INDOOR SCENE RECONSTRUCTION USING THE MANHATTAN ASSUMPTION

By

LIN GUO

Bachelor of Science in Electrical Engineering

Tianjin University

Tianjin, China

2012

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2015

INDOOR SCENE RECONSTRUCTION USING THE

MANHATTAN ASSUMPTION

Thesis Approved:

Dr. Guoliang Fan

Thesis Adviser

Dr. Weihua Sheng

Dr. Damnon Chandler

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my committee members: Guoliang Fan, Weihua Sheng, Damon Chandler, for their willingness to mentor and providing me with excellent atmosphere for research. I would never have been able to finish this thesis without their help.

Special gratitude goes to my advisor, Dr. Guoliang Fan, for his patient guidance, enthusiasm for research and sharing valuable experience. His thoughtful advice on the research and my thesis writing have had more of an impact on my life than he will ever know. Also I would like to thank my fellow graduate students, Aravind, Mahdi, for helping my research work and being the best co-workers.

Finally, I would like to thank my parents and my girlfriend Xuewen Wang. Their understanding and kind encouragement always help me to go through all the difficulties.

This work is supported by the National Science Foundation (NSF) under grant NRI-1427345.

Name: LIN GUO

Date of Degree: JULY, 2015

Title of Study: INDOOR SCENE RECONSTRUCTION USING THE MANHATTAN ASSUMPTION

Major Field: Electrical Engineering

Abstract: Robots are being developed to be co-inhabitants to help the elderly people in an assisted environment. A semantic map can provide robots a lot of information in the environment they cohabit with people. So far, most mapping algorithms have been limited to build maps only based on visible points without much consideration on the occluded parts. This research is two-fold. First, it aims to develop a complete map to help robots gain a deeper insight of the house. The second goal is to reconstruct scenes by mimicking people's indoor understanding. Based on the Manhattan assumption, we propose a technique that separates an indoor scene into major structures and indoor objects. The room structures are reconstructed with ideal planes to render each side of the room. The unseen regions of major structures and objects are generated by extending visible planes. Our system is applied to an artificial kitchen scene and a typical living-room scene. The results show that the generated maps are more complete and semantically meaningful than the ones created by traditional data-driven approaches. Our algorithm has great potential to improve robots' efficiency by accurately locating itself in a cluttered scene and finding useful objects.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

1.1 Motivation and Background

Elderly people have a growing number in US. A number of them tend to live independently or in an assisted living community. Much research has been conducted on developing robots to be co-inhabitants or co-workers. Robot can help the elderly people in their living and monitor their health while keeping their privacy. Previously, robots have to depend on human commands. Ideally, the human-robot interaction (HRI) should be done in a more collaborative or cooperative way. Under this circumstance, assistance robots have a limited sense of the environment that they are in. Recently, a number of researchers have started to focus on developing a 3D map with the help of fast- developed depth sensor. By combining the technique of HRI, and self-localization and mapping (SLAM), many researchers [1], [2] have developed innovative projects. One of the ongoing research projects is about building a smart home for elder people. The objective of this project is to develop a robot that can do daily housework and to monitor the elder in case any emergency would occur. This robot, a human-like co-inhabitant, serves as a housekeeper, and it can do its work without humans' consecutive commands. On contrast, the traditional robot, which follows the owner's commands from time to time, makes people feel being watched with less privacy. To achieve this goal, robots need a semantic indoor map, which provides a good understanding of the house and allows the robot to be aware of the situation. So that it can be

proactive to arrange suitably work. For example, after finding the owner is making breakfast in the kitchen, the robot moves to the bedroom to do some organization. So the semantic map plays an important role for the robot to assist people's living.

Figure 1.1: A robot is serving a senior [3].

People have been trying to increase the accuracy of maps for many years [4]. The fast development of depth sensor makes the 3D mapping become a real popular topic. Depth sensors can provide depth maps, from which we can get point clouds: each pixel in a depth map has a corresponding position in 3D space. Although the point cloud is made of discrete points, it still can provide rich information for 3D modeling and visualization. A lot of work has been done to convert the point cloud to a solid model.

Some research efforts are focused on improving the accuracy of 3D measurements of point clouds in order to show more details. Other researcher try to incorporate some assumption or prior knowledge to build a 3D model from the point cloud. The mostly used assumption is the

Manhattan assumption, which assumes that the major planes in an indoor scene probably follow one of the three major coordinates. Researchers used this assumption to segment the map of the indoor scene to get more indoor information of it. However, the main problem for depth sensors, such as Kinect, is that the sensor can only capture partial appearance of a scene. People can understand scenes better not only because we know what is seen, but also because we can infer what is unseen, especially in an indoor environment where many occlusions exist. It is almost improbable for the point cloud to get every detail of the indoor scene. So our object is to make the computer understand the scene by separating the objects from major room structure and inferring the occluded parts based on the Manhattan assumption.

1.2 Objectives and Approaches

As the first step to build a semantic map for robots, the specific task of our research is to identify the room structure (walls and floor) from the indoor scene point cloud, to build an ideal model of the room structure to make each indoor object have a complete shape, and to build a complete shape for each object from partial data.

The algorithm used in this research is developed by only considering the distribution characteristics of the 3D point cloud, without texture or any color-based cues. We focus on the general shape of the room where the Manhattan assumption can apply. We use the idea of down-sampling to keep the main planes of the scene and to get rid of insignificant details. Then we extract the norm features from the point cloud and use clustering algorithms to find the major coordinates. Afterwards, we find those planes that indicate the indoor structures, such as walls and floor. We reconstruct those planes in an ideal form: perpendicular, smooth, flat and connected. Thus an ideal room is made. After removing the points that are associated with the

room structure, the unseen planes of the indoor objects will be determined by assuming each object has a convex shape.

1.3 Contributions

Building a semantic map is a complex task combining vision, sound and interactions with humans. Our work is one of the fundamental parts of generating a semantic map. Based on the Manhattan assumption, we focused on the general structure of the room, and proposed a structure reconstruction algorithm that considers only the big picture and reduces the influences of trivial details. Moreover, based on the fact that the point cloud cannot show all the indoor planes, a method was introduced to infer each indoor object's possessing space. Our work can make the computer have a good understanding of the indoor scene and provide a good starting point for future research.

In this thesis, Chapter 2 presents some related work and introduces our work in the field of 3D modeling using depth sensors. We will also discuss several different research directions of vision-based scene understanding.

Chapter 3 describes our method applied to a single depth frame of an artificial scene that simulates a smart home for an elderly. This scene shows a typical structure of a kitchen, without any decorations. This is the first step of our algorithm, and our objective is to separate the walls and floor from the point cloud and make ideal ones to replace them.

Chapter 4 is about indoor scene analysis. Our algorithm is applied to a more general case with a large point cloud of a real indoor scene. The techniques of density control and a two-step major coordinate extraction method will be introduced. The dataset is provided by K. Lai et al. [5]. The point cloud is made by fusing multiple Kinect frames without any further refinements. The

objective of this chapter is to analysis general large-scale indoor scenes. We will separate the room structure from the point cloud and generate an ideal room structure model.

In Chapter 5, we are focused on indoor object representation and reconstruction. We compensate the unseen planes of the objects by assuming they have a cube shape. Then use a voxel method to infer the unseen planes to get a rough model of the object. The objective of this chapter is to jointly represent major indoor structures along with individual objects.

Finally, we will present our conclusions and discuss different directions for our further research.

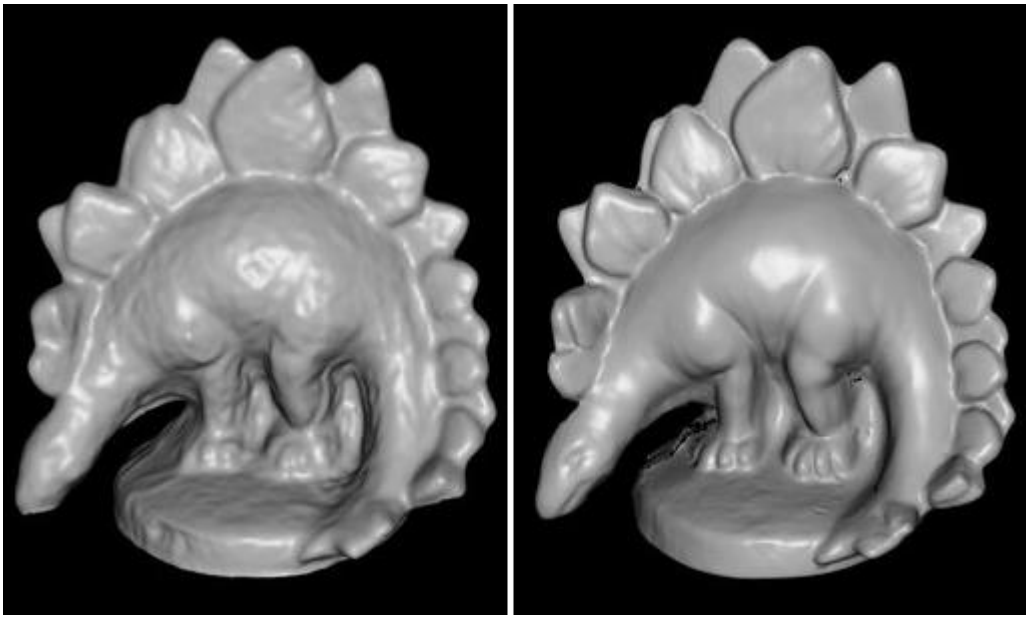
CHAPTER II

RELATED WORK

Our research is based on some existing research works. We will discuss them one by one in the following sections. In particular, a review of the traditional 3D modeling is present. Following that, some recent research with help of depth sensors on indoor reconstruction will be discussed. After it, we will focus on the indoor scene representation, which is found highly valuable in indoor scene reconstruction. At last, we will have a discussion about the Manhattan assumption.

2.1 3D Modeling

It has been a long time since people started trying to build 3D models of real objects and scenes. Numerous reconstruction algorithms have been developed to build 3D models from 2D images. SM Seitz et al. [6] provided several datasets with high quality to be evaluated and benchmark the performance of reconstruction algorithms. They used the Stanford spherical gantry [7] to get the specific latitude and longitude angles. After replacing the object with a chessboard they used the Matlab toolbox for camera calibration [8] to estimate camera parameters. All of the work was to get the camera parameters as accurate as possible. Given the same images and corresponding camera parameters, algorithms developed by researchers with different methods could get evaluated and compared.



(a) (b)

Figure 2.1: (a) Sample reconstruction using point-cloud based method. The surface is not smooth after meshing; (b) Ground truth [9].

Those methods can be roughly classified into three categories. Firstly, voxel based methods [10], [11], [12] , requiring the bounding box of the object, and the output resolution is based on the voxel size because of its quantification effect. Secondly, the method based on deformable polygonal meshes [13], [14], requires a good initial position to start the process of optimization, thus the applicability is reduced. At last, algorithms based on point clouds are simple and effective [15], [16], [17], but required poste-processes to get a solid model from sparse points, so the surface of the 3D model might not be smooth because the points are not highly close to each other (Fig.2.1 (a)). However, all the methods share a common problem: a quite long processing time is required to get a good reconstruction accuracy.

The traditional 3D reconstruction stereo reaches its bottleneck because the input images' corresponding camera parameters are not accurate. Thus researchers started to use the depth information provided by depth sensors to make more accurate models.

2.2 RGB-D Reconstruction

With the help of depth information, many more methods are developed. These methods are either based on expensive equipment such as time of light (TOF) sensors or a very high computational algorithm [18], [19]. Depth-based methods have become more and more popular, the recent explosion of the RGB-D reconstruction mainly contributes to the release of Microsoft Kinect, a depth sensor with very low cost [20].

Henry et al. [2] used Kinect to develop one of the first methods to make a full mapping system for indoor scenes. They used Generalized Iterative Closest Point (GICP) to form multiple frames into one 3D map. But in their procedure, the features they used were extracted from RGB images. The corresponding depth information for each feature point was used to be the initial position during the GICP process. In another word, the depth map provided by Kinect was only used to accelerate the whole process. So essentially, the algorithm is still a color image based 3D method.

To make a better use of the depth map, S. Izadi et al. provided the famous KinectFusion system [21]. They used voxel representation to output a solid 3D model instead of the widely used point cloud representation. The following problem of this method was the extremely expensive computation. Thus it could only be applied in a small area. Much research has been conducted such as Zeng et al. [22] and Keller et al. [23] based on KinectFusion. They tried to extend the mapping area to a bigger size. But when the mapping size was big enough to build a whole room or more, new problems came such as the global consistency and the loop closure problem. Thus, some restrains from indoor scene characters are required to build a better map.

2.3 Indoor Scene Understanding

As one of the main characters of high-level computer vision, computer's understanding of the vision is now paid more and more attention.

Bundle adjustment [24] solved the significant loop closure problem, as shown in Fig. 2.2. When connecting multiple views, the error of camera parameter estimation would get accumulated, thus the loop in the 3D model cannot be closed. The algorithm solved this problem by adjusting the camera parameters of each frame.

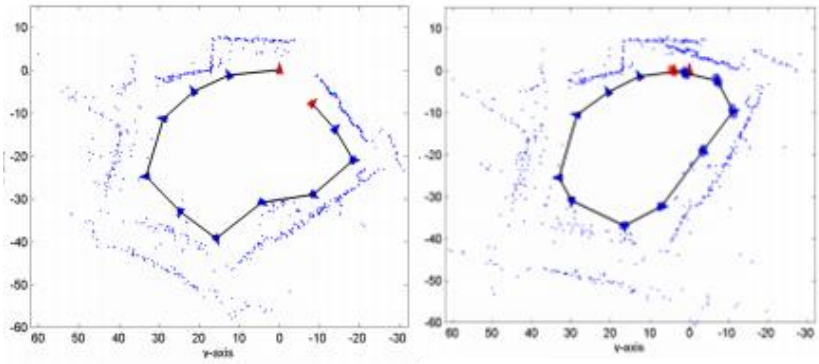


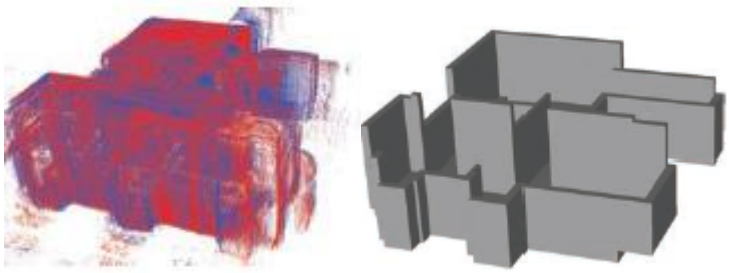
Figure 2.2: Schematic of the loop closure problem: when the contour goes back to the starting point, the loop cannot get closed [25].

