Indoor Modelling Using RGB-D Data

MINA MEHRANFAR
March, 2013

SUPERVISORS:
Dr. K. Khoshelham
Prof. Dr. Ir. M.G. Vosselman

Indoor Modelling Using RGB-D Data

MINA MEHRANFAR
Enschede, The Netherlands, March, 2013

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: GEOINFORMATICS

SUPERVISORS:
Dr. K. Khoshelham
Prof. Dr. Ir. M.G. Vosselman

THESIS ASSESSMENT BOARD:
Prof. Dr. Ir. A. Stein (Chair)
Dr. R. M. Bennett (External Examiner, University of Twente)

ABSTRACT

Various applications of indoor modelling in surveillance and emergency managements, navigations, positioning, robotics, forensics, virtual tours and so many on, brings increasing attentions to this field. Some of these applications need 3D models, e.g. a virtual tour is not possible in 2D models. Also 3D models interact with the users better and efficient. Meanwhile one of the recent technologies to acquire data for this use case, is using the RGB-D sensors. Kinect as one of these sensors, is the most famous and affordable cameras to capture the coloured point cloud.

This research is proposed to develop a method for indoor 3D modelling using Kinect RGB-D data based on Manhattan-World assumptions. First we recognize the planes of indoor, using the characteristics of normal vectors of segments. Then using the grammar based modelling with a cube as the primitive shape, the 3D primitives are generated on all possible planes of indoor. In this way we first select the farthest segments of point cloud from the centroid of indoor, thus the first generated cube is the possible largest cube. Then we reconstruct new cubes using the remained segments and subtract them from the largest cube. In reconstructing the new cubes, we must preserve the orientation parameters of first cube to obey the Manhattan-World assumptions. The final model is written into VRML file and for better visualization of that, we add textures on the faces of model.

The developed method is experimented on three different shaped datasets (L, U, X) and the results show that the method is valid for all of them. Also the method is implemented on real L-shaped point cloud with large registration errors. Despite of those error, all the existing planes are reconstructed in final model. The measurements on the reconstructed model is compared with the real measures to check the accuracy. On the planes with low registration error, we gain satisfactory accuracy.

The developed method covers all the objectives of research, however there are some issues to improve the results. For example, we assumed to have only one plane of floor and ceiling without steps, that in future works, the reconstructed model is better to consider the steps in these two planes and also extending the method to model the scenes which don't obey the Manhattan-World assumptions with non-perpendicular planes to each other is another follow up of this research.

Key Words: Indoor Modelling, Kinect RGB-D data, Plane recognition, Grammar based modelling

ACKNOWLEDGEMENTS

My sincere gratitude and appreciation is to my supervisors; Dr. Khoshelham and Prof. Vosselman for their supports and encouragements. For sure, without the guidance and suggestions of Dr. Khoshelham, this dissertation would not have been possible. The comments that both of my supervisors gave during the research, helped to enrich the work. It was my honor and great pleasure to work with them.

Special thanks to Dr. D. Dos Santos, to help us in preparing the point cloud and registration processes.

I would express my special thanks to all GFM's lecturers and staff, whom shared their knowledge and insights with us with untiring efforts.

I should thank to Dr. Helali, my supervisor from previous educational stage in Iran, who helped in all challenges related to software used for this research and also for his patience and encouragements during the selection of thesis topic.

Last but not least, my appreciations goes to my parents for their morally supports and encouragements and endless love.

This thesis is dedicated to my beloved family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

2D	Two Dimensional
3D	Three Dimensional
AEC	Architecture, Engineering and Construction
ANN	Approximate Nearest Neighbour
CAD	Computer-Aided Design
CIE-Lab	Lab colour space defined by Commission International de l'Eclairage (L stands for luminance, a is the red-green axis and b is the blue-yellow axis)
fps	Frame Per Second
LiDAR	Light Detection And Ranging
LoD	Level of Detail
MATLAB	MATrix LABoratory (software name)
OGC	Open Geospatial Consortium
OSM	Open Street Map
PCM	Point Cloud Mapper (software name)
R	Rotation (matrix)
RANSAC	RANdom SAmple Consensus
RGB-D	Red, Green and Blue- Depth
ToF	Time of Flight
ρ	Perpendicular Distance

1. INTRODUCTION

1.1. Motivation and problem statement

Nowadays with a high demand for accurate and realistic models of urban environments, attention to modelling the interiors of buildings has also grown. This aim also forced Open Geospatial Consortium (OGC) to define the level of detail 4 (LoD4) of the international format CityGML, to represent the large number of details of architectural models from the interior (Budroni and Böhm, 2009).

There are several important applications which need models of the interior: risk assessments and disaster management, architecture, engineering and construction (AEC), navigation, positioning, robotics, forensics, surveillance and emergency management are just some examples where we need indoor models (Khoshelham and Elberink, 2012; Rosser et al., 2012). For example in emergency planning and evacuation simulations, we have to determine the topological and geometrical relationships of the building's inner structures (Zhang et al., 2012). In other cases, we can use these interior models for virtual tours that have various use cases in real state agencies, architectural designs, etc. As these needs and attentions increase, large companies also try to get involved in interior modelling. Google aims to map the interior of public buildings across the world; meanwhile it launched an application on Google maps for mobile devices in which users also can upload their own floor plans to be included in the Google gallery (Google, 2011). Also there is a research on bringing the indoor mapping to Open Street Map (OSM) (Goetz and Zipf, 2011). So we see the need for an effective technique of automation in interior mapping and modelling in favour of time and cost.

The effective way to visualize and represent a model of interior is in 3D; since the floor plan or 2D models of interiors are hard to interpret for most of people and users prefer to look around in 3D models. For example, a virtual tour cannot be done by 2D floor plans and this requires a 3D model. Also for professionals 3D models are more helpful; e.g. they reveal the inconsistencies better than the floor plans (Kashlev, 2008).

Toward this aim there are some researches and works done for 3D modelling of building interiors, which mainly use 3D measurements in the form of point clouds. In researches, Budroni and Böhm have worked on automation in making 3D interior models from Laser data (Budroni and Böhm, 2009, 2010). Although Laser scanner point clouds have good accuracy for this use, but the data accusation by this technique has high costs in money and time. Meanwhile Laser scanners are not handy and easy to carry inside the buildings. On the other hand, the low cost and flexible range cameras are a serious alternative for laser scanners.

One of the techniques to acquire 3D point clouds, where high accuracy is not purposed (Khoshelham and Elberink, 2012), is using the Kinect sensors. These sensors have the capability to capture both RGB data and depth data integrated, in images of 640×480 pixels, at a frame rate of 30 fps (Hobley, 2010). For capturing the depth data in synchronization with the colour image, the sensors project “a pattern of dots with a near infrared laser over the scene and use a detector that establishes the parallax shift of the dot pattern for each pixel in the detector” (Hobley, 2010). As outcome we have high quality coloured point clouds which need to be registered together to be defined related to the reality. This characteristic of Kinect point clouds provides more flexibility for segmentations and recognitions in favour of only depth measurements or only coloured images. As an example, in colour data we may have multiple objects with same or similar colours, which would cause in misidentifications (Hernández-López et al., 2012). Also in

point clouds with only depth measurements we may confuse multiple objects on one plane, such as door and wall, whereas by using the colour information object recognition can be done.
Despite of all these advantages, as we know, all new methods and techniques come with some limitations and incompleteness. The Kinect sensors are no exceptions, and the lower accuracy and resolution (low density of point clouds) in comparison with other LiDAR sensors, brings some difficulties to widespread usage of the technique. Also one of the challenges in data of interiors is the cluttered scenes, in which we may have so many objects inside the area we are capturing the data and this means more planes than the building's planes may exist in the point cloud.
As well there are few works done on using the Kinect data for modelling; which has to solve the mentioned problems and challenges. The basic step in interior modelling is the recognition of main planes of the building interior. Recognizing the planes inside of the building and separating the walls and floor and ceiling from other planes, will lead to reconstruct a 3D CAD model of interior. Then through adding extra information, like as textures of planes or architectural primitives (doors, window, etc.), improves the model visualization and accuracy.

1.2. Research identification

In this research, the aim was to develop a method for recognizing main planes of building interiors, such as walls, floor and ceiling. The assumption to gain the result was to build a Manhattan-World scene, in which planes are horizontal or vertical and create a grid based structure (Budroni and Böhm, 2010). After recognition of planes, we can reconstruct a model that best fits to the recognized planes. The result is a full 3D CAD model of building interior. Figure 1-1 shows an example of a Manhattan-World scene, which we aim to develop a method to reconstruct a model for that.

(a) (b)

Figure 1-1: Modelling an indoor scene, (a): Raw point doud, (b): The expected model to fit the point doud

To enhance the visualization, we also add coloured textures to the model using the RGB data of Kinect sensor.

1.2.1. Research objectives

The main objective of this M.Sc. research is to develop a method for recognition of interior planes of the buildings and then reconstruct a model of the interior. Reaching these objectives will result in the full 3D model of building interior that we also have to represent and visualize it.

The detailed objectives are the followings:

- Reconstructing a model of building interiors based on Manhattan-World assumptions
  - Developing a method to recognize walls, ceiling, floor
  - Reconstructing a 3D model which considers all details about inner planes
- Representing 3D textured model of building interiors
  - Enhancing the visualization of model by texture mapping

1.2.2. Research questions

To reach the objectives of this research, the main questions we had to answer, are as follows:

- Which constraints help to recognize walls, floor and ceiling points from other points in a cloud in Manhattan-World scene?
- How can the model fit to all wall planes, which have steps to each other in one direction?
- How well is the performance of the developed method through the experiments?

1.3. Thesis Structure

This thesis report is organized into seven chapters.

After brief descriptions of problem statement and research identifications in first chapter, we discuss about the related works in the second chapter. This chapter also gives an overview of theories we used to do the research.

The methodology and framework of implementation have been illustrated in chapter three, which discusses the developed method step by step.

Chapter four gives a brief description about the materials (dataset preparation and software) we used to implement the research,

In chapter five we demonstrate the results we got by implementing the developed method of research on datasets we have. Then we discuss about the quality of results in next chapter.

Drawn conclusion of research comes in chapter seven, which gives an overview to the research and provides sight to the future works of research with giving some recommendations.

2. LITERATURE REVIEW

2.1. RGB-D data

Depth cameras have different market names, like as ranging camera, flash LiDAR, time-of-flight (ToF) camera and RGB-D camera. Although the sensing mechanisms are varied for them, but they all provide images and depth information for each pixel at a certain frame rates (Deyle, 2010). Between all these, the RGB-D cameras get more and more fame in last years as they provide colour images beside the depth measurements. Meanwhile they are very low cost and affordable devices, such as Microsoft's Kinect and ASUS's Xtion Pro Live sensors (ICIAR, 2013). The visual information acquired by these sensors is like regular digital camera's result. They also provide almost equal measurement accuracy with the laser 3D range scanners, in ranges up to a few meters, however they outperform the laser scanners in terms of frame rate (Holz et al., 2012).

From consumer-grade view point, the Microsoft's Kinect is the recent technology which was primarily designed for computer game environments. This device captures both colour and depth data integrated in the frame rate up to 30 fps (Khoshelham and Elberink, 2012). The idea is developed on basis of range camera technology by Israeli developer PrimeSense, which developed a system to interpret specific gestures using infrared projector, a camera and a microchip to track the movements (Microsoft, 2010). Figure 2-1 shows the prime diagram of platform developed by PrimeSense in this way. The Kinect device is a horizontal bar of sensors on top of motorized pivot (ifixit, 2010) and involves a RGB camera, depth sensor and multi-array microphone. A view of this device is in Figure 2-2.

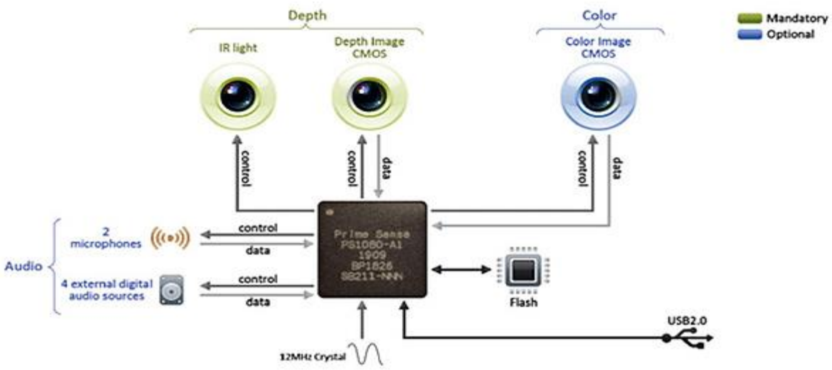


Figure 2-1: Diagram of PrimeSense platform (ifixit, 2010)

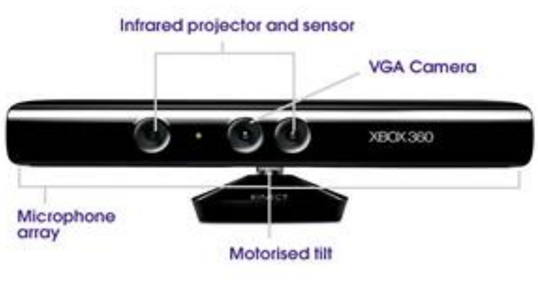


Figure 2-2: Microsoft's Kinect device (EverybodyPlays, 2010)

This device can capture about 300,000 coloured points in each frame that registration of consecutive images can increase the point cloud density. By increasing the distance from sensor, the random error in depth measurements also increases, such that in five meters of range, the error reaches 4 cm. Also the increase in distance results in decrease of depth resolution, as in five meters of range, the point spacing in depth direction is around 7 cm. (Khoshelham and Elberink, 2012)

2.2. Indoor modelling

Several researches focused on building interior modelling and reconstruction of indoors using Laser sensors data. Budroni and Böhm have continuing works on modelling indoor scenes through sweeping planes to find the basic planes of building interiors and also some primitives (Budroni and Böhm, 2009, 2010). In another work on reconstructing indoor scenes, Jenke et al. (2009) used statistical and Bayesian reasoning to model interiors. They described a model with unknown size, orientation and scale, and in an iterative optimization process, they fitted cuboids to the point cloud and through scoring each cuboid, the best model fixes to the point cloud (Jenke et al., 2009). Similar work was done by Dick et al. for modelling architectures from images of exterior facades; but they used model-based object recognitions and optimization of them beside the Bayesian framework (Dick et al., 2004). Although it is about facades modelling but the research describes an effective method to model the whole building with shape primitives. One efficient method is also developed by Adan (2011) to reconstruct cluttered interiors; by detecting walls with only projection of points into 2D space followed by Hough transform and then detecting occluded areas as well as openings and distinguish these two from each other to place primitives (Adan and Huber, 2011).

