Coverage term*. If such objects undergo significant motion, our algorithm might miss them. (c) When visually-similar, partially scanned objects are considered, our method might not produce the correct permutation.

In figure 5.15 we can see the mislabeled thrash bin — the reason for this is the lack of trash bins at the previous time-step.

The other mode of failure happens where all the information required for estimation of the correct arrangement is present, but the method fails to produce completely correct answers. As such we refer to such failure modes as *internal*. We identify three main types of failures in this class (fig. 5.14).

The first issue arises when due to the geometry focus of the proposed method. When scanning of the object is partial, the *Pose Proposal* stage might fail to produce poses for some objects $o_k \in \mathcal{O}$. Since the *Pose Proposal* stage is not able to find a complete set candidate poses, the subsequent stage of *Arrangement Optimization* is not able to estimate correct locations. The correct poses of objects are simply not within the space over which *Arrangement Optimization* is searching. Such cases

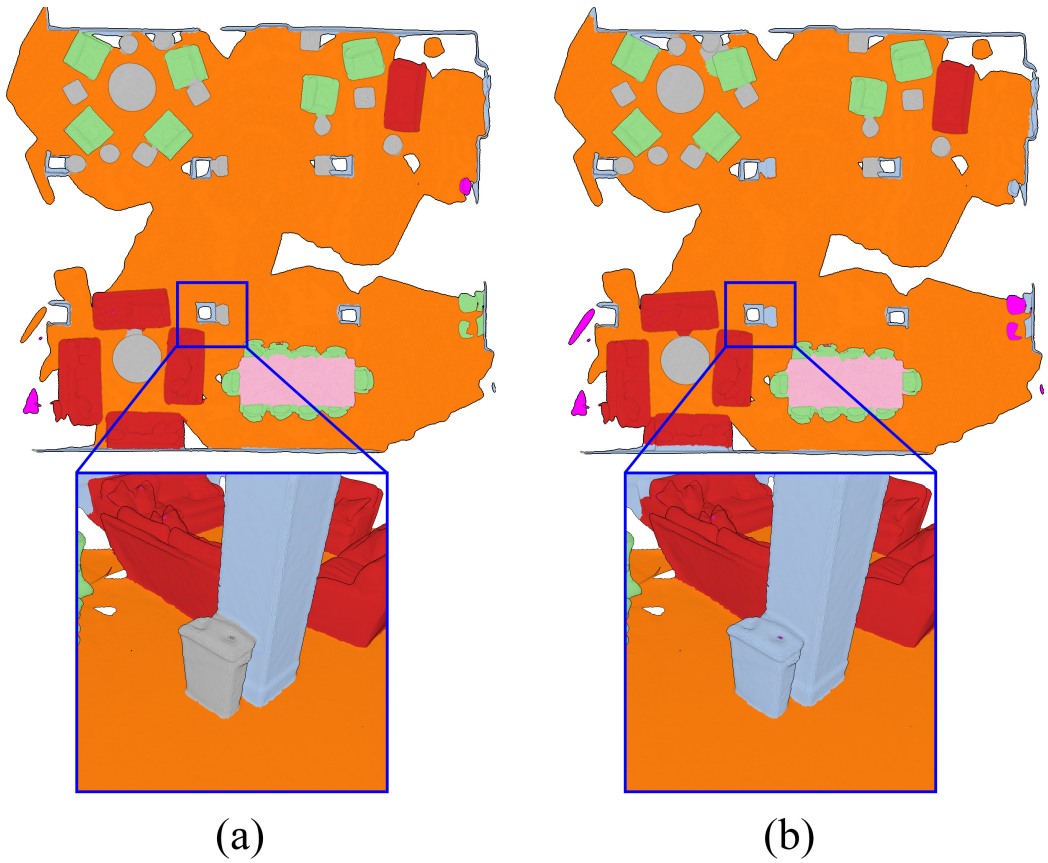


Figure 5.15: (a) Ground truth segmentation. (b) Semantic label prediction. The lack of trash bins in previous timesteps causes our method to mislabel the trash bin as a part of the column.

usually arise at the peripherals of the scene, which were not captured carefully (see figure 5.14a).

A similar issue occurs when the observation of the scene S_{i-1} misses significant parts of the underlying space. In such cases, the objects in the database $\mathcal{O}$ might not be well suited for matching, as they might be incomplete. Such partial objects will not generate meaningful scores for the *Geometry Term*. Additionally, with significant parts of the scene missing at the previous timestep, it might be the case that the scene S_i contains significantly more objects than S_{i-1}. Our method prefers to keep the number of objects approximately constant between S_i and S_{i-1}. This is a result of our formulation of the *Hysteresis Term* which aims to create object identity associations