The limitation of Bundle Adjustment is that it can only work after finishing the round trip. The features in the current frame will be compared with any other past frames to determine if it is a visited position. Thus, it requires more computational resource when dealing with large-scale scenes. To solve this problem, Labbe et al. [26] presented an idea that mimics the memory style of human: the important things would be memorized, the details would be ignored but could be recalled when being focused on. In their method, only some typical features of each frame would

be saved to be compared with the current frame. If the similarity were high, more features of the frame would be loaded to make a further comparison to determine if the loop closure were detected. Their work is very inspiring that they provided a way of solving the reconstruction problems by looking at the general and typical features.

The indoor scenes are all man made, just like the routes and cities. So they share a common typical feature: the Manhattan assumption [27], [28]. With the help of depth sensors, people can get more information from images and to do segmentations [29].

3D laser scan-based approaches are the traditional methods to get accurate depth information. But the laser scanner is high cost and with low frame rate. These makes it very inconvenient to handle. But even after the release of the low cost depth sensor, Kinect, the laser scanner is still very useful in many circumstances. One shortness of Kinect is that it saves every data point in its view. But using laser scanner, people can select the region they need. Xiao et al. [30] presented a project that used a laser scanner to build a structure model of a museum. The data points they made are all walls and floor as shown in Fig. 2.3. With this filtered data, they built the complete structure of the whole museum to make a 3D map to guide the visitors. Their work highlighted the very important character of an indoor scene: the structure.



(a) (b)

Figure 2.3: (a) 3D point cloud captured from an indoor room; (b) A complete reconstructed room model [30].

Cabral et al. [31] studied an algorithm to build the room structures from RGB images by assuming each pixel belonging to one of the three classes: floor, wall or ceiling. It is a bold assumption, but it catches the important character to show an indoor scene. It equals to add some prior knowledge that mimics the way how people understand the room. We can read and understand the scene even if no reconstruction is done for the indoor objects.

(a) (b)

Figure 2.4: (a) A reconstructed room structure including floor and walls; (b) RGB images are attached to each plane. Although indoor objects are not reconstructed at all, people can still have a lot of information from the scene [31].

Object detection is another topic in indoor scene reconstruction. The object detection is closely related to the feature extraction. From the Haar wavelet [32] to recently developed HOG [33] and SIFT [34], features in color images are extracted to represent the objects [35]. The development of cost-effective depth sensors, such as Microsoft Kinect, have helped researchers a lot [36], [37]. Lai et al. [5] proposed an algorithm that detected the object from color frames and then reconstructed the object with depth-information. Tang et al. [38] provided an algorithm that extracts the object directly from depth map by detecting the discontinuities. Their work was based

on the fact that people can identify the separate objects by the different space occupation. For example, people can tell if a book is on the table, but they cannot tell how many books are piled together.

So recent research has proved that, algorithms could do better after learning some treatments of the scene from people [24] , [31], [33], [38]. Computer's understanding of the scene is more important to reconstruct an indoor scene than making a complex algorithm.

2.4 The Manhattan Assumption

The Manhattan Assumption came from an idea in the city plan, which is called the grid plan or the grid street plan. As Fig. 2.5 shows, the streets are perpendicular to each other. So they make grids from the top view.

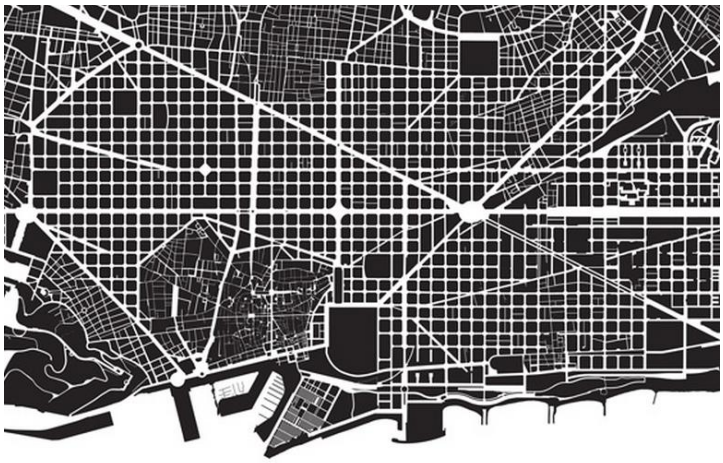


Figure 2.5: The grid planned city map of Barcelona [39].

This idea is later introduced into the area of computer vision, and gets applied in three-dimensional space. Most indoor scenes follow the Manhattan grid: lines are all parallel to one of the 3 axis (major coordinates). So the angles they make are all right angles.

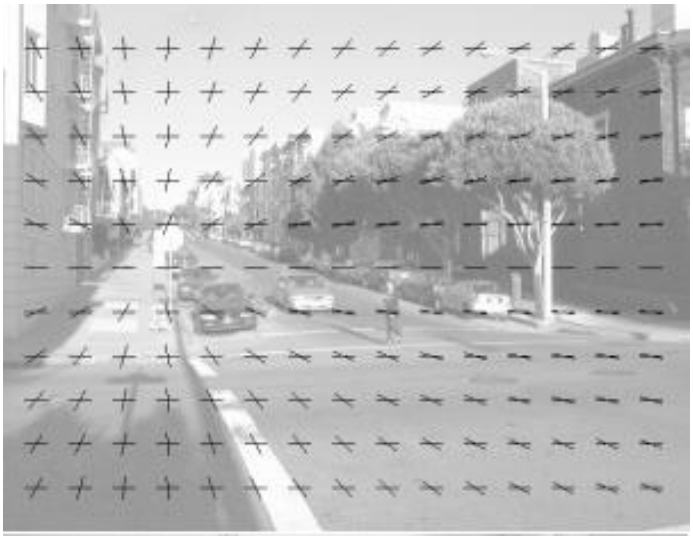
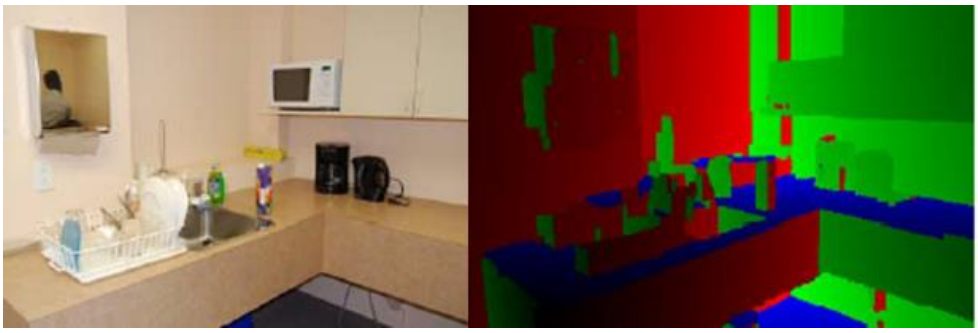


Figure 2.6: The viewer orientation estimation [40].



(a) (b)

Figure 2.7: Indoor scene segmentation using the Manhattan assumption: (a) An input image; (b) The segmented/labeled image [41].

Traditionally, there are two applications that based on the Manhattan assumption. One is to estimate the viewer orientation, as shown in Fig. 2.6. Based on the lines detected in a city view, the relative angle of viewer and the street grid could be estimate. The other application is to segment the indoor scene, as shown in Fig. 2.7. Each pixel in an indoor scene image was assigned with one of the major coordinate in the 3D scene. However, this arbitrary segmentation is not good for 3D reconstruction because it treats each object as a piecewise plane.

CHAPTER III

DEPTH-BASED INDOOR STRUCTURE RECONSTRUCTION

In this chapter, we will talk about the indoor structure reconstruction using depth maps based on the Manhattan assumption [41]. This is a basic approach of using the Manhattan assumption into the area of indoor scene reconstruction. We will find the structure points in point cloud and build an indoor structure from it [30]. The scene is assumed to be a general one that follows the Manhattan assumption without too many trivial parts. In this chapter, there are six sections: (1). The Manhattan assumption in an indoor scene; (2). Feature extraction from depth images; (3). Clustering algorithms; (4). Major coordinate extraction; (5). Structure-object Separation; (6). Experimental Results.

3.1 The Manhattan Assumption in an Indoor Scene

The Manhattan Assumption came from an idea in the city plan, and got applied in computer vision to deal with indoor scenes. From Fig. 3.1 we can see, the indoor structures, such as floor and walls, are big planes that follow the Manhattan assumption. So from top view, indoor scenes are made of different shapes of rectangular, as the floor plan shows in Fig. 3.2.

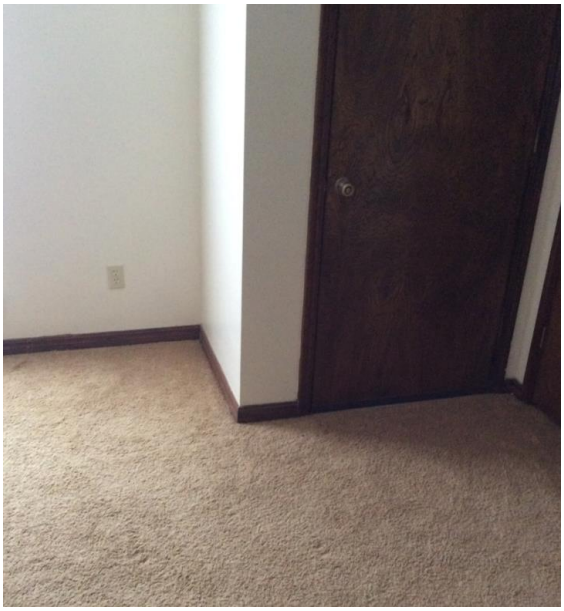


Figure 3.1: Example of indoor structures: floor and walls.

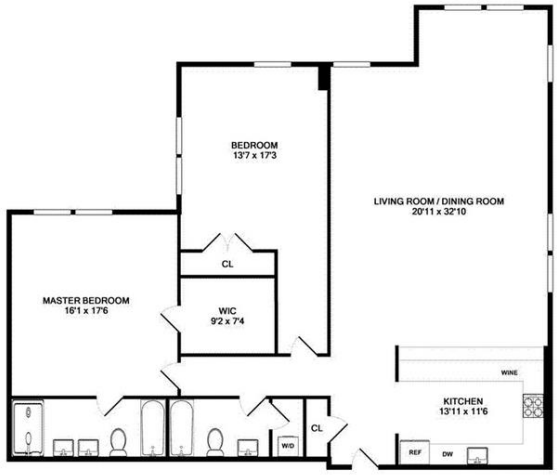


Figure 3.2: Floor plan sample: building's major structures follows the Manhattan assumption [42].

Not only the building structures (walls and floors), but also the indoor objects are mostly designed to follow the Manhattan assumption [41]. People like to decorate their rooms based on

the directions of walls and doors. Thus the indoor objects are also parallel to one of the major coordinates, like Fig. 3.3 shows.

Figure 3.3: A indoor scene sample where furniture and objects generally follow the Manhattan assumption [43].

From the figure above we can see, most indoor objects follow the 3 major coordinates. For example, the sofa is straightly facing the wall. The television is parallel to the wall. Although objects have different kinds of shapes, people tend to put them in a way to follow the Manhattan assumption of the room.

We will apply this assumption to the indoor scene reconstruction with a depth sensor: Kinect [20]. Our work here is not to reconstruct each tiny detail of objects, but to regularize the overall indoor scene with the Manhattan assumption, assign each object its own room to do its own reconstruction. So we will tell walls and floors apart from objects. This gives a better understanding of the indoor scene to the computer.

3.2 Feature Extraction from Depth Images

Based on the Manhattan assumption, each indoor scene has the major coordinates that indicate the orientation of most points. The task of this section is to extract the orientation features from input depth map. The feature we need is the norm of each data point. For each point the norm is calculated by minimizing the deviation of the point and its neighbors. However, the norms represented in Cartesian coordinates are on the surface of the norm sphere. The clustering methods will not work on ball surfaces. So we present the method of using the azimuth-zenith angle map to stand for the norm to do the clustering.

3.2.1 Depth Map Denoising

The depth map provided by depth sensors is very noisy. We use two traditional methods that are widely used [1]: (1). We select the data points that are within the valid range of the sensor; (2). We remove the inconsistent points by checking its 8 nearest neighbors. The sensor we use is Kinect V2. It has a valid range from 0.5m to 4m.

(a) (b)

Figure 3.4: (a) An RBG image of an office; (b) The corresponding depth map from Kinect.

As shown in Fig.3.4, the depth image is very noisy, such as the edge of the filing cabinet shown in the figure. We use the 0.5-4m valid range as shown in Fig. 3.5.

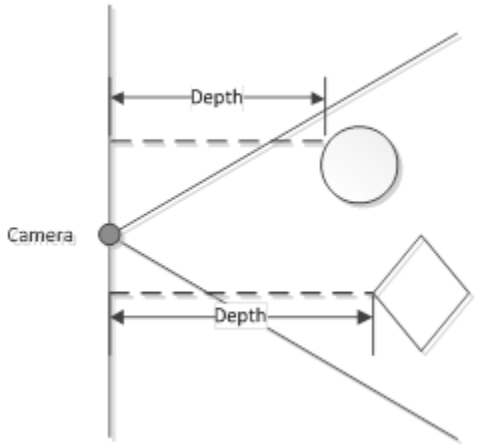


Figure 3.5: The valid range of the Kinect sensor [20].

To remove the points that are outside of Kinect's range, we first project the points in depth map into 3D space using the method provided in [44]. The output point cloud of the algorithm has a unit length that equals to 1mm. And then, we check the distance of each point in 3D to the original point (sensor). If the distance is within the valid range (0.5- 4m), the point will be kept. Otherwise, the point will be deleted.

3.2.2 Definition of the Feature: Azimuth-zenith Map

The general way to represent a norm is using a three-dimensional vector with unit amplitude. Thus the distribution of norms will be on the surface of a ball with the center at original point and radius equals to 1. But the Euclidean distance doesn't work for the clustering algorithms on a ball surface. This fact makes it hard to find the typical norm based on the norms' distribution. We use

the azimuth angle and zenith angle to represent a norm, thus on the azimuth-zenith map, the clustering methods can be applied.

The azimuth angle and zenith angle come from the idea of transforming the Cartesian coordinate system into the spherical coordinate system. The spherical coordinate system is a system that describes the 3D space. It uses 3 parameters to represent a point in space: an azimuth angle (measures from X axis), a polar angle (measures from Z axis which pointing to the zenith), and a distance from the point to the original point.

A point in the spherical coordinate is described as (r, θ, φ). They are the radial distance, azimuth angle, and zenith angle respectively, as Fig. 3.6 shows. Then, any direction can be represented using the azimuth angle together with the zenith angle (θ, φ) after setting the radius r = 1. We let the azimuth angle and zenith angle to be the two axis to make the azimuth-zenith map, as Fig.3.7 shows.