Most of the mentioned researches focus mainly on the recognition of planes of building interior. The method to reconstruct a model using the detected planes is another step in 3D modelling. There are some techniques answering to this necessity, one of which the methods is grammar-based modelling.

In the 70's, Stiny introduced grammar for shapes (Müller et al., 2007; Stiny and Gips, 1971) and then the shape grammars were widely used in architectural designs (Downing and Flemming, 1981; Flemming, 1986), also in geospatial sciences mainly for visualizations (Karnick et al., 2009). In this idea, a shape is generated based on shape primitive component with specific rules (Stiny and Gips, 1971). The rules of shape grammar are developed in urban modelling fields with procedural modelling of buildings using volumetric mass modelling to impose the architectural regularizations (Müller et al., 2006; Müller et al., 2007; Vanegas et al., 2010). Vanegas et al. added Manhattan-World assumption rules to grammar based volumetric modeling to model L-shaped and U-shaped and pushback-shaped buildings (Vanegas et al., 2010). In these researches the main idea is to reconstruct a model based on a shape as a primitive and then adding details to that primitive until the reconstructed model is verified.

The idea of volumetric grammar-based modelling were applied for modelling of building indoors, too. Xiao and Furukawa in their research use a cuboid as a primitive shape with union or difference operations to model the inside of buildings. They first split the 3D space into horizontal slices (floor plan structure in each slice), then using 2D rectangles as primitives, they solve the grammar based modelling in 2D and at the last step using these reconstructions of 2D space, they generate 3D primitives and once again through the union and difference operations, reconstruct a 3D model (Xiao and Furukawa, 2012). This procedure is in Figure 2-3.

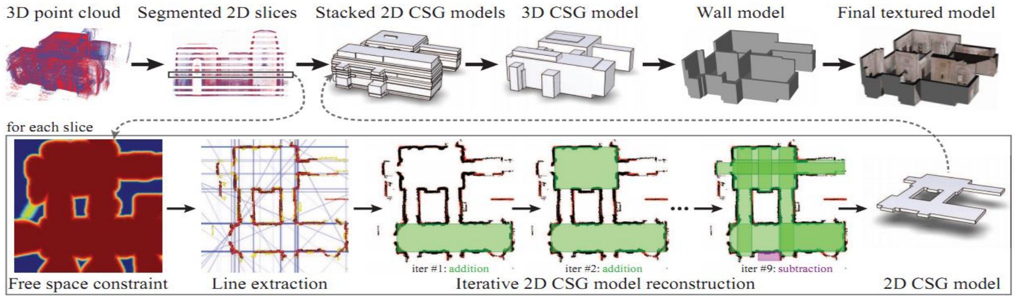


Figure 2-3: System pipeline of reconstructing the inside of building, first row: entire pipeline, second row: grammar based modelling in 2D floor plan structure (Xiao and Furukawa, 2012)

2.3. Using RGB-D data for indoor modelling

There are a few works to use the RGB-D data for indoor mapping and modelling. The researches in this field mainly do a real time mapping of indoors. This technique has various applications in navigation and surveillance.

For example, in robotics, Thrun et al. worked on providing real time map of interiors using the data captured by mobile robot equipped with range and image sensors. They used statistical methods to model rectangular flat surfaces representing walls, doors, etc. and small polygons for non-flat surfaces, and through Expectation Maximization of the surfaces, a map of compact sets of polygons is generated. They also add textures to the polygons using the image data of RGB sensor (Thrun et al., 2004). Their approach is shown in Figure 2-4.

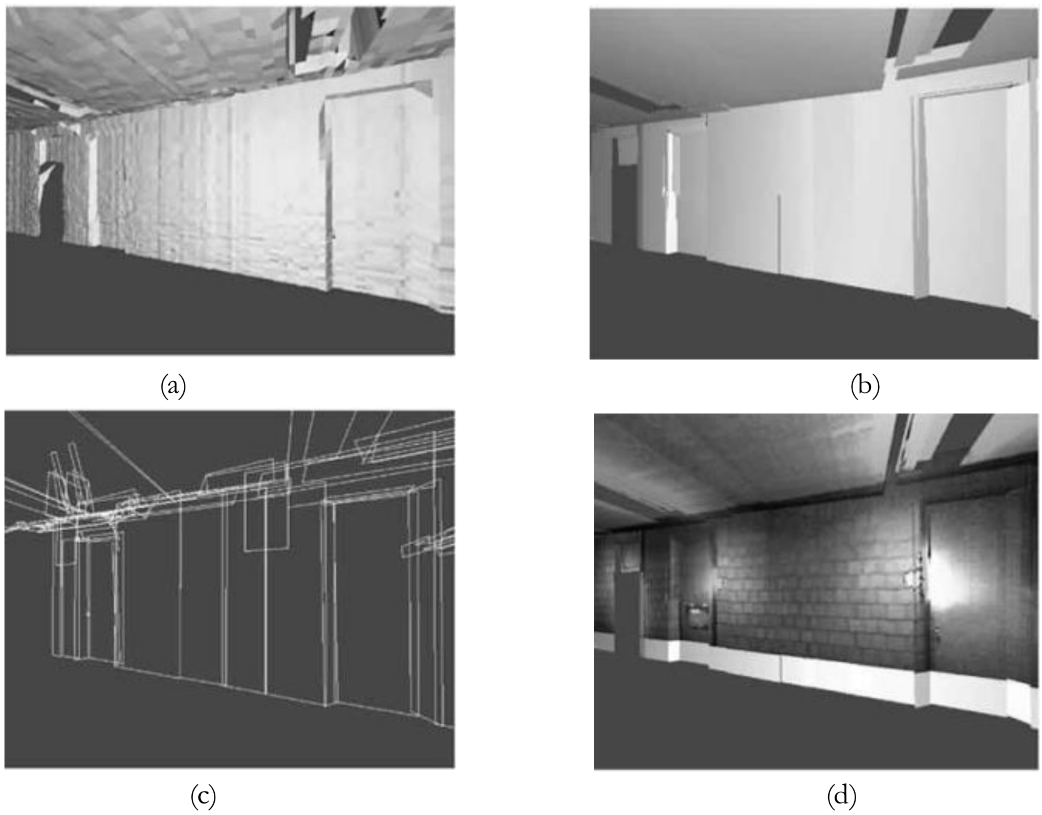


Figure 2-4: Indoor mapping using depth and image data, (a): row data map, (b): Planes detected from map, (c) Plane boundaries, (d): Full texture map (Thrun et al., 2004)

In another work, Henry et al. worked on 3D mapping of indoors. The method they developed is using the optimization algorithm combined with visual features and shape-based alignment. Then they use depth

information to detect the loop-closure and global optimization for map consistency (Henry et al., 2012). The result is dense 3D map in which the frames of the scene are aligned correctly.
The challenging step in modelling is the recognition of planes and objects. The methods developed to recognize the planes can be used in indoor modelling. On the other hand there are good works done on segmentation and recognition out of RGB-D data. Holz et al. introduced a method to recognize real time planes using the RGB-D cameras. In this method the local surface normal are calculated on integral images and then the points are clustered, segmented and classified on both normal space and spherical coordinates. Variation in both normal clusters and distance clusters, results in planes separation (Holz et al., 2012).
Also in a research on objects recognition using Kinect data, Hernandez-Lopez et al. (2012) developed a method for segmentation of points. In this work they use colour data and depth measurements integrated for segmentations and object recognitions. First they segment points in CIE-Lab space after converting the RGB image to this space. Then using the depth data, they reclassify the detected objects with same colour but different planes. (Hernández-López et al., 2012).

2.4. Summary

Indoor modelling have various applications in surveillance and emergency management, navigation, architecture, virtual tours and so on. Thus there are so many researches done to solve the modelling by developing various methods. One of the methods in 3D modelling, is grammar based modelling, which considers a shape as a primitive and reconstructs the model based on the primitives using union and difference operations.
Meanwhile one of the most used datasets to do the modelling of buildings is the point cloud, although most of them captured by laser scanners for indoor modelling. As using the RGB range sensors have impressive growth in creating point clouds in recent years, there are also works which use these colored range data to do the buildings modelling. However for indoor modeling, they mainly focus on detection of planes and faces. The different methods of indoor modelling using laser point clouds and also colored point clouds have been studied. But still there's a need to develop a method to reconstruct a 3D model out of RGB-D data.

3. RESEARCH METHODOLOGY

3.1. Framework

This research is done in two main phases. First we need to recognize the planes of buildings inside. When all the planes (walls, floors, ceiling) are detected, we can reconstruct a 3D model which best fits to the planes. The proposed method for these processes is described in details and step by step in the following. The method was also implemented on captured dataset and experimented on simulated data to see if the developed method is applicable in different shapes of indoors or not.

When we got the 3D model, we used the VRML file format to represent the model in 3D. For better visualization we did texture mapping on the reconstructed model using the RGB data captured by Kinect sensor per point beside the depth measurements and also used the photos taken by ordinary digital camera to map the textures on the 3D model for more realistic visualization.

In the last phase of research, when we got the full 3D model of indoor, we could assess the quality and discuss about the accuracy we can gain by the developed method. This step is discussed in chapter 6.

Figure 3-1 demonstrates an overview of research phases.

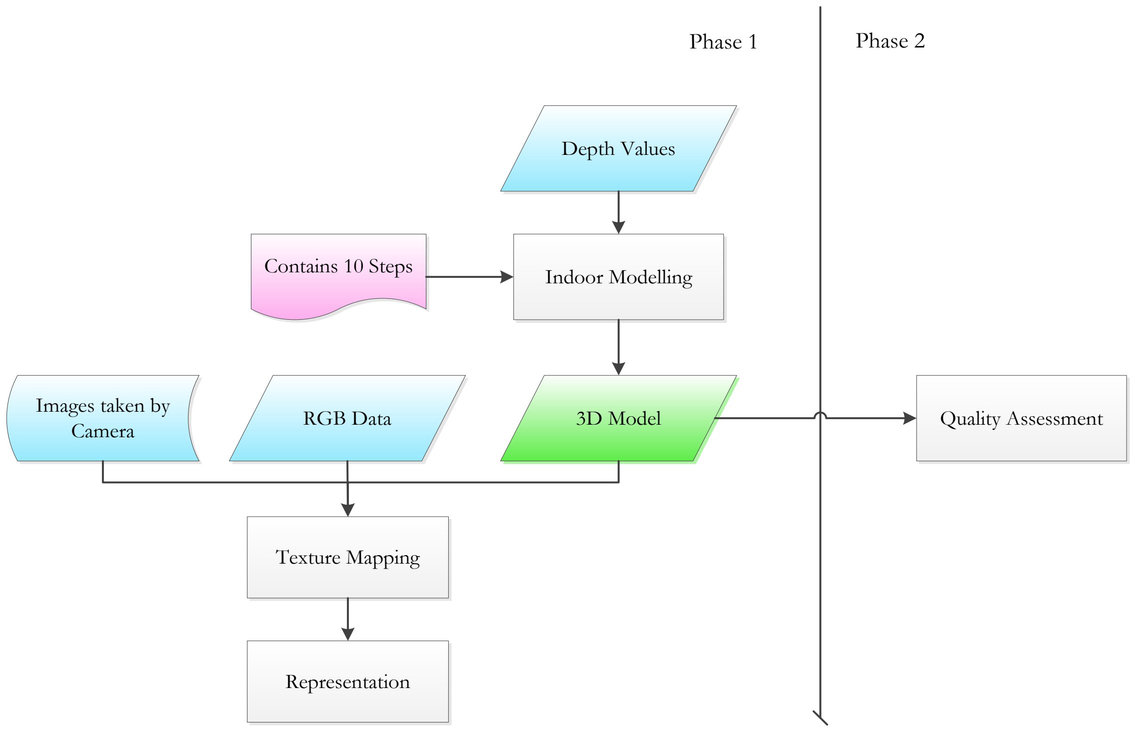


Figure 3-1: General Methodology

3.2. Modelling

The method to model the indoor, starts with recognition of planes of indoor. These planes contain all walls and ceiling and floor. In this research we assume that there is only one plane of ceiling and floor, i.e. we don't have steps on the floor or false ceiling. The walls are perpendicular to floor and ceiling and also to each other to build a grid-based structure (Manhattan-World). When the planes are detected, we need a

3D shape to fit to the planes. Inspired by reality of architectures, a regular shape that could be used for modelling is a cube.
In this research, for recognizing the main planes, we first segmented the point cloud. For each segment the normal vector was calculated and based on these vectors, we classify the segments into six classes representing the main planes of building interior. Then we select the segments from each class correspondent to indoor planes.
After recognition of planes, we can fit a cube to them, but if we have planes outer than a regular cube, some planes would not be considered in cube fitting. To increase the accuracy of model we need also consider these planes. In this research we used grammar-based modelling to involve all planes of indoors in the resulting model.
Inspired by grammar-based modelling ideas, we decided to model the indoors using the shape grammar rules. When we want to fit the cube to the planes of indoor, in the first stage we only consider the farthest planes. Then we have a large cube of indoor. The remaining parts always create rectangular shapes with minimum of two wall planes (Manhattan-World). In the next steps we fit smaller cubes to the remaining parts of scene and subtract these cubes from the largest one (difference operation). In this way all the walls of indoor will be reconstructed in final model.
We may have smaller parts again after second steps. Then we continue the cube fitting to smaller remained parts. The subtraction process starts from the smallest parts (last step) out of its larger (previous to last), until we get the first largest cube. The demonstration of this procedure is in Figure 3-2.