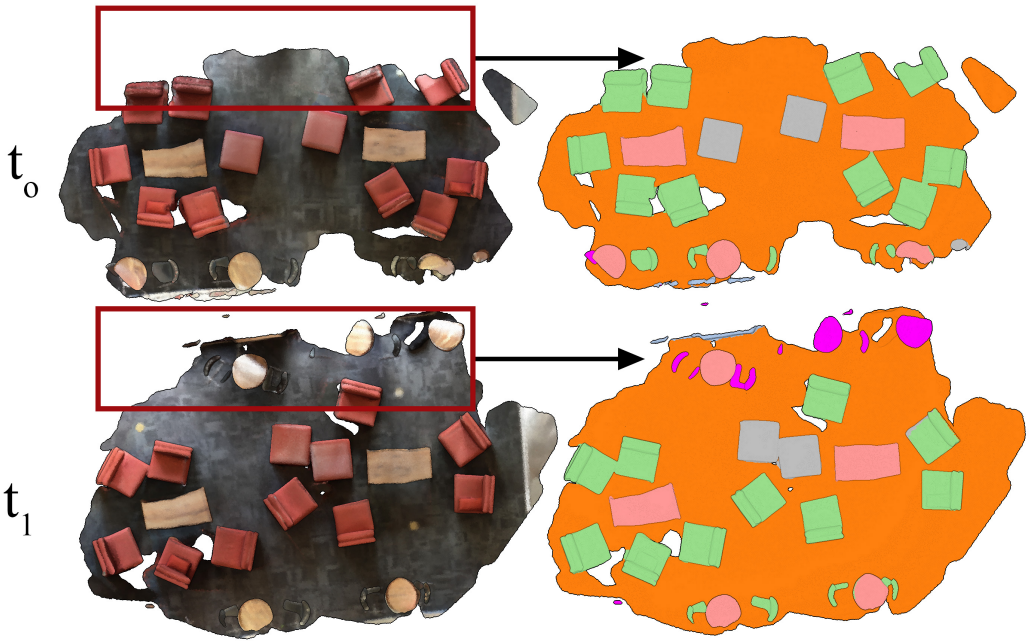


Figure 5.16: Scene capture at t_0 is missing the part scanned in t_i(highlighted). Some objects at t_1 are not detected, due to the lack of overlap between two reconstructions.

between S_i and S_{i-1}, and is penalized when adding many novel objects. In such cases our method might miss certain objects – see figure 5.16

The second is caused by the form of the *Coverage Term*. Larger objects are preferred, as the object's contribution to the overall score is proportional to its size. Smaller objects contribute much less, while at the same time are more likely to move. In section 5.3.4 we outlined our strategy to combat this issue. However, it still does occur in some instances, like the one visualized in figure 5.14b.

Lastly, an incorrect permutation of objects might have a higher objective value than the ground truth one. This effect is a combination of *Geometry Term* providing noisy scores for partial scans of visually similar objects, and their relative spatial proximity, which makes the *Hysteresis Term* a poor discriminator. A concrete example of this general concept is "chairs around the table" case, seen in figure 5.14c. The aforementioned similarity of the objects, combined with the imperfections of the reconstruction process and their relative closeness causes the proposed algorithm to confuse the locations of the objects of the same semantic class.

Overall the performance of our method is still high despite these issues. Moreover, most of the above issues could be resolved with alternative formulations of each of the objective function's terms. Learned alternatives for these terms are an interesting direction for future work. A more fundamental issue comes from the case of an object outside $\mathcal{O}$ appearance. Without a similar object example in the database, our method either mislabels the novel object or provides no labels. Detection of such cases is also an exciting future direction. One way of resolving it would be to put the user in the loop or combine Rescan algorithm with supervised approaches for instance segmentation.

5.5 Conclusion

We present an algorithm for estimating the semantic instance segmentation for an RGBD scan of an indoor environment. The proposed algorithm is inductive – using a temporal scene model which subsumes previous observations, an instance segmentation of the novel observation is inferred and used to update the temporal model. Our experiments show better performance on a novel benchmark dataset when compared to a strong baseline. New directions for future work include replacing the terms of our objective function with learned ones. Going further, we should also consider investigating RNN architectures when more massive datasets become available.

Chapter 6

Conclusion

This dissertation describes a set of algorithms for improving various stages of RGBD reconstruction pipeline, allowing for reconstruction of spaces larger than the previous methods were capable of handling. We also present a method able to utilize the result of the robust reconstruction pipeline, where multiple reconstructions of the same space are associated across time.

6.1 Summary

In this work, we have presented three methods that address different problems arising during the processing of the RGBD data.

First, we present a calibration method for PrimeSense class of devices. We show how to estimate all properties of the device, from the intrinsic parameters of both color and depth camera, through the extrinsic relationship between the two, to the estimation of the undistortion field. The proposed method is efficient and can be performed at a low cost, only requiring a print-out of a standard calibration target. We showcase how the depth undistortion improves the effective range of a device from which depth measurements can be used by comparing the tracking quality between original and undistorted versions of the input data. Additionally, the proposed cal-

ibration procedure has been used to improve the quality of data in the widely used Scannet dataset [Dai et al., 2017].

Second, we present a method that is able to reconstruct large spaces from long RGBD sequences [Halber and Funkhouser, 2017]. The robustness of our method is due to two major factors: the fine-to-coarse optimization strategy, and the incorporation of priors regarding man-made spaces into the energy function. The method can improve the quality of a tracking algorithm by first creating both low-level (in the form of closest-point correspondences) and high-level (in the form of planar structures) constraints, along local regions over the trajectory. The local regions start small, but progressively grow, until the entire trajectory is considered. We observe that the proposed method is displaying loop-closure like a behavior, even though no explicit visual loop-closure solution is employed. We evaluate our algorithm on a newly created dataset of correspondences and show that our method is able to outperform existing solutions.

Lastly, we present "Rescan", a system of associating the data between different reconstructions of the same space. Given a three-dimensional reconstruction with some per-vertex information, and other reconstruction of the same space, we propose a system for transferring the information between different observations. We assume that the scene contents can change considerably, with a significant object motion and object disappearance. Our method is able to establish an association between such reconstructions by using a geometric fitting strategy. We showcase the effectiveness of our method on the segmentation transfer task. As the transfer between the reconstructions of the same space taken at sparse temporal intervals is a new task, we introduce a dataset of the labeled scene with associations across time. We show that the proposed method is able to achieve results competitive with existing work on standard tasks, like semantic and instance segmentation. At the same time,

our method is much more capable at the segmentation transfer task then a baseline approach based on top learning methods.