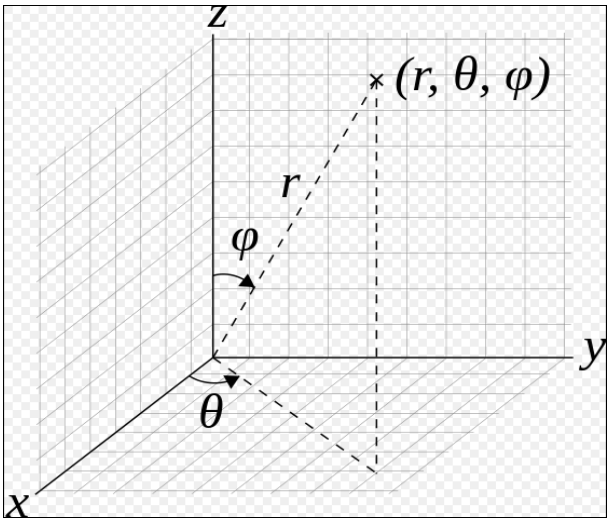
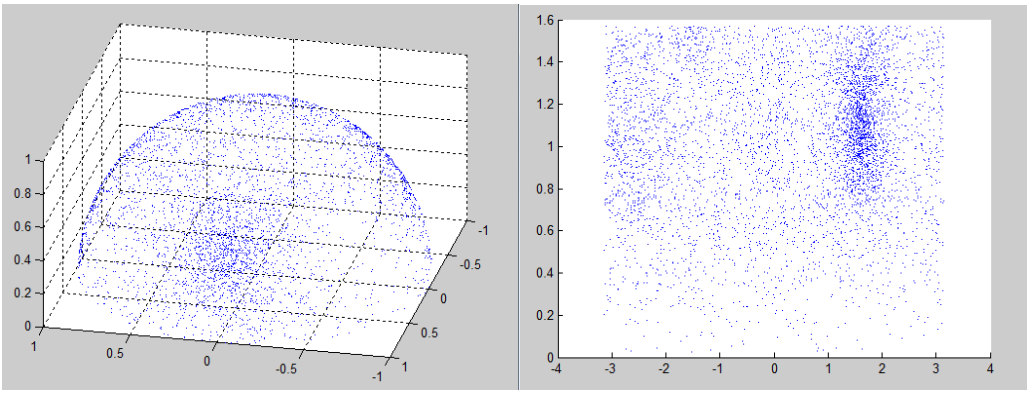


Figure 3.6: The spherical coordinate is shown in a Cartesian coordinate system.

We flip the norm pointing – z to + z, to remove the repetition. Thus the norms are distributed in a hemisphere, as Fig. 3.7 (a) shows. The norm features are azimuth angle θ and zenith angle φ pairs, as Fig. 3.7 (b) shows.



(a) (b)

Figure 3.7: The same norm set in two coordinate systems: (a) The Cartesian coordinate; (b) The azimuth-zenith coordinate.

For each norm described as (x,y,z) in Cartesian coordinates, the functions to transform it to spherical coordinates using azimuth angle θ and zenith angle φ is:

$$\begin{cases} \theta = \tan^{-1}\left(\frac{y}{x}\right), & (x > 0) \\ \theta = \tan^{-1}\left(\frac{y}{x}\right) + \pi, (x < 0; y > 0) \\ \theta = \tan^{-1}\left(\frac{y}{x}\right) - \pi, (x < 0; y < 0) \end{cases} \quad (3.1)$$

$$\varphi = \cos^{-1}(z) \quad (3.2)$$

3.2.3 Feature Extraction from Depth Map.

To get the norm for each data point, we apply the N-nearest neighbor method to the depth map. To get the general norm of each plane, we need to get rid of the influence of local texture. As Fig. 3.8 shows, the wall and carpet sometimes have the texture that makes them no longer perfectly flat, although people always treat them as flat planes in the indoor scene.

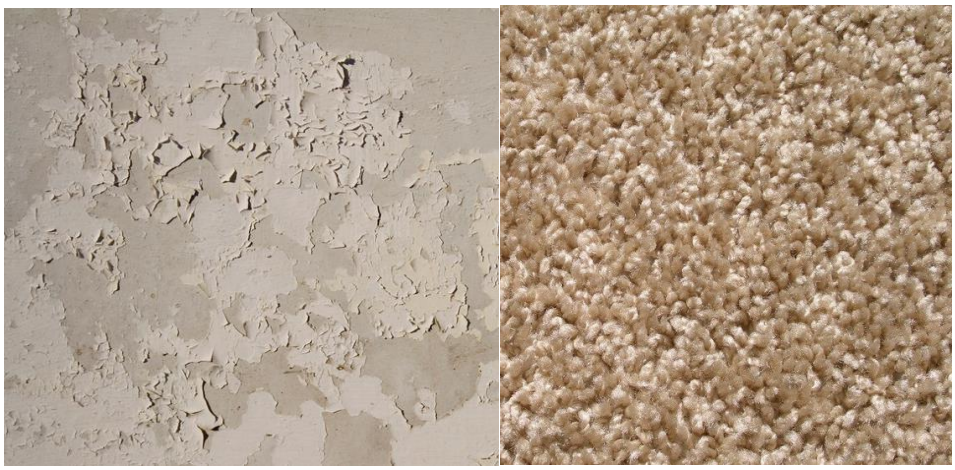


Figure 3.8: Wall and carpet texture examples.

The N-nearest neighbors take the local distribution into account to smooth this effect to some degree, but this method only works when the distances among points are almost the same. But the depth map can be taken from any angle; the points' density for each plane varies.

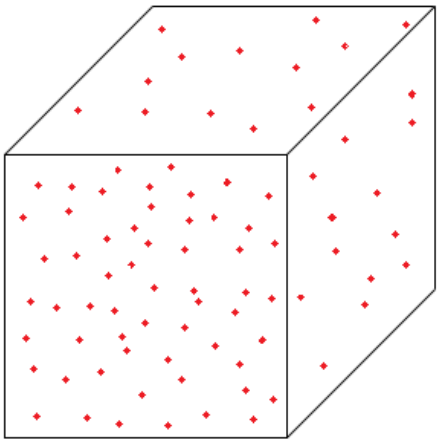
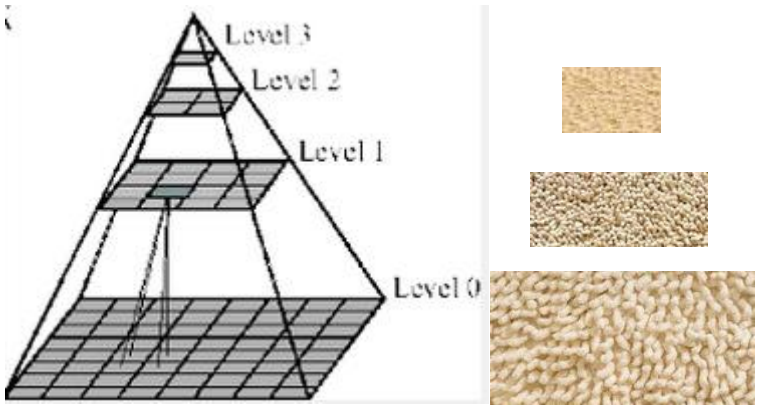


Figure 3.9: Points are dense in the front plane, but sparse on the side.

The widely used Gaussian filter can remove noises with Gaussian distribution [45]. But in the cases shown in Fig. 3.8, the fluctuation may not be Gaussian. Moreover, the filters would blurry the edges in the image. To overcome this problem we use the method to down-sample the depth image first, and then load the 3D location of the points in the down-sampled image from original depth map. The relationship of the depth map and the 3D position of each pixel is based on the original resolution of the depth map, thus the loading is necessary and important. This process works similarly as people always zoom in to see the details and zoom out to see the general shape.



(a) (b)

Figure 3.10: (a) The idea of down-sampling; (b) The example of down-sampling the carpet texture: local texture is reduced after down-sampling.

3.3 Clustering Algorithms

Indoor scenes always have three major coordinates based on the Manhattan assumption. But in reality, the norms of an indoor scene are usually not pointing to the major coordinates directly. They distribute around the major coordinates to make clusters. Our task is to extract the major coordinates from those clusters with help of clustering algorithms.

There are two kinds of clustering algorithms used in our method: 1. Based on knowing that there are three major coordinates on the indoor scene, we use parametric methods including the EM algorithm [46] and the K-means algorithm [47]; 2. Based on not knowing how many big planes are there in the scene, we use a non-parametric method: the mean shift algorithm [48].

3.3.1 Parametric Methods

The Expectation-Maximization (EM) algorithm is widely used. It is a method to find the best parameters by maximizing a likelihood or a posteriori of the model with hidden variables.

Given a set of observations *X*, the missing data *C* (classes), the objective function for EM clustering method is defined as:

$$L(\theta; X) = \sum_C p(X, C|\theta) \qquad (3.3)$$

Where θ is the parameters of each class in *C*.

The process of EM algorithm has two steps that will be processed in each iteration: E-step and M-step. E-step means expectation step, it makes the expectation of likelihood (usually uses log-likelihood for computation) based on the parameters' current value. M-step means maximization, it returns the updated parameters by maximizing the likelihood function provided by the E-step. Then, in the next generation, the E-step will be processed by using the updated parameters. Each step uses arbitrary values of data, assuming they are given, then uses them to get a better estimation. Thus with the two steps updating better results alternatively, the resulting values will converge to fixed points.

EM algorithm is very sensitive to the initialization. It may get stuck at local maxima. So we use a quick algorithm- K-means algorithm to do the initialization for EM clustering.

K-means clustering is to group the data into k clusters so that for each cluster. Its elements are concentrated around the center, while others are separated apart.

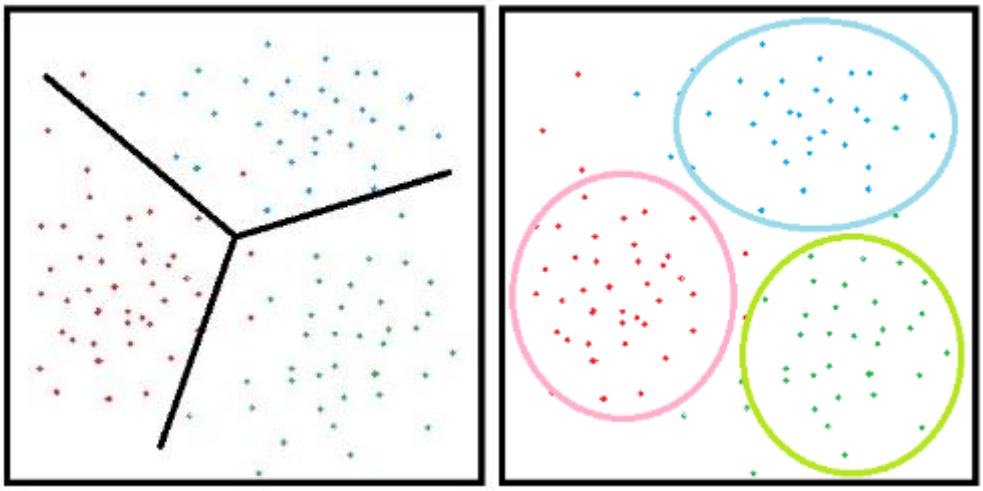
The objective function is defined as:

$$\underset{C}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \qquad (3.4)$$

Where μ_i is the center point for the class C_i

For each iteration, each cluster will generate its own center to represent it. To reclassify the data points to those new centers of all the data. Then new clusters for the next generation is formed. Repeat the process until it converges. During this process, the Euclidean distance is used.

However, the K-mean classification is a clustering with hard limits and the clusters are separated by lines, which is not a good method for data. The EM clustering is a better approach by adding Gaussian distribution assumption to data and can better deal with ambiguous data points.



(a) (b)

Figure 3.11: Comparison of two clustering algorithms. (a) K-means and (b) EM clustering.

A comparison of the K-means clustering and the EM clustering is shown in Fig. 3.11. The K-means algorithm is a linear segment method. It is quick but rough. The EM algorithm estimates each point-cluster classification probability. The EM algorithm can provide a better result by

considering the missing data but it is very slow. So we use the K-means method to make a quick initialization for EM clustering.

3.3.2 Non-parametric Methods

The Non-parametric method we use is the mean shift method. The mean shift algorithm is a mode-seeking algorithm. It assumes that the given data distribution follows an underlying probability density function. It is used to locate the maxima of this density function. In our case, we don't know how many big planes are there in the given indoor scene. This algorithm can find the number of the planes and the positions of them.

The mean shift algorithm defines a kernel to give weights to neighbor points to compute the local mean. Apply this kernel to a data point to get the mean of it, and then shift the kernel to the mean for next iteration. So the local search will move to a denser area after each iteration. Repeat the process for all data points until it converges to several local peaks.

In our algorithm, we use the mean shift algorithm with Gaussian kernel. Given n points, the centroid of next generation y_i^{t+1} is calculated with each point x_j. t is the iteration number and h is the standard deviation. The process is as the formula described below:

$$y_i^{t+1} = \frac{\sum_{j=1}^{n} x_j e^{\frac{-\left|y_i^t - x_j\right|^2}{h^2}}}{\sum_{j=1}^{n} e^{\frac{-\left|y_i^t - x_j\right|^2}{h^2}}} \qquad (3.5)$$

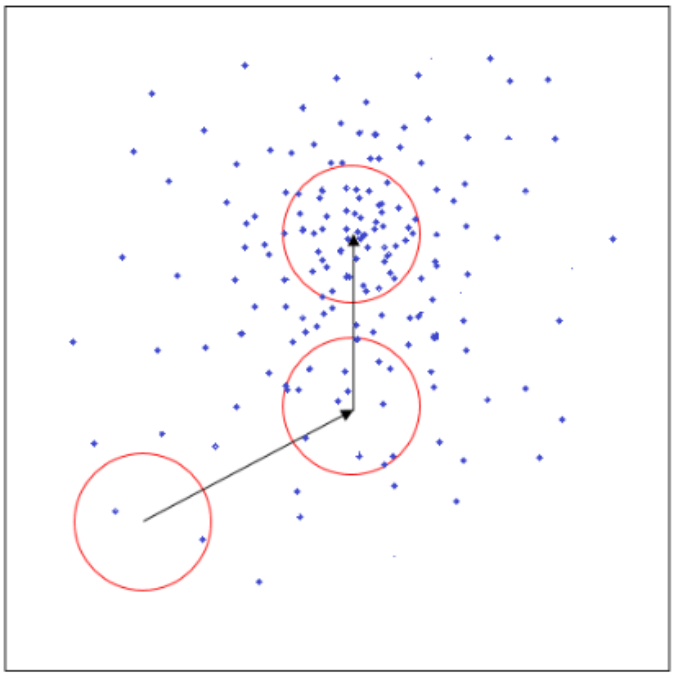


Figure 3.12: The illustration of the mean-shift algorithm: the red circle is the window to do the local average, each circle means one iteration. The red line shows the movement of the circle's center. The window circle started from a low density area and moves to the high density area, thus the local peak is found.