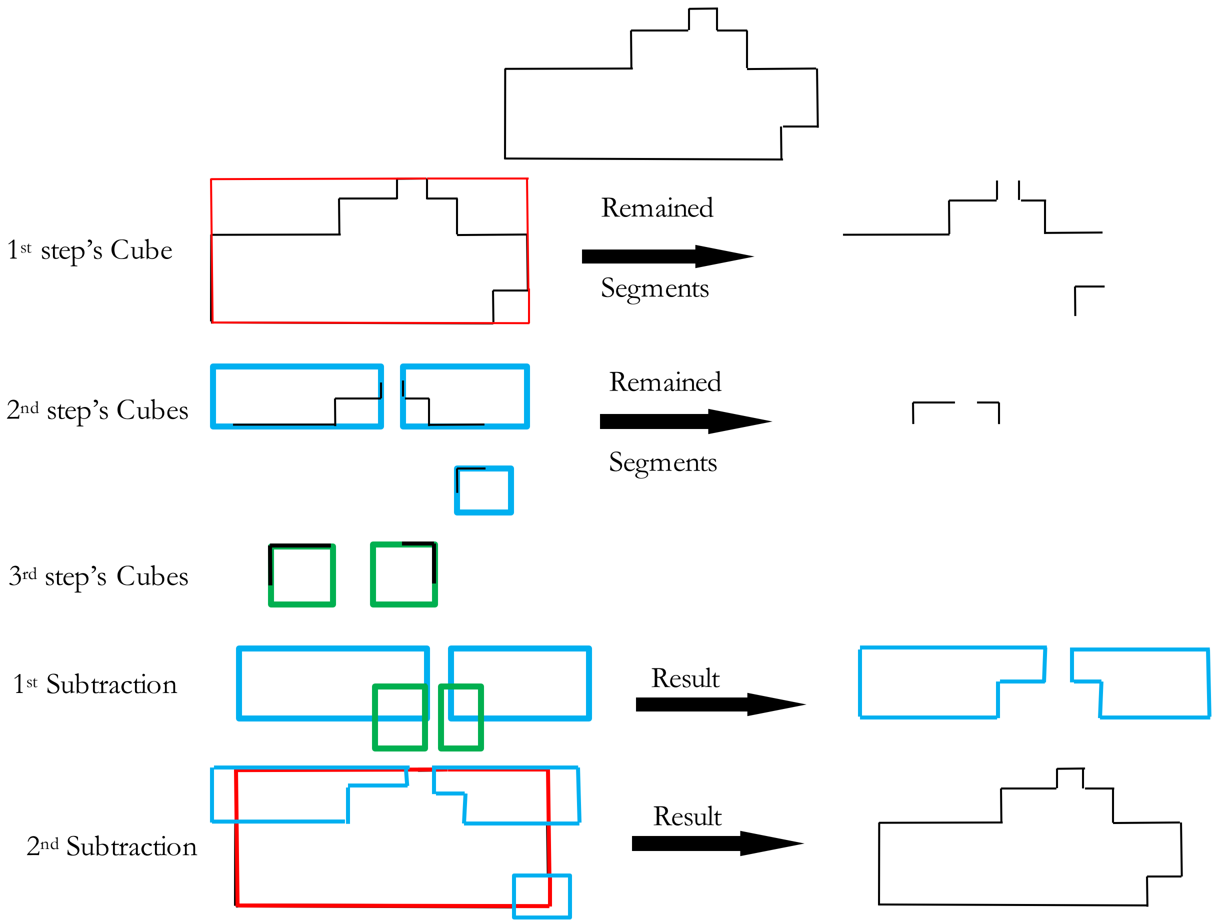


Figure 3-2: Modelling procedure step by step

All these steps from segmentation to get the final model are demonstrated in Figure 3-3. The details of each step are described in the following.

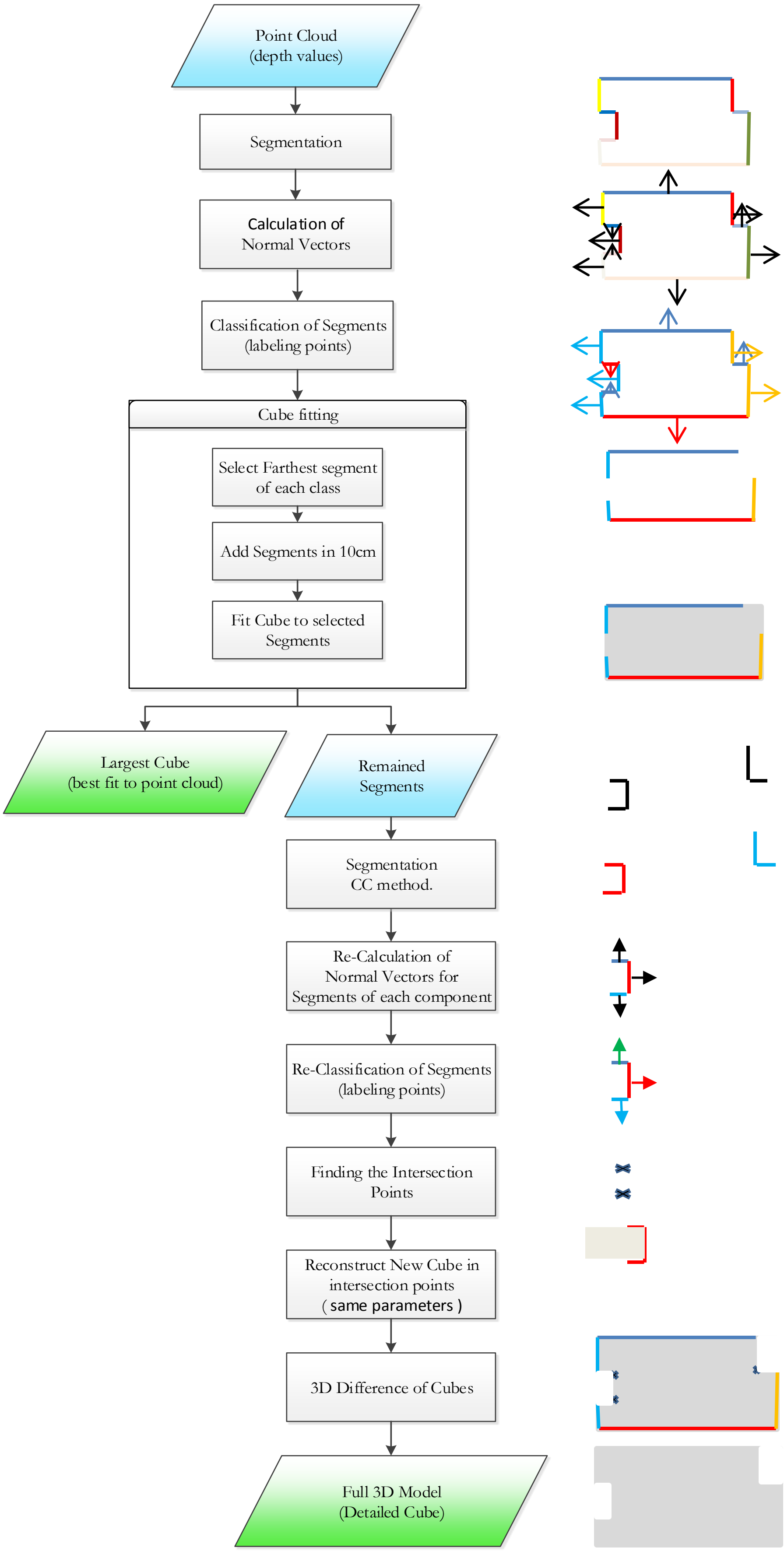


Figure 3-3: Methodology 3D modelling of indoors

3.2.1. Segmentation

The first step of implementation was to segment the point cloud into planes to calculate normal vector elements of each segment. For this purpose the surface growing method was chosen to segment the points.

The surface growing algorithm first finds nearby points with the best fit into a plane. These points play the role of seeds and then generate the segments with adjacent non-classified points if their distances to the plane or locally defined smooth surface are less than a threshold distance (Vosselman, 2009).

The segmentation of point cloud was done in PCM software. In PCM software, there are ready tools to do segmentation based on the mentioned method. The only thing to do is to configure the parameters of the segmentations. For surface growing segmentation the main parameters are the radius of growing surface and distances to the surface; which need to be adjusted according to the accuracy of sensor and point cloud. There are also parameters related to selection of seed points. All the parameters related to distance in PCM have to be set into the unit of point's coordinates in the cloud. Figure 3-4 shows the parameters we used in this research.

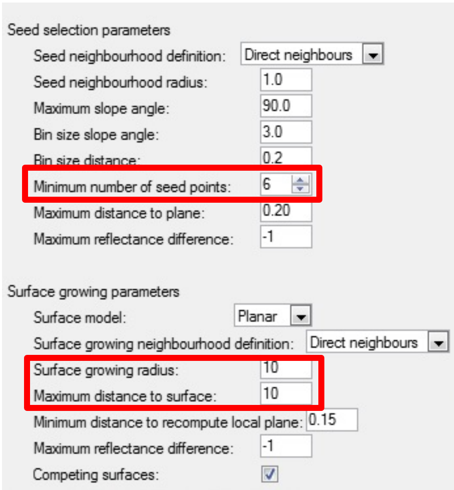


Figure 3-4: Segmentation parameters in PCM software

3.2.2. Normal vector estimation

To estimate the normal vector for each segment we tried two methods. First we estimated the normal vector for all points of each segment. To do this, we find six nearest neighbour points to the selected point. Then through fitting a plane to these seven points, the normal vector could be calculated and assigned to the selected point. When all points of one segment had normal vector elements, by averaging the components we could assign the mean normal vector to the segment. The process of finding n nearest neighbour was done with approximate nearest neighbour (ANN) algorithm, which had ready toolbox programmed in MATLAB. We could also classify points based on directions of normal vectors, then we don't even need segmentation. But in this method all points are involved in the calculation of normal vectors, despite of any error may took place in assigning outlier points to the segments.

The second method to estimate the normal vector of segments tried to cover the mentioned shortcoming of previous one (separation of outlier points). In this method we fitted plane to the points of segment considering the separation of inliers and outliers. RANSAC[1] algorithm used to fit the plane to inliers. In this algorithm at first, at least three points are selected randomly and a plane fits to them. Then by comparing the threshold distance with distance of points of cloud, the inlier points to the plane are selected. This process repeats and each time the generated plane is compared to the previous plane until

[1] RANdom SAmple Consensus

the selected plane satisfies the stop rules (Yang and Förstner, 2010). When the plane best fits to points of each segments, the parameter of plane can be calculated. These parameters are three elements of normal vector and perpendicular distance of the plane to the origin of coordinate system.

In this research, the second method is used to continue the implementation through writing codes in MATLAB.

3.2.3. Classification of planes

Scattering the normal vector of segments shows the variety between them. The segments with similar elements of normal vector create separate clusters in this scatter, as the normal vector of segments with same direction are approximately same.

On the other hand we know the normal vectors of segments belonging to the defined walls based on the research assumptions. In an ideal clustering, that normal vectors of indoor planes are along the coordinate system axes, we can say that:

- The normal vector of front wall is along the Y axis ($n = (0, 1, 0)^T$).
- The normal vector of right wall is along the X axis ($n = (1, 0, 0)^T$).
- The normal vector of ceiling is along the Z axis, ($n = (0, 0, 1)^T$).
- Back wall is apposite the front wall, left one is apposite the right one and floor is in apposite of ceiling.

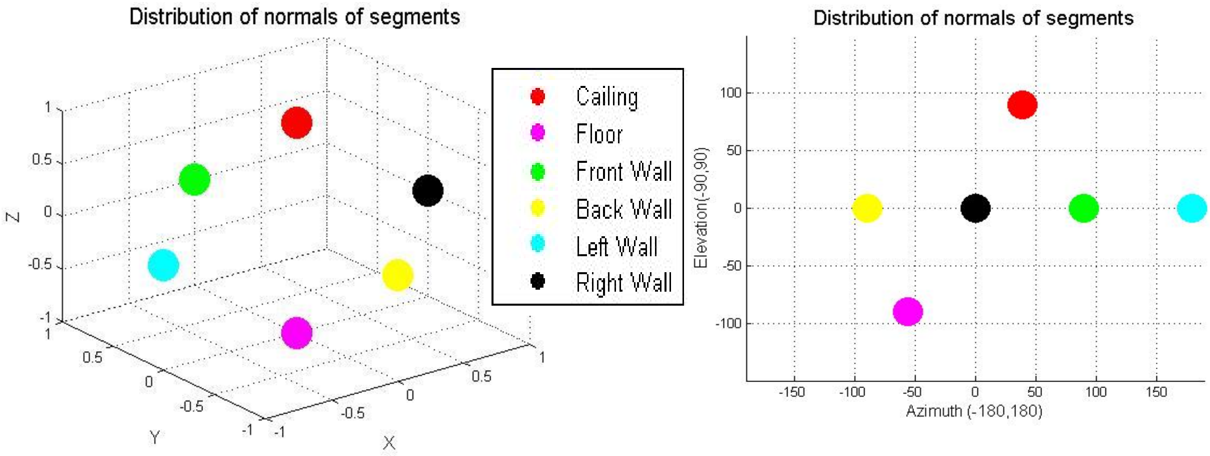
Considering these characteristics of normal vectors, we can classify them into six main groups; one for ceiling, one for floor and four classes for walls. The uncertainty and errors due to the remained rotation in the point cloud, can also be involved in the constraints for classification. Table 3-1 show the constraints used in our research. n = (n_x, n_y, n_z) is the denotation of normal vector.