6.2 Future Work

In terms of the directions of future work, there are numerous ways to improve and build on the research presented in this document. We first consider some possible improvements to the individual systems we have presented, and then we discuss more general directions that future work could follow.

Acquisition. In Chapter 3 we have described a system for improving the quality of depth data produced by the PrimeSense class of devices. These technologies continue to advance, and while it is interesting to understand the underlying depth generation algorithm further to design more robust denoising strategies, such a bottom-up strategy might be rendered invalid by a new iteration of devices. At the same time, no system is perfect. As such, it is valid to explore directions that do not make strong assumptions on the capture algorithm. One example is a self-calibration strategy that would use the color information to reconstruct a sparse set of well-localized 3D positions. Such a sparse set of positions could be used to populate the low-frequency distortion field without the need for a separate calibration stage. Another direction is to use deep learning to improve the quality of depth maps. While the creation of a dataset that contains aligned image pairs from a high-quality sensor and a commodity sensor is challenging, it would be interesting to explore domain translation approaches in the context of depth maps.

Reconstruction. In terms of the reconstruction from RGBD data, both the method presented in Chapter 4 and a multitude of alternatives can generate high-quality geometry of the scene. At the same time, much less effort has been put into

estimating a high-quality color and textures for the resulting reconstructions. More careful design and optimization of the color information would help to increase the visual fidelity of the captured scenes. Going further, using the estimated geometry combined with a light estimation could be used to predict the actual albedo of surfaces — this way the resulting geometry could be used in applications like 3D animation and video-games. Another direction for improvement of the proposed system is the efficiency aspect. The fine-to-coarse approach is inherently offline, but it could be incorporated as a corrective parallel process. As such, it would be running alongside a real-time SLAM system, correcting drift and loss of tracking.

Understanding. The Rescan project presented in Chapter 5 is a proof-of-concept of a system that exploits redundant information available in multiple reconstructions of the same scene. In this project, we are able to show that multiple reconstructions of the same scene can be used to build a persistent model of a scene, just by using geometric information. Follow-up work should look at improving the quality of associations between different timesteps by incorporating different data modalities, as well as possibly learned features. As such, a venue for future work is a creation of a larger dataset with temporal associations. Another obvious direction for future research is addressing the special case when the object dataset is empty at the start. In our work, we alleviate this issue using ground truth segmentation. This is a clear limitation and should be addressed in future work by co-segmentation or learned solutions to propose the initial segmentation.

Finally, the research presented in this document looks at the problem of reconstruction and understanding as separate stages of the RGBD processing pipeline. This separation is a clear limitation as semantic understanding can assist the reconstruction pipeline, by providing high-level features in terms of object categories, or even object parts. While there already exist some systems attempting to look at

this coupling like work by [Salas-Moreno et al., 2013], they are limited to a set of pre-scanned objects. Others, like [Narita et al., 2019] only look to perform semantic segmentation as the reconstruction is generated, and do not improve the tracking using the semantic information. Future RGBD pipelines should look at a tighter coupling of the two methods and establishing how the semantic and instance information can be leveraged in a camera pose estimation task. An additional advantage of such a system would be that a resulting reconstruction already has semantic information embedded.

Bibliography

[Adams, 2002] Adams, M. D. (2002). Coaxial range measurement - current trends for mobile robotic applications. *IEEE Sensors Journal*, 2(1):2–13.

[Agarwal et al., 2011] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. (2011). Building rome in a day. *Commun. ACM*, 54(10):105–112.

[Aiger et al., 2008] Aiger, D., Mitra, N. J., and Cohen-Or, D. (2008). 4-points congruent sets for robust surface registration. *ACM Transactions on Graphics*, 27(3):#85, 1–10.

[Ambruş et al., 2014] Ambruş, R., Bore, N., Folkesson, J., and Jensfelt, P. (2014). Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1854–1861.

[Anand et al., 2012] Anand, A., Koppula, H. S., Joachims, T., and Saxena, A. (2012). Contextually guided semantic labeling and search for 3d point clouds. *IJRR*.

[Andersen et al., 2012] Andersen, M., Jensen, T., Lisouski, P., Mortensen, A., Hansen, M., Gregersen, T., and Ahrendt, P. (2012). Kinect depth sensor evaluation for computer vision applications. *Technical Report Electronics and Computer Engineering*, 1(6).

[Angeli et al., 2008] Angeli, A., Filliat, D., Doncieux, S., and Meyer, J.-A. (2008). Fast and incremental method for loop-closure detection using bags of visual words. *Robotics, IEEE Transactions on*, 24(5):1027–1037.

[Anguelov et al., 2002] Anguelov, D., Biswas, R., Koller, D., Limketkai, B., and Thrun, S. (2002). Learning hierarchical object maps of non-stationary environments with mobile robots. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 10–17. Morgan Kaufmann Publishers Inc.

[Avetisyan et al., 2019] Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A. X., and Nießner, M. (2019). Scan2cad: Learning cad model alignment in rgb-d scans.

[Bartoli and Sturm, 2003] Bartoli, A. and Sturm, P. (2003). Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *International Journal of Computer Vision*, 52(1):45–64.

[Batlle et al., 1998] Batlle, J., Mouaddib, E. M., and Salvi, J. (1998). Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern Recognition*, 31:963–982.

[Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359.

[Besl and McKay, 1992] Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Trans. PAMI*, 14(2):239–256.

[Birdal and Ilic, 2015] Birdal, T. and Ilic, S. (2015). Point pair features based object detection and pose estimation revisited. In *2015 International Conference on 3D Vision*, pages 527–535.