3.4 Major Coordinate Validation

This section talks about the method we use to validate the coordinate extract from the features. The major coordinate we extract from the norms distribution is a general representative. We test it by matching it with the local norms extracted from some patches of the down-sampled depth map. Each patch starts from a random point, then expands by adding nearby points that have the same norm as the starting point. The expansion ends when the patch size reaches a preset threshold, 10 points for example. If the differences between the major coordinates and local norms are close, it means the norms we extract are representing the planes in the scene. Otherwise

it means the result is biased to some degree, a bigger step to do the down-sampling is required to get a more simplified scene by removing the trivial parts. The flow chart is shown in Fig. 3.13.

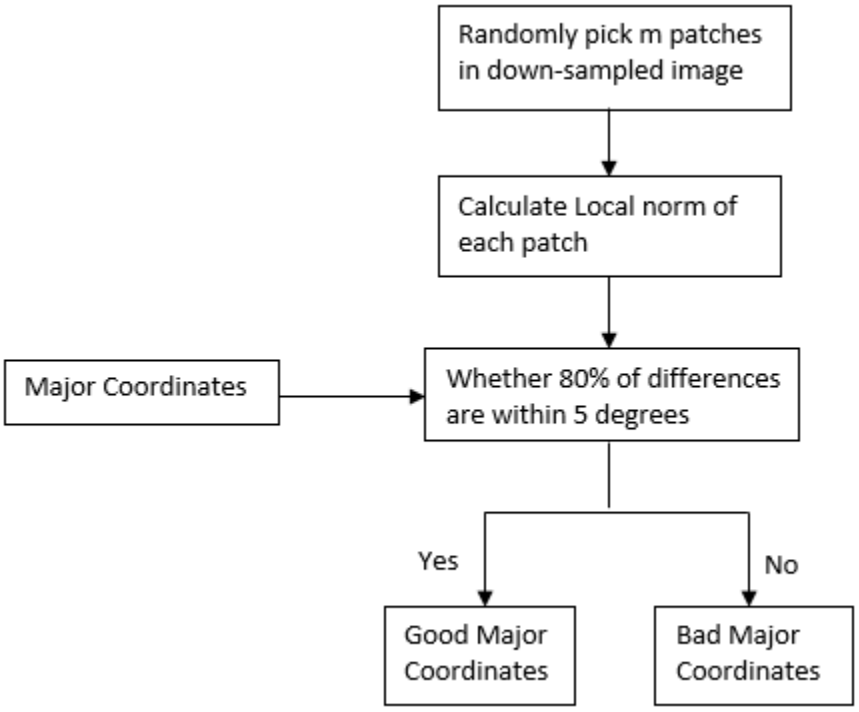


Figure 3.13: The flowchart of major coordinate validation.

3.5 Major Coordinate Extraction Algorithm

In summary, the major coordinate extraction starts with the original depth map. After denoising (see Section 3.2.1), the input depth map is down-sampled. As illustrated in the section 3.2.3, we believe the down sampled points can better show the general shape. We extract features (section 3.2) from the point cloud of the down-sampled depth map and use the EM algorithm (section 3.3) to get the three major coordinates' candidate. This candidate will be verified by checking if it agrees with some random patches in the down-sampled depth map (section 3.4). Otherwise we will use a bigger step to down-sample the input depth map because the previous down-sampled

effect is not enough to show the general shape. The flow chart of our algorithm is shown in Fig. 3.14.

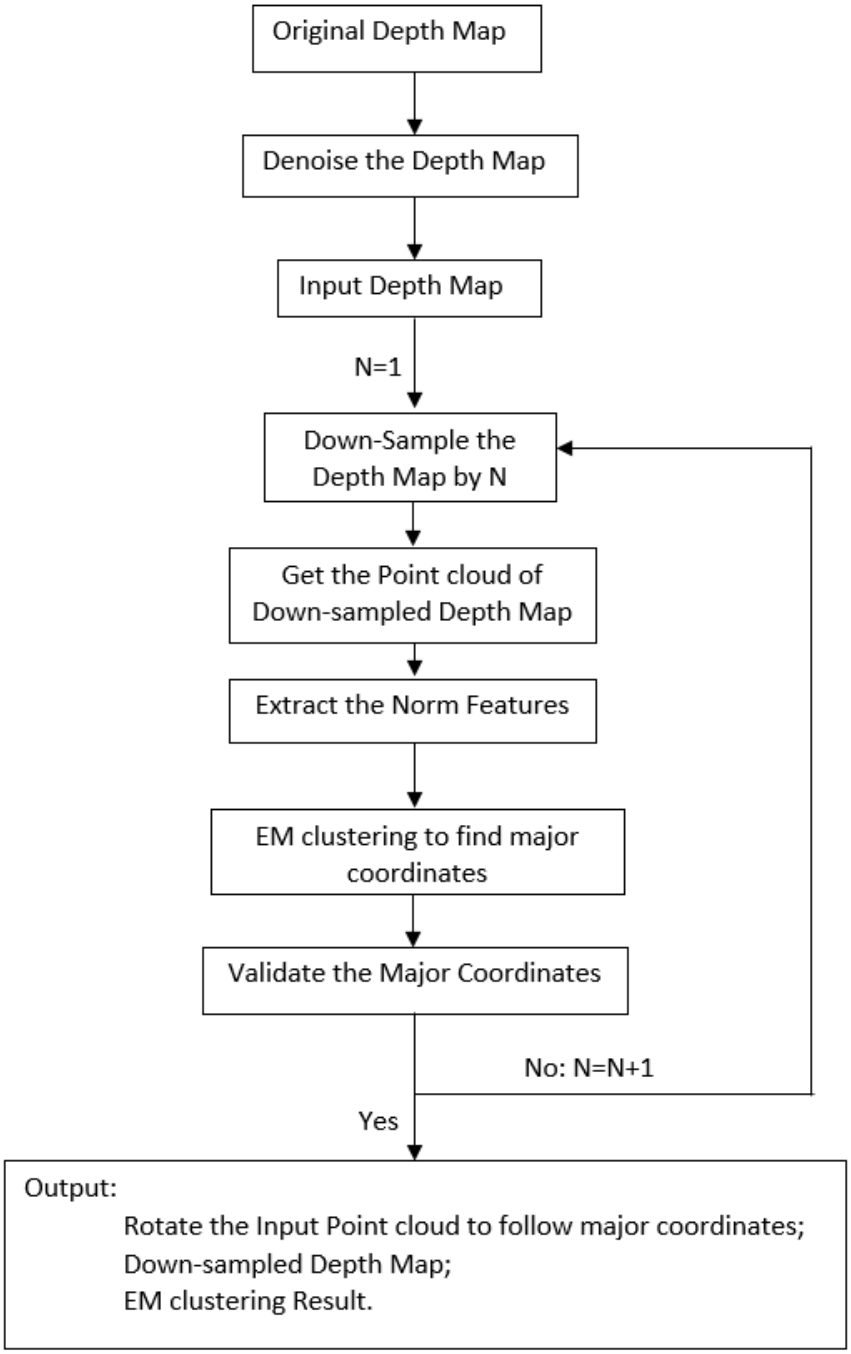


Figure 3.14: The flowchart of the proposed depth-based major coordinate extraction algorithm.

We use the EM algorithm to do the clustering of features which is initialized by the K-means. Normally, the K-means algorithm is sensitive to the initial center assumptions in the first generation. But in our case, the three major norms in the indoor scene are perpendicular to each other and thus well separated on the azimuth-polar map. This means the algorithm will not get stuck at the local maxima. Moreover, the depth map for this algorithm is not a random one, it shows the big planes in an indoor scene. Thus, in our experiments, the K-means method is not easily stuck on the local minima. Rerun the algorithm several times to find the stable solution.

After finding the major coordinate, the point cloud will be rotated to follow them. Thus the big planes, such as walls, floors and table surface, will be perpendicular to one of the 3 major coordinates.

3.6 Structure-Object Separation

This step is to reconstruct the general structures of a room, while the indoor objects are kept unharmed. For example, there's a table standing on a carpet. The table will be represented with the original point cloud. But the carpet will be reconstructed with a perfect flat plane, because the carpet is the major structure by playing the role as the floor. On the other hand the table will be kept with the original point cloud because we do not want to lose any details of the indoor objects. Thus, other methods can be applied to do the reconstruction or recognition of the table. The separation of indoor objects and how to build representative cubes for each object will be explained in the next chapter.

To find the basic structure of the indoor scene, the mean shift algorithm will be applied to each of the 3 major coordinates. Then the potential planes for each coordinate will be found. In the Manhattan assumption, all the planes belongs to one of the major coordinates. So the floor and

walls we need are among those potential planes. The flow chart to do this job is showed in Fig. 3.15.

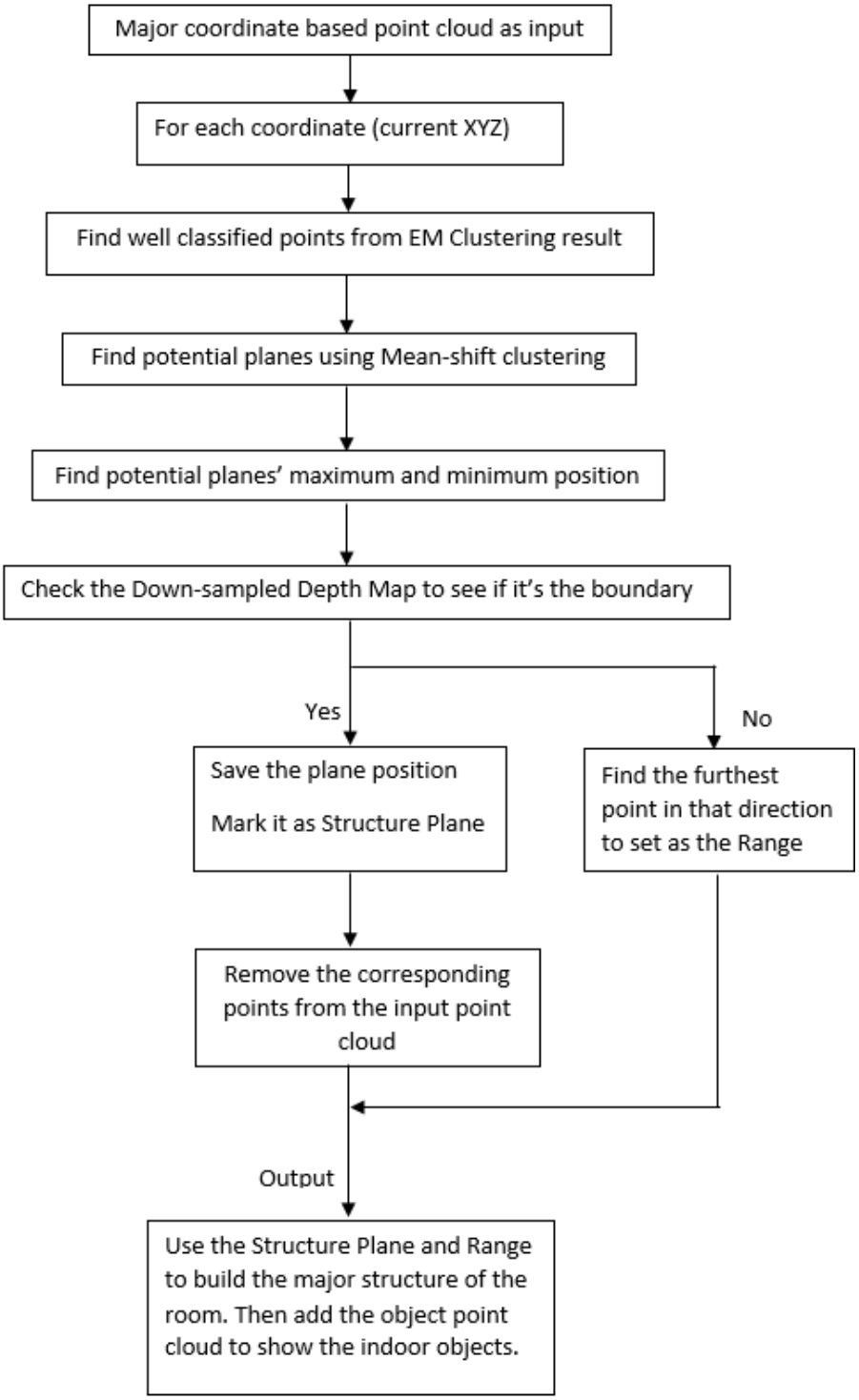


Figure 3.15: The flowchart of the proposed structure-object separation algorithm.

3.7 Experimental Results

We use an artificial kitchen scene to do the experiment to simplify the problem and get rid of trivial influences. The kitchen is built for the project “Building a smart home for the elders”. In the artificial kitchen, the objects include a washing machine, a stove, a sink and a refrigerator. These objects are made from paper boxes. Some typical features for each object are added but no texture, as Fig.3.16 shows. Some big plastic sponges are added behind the objects as the walls. This is an ideal kitchen scene that shows the general kitchen appearance. We will use a single depth map from the Kinect sensor to test our algorithms. Our goal is to find the major coordinates of the scene and then separate the structure (wall and floor) from the indoor objects.

Figure 3.16: An artificial kitchen.

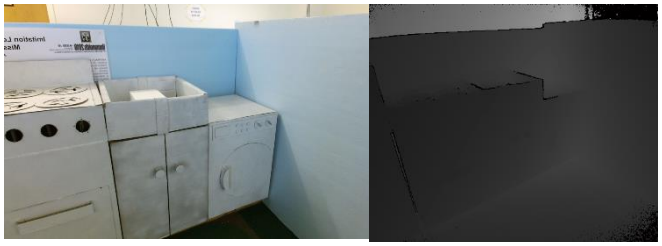


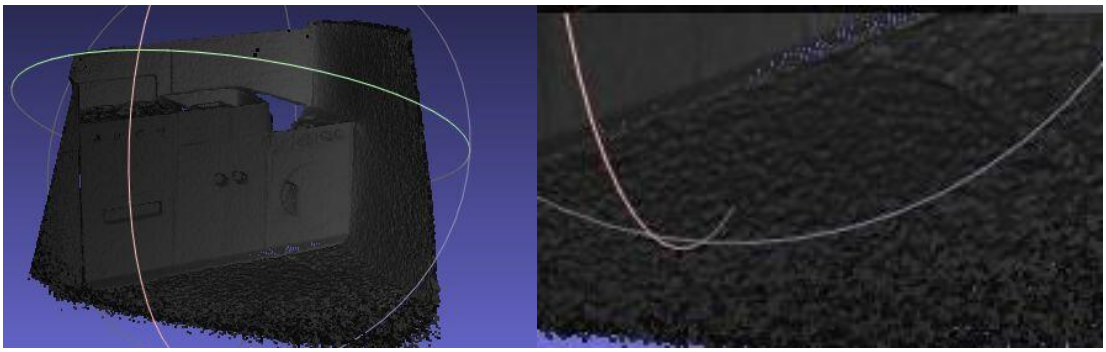
Figure 3.17: The input RGB frame (left) and the corresponding depth map (right).