Table 3-1: Constraints to classify the segments based on normal vectors

FOR	NORMAL VECTOR	CONSTRAINNTS		
	n	n_x	n_y	n_z
CEILING	$[0\ 0\ 1]^T$	-0.2≤ ≤0.2	-0.2≤ ≤ 0.2	0.8≤ ≤1
FLOOR	$[0\ 0\ -1]^T$	-0.2≤ ≤0.2	-0.2≤ ≤0.2	-1≤ ≤-0.8
FRONT WALL	$[0\ 1\ 0]^T$	-0.2≤ ≤ 0.2	0.8≤ ≤1	-0.2≤ ≤ 0.2
BACK WALL	$[0\ -1\ 0]^T$	-0.2≤ ≤ 0.2	-1≤ ≤-0.8	-0.2≤ ≤ 0.2
RIGHT WALL	$[1\ 0\ 0]^T$	0.8≤ ≤1	-0.2≤ ≤ 0.2	-0.2≤ ≤ 0.2
LEFT WALL	$[-1\ 0\ 0]^T$	-1≤ ≤-0.8	-0.2≤ ≤ 0.2	-0.2≤ ≤ 0.2

Then we can determine the true classified segments into each class based on some other considerations. For example we know that the ceiling is only one plane, so in the class of ceiling the segments must be in similar perpendicular distance of the origin. As the origin of coordinate system is on the centroid of point cloud, this distance must be around the half of room height. It is also equal to the distance floor segments of the origin (based on research assumption we don't have steps on the floor.). With this constraint we can clean the classes out of wrong segments belonging to the cluttered objects of indoor, i.e. all the segments with distances (approximately) smaller than the half of the building height must be removed from ceiling and floor classes.

Figure 3-5 shows how the normal vectors of segments for an ideal cube create separate clusters beside each other.



(a) (b)

Figure 3-5: Distribution of normal vectors of segments; (a): Normal vectors in 3D coordinate system, (b): Normal vectors in 2D coordinate system (azimuth and elevation)

3.2.4. Cube Fitting

Up to this step, all the indoor planes were recognized and labelled. As discussed before, for fitting the cube, first the farthest planes must be selected to reconstruct a large cube containing whole indoor. In next steps we could subtract the additional parts out of this large cube to model the details.

Using the perpendicular distances of segment planes to the origin, we can detect the farthest segment in each class of labelled segments. Due to the accuracy of sensor and registration errors in dataset, we may have segments with low differences in their perpendicular distances. Also, inspired by the reality, we know that there's almost no two planes with the difference of distances lower than 10 cm. So when we select the farthest segment of each class, we then add all the neighbor segments within the 10 cm of it. Then we create a matrix of points correspondent to the selected segments in a specific order, in which points of one class come after each other and then the next class's points are put. For each class we assign an id representing the faces in cube.

Fitting cube to the selected segments, itself contains some steps as follow:

- Creating a unit box with the centroid on the origin of coordinate system,
- Assigning correspondence labels between the points in the point matrix and the faces of the unit cube
- Estimating the parameters of the transformation matrix that transforms the unit cube such that each face of the transformed cube fits to its corresponding points
- Calculating the cube vertices through the estimated parameters which define the cube that best fits the points

This algorithm is written in MATLAB. When we have the vertices, we can visualize the cube using that vertices and the face ids on the scatter of point cloud to get a representation of cube in MATLAB. The vertices and parameters of transformation and also the planes of floor and ceiling must be stored for next steps.

Also we can extract those segments of each class that were not involved in fitting the largest cube to point cloud. These remaining segments will reconstruct new smaller cubes to be subtracted of the largest one, to model the details of indoor.

3.2.5. Adding details of indoor

This step is mainly the repetition of the previous procedure, with only some differences. First we put all the remaining segments of each class beside each other. All these points have to be re-segmented into separate components, because we may have more than one remaining cube. In this situation each cube has its own segment which we need to separate them out of each other into components and repeat modelling separately on each component. This time connected component method is going to be used in segmentation. This method splits the points into separate components. Each component may have more than one smooth surface, but all of the surfaces are connected to each other based on threshold distance. This is done in PCM. The main parameter of connected component segmentation is distance between points.

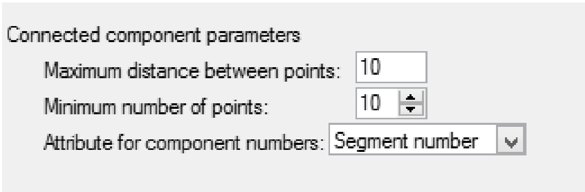


Figure 3-6: Parameters of Connected Component segmentation on PCM

Then we translate the coordinates of points to new coordinate system with the origin on the centroid of component. For segments of each component, the normal vector are calculated in new coordinate system. As we have only wall's segments, we classify the segments into four class of walls (if there are). Once more the farthest segment plus the segments in 10 cm of it, are selected from each class. New planes are fit to the selected segments.

Through the planes of floor and ceiling from the previously fitted cube and new planes we can calculate the intersection points of new planes with the cube. The intersection point is on the planes of two walls and ceiling/floor.

We have to reconstruct new cubes in these intersection points. But as we have less planes than six necessity, we can translate the previous cube to the intersection points. Meanwhile, to reserve the Manhattan-World assumptions, the new cube(s) and the first cube must be in same rotation. Also we have only one plane of ceiling and floor, so the scale along the Z direction must be preserved. The only unstable parameters can be scale factors along the X, Y direction and translation vector. The translation vector can easily be calculated through translating the correspondent point of first cube to one of the intersection points.

To calculate the scale factors along X, Y direction:

- If we have only one intersection point on floor (ceiling), then these scale factors can be set to infinite (a large number) ,
- If we have two intersection points on floor, which are along the X axis, then the X scale factor can be calculated through the X-length between the intersection points, the Y scale factor can set to infinite,
- If there are two intersection points on floor, which are along the Y axis, then the Y scale factor can be calculated through the Y-length between the intersection points, the X scale factor can set to infinite.

Then we have new cube reconstructed in intersection point, which can be subtracted from first cube.

If we had remained segments from second step, then we should repeat the mentioned procedure again before subtraction, until there was less than two planes remained at the end of process. The vertices of new cubes are also stored. Although the steps are repeated, but the subtraction of connectors out of cubes must start from the last (smallest) one from its larger to get the all details on walls. This process was demonstrated in Figure 3-2.

The subtraction of cubes can be done in Arc Scene with the 3D Difference toolbox. This toolbox can calculate the volumetric difference of 3D shapes. The input file for this operation can be in .3ds format. When we have the vertices of cubes, we can easily create the VRML files for each cube. These VRML files can be converted into .3ds files to be imported into Arc Scene.

3.3. Representation

After subtraction of cubes, we have the vertices of full model. Then we can write the VRML file of model by putting the coordinates of vertices and defining the indices of points which are correspondent to each face of model. Web browsers with specified plugin or any free VRML viewer can lunch the VRML file and represent the model in 3D.

In fact, we can export the result of 3D Difference in Arc Scene into VRML file. But this file is so complicated to work with or edit it. We want to assign a texture for each face of model for better visualization, so with writing the faces with point indices, we can also dedicate an image (texture) to the face in VRML.

To create a texture image of faces for model, we can use the RGB values captured by Kinect Sensor. The description of procedure is in the following.

One other way to represent the model is in SketchUp software, which beside the visualization, lets the operator to edit, measure, map the texture, etc. on the model. The .3ds model can be imported into SketchUp. We used SketchUp to map the photos taken by camera on the model, to get a realistic view of indoor.

3.3.1. Texture mapping with RGB values

We have RGB values beside the depth measurements per point. To map these data on the reconstructed model, we first mapped an empty image with pixel size of 1×1 cm^2 on the planes of reconstructed 3D model. Then we assigned the coordinate to the center of each pixel through the model coordinates (coordinates of vertices). The sketch for this process is in Figure 3-7: Sketch of calculating pixel coordinatesFigure 3-7.

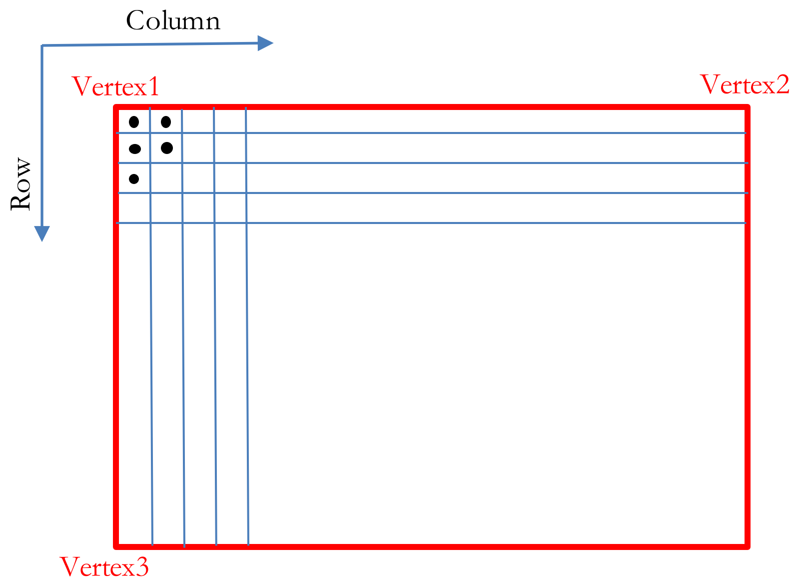


Figure 3-7: Sketch of calculating pixel coordinates

In next step we find three nearest neighbor to the each coordinates of pixel's centers. The process of selecting nearest points were done based on ANN algorithm. These nearest neighbor points have RGB values that the average of those values were assigned to the pixel on the empty image.
At last when the image filled with the RGB values, we saved the image in .jpg format. These images can be represented on the 3D model using VRML format.

3.3.2. Texture mapping with photos

The RGB values per points beside the depth measurements, may not represent realistic colours of indoor. Thus to give better visualization of the indoor model, high resolution photos were taken with digital camera. These photos were mapped as a texture on the faces of 3D model on SketchUp software. There we can import the photos as a texture to add them to each face.

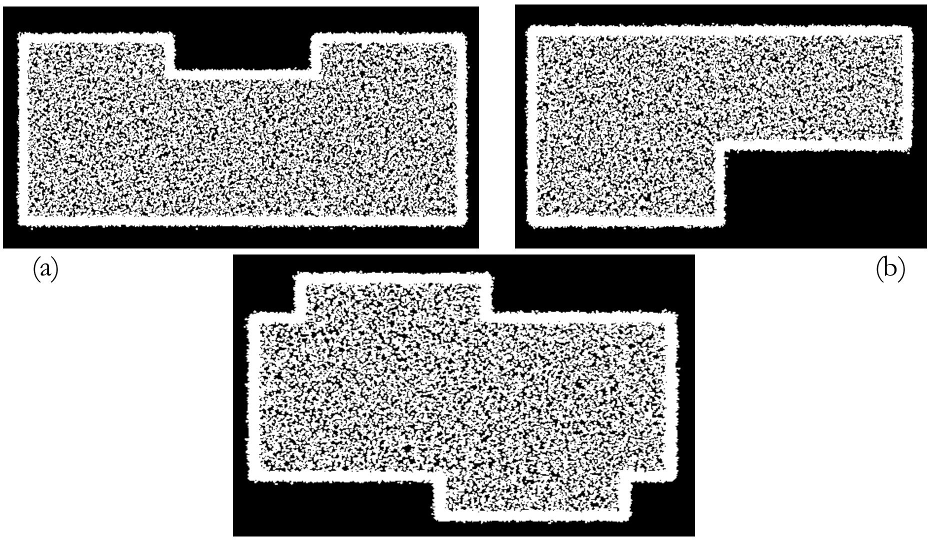
3.4. Summary

In this chapter, we first illustrated the framework of research into a flowchart. The research proposed into two main phases of modelling and quality assessment, which the last step is described in the Chapter 6 for the results of implementation.
The modelling process itself, contains ten main steps which demonstrated in a separate flowchart. First we segment the point cloud, and calculate the normal vector for each segment. Based on these normal vectors, we can classify the segments into six classes. We select the proper segments based on the perpendicular distances from the coordinate system origin for fitting the cube. With the first cube, we can determine the segments which are not involved in the model. So we repeat the process for remained segments to reconstruct new smaller cubes. By subtraction of smaller cubes out of the large cube, the reconstructed model will consider all the planes of indoor.
To represent the reconstructed 3D model, the VRML file format is used, which also have capability to display textures for faces of model. To enhance the visualization of model, the textures, using RGB values in point cloud and photos taken by camera, map on the faces of model.

4. MATERIALS

4.1. Data

To implement the developed method we used both simulated data and real captured data. We created simulated datasets in shapes of U, L and X shape to check the validity of developed method in terms of accuracy and performance. Figure 4-1 shows point clouds of simulated datasets.



(a) (b)

(c)

Figure 4-1: Simulated point clouds in different shapes, (a): U shaped, (b): L shaped, (c): X shaped

The real data was captured by Kinect sensor from a room interior in ITC faculty. The room is L-shaped. Figure 4-2 shows the point cloud of this room. This point cloud consists of 1,376,134 points with X, Y, Z values and also R, G, B values. In some parts, like as on the floor under the sensor or on the ceiling we have gaps. There is a bookcase in front of one of walls that causes gap in almost half of the wall. One of the walls is almost covered by windows, so we don’t have depth measurements for that parts.

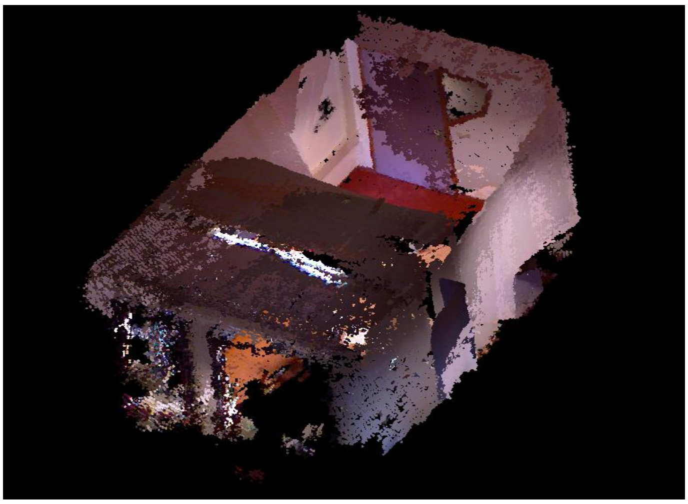


Figure 4-2: The captured point cloud of a room

4.2. Data Preparation

4.2.1. Registration

Each captured frame with Kinect, contains one RGB image of view, one disparity image of the view and one infrared image. Based on the disparity images, we can calculate the depth values (distances to the sensor)(Khoshelham, 2011). To align all captured frames and bring them into one single coordinate system, registration has to be done (Pulli, 1999). By registration of all disparity images the point cloud with X, Y, Z coordinate values is constructed.