[Biswas et al., 2002] Biswas, R., Limketkai, B., Sanner, S., and Thrun, S. (2002). Towards object mapping in non-stationary environments with mobile robots. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 1014–1019. IEEE.

[Bore et al., 2017] Bore, N., Ekekrantz, J., Jensfelt, P., and Folkesson, J. (2017). Detection and tracking of general movable objects in large 3d maps. *arXiv preprint arXiv:1712.08409*.

[Bosche et al., 2009] Bosche, F., Haas, C. T., and Akinci, B. (2009). Automated recognition of 3d cad objects in site laser scans for project 3d status visualization and performance control. *Journal of Computing in Civil Engineering*, 23(6):311–318.

[Boykov et al., 2001] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239.

[Brown and Rusinkiewicz, 2007] Brown, B. and Rusinkiewicz, S. (2007). Global non-rigid alignment of 3D scans. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3).

[Brown et al., 2008] Brown, B. J., Toler-Franklin, C., Nehab, D., Burns, M., Dobkin, D., Vlachopoulos, A., Doumas, C., Rusinkiewicz, S., and Weyrich, T. (2008). A system for high-volume acquisition and matching of fresco fragments: Reassembling Theran wall paintings. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 27(3).

[Caruso et al., 2015] Caruso, D., Engel, J., and Cremers, D. (2015). Large-scale direct slam for omnidirectional cameras. In *International Conference on Intelligent Robots and Systems (IROS)*.

[Chang et al., 2017] Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y. (2017). Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*.

[Chang et al., 2015] Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012.

[Chen et al., 2013] Chen, J., Bautembach, D., and Izadi, S. (2013). Scalable real-time volumetric surface reconstruction. *ACM Trans. Graph.*, 32(4):113:1–113:16.

[Chen and Medioni, 1991] Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729 vol.3.

[Chen et al., 2014] Chen, Z., Lam, O., Jacobson, A., and Milford, M. (2014). Convolutional neural network-based place recognition. *arXiv preprint arXiv:1411.1509*.

[Choi et al., 2015] Choi, S., Zhou, Q.-Y., and Koltun, V. (2015). Robust reconstruction of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Corsini et al., 2012] Corsini, M., Cignoni, P., and Scopigno, R. (2012). Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):914–924.

[Curless and Levoy, 1996] Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA. ACM.

[Dai et al., 2017] Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

[Dai and Nießner, 2018] Dai, A. and Nießner, M. (2018). 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Dai et al., 2016a] Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., and Theobalt, C. (2016a). Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *arXiv preprint arXiv:1604.01093*.

[Dai et al., 2016b] Dai, A., Qi, C. R., and Nießner, M. (2016b). Shape completion using 3d-encoder-predictor cnns and shape synthesis. *CoRR*, abs/1612.00101.

[Dai et al., 2018] Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., and Nießner, M. (2018). Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

[Delong et al., 2012] Delong, A., Osokin, A., Isack, H. N., and Boykov, Y. (2012). Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96(1):1–27.

[Di Cicco et al., 2015] Di Cicco, M., Iocchi, L., and Grisetti, G. (2015). Non-parametric calibration for depth sensors. *Robot. Auton. Syst.*, 74(PB):309–317.

[Dou et al., 2012] Dou, M., Guan, L., Frahm, J.-M., and Fuchs, H. (2012). Exploring high-level plane primitives for indoor 3d reconstruction with a hand-held rgb-d camera. In *Asian Conference on Computer Vision*, pages 94–108.

[Drost et al., 2010] Drost, B., Ulrich, M., Navab, N., and Ilic, S. (2010). Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 998–1005.

[Dzitsiuk et al., 2016] Dzitsiuk, M., Sturm, J., Maier, R., Ma, L., and Cremers, D. (2016). De-noising, stabilizing and completing 3d reconstructions on-the-go using plane priors. *CoRR*, abs/1609.08267.

[Ekekrantz et al., 2016] Ekekrantz, J., Bore, N., Ambrus, R., Folkesson, J., and Jensfelt, P. (2016). Towards an adaptive system for lifelong object modelling. *ICRA Workshop: AI for Long-term Autonomy*.

[Elghor et al., 2015] Elghor, H. E., Roussel, D., Ababsa, F., and Bouyakhf, E. H. (2015). Planes detection for robust localization and mapping in rgb-d slam systems. In *3D Vision (3DV), 2015 International Conference on*, pages 452–459.

[Engel et al., 2016] Engel, J., Koltun, V., and Cremers, D. (2016). Direct sparse odometry. In *arXiv:1607.02565*.

[Engel et al., 2014] Engel, J., Schöps, T., and Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*.

[Estrada et al., 2005] Estrada, C., Neira, J., and Tardos, J. (2005). Hierarchical slam: Real-time accurate mapping of large environments. *Transactions on Robotics*, 21(4):588–596.

[Fan et al., 2016] Fan, X., Zhang, L., Brown, B., and Rusinkiewicz, S. (2016). Automated view and path planning for scalable multi-object 3D scanning. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 35(6).

[Fanello et al., 2016] Fanello, S. R., Rhemann, C., Tankovich, V., Kowdle, A., Escolano, S. O., Kim, D., and Izadi, S. (2016). Hyperdepth: Learning depth from structured light without matching. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5441–5450.

[Fanello et al., 2017] Fanello, S. R., Valentin, J., Rhemann, C., Kowdle, A., Tankovich, V., Davidson, P., and Izadi, S. (2017). Ultrastereo: Efficient learning-based matching for active stereo systems. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6535–6544.

[FARO Technologies, 2019] FARO Technologies, I. (2019). Faro focus scanner. [Online; accessed 14-May-2019].