The input depth map is shown in Fig.3.17. A corresponding reconstruction result using Kinect Fusion [21] is provided in Fig.3.18. From the reconstruction result we can see, even using a good algorithm, the planes that are directly built from depth map are not flat. The connection of planes is not the right angle. This inaccuracy comes from the Kinect sensor. Our algorithm will be applied to overcome this inaccuracy.

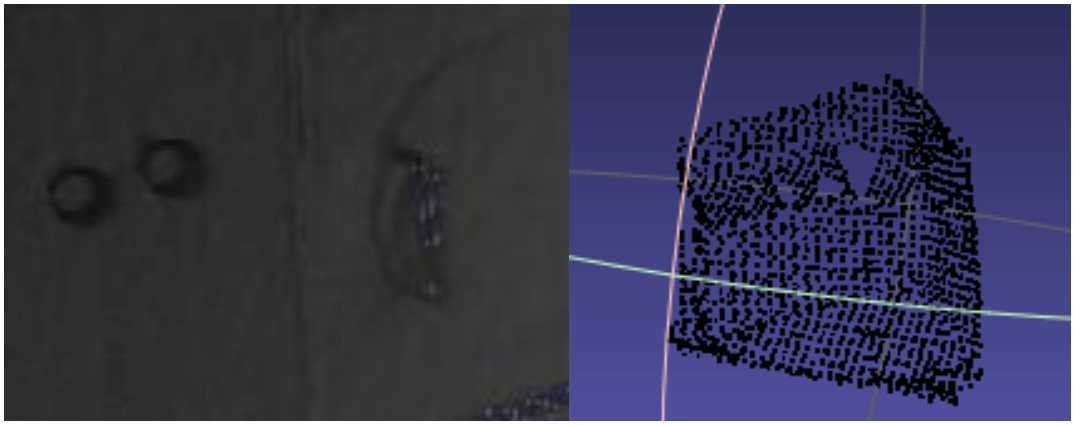
(a) (b)

Figure 3.18: The 3D reconstruction result from Kinect Fusion [21]: (a) Front view of the scene; (b) Top view of the scene. It is shown that the sharp corners become a rounded shape due to the imperfect point cloud data.

In our algorithm, the first step is to down-sample the input depth map and extract the norm features. Then we use a down-sampling method to find the major coordinate of the scene and rotate the point cloud to follow it. The down-sampling process can remove the influence of local details, as shown in Fig. 3.19 (b) and (c).



(a) (b)



(c) (d)

Figure 3.19: Major coordinate extraction: (a) Original point cloud; (b) The floor of original point cloud is not flat; (c) Local detail of the scene; (d) A partial point cloud after down-sampling.

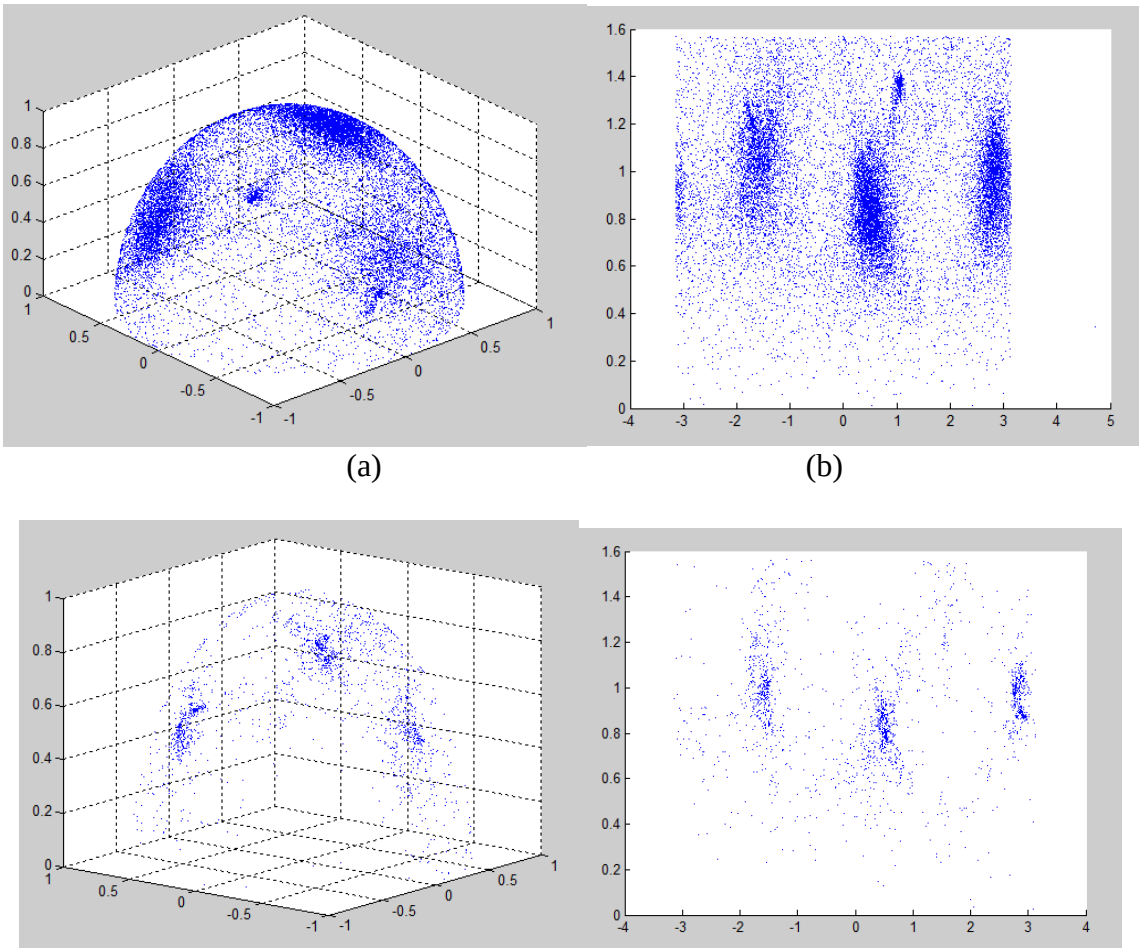


Figure 3.20: Illustration of down-sampling: (a) Normal distribution of the original point cloud in the Cartesian coordinate system; (b) The corresponding azimuth-zenith map of (a); (c) Recalculated norm distribution after down-sampling by factor 3; (d) The corresponding Azimuth-zenith map of (c).

Fig. 3.19 and 3.18 show that the down-sampling method can remove the influence of trivial details (Fig. 3.19 (b) and (c)) of the scene. Those details can make the norm distribution less concentrated (Fig. 3.20 (b)). After down-sampling, the distribution of norm gets clean. Thus the three major coordinates can be found.

The second step is to identify the wall and floor, and separate them apart. We replace the structure points with ideal planes. The indoor objects are kept in their original position. The results are provided to make comparisons in Fig.3.21.

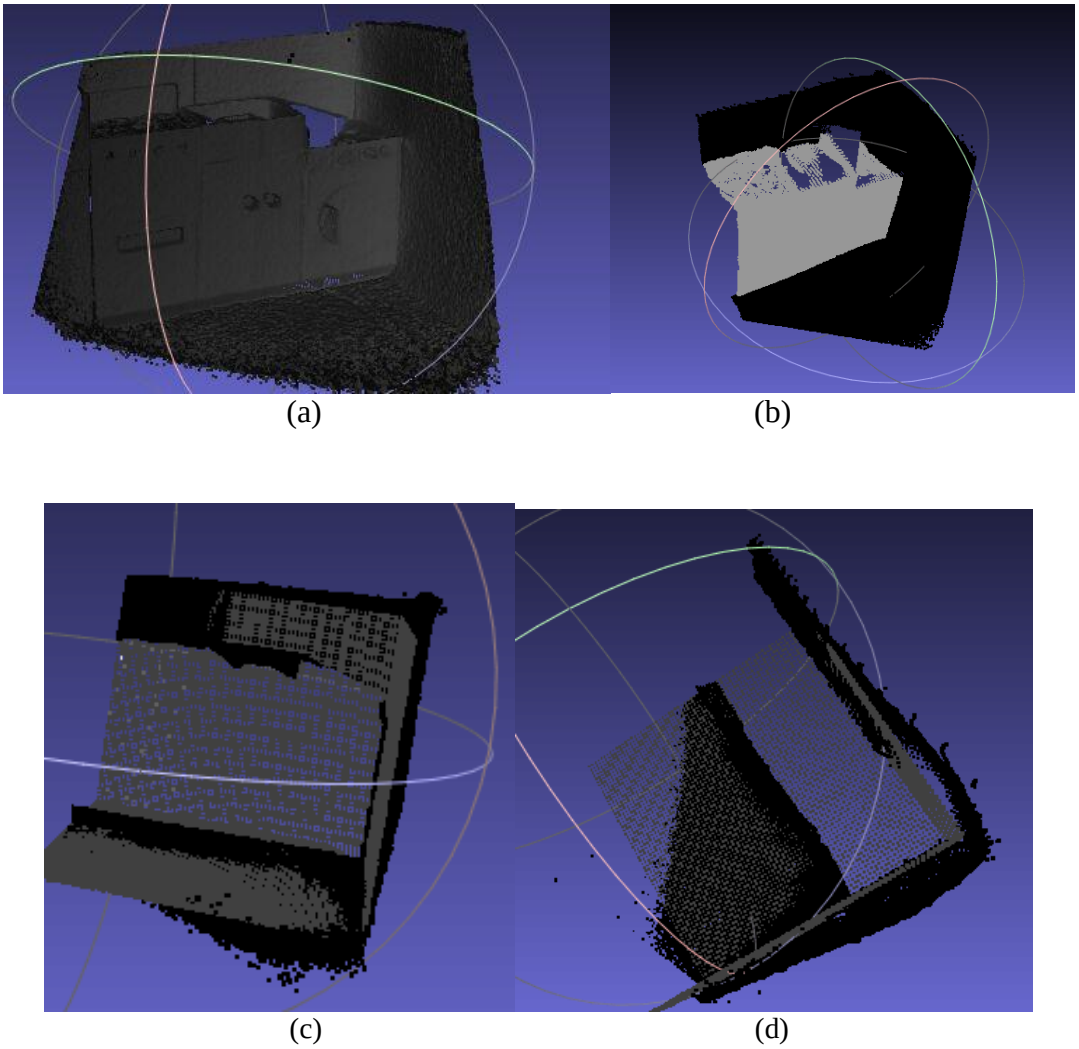
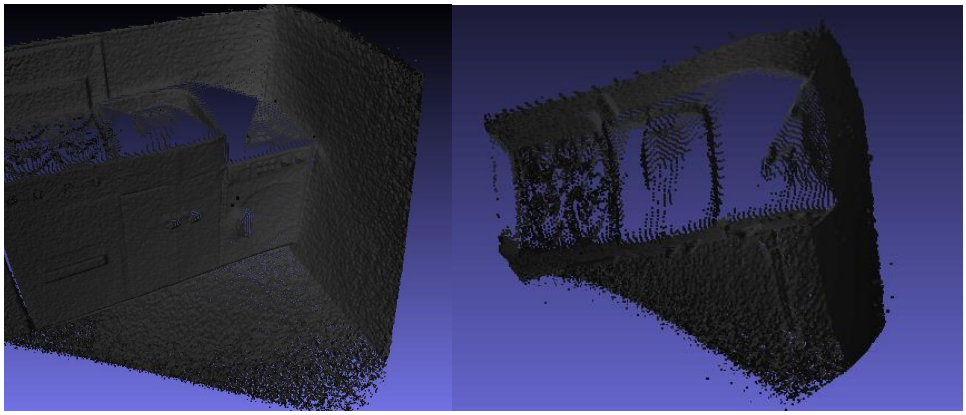
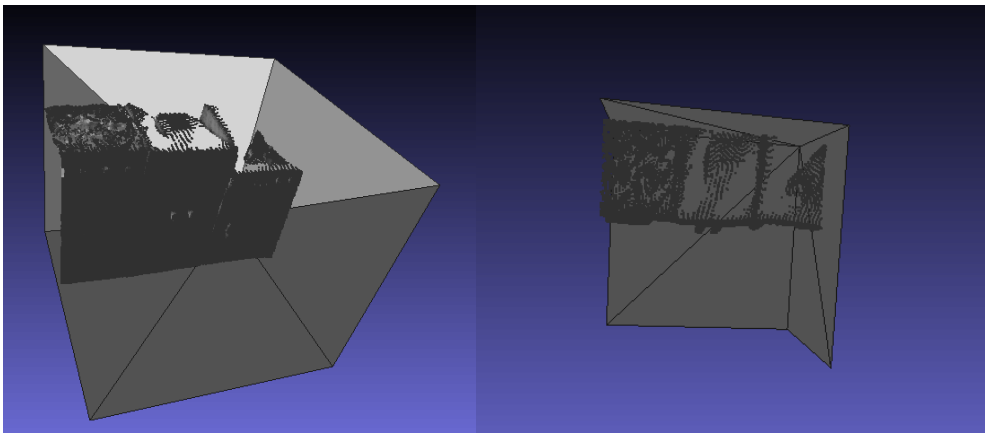


Figure 3.21: (a) The original point cloud; (b) Structure points are shown in black and indoor objects are shown in white; (c) and (d) original structure points are shown in black points; reconstructed ideal planes are shown in grey.

In Fig. 3.21, it shows that the original point cloud is separated into *structure points* and *indoor object points*. The *structure points* are reconstructed using ideal planes as Fig. 3.22 (c) and (d) show. The non-perpendicular corner problem (Fig.3.19 (b) and Fig. 3.21 (d) black) is solved (Fig. 3.21 (d) grey). The reconstruction result is shown in Fig. 3.22.



(a) (b)



(c) (d)

Figure 3.22: (a) and (b) are the input point cloud; (c) and (d) structure points are reconstructed f ideal planes.

Our method can make the depth map more meaningful by reconstructing the hidden structures to form the whole scene into a complete environment. The rounded corners are fixed based on the prior knowledge that we know the walls are perpendicular to each other in indoor scenes. So every structure in the scene is rectangular shaped.