Although the registration of frames is the first step in data preparation, but it's not yet a fully solved problem. For automatic registering of the scans, we need either of discrete object (point) matching, e.g. in (Bae and Lichti, 2008), or surface matching (Huber and Hebert, 2003), or correlation matching on rotationally aligned pairs (Makadia et al., 2006), line matching methods or any other method with combination of those . When we have a uniform surface with no change of depth values or textures, e.g. a plane of white wall, then automatic registration step fails in finding matches. This makes the resulting point cloud to have registration error, which means the low accuracy. This error of registration then influences the next steps like modelling based on the point cloud.

As well, the point cloud for our use in this research had (big) errors due to registration. For example, we did know the ceiling is only one plane in our point cloud, but in segmentation of the point cloud we had more than one segment for the ceiling with distances even more than 30 cm. It is to be mentioned that we tried to cover big errors by sticking some segments manually to each other (unifying the segment numbers of them).

4.2.2. Rotation

In some steps of implementation, like as for classification of normal vectors, the angles have to be considered. As we assumed that the main planes are horizontal and vertical, we need to be sure about the orientation of point cloud; because the normal vectors must be almost along the coordinate system axes. On the other hand, we know that the Kinect sensor was not levelled during data capturing, although the observed rotation angels around the coordinate system axes were low. So we need to check the orientation of point cloud and set the normal vectors along the axes (orient) beforehand in data preparation processes.

We can calculate the approximate rotation matrix of the point cloud by checking the rotation of normal vectors around the axes, for certain segments. So some segments have been selected manually which belong to the determined planes of room, e.g. one segment for left wall and one for front wall and etc.. Then the normal vector of these segments are compared with the expected normal vectors and through the transformation of these two set, the rotation matrix can be calculated.

During the registration of point cloud, the origin of coordinate system is set on the centroid of point cloud and the normal vectors are in outward direction of the origin. We call the planes based on their normal vectors, and the division is into six labels. The rules for labelling in a point cloud with no rotation, are:

- Front wall: the normal vector is along the Y axis ($n = (0, 1, 0)^{\mathrm{T}}$).
- Right wall: the normal vector is along the X axis ($n = (1, 0, 0)^{\mathrm{T}}$).
- Ceiling: The normal vector is along the Z axis, ($n = (0, 0, 1)^{\mathrm{T}}$).
- Back wall is apposite the front wall, left one is apposite the right one and floor is in apposite of ceiling.

Based on these assumptions, we can select some segments correspondent to planes of each class manually and calculate the normal vectors of those segments. The transformation of the calculated normal vectors to the expected ones will give us the rotation elements.

This rotation matrix is calculated based on following equation:

$$\mathrm{n}_j^1 = \mathrm{R}\,\mathrm{n}_j^2 \tag{4-1}$$

The prove for this can be done as following (Khoshelham, 2010):

Rotation of normal vector, means the rotation of plane. Consider that $\mathrm{n} = (\mathrm{n}_1,\ \mathrm{n}_2,\ \mathrm{n}_3)^{\mathrm{T}}$ is the normal vector of plane and π is the plane with normal vector of n and ρ as the perpendicular distance from the origin of coordinate system. Each point on this plane can be defined as:

$$\pi^{\mathrm{T}} x = 0 \tag{4-2}$$

When we transform the plane *j*, that contains the points from second scan to the first, we can write:

$$\pi_j^1 = \mathrm{H}\,\pi_j^2 \tag{4-3}$$

$$\mathrm{H} = \begin{bmatrix} s\mathrm{R}_{3\times3} & \mathrm{t}_{3\times1} \\ 0_{1\times3}^{\mathrm{T}} & 1 \end{bmatrix} \tag{4-4}$$

Where H is the similarity transformation matrix and s indicates the scale, R is the rotation matrix and t is the translation vector. Rewriting the Equation (4-3) with the plane parameters results in:

$$\begin{bmatrix} n_j^1{}_{3\times1} \\ \rho_j^1 \end{bmatrix} = \begin{bmatrix} s\mathrm{R}_{3\times3} & \mathrm{t}_{3\times1} \\ 0_{1\times3}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} n_j^2{}_{3\times1} \\ \rho_j^2 \end{bmatrix} \tag{4-5}$$

In equation (4-5), for our use case, the scale factor is equal to one and we don't have translation vector. So this equation with these assumptions will lead to equation 4-1.

We calculate this rotation matrix based on transformation of selected segment's planes to the expected planes and then using this matrix we can rotate the whole point cloud along the coordinate system axes. Note that this will give approximate rotation angles of point cloud, but then we can ensure the rotation of point cloud along the axes is as low as to be negotiated.

4.3. Software

The method developed by writing codes in MATLAB to implement the algorithms. The license for this software was available by the ITC faculty.

For segmentation of point cloud, the Point Cloud Mapper (PCM) software has been used.

For visualization of 3D model and representing it, we wrote the files in VRML format. Any VRML viewer, e.g. FreeWRL Launcher (free access), can represent the model. This format can be easily converted to other 3D file formats, like as .3ds (3D Studio) format.

This conversion is needed to import the model into Arc Scene and SketchUP. We used the 3D Difference toolbox in Arc Scene to subtract the cubes from each other. SketchUp was used for texture mapping of photos on the model.

5. RESULTS

5.1. Indoor modelling

The developed method was implemented on three simulated datasets which are resembling three famous shapes of indoors. Then to assess the performance and quality of results for developed method the implementation was done on real point cloud captured with Kinect. The results for each step of modelling process is in below.

5.1.1. Implementation on simulated data

Here we will show the results of modelling steps for each of datasets separately.

5.1.1.1. L-Shaped point cloud

First step is to segment the point cloud in PCM. The parameters are set due to the errors of point cloud. We created these simulated point clouds with noise of 0.5 cm. then the normal vectors are calculated and classified into six classes. Based on the perpendicular distance of segments from origin of coordinate system, the farthest segment has been chosen and a cube is fitted to selected segments. The results for these steps are in Figure 5-1.

.

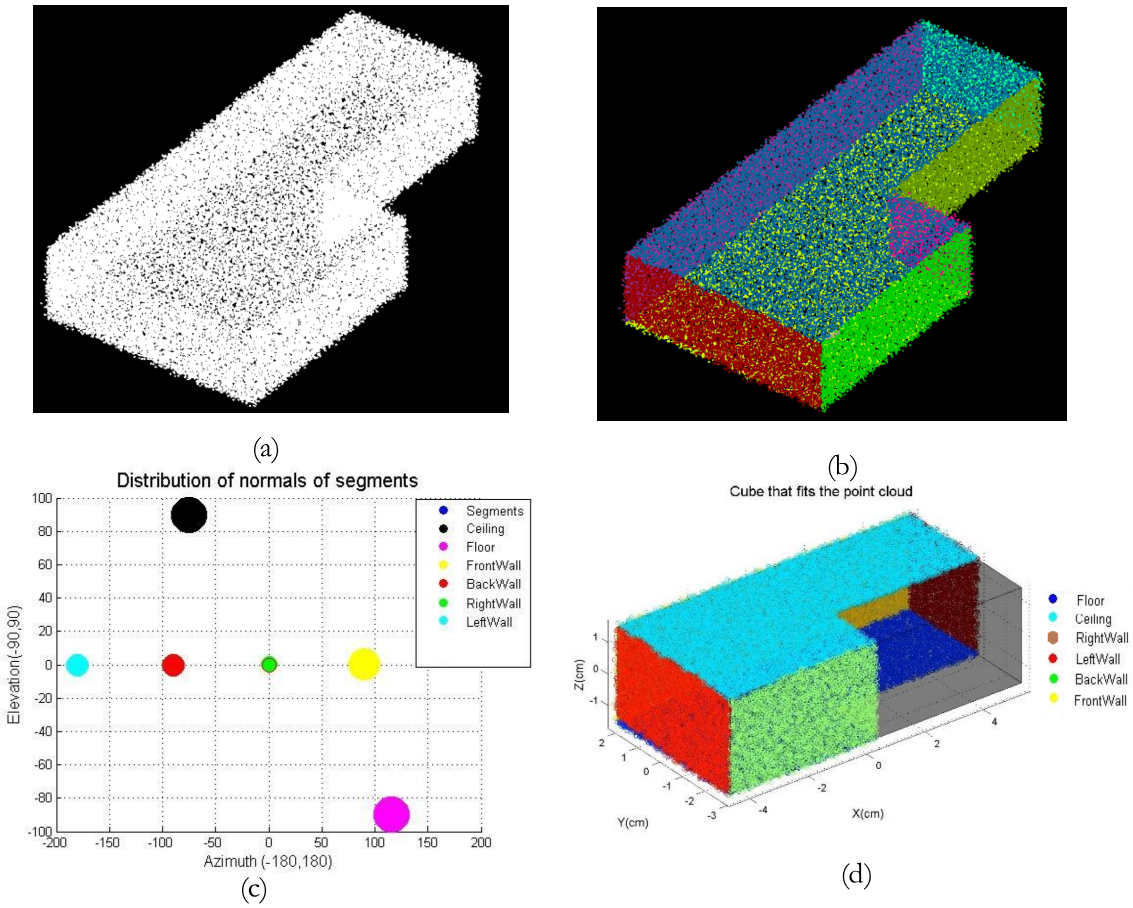


Figure 5-1: Results for first stage of modelling for L-shaped data, (a): Point cloud, (b): Segmentation in PCM, (c): Scatter of normal vectors of segments, (d): The largest fitted cube to point cloud

After fitting the cube to selected segments, we save the vertices and also export the remained segments. The new coordinate system is calculated for the remained segments with the origin on centroid of remained points. Normal vectors of segments are calculated and segments are classified. The intersection points of the remained planes with floor/ ceiling are calculated. The reconstructed cube of previous stage is translated to the intersection point. As we have only one intersection point, the all transformation parameters of previous cube are preserved. The 3D difference of 2 cube will result in a full 3D model. These results are in Figure 5-2.

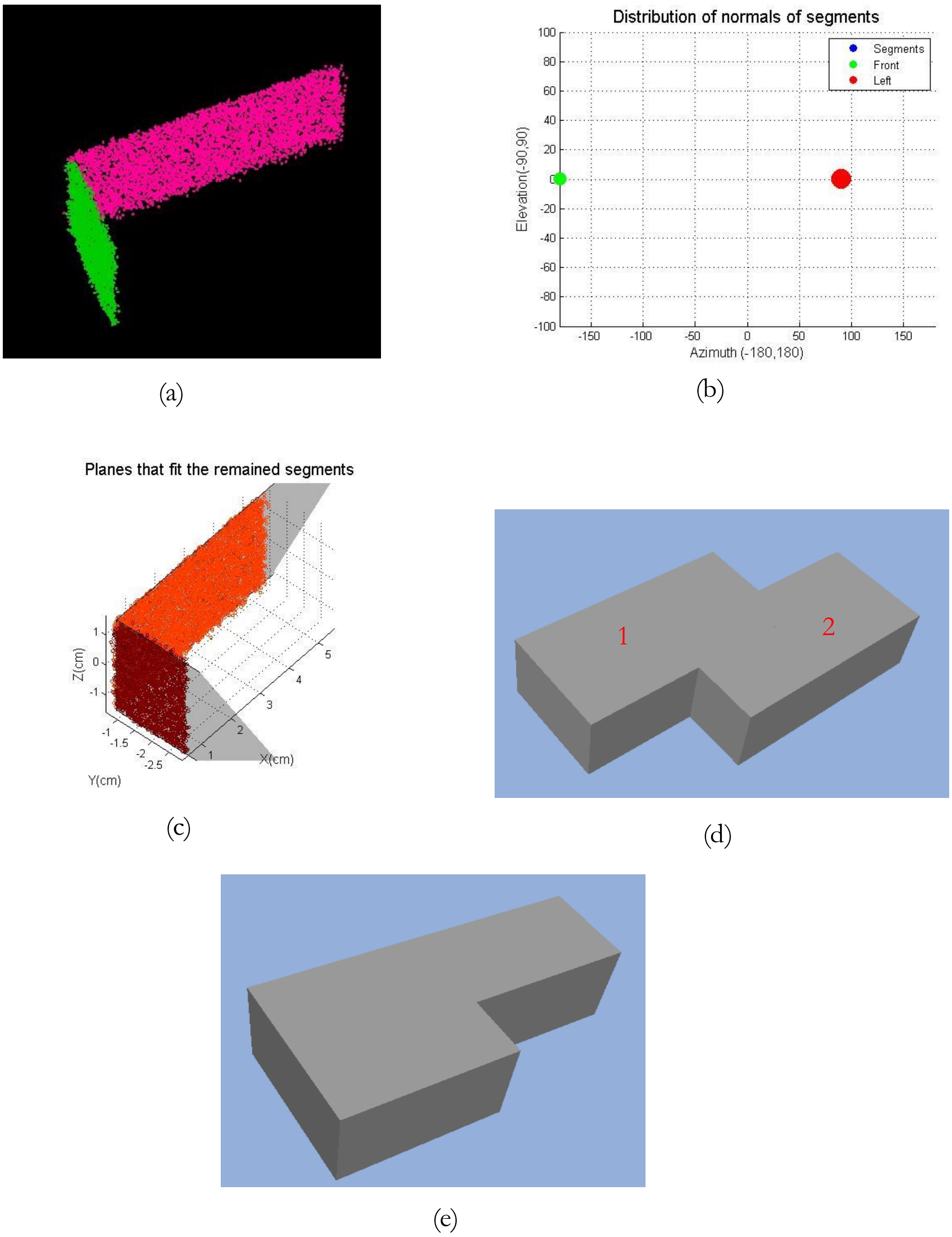


Figure 5-2: Results of second stage of modelling for L-shaped point cloud, (a): Remained segments, (b): Scatter of normal vectors of remained segments, (c): Planes fitted to remained segments, (d): Second cube in intersection point, (e): The full 3D model

5.1.1.2. U-Shaped point cloud

The same process is repeated for this point cloud. The results to get the first cube are in Figure 5-3.