[Fäulhammer et al., 2017] Fäulhammer, T., Ambruş, R., Burbridge, C., Zillich, M., Folkesson, J., Hawes, N., Jensfelt, P., and Vincze, M. (2017). Autonomous learning of object models on a mobile robot. *IEEE Robotics and Automation Letters*, 2(1):26–33.

[Fehr et al., 2017] Fehr, M., Furrer, F., Dryanovski, I., Sturm, J., Gilitschenski, I., Siegwart, R., and Cadena, C. (2017). Tsdf-based change detection for consistent long-term dense reconstruction and dynamic object discovery. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5237–5244. IEEE.

[Finman et al., 2014] Finman, R., Whelan, T., Paull, L., and Leonard, J. J. (2014). Physical words for place recognition in dense rgb-d maps. In *ICRA workshop on visual place recognition in changing environments*.

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

[Freedman et al., 2007] Freedman, B., Machline, A. S. M., and Arieli, Y. (2007). Depth mapping using projected patterns.

[Frese et al., 2005] Frese, U., Larsson, P., and Duckett, T. (2005). A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):1–12.

[Furukawa and Ponce, 2010] Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376.

[Gallagher et al., 2009] Gallagher, G., Srinivasa, S. S., Bagnell, J. A., and Ferguson, D. (2009). Gatmo: A generalized approach to tracking movable objects. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2043–2048. IEEE.

[Gelfand et al., 2003] Gelfand, N., Ikemoto, L., Rusinkiewicz, S., and Levoy, M. (2003). Geometrically stable sampling for the icp algorithm. In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, pages 260–267.

[Gelfand et al., 2005] Gelfand, N., Mitra, N. J., Guibas, L. J., and Pottmann, H. (2005). Robust global registration. In *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Gokturk et al., 2004] Gokturk, S. B., Yalcin, H., and Bamji, C. (2004). A time-of-flight depth sensor - system description, issues and solutions. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 35–35.

[Golparvar-Fard et al., 2012] Golparvar-Fard, M., Pena-Mora, F., and Savarese, S. (2012). Automated progress monitoring using unordered daily construction photographs and ifc-based building information models. *Journal of Computing in Civil Engineering*, 29(1):04014025.

[Graham et al., 2018] Graham, B., Engelcke, M., and van der Maaten, L. (2018). 3d semantic segmentation with submanifold sparse convolutional networks. *CVPR*.

[Grisetti et al., 2010] Grisetti, G., Kümmerle, R., Stachniss, C., and Burgard, W. (2010). A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43.

[Halber and Funkhouser, 2017] Halber, M. and Funkhouser, T. (2017). Fine-to-coarse global registration of rgb-d scans. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

[Hall-Holt and Rusinkiewicz, 2001] Hall-Holt, O. and Rusinkiewicz, S. (2001). Stripe boundary codes for real-time structured light range scanning of moving objects. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 359–366 vol.2.

[Han and Golparvar-Fard, 2015] Han, K. and Golparvar-Fard, M. (2015). Bim-assisted structure-from-motion for analyzing and visualizing construction progress deviations through daily site images and bim. In *Congress on Computing in Civil Engineering, Proceedings*, volume 2015, pages 596–603.

[Handa et al., 2014] Handa, A., Whelan, T., McDonald, J., and Davison, A. (2014). A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

[Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

[Hattori and Sato, 1996] Hattori, K. and Sato, Y. (1996). Accurate rangefinder with laser pattern shifting. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 3, pages 849–853 vol.3.

[Hawes et al., 2017] Hawes, N., Burbridge, C., Jovan, F., Kunze, L., Lacerda, B., Mudrova, L., Young, J., Wyatt, J., Hebesberger, D., Kortner, T., Ambrus, R., Bore, N., Folkesson, J., Jensfelt, P., Beyer, L., Hermans, A., Leibe, B., Aldoma, A., Faulhammer, T., Zillich, M., Vincze, M., Chinellato, E., Al-Omari, M., Duckworth, P., Gatsoulis, Y., Hogg, D. C., Cohn, A. G., Dondrup, C., Pulido Fentanes, J., Krajnik, T., Santos, J. M., Duckett, T., and Hanheide, M. (2017). The strands project: Long-term autonomy in everyday environments. *IEEE Robotics Automation Magazine*, 24(3):146–156.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask R-CNN. *CoRR*, abs/1703.06870.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

[Henry et al., 2010] Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2010). Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *International Symposium on Experimental Robotics (ISER)*.

[Herbst et al., 2014] Herbst, E., Henry, P., and Fox, D. (2014). Toward online 3-d object segmentation and mapping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3193–3200. IEEE.

[Herrera C. et al., 2012] Herrera C., D., Kannala, J., and Heikkilä, J. (2012). Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2058–2064.

[Horn and Kiryati, 1997] Horn, E. and Kiryati, N. (1997). Toward optimal structured light patterns. In *Proceedings. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No.97TB100134)*, pages 28–35.

[HP, 2019] HP, L. (2019). Hp 3d scan. [Online; accessed 23-May-2019].

[Hua et al., 2016] Hua, B.-S., Pham, Q.-H., Nguyen, D. T., Tran, M.-K., Yu, L.-F., and Yeung, S.-K. (2016). Scenenn: A scene meshes dataset with annotations. In *International Conference on 3D Vision (3DV)*.

[Huang et al., 2018] Huang, J., Zhang, H., Yi, L., Funkhouser, T., Nießner, M., and Guibas, L. (2018). Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. *arXiv preprint arXiv:1812.00020*.

[Iddan and Yahav, 2001] Iddan, G. J. and Yahav, G. (2001). Three-dimensional imaging in the studio and elsewhere. volume 4298, pages 48–55.