Our algorithm works well in this experiment. But a single depth map has a limited capacity, we need to extend the method to a point cloud for a bigger scene. Moreover, the artificial scene is more ideal than real scenes. We will test and upgrade our method in the next chapter.

CHAPTER IV

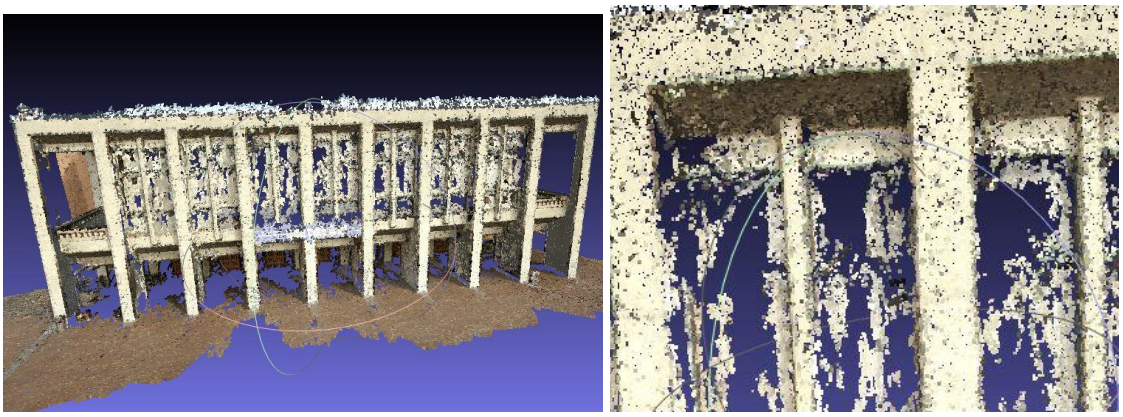
POINT CLOUD-BASED INDOOR SCENE REPRESENTATION

The method provided in chapter three is a basic method. In this chapter, we will apply the method to more complicated indoor scenes, which are point clouds that generated from multiple frames. These scenes are large-scale indoor scenes with many objects that do not follow the Manhattan assumption. In this chapter, the basic method we provide in chapter three will be refined to fit these complicated circumstances.

4.1 Point Cloud Density Control

4.1.1 Point Cloud Introduction

A point cloud is a set of points describing a scene in a same 3D coordinate. The point cloud usually represents the surface of objects. Point clouds can be generated by 3D sensors such as Kinect, or created by many algorithms with multi-views.



(a) (b)

Figure 4.1: (a) A real point cloud example that covers a large scene. (b) A zoomed-in portion which reveals non-uniform and non-continuous point distribution [15].

Point clouds are made of discrete points. When it's dense enough, it is good for visualization, but it is not good to turn into 3D applications directly. Some processes are needed, such as polygon mesh or triangle mesh models by combining nearby points.

The point cloud obtained by Kinect comes from matching different frames of depth maps. Many work in this area has been done to improve the accuracy. The point clouds we used are with basic alignment but without further refinement [5].

4.1.2 Density Control Using Cell Grid

The point cloud comes from matching multiple-frames. An object with more frames has more points. This means the points in each space volume are not the same. Although the object is better represented where the point cloud is very dense, it biased the general distribution of the norm. The total scene in general shall have each part equally weighted because everywhere has the same

importance. So it is necessary to have the density control when doing an indoor scene reconstruction.

The method we used is the cell grid, which means to segment the whole point cloud into many cells with equal size. For each cell, only one representative point can be kept [15]. The position of that point is determined by averaging all the original points within the cell, as Fig. 4.2 shows.

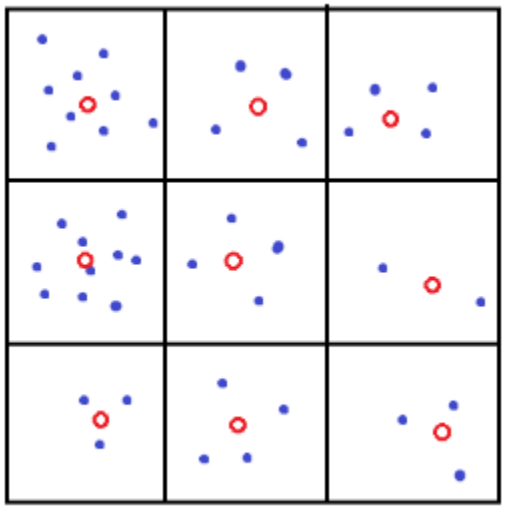


Figure 4.2: The illustration of density control. The point data are first segmented into cells and then each cell is represented one point inside.

4.2 Measures of Large-scale Indoor Scene

4.2.1 Identification of Walls

In an indoor scene, walls are the boundaries that make the shape of room and limit the accessible space of the room. In the previous chapter, the room is simplified to be a box. But in reality it is

not. There are many designs that add walls in a room to decorate, or separate some rooms out from a big room, as Fig. 4.3 shows.

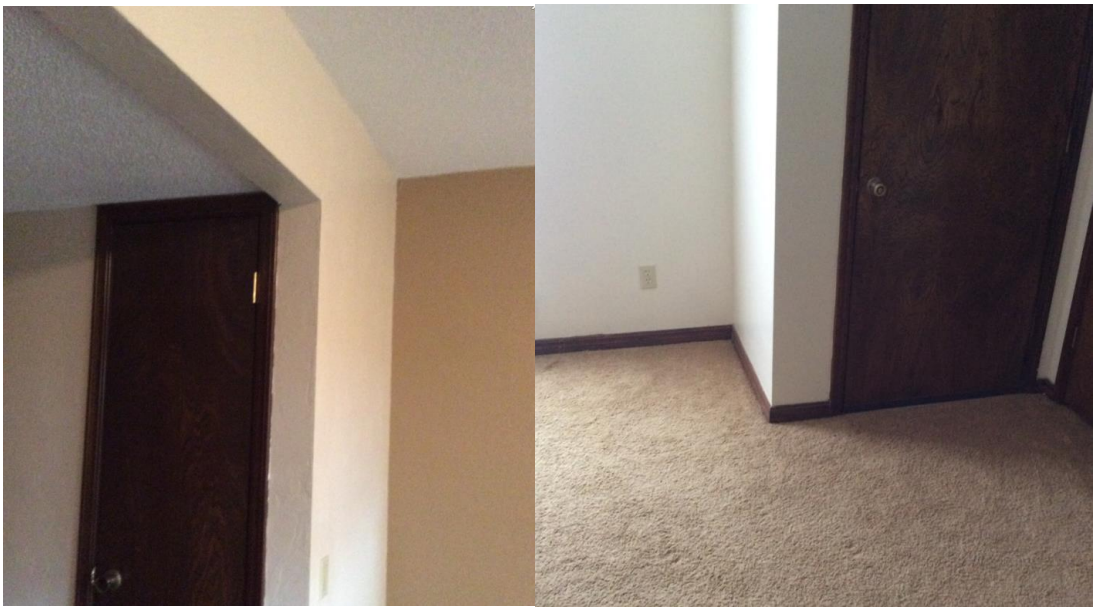


Figure 4.3: Rooms are not perfect cubes.

So we will treat the plane as a wall if it has a height over a threshold *ht*. This means, if the plane is tall enough, we will treat it as a part of the room's structure.

4.2.2 Vertical Norm Dominance

In a large-scale indoor scene, there will be some planes not following major coordinates, which density control cannot handle. The horizontal norms that come from walls may not in the dominant position. But the floor is always the most stable and biggest plane. We will find the vertical norm (comes from floor) in the first step and try to find the other planes later. One main character of the indoor scene is that the floor always has the biggest size, so it is very easy to get

identified. Because of the gravity, the floor is always the foundation of other objects. So it is very important and convenient to reconstruct the scene with help of the floor.

As mentioned in chapter three, the indoor objects must stand on the planes. Here we made an assumption: every big object is standing on the floor. This assumption is actually reasonable because when looking at an indoor scene, people usually focus on the big stuff and treat what on them are just accessories. For example: people will treat the sofa and the cushion on it as a whole. This assumption is already applied in room interior decoration. The widely used floor plan is based on it.

4.3 Large-scale Scene Major Coordinate Extraction

We made a feed-back system to determine the step size of the cell in the density control. The cell size works similarly to the down-sampling method described in chapter three. After extracting the norm features, we found the vertical norm first. The validation process is the same as described in Section 3.4. We removed the well-classified points of the vertical coordinate based on the EM clustering. The next step is to find the horizontal norm pair that describes the walls. We used the mean shift algorithm to find all potential norms, because there might be big planes that did not follow the major coordinates. After checking the perpendicular restrains, the horizontal norm pairs were made. The pair with the most points would be the horizontal major coordinates because most indoor objects followed the major coordinates. The flow chart is shown in Fig. 4.4.

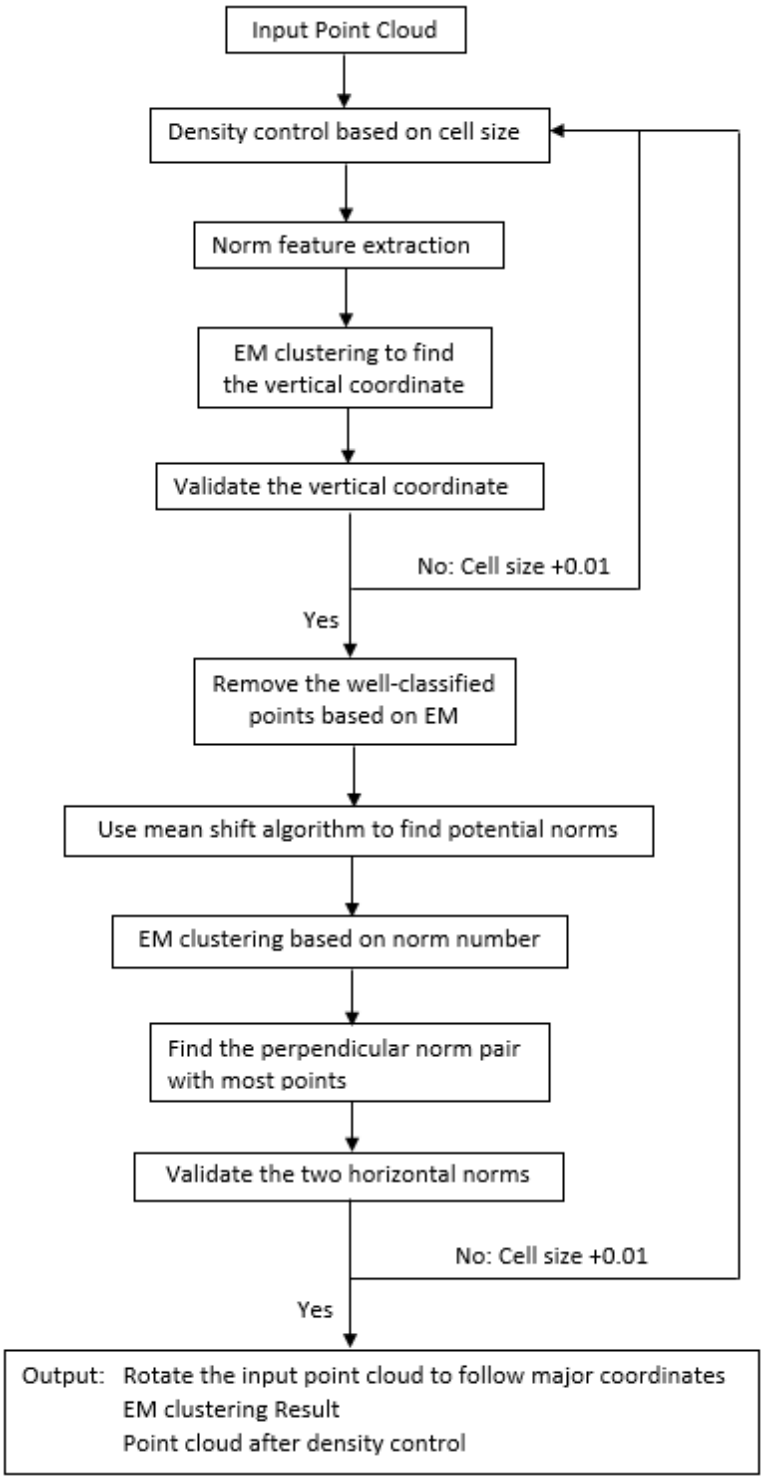


Figure 4.4: The flowchart of the large-scale major coordinate extraction.

4.4 Experimental Results

The point cloud we used for our experiment is based on the research of K. Lai et al. [5]. The dataset is created by them. It was originally used for object recognition. The point cloud is obtained by aligning multiple video frames. The scenes they provide are indoor scenes that contain big furniture (sofa, coffee table, chair), and many small items (bowls, soda cans, caps). A sample point cloud is shown in Fig.4.5. Our task is to identify the room structure and reconstruct it using ideal planes while keeping the indoor objects as their original shape.