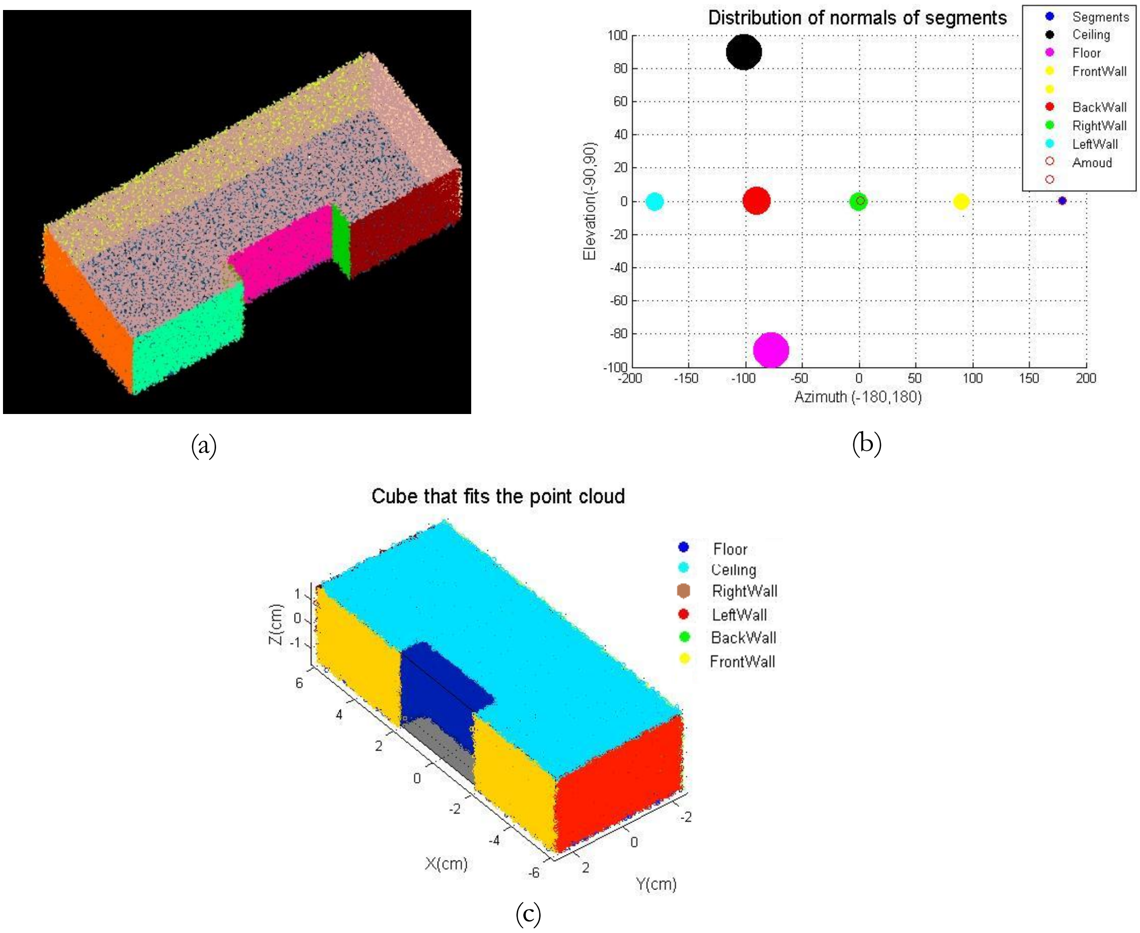


Figure 5-3: Results of first stage of modelling for U-shaped point doud, (a): Segmentation in PCM, (b): Scatter of normal vectors of segments, (c): The largest cube fitted to point doud

Then the remaining segments create one component. The new coordinate system is calculated for this component. The classification of segments is done based on calculated normal vectors. This time we have two intersection points of remained segments with floor/ ceiling. So beside the translation of cube to the intersection point, we have to change the scale of new cube. This scale factor is calculated using the distance between the two intersection points. The results are demonstrated in Figure 5-4 and Figure 5-5.

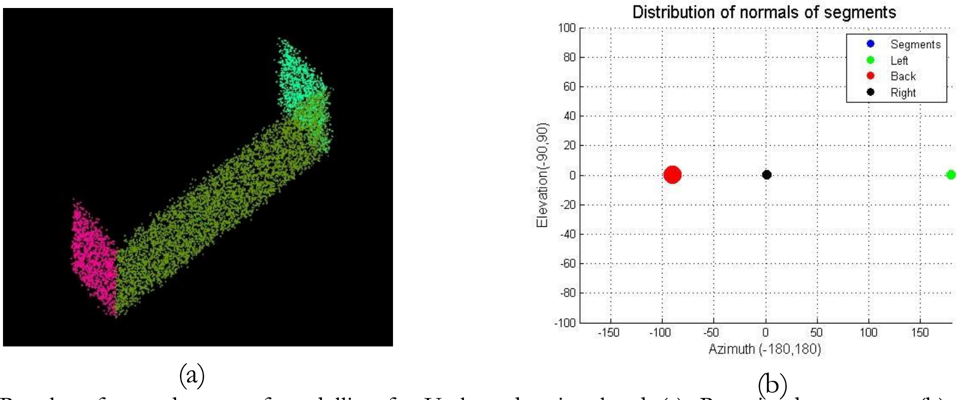


Figure 5-4: Results of second stage of modelling for U-shaped point doud, (a): Remained segments, (b): Scatter of normal vectors of remained segments

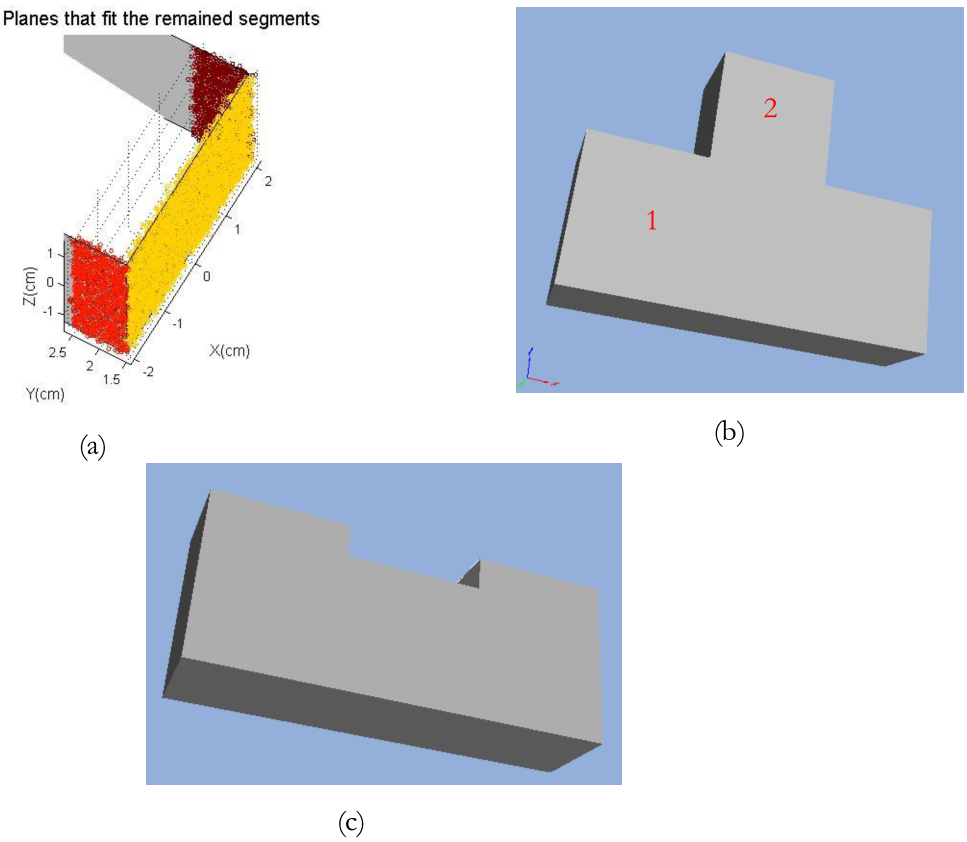


Figure 5-5: Results of second stage of modelling for U-shaped point cloud, (a): Planes fitted to remained segments, (b): The second cube on first cube, (c): Full 3D model

5.1.1.3. X-Shaped point cloud

The similar process was followed for X-shaped point cloud. In this point cloud, in fact we have 4 L-shaped structure. The first stage is the same as the others, that we get results like as …..

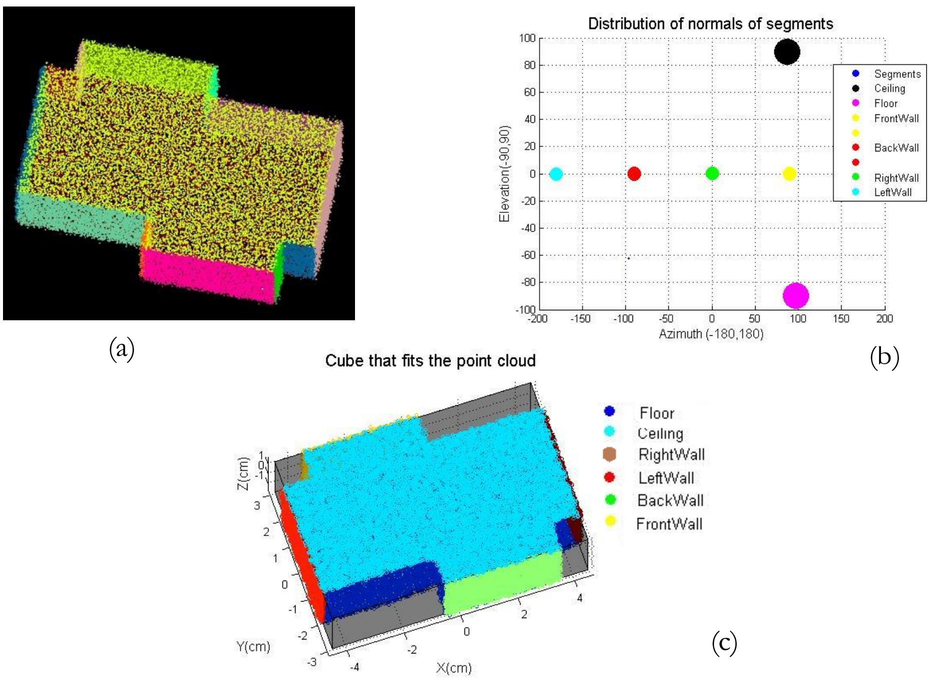


Figure 5-6: Results of first stage of modelling for X-shaped point cloud, (a): Segmentation in PCM, (b): Scatter of normal vectors of segments, (c): The largest cube fitted to point cloud

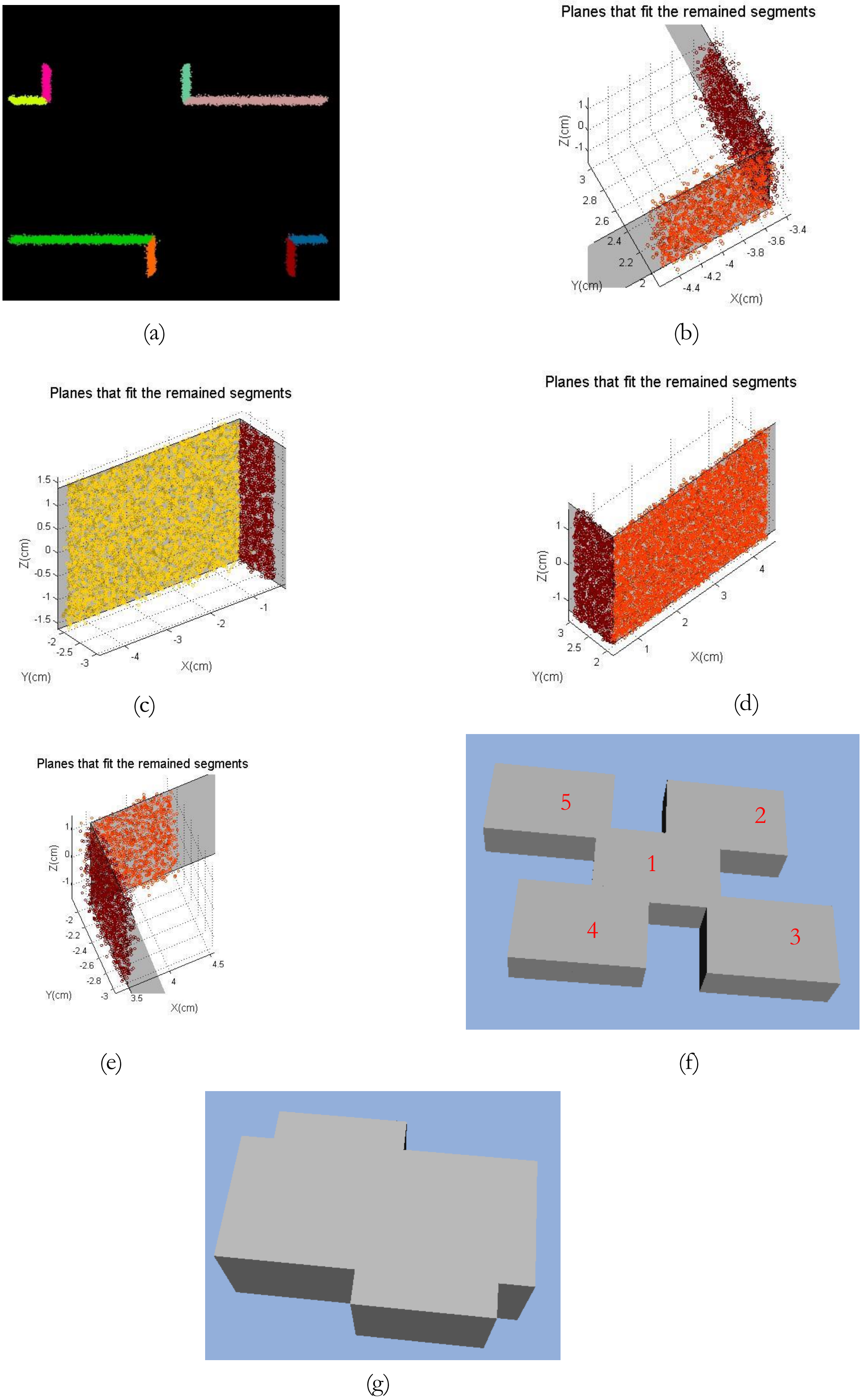


Figure 5-7: Results of second stage of modelling for X-shaped point cloud, (a): Remained segments, (b): Planes fitting to remained segments of up-left, (c): Planes fitting to remained segments of down-left, (d): Planes fitting to remained segments of up-right, (e): Planes fitting to remained segments of down-right, (f): The new cubes on first cube, (g): The full 3D model

5.1.2. Implementation on captured data

The result of surface growing segmentation on the captured point cloud, was 82 segments (Figure 5-8). The segment numbers for points were saved into .txt file to go to the next step.