[Intel, 2019] Intel, C. (2019). Intel realsense. [Online; accessed 23-May-2019].

[Johnson and Hebert, 1999] Johnson, A. E. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449.

[Johnson et al., 2016] Johnson, J., Karpathy, A., and Fei-Fei, L. (2016). Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Kahler et al., 2015] Kahler, O., Prisacariu, V. A., Ren, C. Y., Sun, X., Torr, P. H. S., and Murray, D. W. (2015). Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. *IEEE Transactions on Visualization and Computer Graphics (Proceedings International Symposium on Mixed and Augmented Reality 2015*, 22(11).

[Karsch et al., 2014] Karsch, K., Golparvar-Fard, M., and Forsyth, D. (2014). Constructaide: analyzing and visualizing construction sites through photographs and building models. *ACM Transactions on Graphics (TOG)*, 33(6):176.

[Kazhdan and Hoppe, 2013] Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29:1–29:13.

[Keller et al., 2013a] Keller, M., Lefloch, D., Lambers, M., Izadi, S., Weyrich, T., and Kolb, A. (2013a). Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision – 3DV*, pages 1 – 8, DOI:10.1109/3DV.2013.9.

[Keller et al., 2013b] Keller, M., Lefloch, D., Lambers, M., Izadi, S., Weyrich, T., and Kolb, A. (2013b). Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *Proceedings of Joint 3DIM/3DPVT Conference (3DV)*, page 8.

[Khoury et al., 2017] Khoury, M., Zhou, Q.-Y., and Koltun, V. (2017). Learning compact geometric features. In *International Conference on Computer Vision (ICCV)*.

[Klein and Murray, 2007] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan.

[Klingensmith et al., 2015] Klingensmith, M., Dryanovski, I., Srinivasa, S., and Xiao, J. (2015). Chisel: Real time large scale 3d reconstruction onboard a mobile device. In *Robotics Science and Systems 2015*.

[Knapitsch et al., 2017] Knapitsch, A., Park, J., Zhou, Q.-Y., and Koltun, V. (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4).

[Krajník et al., 2017] Krajník, T., Fentanes, J. P., Santos, J. M., and Duckett, T. (2017). Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments. *IEEE Transactions on Robotics*, 33(4):964–977.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

[Kuhn, 1955] Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

[Lai et al., 2014] Lai, K., Bo, L., and Fox, D. (2014). Unsupervised feature learning for 3d scene labeling. In *IEEE International Conference on Robotics and Automation*, page 3050–3057.

[Lee and C. Fowlkes, 2017] Lee, M. and C. Fowlkes, C. (2017). Space-time localization and mapping. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3932–3941.

[Levoy et al., 2000] Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., and Fulk, D. (2000). The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 131–144, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[Li et al., 2011] Li, Y., Wu, X., Chrysanthou, Y., Sharf, A., Cohen-Or, D., and Mitra, N. J. (2011). Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics*, 30(4):52:1–52:12.

[Li Zhang et al., 2002] Li Zhang, Curless, B., and Seitz, S. M. (2002). Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*, pages 24–36.

[lim Low, 2004] lim Low, K. (2004). Linear least-squares optimization for point-to-plane icp surface registration. Technical report, University of North Carolina at Chapel Hill.

[Liu and Furukawa, 2019] Liu, C. and Furukawa, Y. (2019). Masc: Multi-scale affinity with sparse convolution for 3d instance segmentation.

[Long et al., 2015] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

[Ma et al., 2016] Ma, L., Kerl, C., Stueckler, J., and Cremers, D. (2016). Cpa-slam: Consistent plane-model alignment for direct rgb-d slam. In *Int. Conf. on Robotics and Automation*.

[Martin-Brualla et al., 2015] Martin-Brualla, R., Gallup, D., and Seitz, S. M. (2015). Time-lapse mining from internet photos. *ACM Transactions on Graphics (TOG)*, 34(4):62.

[Martinez and Stiefelhagen, 2013] Martinez, M. and Stiefelhagen, R. (2013). Kinect unleashed: Getting control over high resolution depth maps. In *Proceedings of the 13. IAPR International Conference on Machine Vision Applications, MVA 2013, Kyoto, Japan, May 20-23, 2013*, pages 247–250.

[Martinez and Stiefelhagen, 2014] Martinez, M. and Stiefelhagen, R. (2014). Kinect unbiased. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5791–5795.

[MathWorks, 2019a] MathWorks (2019a). Single camera calibrator app. [Online; accessed 14-May-2019].

[MathWorks, 2019b] MathWorks (2019b). Stereo camera calibrator app. [Online; accessed 14-May-2019].

[Mattausch et al., 2014] Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., and Pajarola, R. (2014). Object detection and classification from large-scale cluttered indoor scans. *Computer Graphics Forum*, 33(2):11–21.

[Matterport, 2019] Matterport, I. (2019). Matterport pro2 3d camera. [Online; accessed 21-May-2019].

[Matzen and Snavely, 2014] Matzen, K. and Snavely, N. (2014). Scene chronology. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Mellado et al., 2014] Mellado, N., Aiger, D., and Mitra, N. J. (2014). Super 4pcs fast global pointcloud registration via smart indexing. *Computer Graphics Forum*, 33(5):205–215.

[Microsoft, 2019] Microsoft (2019). Kinect for windows. [Online; accessed 23-May-2019].

[Mitra et al., 2004] Mitra, N. J., Gelfand, N., Pottmann, H., and Guibas, L. (2004). Registration of point cloud data from a geometric optimization perspective. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, pages 22–31, New York, NY, USA. ACM.