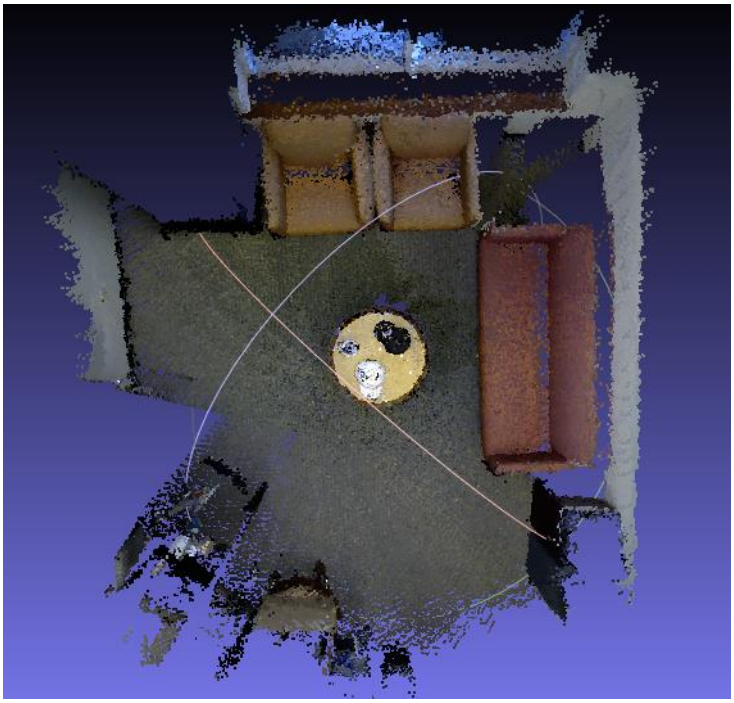
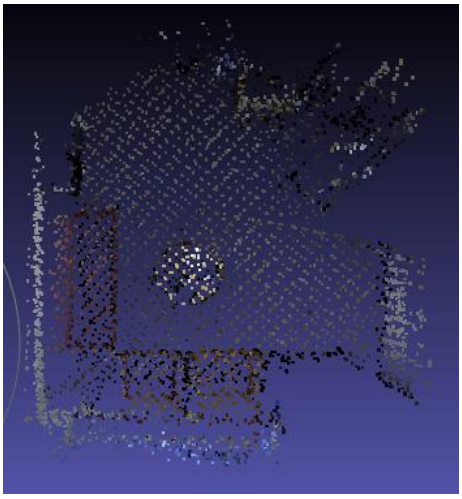
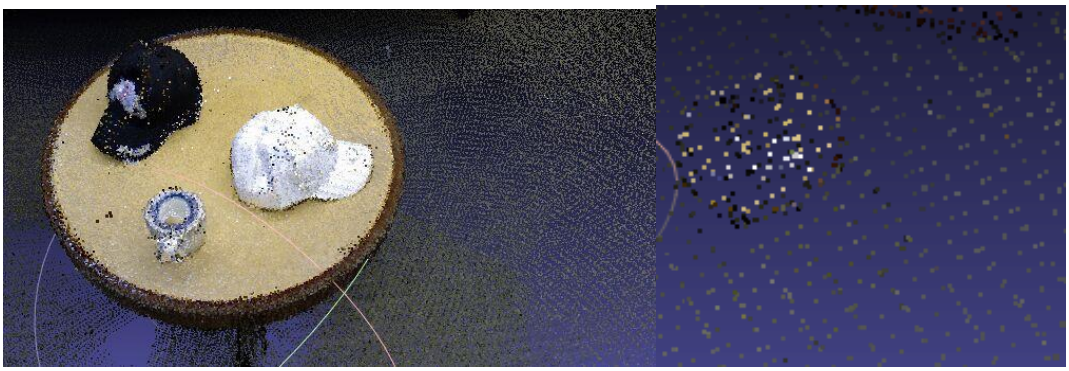


Figure 4.5: A real point cloud of an indoor room [5].

The method of density control can help to make each part of the scene equally weighted, as shown in Fig. 4.6. The point cloud was made by going around the coffee table, thus the points that indicate the coffee table are denser than other parts. After density control, the coffee table has the same point density as other regions.



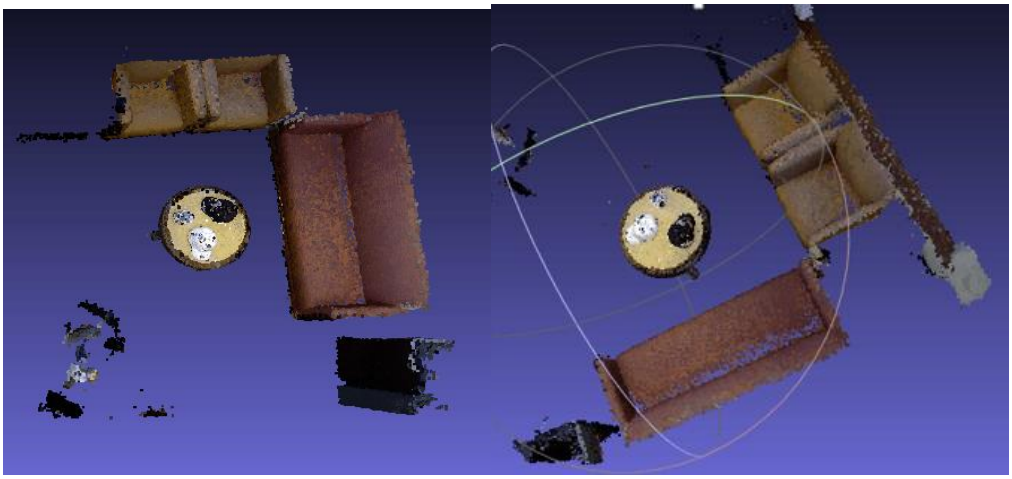
(a)



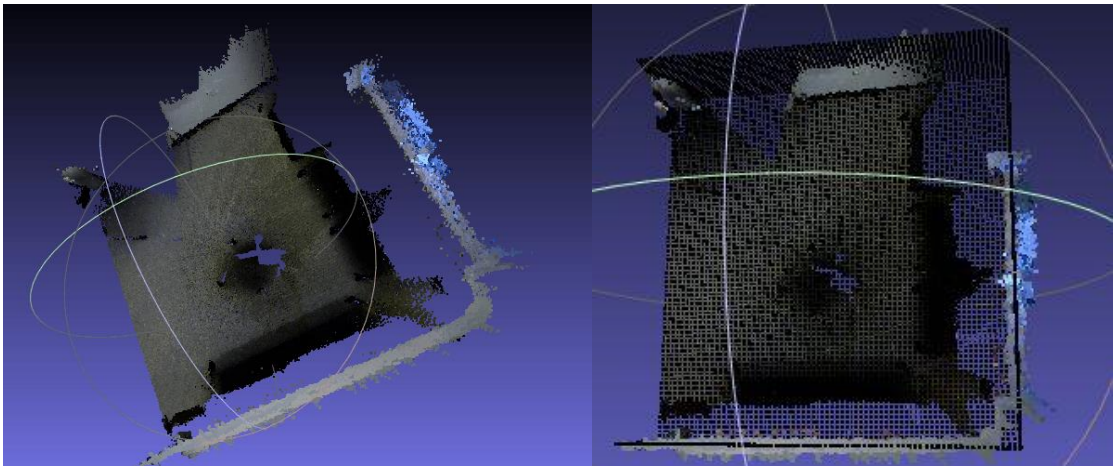
(b) (c)

Figure 4.6: (a) The down-sampled point cloud; (b) The point cloud of the coffee table; (c) The coffee table after density control.

When doing the *structure-object* separation, we found the vertical nom and the floor points first. And then the horizontal coordinates were extracted and the point cloud would be rotated to follow the major coordinates.



(a) (b)

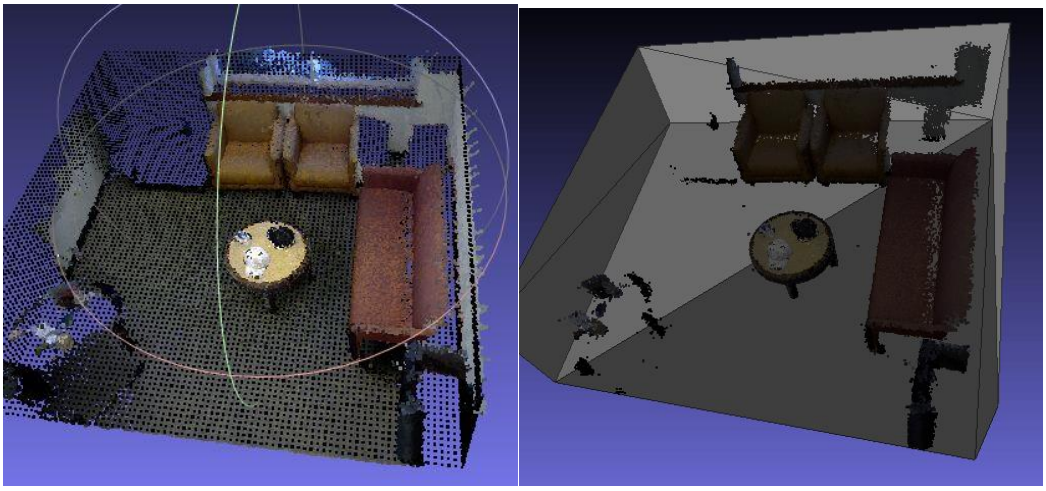


(c) (d)

Figure 4.7: The indoor objects are shown in (a) and (b). The points of room structures are shown in (c). The generated room structure (black) is shown along with the original point cloud in (d).

In Fig. 4.7 (c), it is shown that the *structure* in original point cloud has many holes. The reconstructed *structures* are shown in a point cloud in Fig. 4.7 (d) to render inputs and reconstructions simultaneously. It shows that our algorithm can infer the unseen parts of the

structure and give the room a complete shape. The final reconstructions use ideal planes as shown in Fig. 4.8.



(a) (b)

Figure 4.8: Reconstruction results: (a) Input point cloud and reconstructed *structures*; (b) *Structure* points are substituted with ideal planes.

The experiment also shows some problems of our method. When the wall is represented in a single layer of points, the structure may get disconnected to the objects attached to it (Fig. 4.8 (a) red sofa). In reality, each wall shall have a thickness. Another problem is from the density control. It equalizes each part of the scene, but also the unconfident regions. For example, in Fig. 4.8 (b), the right side wall has a lot of outliers, but the process of density control keeps them (Fig. 4.6 (a) left side). Thus, the position of the wall which is found by the mean shift algorithm, is biased to a wrong direction. These two problems work together to make the algorithm unable to handle sills (Fig. 4.7 (d) right side). Sills are always with a thickness, and the windows may contain many bad points because of reflection. So the reconstructed structure plane is dragged to

the outside of the room. That explains the wall behind yellow chairs classified as *indoor objects*: because the algorithm considered there was a wall behind it.

To solve these problems, a better density control method is needed by checking the point number and consistency in each cube. A prior knowledge in room reconstruction also shall be added to deal with doors and windows.

CHAPTER V

OBJECT DETECTION AND REPRESENTATION

In the previous two chapters, we focused on the detection and reconstruction of indoor structures. The room is reconstructed using ideal planes. The process we provide can remove the structure points from the original input and leave the indoor objects. In this chapter, we will focus on the reconstruction of indoor objects by inferring the unseen planes using voxel-based methods.

5.1 Voxel Representation

A *voxel* a unit cube in three-dimensional space on a regular grid [49]. Similar to the idea of a *pixel* in an image, a *voxel* is the smallest change in volume. The size of *voxel* determines the accuracy of a reconstruction. Big voxels can give a general shape of the reconstructed object; small voxels can provide more details but require more computation effort.

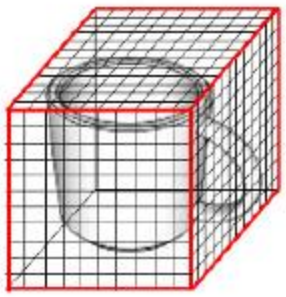


Figure 5.1: Sample voxel representation of a cup [5].

We use voxel representation to do the reconstruction of indoor objects because we want to infer the volume of objects from the discrete point cloud.

5.2 Inferring Hidden Plane of an Object

Room-Object separation and cube construction: based on the hypothesis plane and the original point cloud, we will build the room structures first, form walls and floor. Then each object in the room will be presented as a cube (6-faces). After checking each hypothesis plane's rough position together with the original point cloud, the best fit of plane-plane matching is found. During this process, more planes will be built, those are the hidden planes that we could not see but exist.

This kind of labeling method is carrying the feature that the wall and floor are working as boundaries of an indoor scene. Although weird shaped rooms exist, most rooms' walls are all following the major coordinates. Traditionally, the wall and floor selection is based on feature classification. But in the 3D indoor scene point cloud, the walls and floor are easier to find.

After identifying the walls and floor as described in Chapter three and four, the rest are the indoor objects, the point cloud will be checked based on the hypothesis planes from the mean-shift

algorithm. For each hypothesis plane, the corresponding nearby perpendicular hypothesis plane will be checked to see if they belong to the same object cube.

Clearly, not all faces of an object cube can be seen. There are three parameters that shall be determined for each cube: its length, width and height, which also means the amplitude along 3 axis. Moreover, the indoor objects also follow the law of gravity: the object will not floating in the air. So every object will be standing on a plane. Another assumption we made is if it is hard to tell the distance of the object to the wall behind it, the object will be next to the wall. Thus, any two clear plane or one plane near a wall will be enough to create a cube to represent the object.

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

Figure 5.2: Cube assignment: (a) One face available, the cube that represents the object will be attached to the wall (b); (c) Two faces available, it's enough to create a cube (d).

Given the histogram of density $D_i^j(x_i)$ along major coordinate i for the points in potential plane j, the target function to determine the plane edge x_i is given below, where g is the step size of histogram and γ is the threshold to find where plane density change sharply.

$$\left|D_i^j(x_i) - D_i^j(x_i + g)\right| > \gamma \qquad (5.1)$$

However, this method can only be applied to cube-shaped objects. For those objects with smooth curves, such as sofas, this method would not work well.

5.3 Voxel-based Object Completion

After having the point cloud in major coordinates as described in Chapter three and four, the big planes in the scene would be perpendicular to one of the major coordinates. With the help of the "norm" information from previous steps, we project the points with same norm class onto its corresponding perpendicular coordinate, thus the histogram of this projection will show where the potential plane is. We use the mean shift algorithm [4] to find the number of the potential plane and its position. Then track back to the point cloud to see the rough position and size of each plane to make the plane a confident one. This process will be applied to each coordinate, and thus the hypothesis planes will be formed for each major coordinate.

Based on the confident plane's position, the cube that represents the object is generated as illustrated in section 5.2. To deal with the problem that the original point cloud and the generated cube may get overlapped, we propose a voxel-based object completion method. As shown in Fig. 5.3, the point cloud of the reconstructed object is segmented into voxels. If a voxel contains both reconstructed points and original points, the reconstructed ones are eliminated to preserve the local structure provided by original data points.

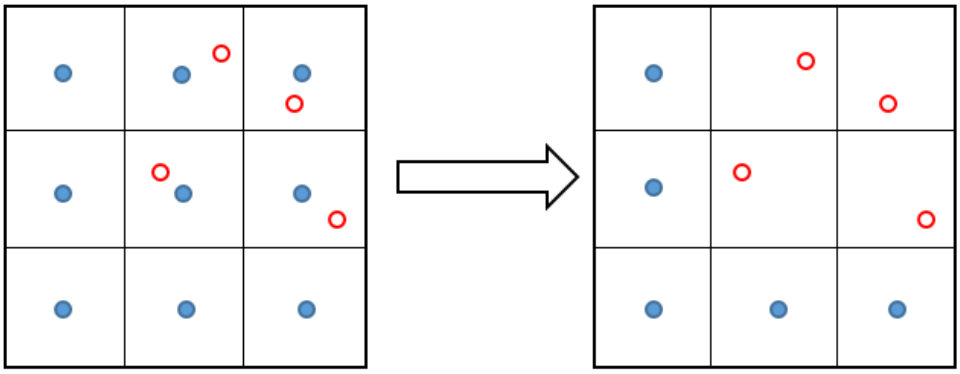
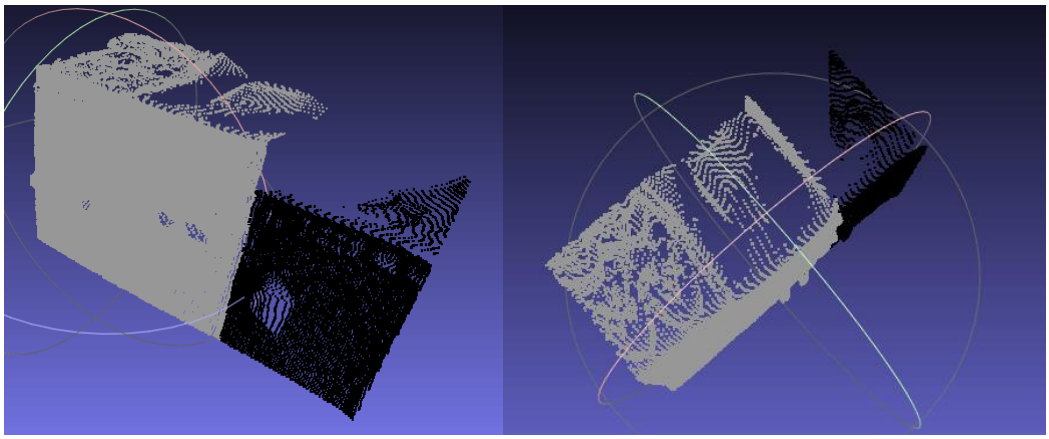


Figure 5.3: Voxel-based object completion.