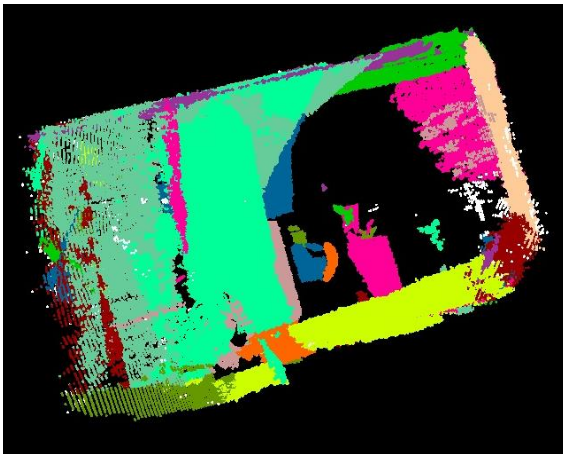


Figure 5-8: Segments of original point doud

Calculation of normal vectors for points, shows that the vectors for points of same planes are in same direction. Figure 5-9 shows this fact. Using this characteristic, we decided to classify the segments based on the verities on normal vectors. Then the normal vectors were calculated for each segments.

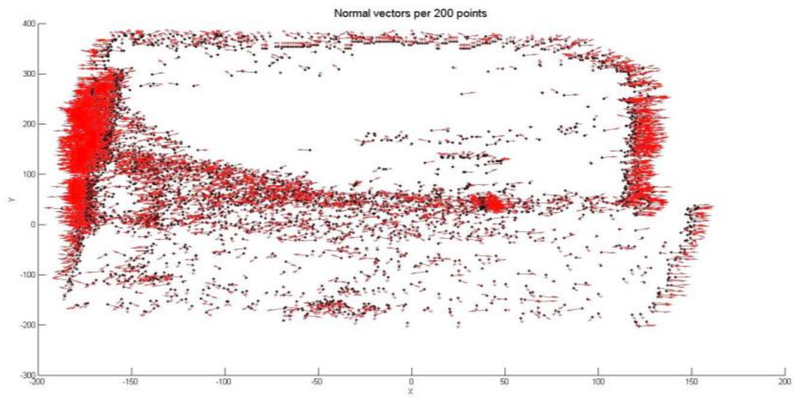


Figure 5-9: The directions of normal vectors of points (per 200 points)

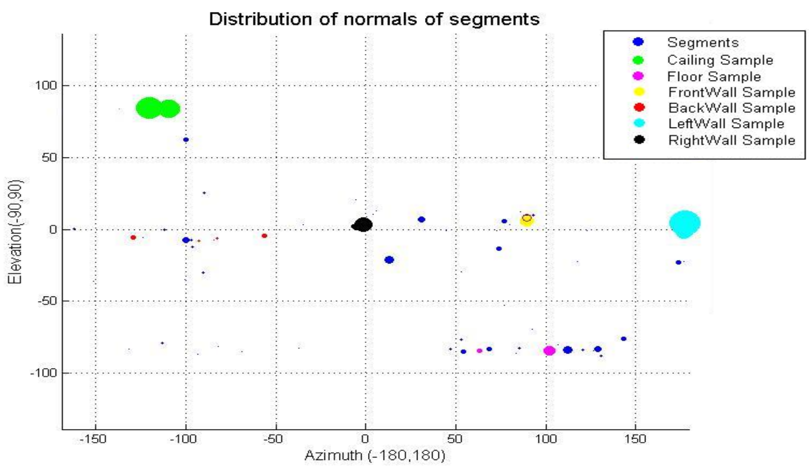


Figure 5-10: Scatter of normal vectors of segments

Classification of normal vectors were done with constraints on the elements of the vectors. These constraints also involved some uncertainties about the angles.

When the segments for each class of planes were selected, we can fit the cube to them. But we may have more than one segment in each class. So we selected the farthest segment of the origin in each class. To cover the inaccuracy of registration and sensor in dataset, the segments within the 10 cm of the farthest segment also added to the selection list.

In this stage the cube fitted to the selected segments. The result is in Figure 5-11.

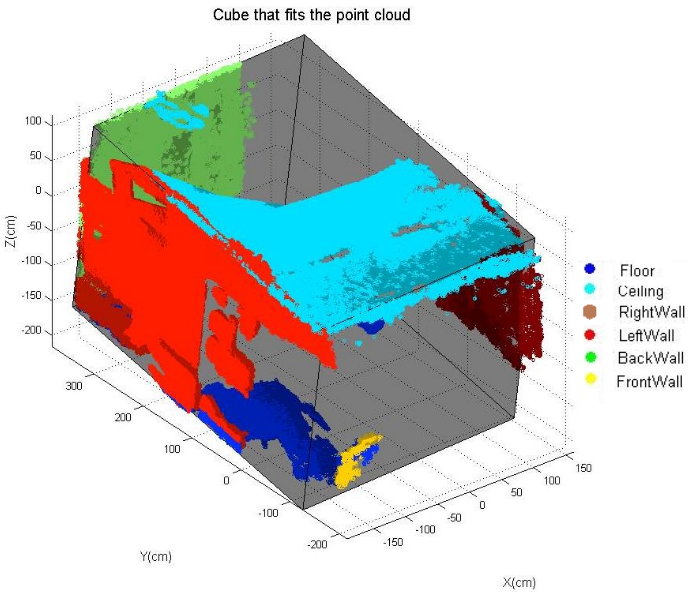
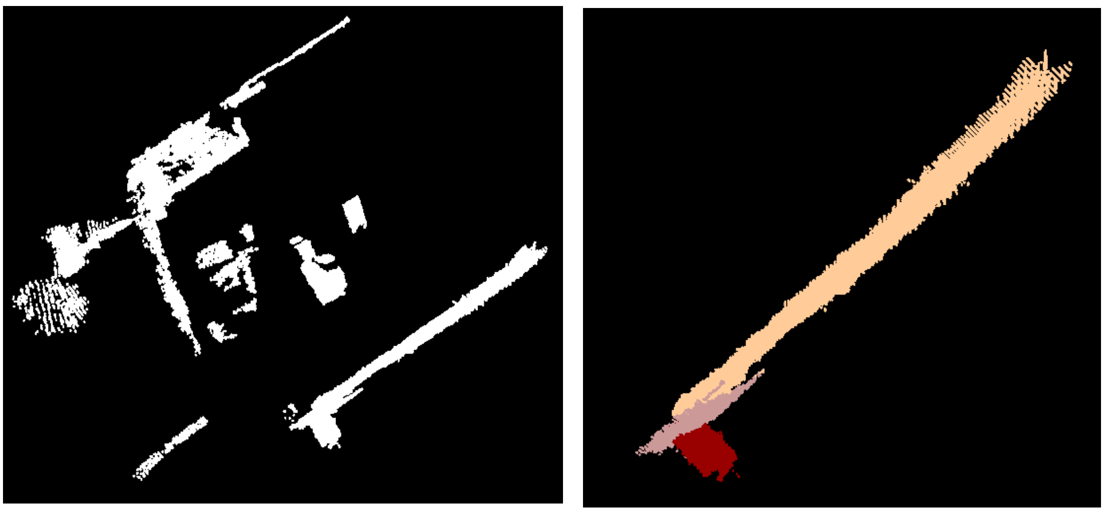


Figure 5-11: The Largest cube which fits the point cloud

In this step, we knew which segments of labelled classes are not involved in the reconstructed cube. These segments go into the PCM and the connected component segmentation is done on them. Here we can separate the favorite component which contains the remaining walls of indoor scene. After separation, we do surface growing segmentation on it to segment the surfaces.



(a) (b)

Figure 5-12: Re-segmentation of remained points, (a): Connected components, (b): Surface growing on one component

Here the normal vector for each segment is calculated. Through the classifying the normal vectors, we can label segments in four walls (if there are). Then in each class we select the farthest segment plus the segments within 10 cm of it and fit a plane to the points of selected segments with RANSAC algorithm.

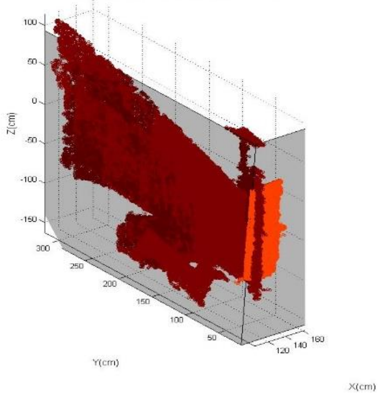


Figure 5-13: Planes of remained segments to cut the cube in extra parts

Now we can find the intersection points of wall's planes and floor/ceiling. We select the plane of two perpendicular walls (using the labels) and once intersect them with floor and once with the ceiling planes. These points are the placement of new cube. As we have only one intersection point on floor, we translate the cube with all preserved parameters to the intersection point. Figure 5-14 shows new cube on the first large cube.

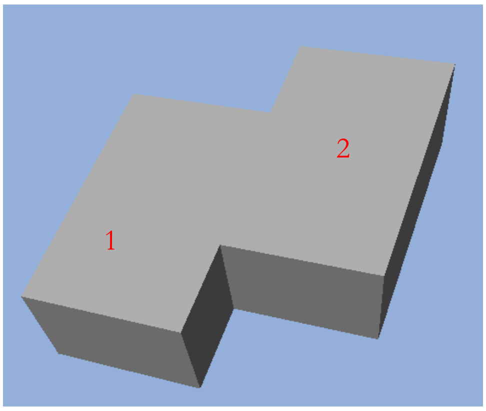


Figure 5-14: The cutting planes on the larger cube

We can subtract this new cube from the first cube and get the full 3D model. The consistent reconstructed 3D model with all details is shown in Figure 5-15.

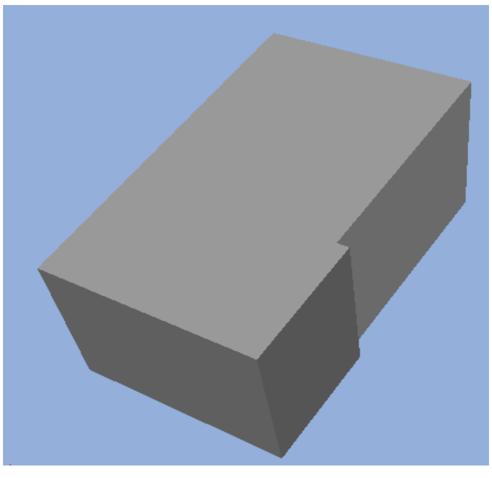


Figure 5-15: The full 3D model of indoor

5.2. Texture mapping

For generating the textures, we first used the RGB values of the point cloud; in which for the coordinates of centres of pixels in an empty image, we find three nearest neighbour from the point cloud. The average of R, G and B values of these neighbour points are assigned to the pixel. In this way we get a low resolution image with low accurate RGB values. The textured image is mapped on the faces of model using the VRML denotations. The result for one of the faces of the reconstructed model is in Figure 5-16. As the image shows, the texture is mapped on top of faces, however they are for inside of the model.

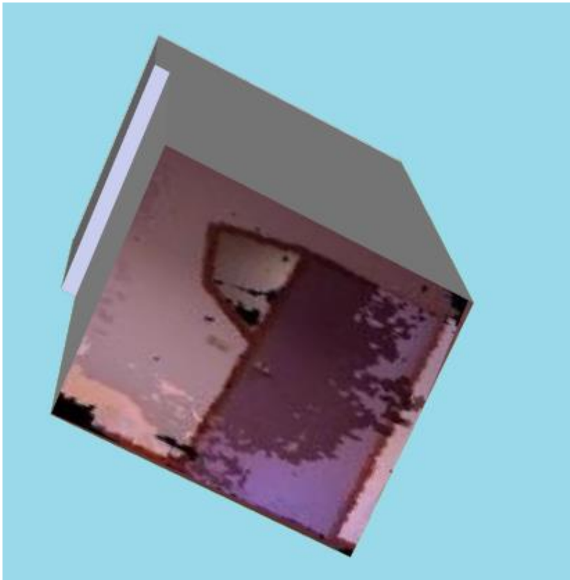


Figure 5-16: Texture mapping using the RGB values of point doud

To get better and realistic visualization, then we map the photos of digital camera on the faces in SketchUp. In this way we can put the textures inside the model. Figure 5-17 shows one view of the result of this method.

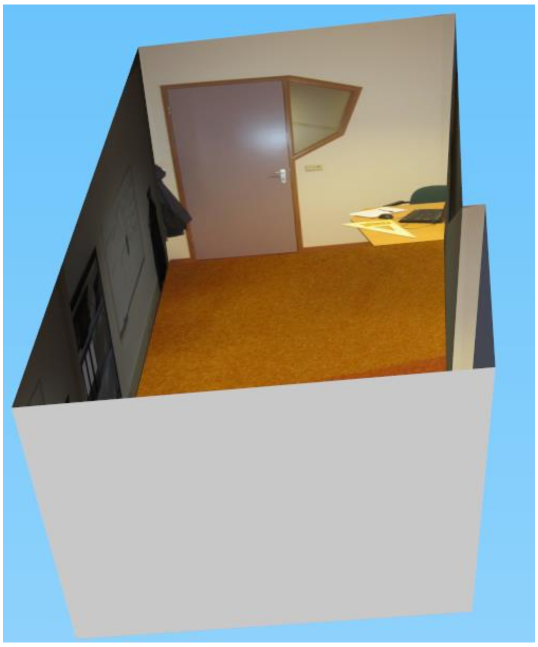


Figure 5-17: Texture mapped on model using photos taken by camera in SketchUp (ceiling off)

6. ANALYSIS

6.1. Accuracy of Model

The accuracy of reconstructed model can be discussed in two aspects. First, during the fitting the cube to point cloud, we select inlier points and then calculate the transformation matrix of unit cube to these cubes. The distances of all points of point cloud from the reconstructed cube can show the average difference of the cube and real indoor. This distance difference is shown in a histogram for the captured point cloud of the room in Figure 6-1.