[Monszpart et al., 2015] Monszpart, A., Mellado, N., Brostow, G. J., and Mitra, N. J. (2015). Rapter: Rebuilding man-made scenes with regular arrangements of planes. *ACM Trans. Graph.*, 34(4):103:1–103:12.

[Mur-Artal et al., 2015] Mur-Artal, R., Montiel, J. M. M., and Tardós, J. D. (2015). ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163.

[Narita et al., 2019] Narita, G., Seno, T., Ishikawa, T., and Kaji, Y. (2019). Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. *CoRR*, abs/1903.01177.

[Newcombe et al., 2015] Newcombe, R., Fox, D., and Seitz, S. (2015). Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

[Newcombe et al., 2011] Newcombe, R. A., Davison, A. J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., and Fitzgibbon, . A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE.

[Newcombe et al., 2011] Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). Dtam: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision*, pages 2320–2327.

[NextEngine, 2019] NextEngine, I. (2019). Nextengine scanner camera. [Online; accessed 21-May-2019].

[Nguyen et al., 2012] Nguyen, C. V., Izadi, S., and Lovell, D. (2012). Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, pages 524–530.

[Nguyen et al., 2015] Nguyen, T., Reitmayr, G., and Schmalstieg, D. (2015). Structural modeling from depth images. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1230–1240.

[Nguyen et al., 2007] Nguyen, V., Harati, A., and Siegwart, R. (2007). A lightweight slam algorithm using orthogonal planes for indoor mobile robotics. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 658–663. IEEE.

[Nießner et al., 2013] Nießner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. (2013). Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*.

[Park et al., 2017] Park, J., Zhou, Q.-Y., and Koltun, V. (2017). Colored point cloud registration revisited. In *ICCV*.

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

[Posdamer and Altschuler, 1982] Posdamer, J. L. and Altschuler, M. D. (1982). Surface measurement by space-encoded projected beam systems. *Computer Graphics and Image Processing*, 18(1):1–17.

[Pulli, 1999] Pulli, K. (1999). Multiview registration for large data sets. In *Proceedings of the 2Nd International Conference on 3-D Digital Imaging and Modeling*, 3DIM'99, pages 160–168, Washington, DC, USA. IEEE Computer Society.

[Qi et al., 2017a] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Qi et al., 2017b] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*.

[Ranftl et al., 2016] Ranftl, R., Vineet, V., Chen, Q., and Koltun, V. (2016). Dense monocular depth estimation in complex dynamic scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4058–4066.

[Ratter and Sammut, 2015] Ratter, A. and Sammut, C. (2015). Local map based graph slam with hierarchical loop closure and optimisation.

[Rebolj et al., 2017] Rebolj, D., Pučko, Z., Babič, N. Č., Bizjak, M., and Mongus, D. (2017). Point cloud quality requirements for scan-vs-bim based automated construction progress monitoring. *Automation in Construction*, 84:323–334.

[Riegler et al., 2017] Riegler, G., Ulusoy, A. O., and Geiger, A. (2017). Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Rusinkiewicz, 2019] Rusinkiewicz, S. (2019). A symmetric objective function for ICP. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 38(4).

[Rusinkiewicz et al., 2002] Rusinkiewicz, S., Hall-Holt, O., and Levoy, M. (2002). Real-time 3D model acquisition. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 21(3):438–446.

[Rusinkiewicz and Levoy, 2001] Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152.

[Rusu et al., 2009] Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217.

[Salas-Moreno et al., 2014] Salas-Moreno, R., Glocken, B., Kelly, P., and Davison, A. (2014). Dense planar slam. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 157–164.

[Salas-Moreno et al., 2013] Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P. H. J., and Davison, A. J. (2013). SLAM++: simultaneous localisation and mapping at the level of objects. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 1352–1359.

[Santos et al., 2017] Santos, J. M., Krajník, T., and Duckett, T. (2017). Spatio-temporal exploration strategies for long-term autonomy of mobile robots. *Robotics and Autonomous Systems*, 88:116–126.

[Schindler et al., 2007] Schindler, G., Dellaert, F., and Kang, S. B. (2007). Inferring temporal order of images from 3d structure. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7.

[Schulz and Burgard, 2001] Schulz, D. and Burgard, W. (2001). Probabilistic state estimation of dynamic objects with a moving mobile robot. *Robotics and Autonomous Systems*, 34(2-3):107–115.

[Shotton et al., 2011] Shotton, J., Fitzgibbon, A., Blake, A., Kipman, A., Finocchio, M., Moore, B., and Sharp, T. (2011). Real-time human pose recognition in parts from a single depth image. In *CVPR*. IEEE. Best Paper Award.

[Silberman et al., 2012] Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *Proc. European Conf. on Comp. Vision*.

[Smisek et al., 2011] Smisek, J., Jancosek, M., and Pajdla, T. (2011). 3d with kinect. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1154–1160.

[Snavely et al., 2006] Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA. ACM Press.

[Song and Xiao, 2013] Song, S. and Xiao, J. (2013). Tracking revisited using rgbd camera: Unified benchmark and baselines. In *Proceedings of the IEEE international conference on computer vision*, pages 233–240.

[Song et al., 2017] Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T. (2017). Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*.

[Stotko, 2016] Stotko, P. (2016). State of the art in real-time registration of rgb-d images. In *CESCG*.

[Stühmer et al., 2010] Stühmer, J., Gumhold, S., and Cremers, D. (2010). Real-time dense geometry from a handheld camera. In Goesele, M., Roth, S., Kuijper, A., Schiele, B., and Schindler, K., editors, *Pattern Recognition*, pages 11–20, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Sturm et al., 2012] Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*.

[Szeliski, 2010] Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition.