5.4 Experimental Results

We tested our algorithm by reconstructing the objects from the experiments in chapter three. The identified objects are shown in Fig. 5.4. The objects are not in their complete form. From the top view we can see, the top planes are very sparse.

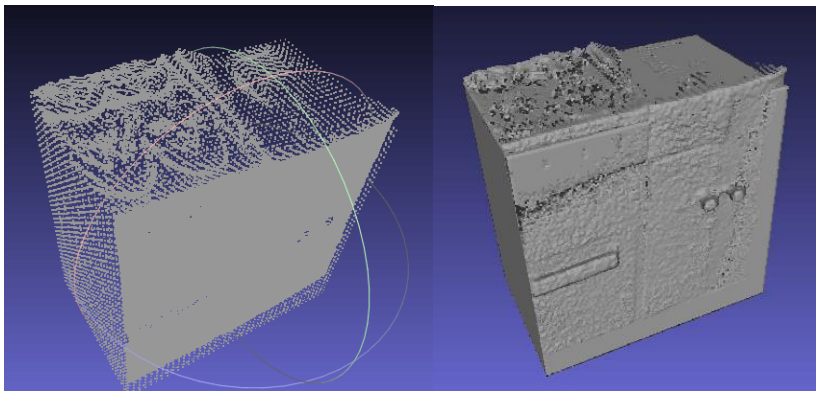


(a) (b)

Figure 5.4: (a) Identified two objects are shown in white and black; (b) top view of (a).

As described in section 5.2, the confident planes are used to separate each object. But the object shown in white points actually has two parts. It cannot separate the two parts (a stove and a sink) for now because in this experiment, we only have one depth map, the data is limited, and the two structures are very alike in shape and attach to each other in space.

For each object, we use the front plane where the points are dense to generate a cube-shaped container. The container describes the general shape of the object. Then, the method described in section 5.3, a complete model of the object is generated. The model has a cubic shape and preserves details provided by the original point cloud, such as the handles under the sink.



(a) (b)

Figure 5.5: (a) A cube-shaped container defines the general shape of the object; (b) Generated 3D model of the object.

The final reconstruction results of the artificial kitchen scene is shown in Fig. 5.6. Indoor structures are reconstructed using ideal planes and indoor objects are with complete shapes. Our algorithm made good results of the kitchen scene. But for the data set used in Chapter four, it has many non-cubic based objects, such as sofas and a round coffee table with a cap and a cup on it.

These objects cannot be approached with cube containers. Moreover, our algorithm is based on confident big planes. These objects have many curves and even without big planes. To solve this problem, some training models may be needed. The objects can be classified to the model they belong to. Then our object completion algorithm can be implemented to generate the unseen parts of the object.

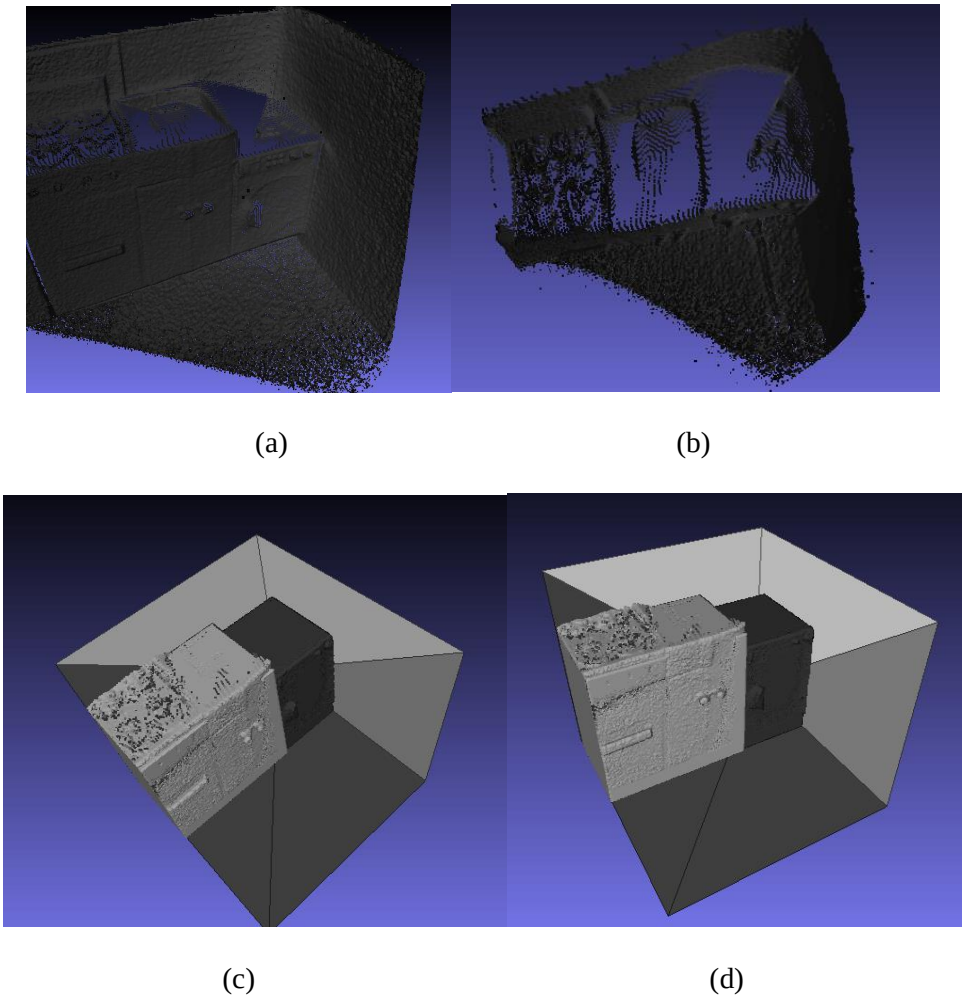


Figure 5.6: (a) and (b) show the original point cloud; (c) and (d) present the final reconstruction results.

CHAPTER VI

CONCLUSION AND FUTURE RESEARCH

6.1 Conclusion

In this study, an indoor scene reconstruction algorithm was developed to reconstruct the indoor structure and indoor objects separately. This method can provide a complete indoor scene reconstruction by getting some understandings of the structure of the room first and identifying the indoor objects in the second place.

We used the Manhattan assumption to regularize the indoor scene. To solve the problem that local texture might not show the general shape, we proposed a feedback system to determine the down-sampling scaler. After down-sampling, the major structure could be kept; the details would be ignored. Thus the major coordinates of the scene were extracted. We found the points that indicated the indoor structure from the input depth map, reconstructed the indoor structure with ideal planes to make a complete cube-shaped indoor structure. Moreover, to deal with the large-scale complex indoor scene, the algorithm was adjusted to extract the vertical norm first from the floor and extract the horizontal norm after it. For the objects in an indoor scene, we proposed a method to infer the unseen planes to make each object to be a complete model rather than several discrete patches. The experiments showed that this method did well on cube-shaped objects, but not on round-shaped objects.

The original objective was to reconstruct an indoor scene to make the computer understand it. As the first step to build a semantic map, our method basically achieves the objective. There are still some spaces to improve, especially on the round-shaped object reconstruction.

6.2 Future Research

The final objective of our future research is to build a semantic map. In this work, the indoor objects are reconstructed in a low resolution; it would be necessary to make the 3D models with higher accuracy. It would be interesting to apply our method in multi-views, to regularize the scene to get a more accurate map before triggering loop closure based on the Manhattan assumption. Moreover, more information of the scene would be considered such as humans' motion and the sound made by each object. This leads to a new hybrid affordance-based and appearance-based object recognition approach for indoor semantic mapping.

REFERENCES

[1] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *The International Journal of Robotics Research,* vol. 34, pp. 598-626, 2015.

[2] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *The International Journal of Robotics Research,* vol. 31, pp. 647-663, 2012.

[3] *prototype robot for assisted living*. Available: http://www.paneuropeannetworks.com/science-technology/prototype-robot-for-assisted-living/

[4] D. A. Forsyth and J. Ponce, "A Modern Approach," *Computer Vision: A Modern Approach,* 2003.

[5] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1817-1824.

[6] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, 2006, pp. 519-528.

[7] M. Levoy, "The stanford spherical gantry," ed, 2002.

[8] J. Heikkila and O. Silvén, "A four-step camera calibration procedure with implicit image correction," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 1106-1112.

[9] J.-Y. Guillemaut and A. Hilton, "Joint multi-layer segmentation and reconstruction for free-viewpoint video applications," *International journal of computer vision,* vol. 93, pp. 73-100, 2011.

[10] O. Faugeras and R. Keriven, *Variational principles, surface evolution, pde's, level set methods and the stereo problem*: IEEE, 2002.

[11] A. Hornung and L. Kobbelt, "Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, pp. 503-510.

[12] J.-P. Pons, R. Keriven, and O. Faugeras, "Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score," *International Journal of Computer Vision,* vol. 72, pp. 179-193, 2007.

[13] A. Zaharescu, E. Boyer, and R. Horaud, "Transformesh: a topology-adaptive mesh-based approach to surface evolution," in *Computer Vision–ACCV 2007*, ed: Springer, 2007, pp. 166-175.
[14] C. H. Esteban and F. Schmitt, "Silhouette and stereo fusion for 3D object modeling," *Computer Vision and Image Understanding,* vol. 96, pp. 367-392, 2004.
[15] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 32, pp. 1362-1376, 2010.
[16] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 27, pp. 418-433, 2005.
[17] M. Habbecke and L. Kobbelt, "Iterative multi-view plane fitting," in *Int. Fall Workshop of Vision, Modeling, and Visualization*, 2006, pp. 73-80.
[18] O. Hall-Holt and S. Rusinkiewicz, "Stripe boundary codes for real-time structured-light range scanning of moving objects," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2001, pp. 359-366.
[19] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, "3D shape scanning with a time-of-flight camera," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 1173-1180.
[20] *Microsoft Kinect*. Available: https://msdn.microsoft.com/en-us/library/hh973078.aspx
[21] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, 2011, pp. 127-136.
[22] M. Zeng, F. Zhao, J. Zheng, and X. Liu, "A memory-efficient kinectfusion using octree," in *Computational Visual Media*, ed: Springer, 2012, pp. 234-241.
[23] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3D reconstruction in dynamic scenes using point-based fusion," in *3D Vision-3DV 2013, 2013 International Conference on,* 2013, pp. 1-8.
[24] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Vision algorithms: theory and practice*, ed: Springer, 2000, pp. 298-372.
[25] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *Robotics, IEEE Transactions on,* vol. 24, pp. 946-957, 2008.
[26] M. Labbe and F. Michaud, "Appearance-based loop closure detection for online large-scale and long-term operation," *Robotics, IEEE Transactions on,* vol. 29, pp. 734-745, 2013.
[27] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on,* 1999, pp. 941-947.
[28] H. Wildenauer and M. Vincze, "Vanishing point detection in complex man-made worlds," in *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, 2007, pp. 615-622.

[29] O. Barinova, V. Lempitsky, E. Tretiak, and P. Kohli, "Geometric image parsing in man-made environments," in *Computer Vision–ECCV 2010*, ed: Springer, 2010, pp. 57-70.
[30] J. Xiao and Y. Furukawa, "Reconstructing the world's museums," *International Journal of Computer Vision*, vol. 110, pp. 243-258, 2014.
[31] R. Cabral and Y. Furukawa, "Piecewise planar and compact floorplan reconstruction from images," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014, pp. 628-635.
[32] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 193-199.
[33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 886-893.
[34] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, 1999, pp. 1150-1157.
[35] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 4, pp. 51-52, 2001.
[36] B. Sabata, F. Arman, and J. K. Aggarwal, "Segmentation of 3D range images using pyramidal data structures," *CVGIP: Image Understanding*, vol. 57, pp. 373-387, 1993.
[37] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 821-826.
[38] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, *et al.*, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *Computer Vision–ACCV 2012*, ed: Springer, 2013, pp. 525-538.
[39] *Urban Resolve* Available: http://urbanresolve.tumblr.com/post/46259926486/salixtance-barcelona-city-grid
[40] J. M. Coughlan and A. L. Yuille, "The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference," in *NIPS*, 2000, pp. 845-851.
[41] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 1422-1429.
[42] *Simple Blank House Floor Plan*. Available: http://galleryhip.com/simple-blank-house-floor-plan.html
[43] *Home designing*. Available: http://www.home-designing.com/2015/06/modern-apartment-designs-by-phase6-design-studio
[44] *Kinect2 Matlab tool*. Available: http://www.codeproject.com/Tips/819613/Kinect-Version-Depth-Frame-to-mat-File-Exporter
[45] P. J. Burt, "Fast filter transform for image processing," *Computer graphics and image processing*, vol. 16, pp. 20-51, 1981.

[46] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1-38, 1977.
[47] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Applied statistics*, pp. 100-108, 1979.
[48] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 603-619, 2002.
[49] J. Ashburner and K. J. Friston, "Voxel-based morphometry—the methods," *Neuroimage*, vol. 11, pp. 805-821, 2000.

VITA

Lin Guo

Candidate for the Degree of

Master of Science

Thesis: INDOOR SCENE RECONSTRUCTION USIND THE MANHATTAN ASSUMPTION

Major Field: Electrical Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2015.

Completed the requirements for the Bachelor of Science in Electrical Engineering at Tianjin University, Tianjin, China in 2012.

Experience:

Visual Computing and Image Processing Lab (VCIPL), OSU Stillwater, OK
Research Assistant, Sep. 2014 – Present.

Professional Memberships:
NA