[Taguchi et al., 2013] Taguchi, Y., Jian, Y.-D., Ramalingam, S., and Feng, C. (2013). Point-plane slam for hand-held 3d sensors. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5182–5189.

[Tang and Feng, 2015] Tang, Y. and Feng, J. (2015). Hierarchical multiview rigid registration. *Comput. Graph. Forum*, 34(5):77–87.

[Tatarchenko et al., 2018] Tatarchenko, M., Park, J., Koltun, V., and Zhou., Q.-Y. (2018). Tangent convolutions for dense prediction in 3D. *CVPR*.

[Teichman et al., 2013] Teichman, A., Miller, S., and Thrun, S. (2013). Unsupervised intrinsic calibration of depth sensors via SLAM. In *Robotics: Science and Systems*.

[Trevor et al., 2012] Trevor, A., Rogers III, J., and Christensen, H. (2012). Planar surface SLAM with 3D and 2D sensors. In *ICRA*.

[Triggs et al., 2000] Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, UK. Springer-Verlag.

[Turkan et al., 2012] Turkan, Y., Bosche, F., Haas, C. T., and Haas, R. (2012). Automated progress tracking using 4d schedule and 3d sensing technologies. *Automation in Construction*, 22:414–421.

[Tuttas et al., 2017] Tuttas, S., Braun, A., Borrmann, A., and Stilla, U. (2017). Acquisition and consecutive registration of photogrammetric point clouds for construction progress monitoring using a 4d bim. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 85(1):3–15.

[Valgaerts et al., 2012] Valgaerts, L., Bruhn, A., Mainberger, M., and Weickert, J. (2012). Dense versus sparse approaches for estimating the fundamental matrix. *International Journal of Computer Vision*, 96:212–234.

[Wang and Thorpe, 2002] Wang, C.-C. and Thorpe, C. (2002). Simultaneous localization and mapping with detection and tracking of moving objects. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2918–2924. IEEE.

[Wang et al., 2003] Wang, C.-C., Thorpe, C., and Thrun, S. (2003). Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 842–849. IEEE.

[Wang et al., 2016] Wang, H., Wang, J., and Wang, L. (2016). Online reconstruction of indoor scenes from rgb-d streams. In *IEEE CVPR*.

[Weingarten and Siegwart, 2006] Weingarten, J. and Siegwart, R. (2006). 3D SLAM using planar segments. In *IROS*, page 3062–3067.

[Whelan et al., 2012] Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., and McDonald, J. (2012). Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia.

[Whelan et al., 2014] Whelan, T., Kaess, M., Johannsson, H., Fallon, M., Leonard, J., and McDonald, J. (2014). Real-time large scale dense RGB-D SLAM with volumetric fusion. *Intl. J. of Robotics Research, IJRR*.

[Whelan et al., 2015a] Whelan, T., Leutenegger, S., Salas-Moreno, R. F., Glocker, B., and Davison, A. J. (2015a). ElasticFusion: Dense SLAM without a pose graph. In *Robotics: Science and Systems (RSS)*, Rome, Italy.

[Whelan et al., 2015b] Whelan, T., Leutenegger, S., Salas-Moreno, R. F., Glocker, B., and Davison, A. J. (2015b). ElasticFusion: Dense SLAM without a pose graph. In *Robotics: Science and Systems (RSS)*, Rome, Italy.

[Xiao et al., 2013a] Xiao, J., Owens, A., and Torralba, A. (2013a). Sun3d: A database of big spaces reconstructed using sfm and object labels. *Computer Vision, IEEE International Conference on*, 0:1625–1632.

[Xiao et al., 2013b] Xiao, J., Owens, A., and Torralba, A. (2013b). SUN3D database: Semantic RGB-D bundle adjustment with human in the loop. In *Proc. Int. Conf. on Comp. Vision*.

[Yan et al., 2014] Yan, F., Sharf, A., Lin, W., Huang, H., and Chen, B. (2014). Proactive 3d scanning of inaccessible parts. *ACM Trans. Graph.*, 33(4):157:1–157:8.

[Young et al., 2017] Young, J., Basile, V., Suchi, M., Kunze, L., Hawes, N., Vincze, M., and Caputo, B. (2017). Making sense of indoor spaces using semantic web mining and situated robot perception. In *European Semantic Web Conference*, pages 299–313. Springer.

[Yu et al., 2015] Yu, F., Xiao, J., and Funkhouser, T. (2015). Semantic alignment of lidar data at city scale. In *28th IEEE Conference on Computer Vision and Pattern Recognition*.

[Zeng et al., 2016] Zeng, A., Song, S., Nießner, M., Fisher, M., and Xiao, J. (2016). 3dmatch: Learning the matching of local 3d geometry in range scans. *arXiv preprint arXiv:1603.08182*.

[Zhang et al., 2015] Zhang, Y., Xu, W., Tong, Y., and Zhou, K. (2015). Online structure analysis for real-time indoor scene reconstruction. *ACM Transactions on Graphics (TOG)*, 34(5):159.

[Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.

[Zhou and Koltun, 2013] Zhou, Q.-Y. and Koltun, V. (2013). Dense scene reconstruction with points of interest. *ACM Transactions on Graphics*, 32(4).

[Zhou and Koltun, 2014] Zhou, Q.-Y. and Koltun, V. (2014). Color map optimization for 3d reconstruction with consumer depth cameras. *SIGGRAPH Conf. Proc.*

[Zhou and Koltun, 2015] Zhou, Q.-Y. and Koltun, V. (2015). Depth camera tracking with contour cues. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Zhou et al., 2016] Zhou, Q.-Y., Park, J., and Koltun, V. (2016). Fast global registration. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 766–782, Cham. Springer International Publishing.