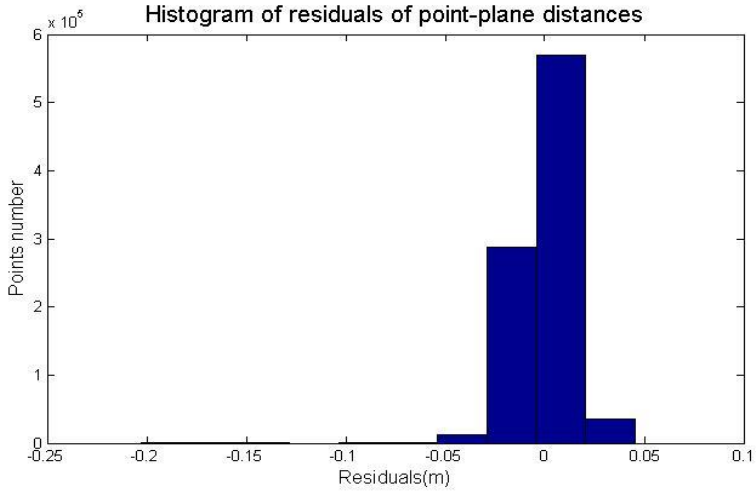


Figure 6-1: Histogram of residuals for distances between points and planes

As this histogram shows, the mode of residuals are almost zero; i.e. we don't have so much residuals and the fitted plane almost covered all the points. The remaining residuals are smaller than 5 cm, which is also smaller than the error we were expecting from registration.

Another factor in discussing about the accuracy, can be the difference between real measurements of room length, width and height and the measurements of correspondents on the model. Table 6-1 show these differences for captured dataset.

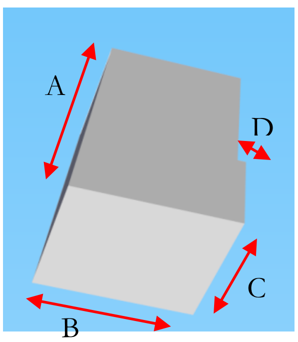


Table 6-1: Differences between room and model's length, width and height

	Cube Model (cm)	Manual measurement (cm)
A	507.8	508.0
B	327.6	320.0
C	267.9	270.0
D	17.6	33.0

As the values shows, we have small errors in the first fitted cube. The length and width and height of indoor are nearly same on the model and reality. The difference in width (B) is higher than the errors of other lengths. On the other hand the length of wall perpendicular to the right wall is very smaller than the real measure.

With these two observation we can say that in right side of the room, we have large error. This can be partly because of being texture less of the walls in this side of the room. The walls were all white, with no objects or variations on them. This were causing the registration to fail in finding matches on these walls and registration for most frames of this side of room was done manually in selecting the match points.

6.2. Acuracy of Texture mapping

In this research, we used the RGB values in the point cloud. These values are not accurate and available for whole the indoor. So the accuracy of RGB values are low, although the spatial accuracy (coordinates of centre for each pixel) are calculated accurately.

In the other method, the RGB images were taken with digital camera and photos are in high resolution, but as we map these photos manually on the face only for giving a better visualization, we can't discuss about the spatial accuracy of the textures. Because the four corners of images are assigned to corners of the faces manually.

7. CONCLUSIONS and RECOMMENDATIONS

7.1. Summary

In this research we developed a method to reconstruct a 3D model of indoors using Kinect point clouds based on Manhattan-World assumptions. The research another assumption was that the ceiling and floor only have one plane (without steps).
To reconstruct the model, first the planes were recognized using the characteristics of normal vectors and perpendicular distances of segments from the origin of coordinate system. When the planes are detected, the largest cube which could fit the whole point cloud of indoor was reconstructed. Some planes of walls were not involved in fitting the large cube. These segments were classified again in the coordinate system of origin on the centroid of remained points. Planes were fitted to the segments and intersections of these planes with floor/ ceiling plane were calculated. Then new cube can be reconstructed in these intersection points.
At the end we added textures on the faces of model. These textures once generated with RGB measurements of Kinect per point, available in point cloud file, however the accuracy of these RGB values are low. The main idea to use the colour information of Kinect can be using the RGB images taken with Kinect in each captured frame, but unfortunately we couldn't access to these images. Then we used the RGB values of point cloud to enhance the visualization of model and also to show the method of using RGB values for model. Also for realistic textures on the model, we used high resolution photos taken by ordinary digital camera.

7.2. Conclusion

According to the developed method and results of implementation, following conclusions were drawn:

- As the normal vector's elements for segments in same direction to the origin of coordinate system, are almost the same, we can classify the segments based on the constraints on these elements.
- As the origin of coordinate system is on the centroid of point cloud, the perpendicular distance of ceiling and floor segments must be half of the indoor height. With this constraint we can clean the point cloud out of the objects on the floor (desk, chair, etc.).
- One of the effective methods to model all the details of indoors is grammar based modelling, which considers one shape as primitive (cube in this research) and generates these primitives with all possible planes. Then using the subtraction or union operation on the primitives, the full model is reconstructed.
- One of the features to select proper segments for fitting cube to them, is perpendicular distance from origin, which the constraint for selection is largest distance.
- If we select the farthest segments to generate the first primitive, it will be the largest primitive and next generated primitives must be subtracted of that.
- To preserve the Manhattan-World assumption, the new reconstructed cubes in intersection points, must be in same rotation statue with the first cube. For this, the only variable parameters of transformation matrix to create a new cube are translation vector and scale factors in X and Y directions.

7.2.1. Answers to research questions

With studying the related works and the considerations on developed method, the answers for the research questions are as follow:

- Which constraints help to recognize walls, floor and ceiling points from other points in a cloud in Manhattan-World scene?

In a properly oriented point cloud, the normal vectors of planes of the indoor, are along the coordinate system axes. This can help to separate the planes with normal vectors in direction of axes, from other segments. Then for floor and ceiling, using the constraint on the perpendicular distances of segments from the origin of coordinate system, the main plans are separated from other object's planes. For walls, in the second stage of modelling, when the remained segments go into re segmentation with connected components method, the components with the wall planes are separated from other components.

- How can the model fit to all wall planes, which have steps to each other in one direction?

Using grammar based modelling methods, we can fit a primitive shape to the all existing planes in each level. Then using the subtraction or union operation on the generated primitives, the final reconstructed model contains all the existing walls.

- How well is the performance of the developed method through the experiments?

As the developed method was experimented on three different shapes of indoor point clouds, we can say that the method is valid for all Manhattan-World scenes. Also implementation on the real point cloud acquired with Kinect sensor, with large registration errors, proved that method can model details of indoor.

The residual of distances between points and fitted planes in cube fitting step, shows that the almost all the points are involved in plane fitting process, as the residuals are mostly near to zero. The only point on accuracy of resulted model, is the difference of measures on the model and on the reality for some faces; this is mainly because of registration errors on that planes.

7.2.2. Contribution

One of the main differences between Kinect data and Laser data is the accuracy of depth measurements. The lower accuracy and density of Kinect point clouds cause challenges in detecting points belonging to planes. Similar works on plane recognition have been done with laser data; e.g. (Adan and Huber, 2011; Budroni and Böhm, 2010; Jenke et al., 2009), but automation of the methods for using Kinect data in recognition and reconstruction was one of the new purposes that this research reached, despite of the low accuracy of data.

One other characteristic of Kinect datasets, is the orientation of the point clouds. Laser sensors are considered to be levelled before capturing data, so the orientation parameters are known through fixing the vertical axis of the coordinate system along the local vector of gravity (Budroni and Böhm, 2010). But as Kinect sensors are operated by hand and don't need to be levelled for data capturing, the orientation parameters are unknown. Normally, the operator tries to keep the sensor levelled, but it may have little tilts along the axis. In this project, to overcome this limitations of Kinect point clouds, we determined the approximate orientation parameters before plane recognitions. Also we used reliable confidence interval in classifying the segments based on the normal vector's elements. Once more during the fitting cube to the point cloud, accurate parameters of transformation matrix, including rotation parameters were calculated.

7.3. Recommendations

Although the developed method, could reach all the objectives of research and answer the research questions, there are still open research issues. Some recommendations for future works can be as the following:

- Extend the method to model the possible steps on the floor and false ceilings: in real architectures we almost always have steps, so the first step to improve the reconstructed model can be considering the steps on planes of floor/ ceiling.

- Extend the method to model the scenes that doesn't obey Manhattan-World assumptions: for this objective, first the non-perpendicular planes to each other must be recognized.
- Add architectural primitives (doors, windows, etc.) to the model using colour information of Kinect sensors: for adding these primitives to the model, usually openings and gaps in the point cloud are needed. So for these methods, the doors must be opened, and no curtains cover the windows. However using the colour information of Kinect sensors, all the doors without being open and windows even with curtains can be modelled. In the RGB image of Kinect, we can find these primitives and search for the shape of them into point cloud.
- Texture mapping using the RGB images of Kinect: in this research we didn't have access to these images and we used the RGB values of point cloud instead of RGB images. The accuracy of generated textures are low in this way, although the method for generating the texture image is almost the same.

LIST OF REFERENCES

Adan, A., Huber, D., 2011. 3D Reconstruction of Interior Wall Surfaces under Occlusion and Clutter, 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on, pp. 275-281.

Bae, K.-H., Lichti, D.D., 2008. A method for automated registration of unorganised point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 63(1), 36-54.

Budroni, A., Böhm, J., 2009. Toward automatic reconstruction of interiors from laser data 3D Virtual Reconstruction and Visualization of Complex Architectures/ ISPRS XXXVIII(5/W1), 7.

Budroni, A., Böhm, J., 2010. Automatic 3d modelling of indoor Manhattan-world scenes from Laser data. ISPRS Commission V Mid-Term Symposium 'Close Range Image Measurement Techniques XXXVIII(Part 5), 115-120.

Deyle, T., 2010. Low-Cost depth cameras (aka Ranging cameras or RGB-D Cameras) to Emerge in 2010?, Hizook {Robotics News for Academic & Professionals}. Hizook.

Dick, A.R., Torr, P.H.S., Cipolla, R., 2004. Modelling and Interpretation of Architecture from Several Images. International Journal of Computer Vision 60(2), 111-134.

Downing, F., Flemming, U., 1981. The bungalows of Buffalo. Environment and Planning B 8(3), 269-293.

EverybodyPlays, 2010. Countdown to Kinect - Kinect FAQ. everybodyplays.co.uk.

Flemming, U., 1986. More than the sum of parts: the grammar of Queen Anne houses. Environment and Planning B: Planning and Design 14(3), 323-350.

Goetz, M., Zipf, A., 2011. Extending OpenStreetMap to Indoor Environments: Bringing Volunteered Geographic Information to the Next Level, in: Rumor, M., Zlatanova, S., LeDoux, H. (Eds.), Urban and Regional Data Management: Udms, Delft, The Netherlands, pp. 47-58.

Google, 2011 A new frontier for Google Maps: mapping the indoors, Google Official Blog; Insights from Googlers into our products, technology and the Google culture. Google.

Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D., 2012. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. The International Journal of Robotics Research 31(5), 647-663.

Hernández-López, J.-J., Quintanilla-Olvera, A.-L., López-Ramírez, J.-L., Rangel-Butanda, F.-J., Ibarra-Manzano, M.-A., Almanza-Ojeda, D.-L., 2012. Detecting objects using color and depth segmentation with Kinect sensor. Procedia Technology 3(0), 196-204.

Hobley, S., 2010. What the Kinect sensor actually does. http://www.stephenhobley.com/blog/2010/12/04/what-the-kinect-sensor-actually-does/ (accessed 11 August).

Holz, D., Holzer, S., Rusu, R., Behnke, S., 2012. Real-Time Plane Segmentation Using RGB-D Cameras, in: Röfer, T., Mayer, N.M., Savage, J., Saranlı, U. (Eds.), RoboCup 2011: Robot Soccer World Cup XV. Springer Berlin Heidelberg, pp. 306-317.

Huber, D.F., Hebert, M., 2003. Fully automatic registration of multiple 3D data sets. Image and Vision Computing 21(7), 637-650.

ICIAR, 2013. Recent Advances on RGB-D Camera Applications. (accessed Feb. 5, 2013).

ifixit, 2010. Microsoft Kinect Teardown. http://www.ifixit.com/Teardown/Microsoft+Kinect+Teardown (accessed Feb. 5, 2013).

Jenke, P., Huhle, B., Straßer, W., 2009. Statistical Reconstruction of Indoor Scenes, in: Chen, M., Skala, V. (Eds.), The 17th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision; WSCG. University of West Bohemia Plzen, Czech Republic pp. 17-24.

Karnick, P., Jeschke, S., Cline, D., A., R., Wentz, E., Wonka, P., 2009. A Shape Grammar for Developing Glyph-based Visualizations. Computer Graphics Forum 28, 2176-2188.

Kashlev, D., 2008. Efficient 3D building model generation from 2D floor plans, Department of Electrical engineering and computer scince. Massachusetts Institute of Technology, p. 87.

Khoshelham, K., 2010. Automated localization of a laser scanner in indoor environments using planar objects, Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on, pp. 1-7.

Khoshelham, K., 2011. Accuaracy analysis of Kinect depth data, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS) 2011 Workshop. ISPRS Calgary , Canada, pp. 133-138.

Khoshelham, K., Elberink, S.O., 2012. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. Sensors 12(2), 1437-1454.
Makadia, A., Patterson, A.I., Daniilidis, K., 2006. Fully Automatic Registration of 3D Point Clouds, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, pp. 1297-1304.
Microsoft, 2010. PrimeSense Supplies 3-D-Sensing Technology to “Project Natal” for Xbox 360. Microsoft Press.
Müller, P., Wonka, P., Haegler, S., Ulmer, A., Gool, L.V., 2006. Procedural modeling of buildings. ACM Trans. Graph. 25(3), 614-623.
Müller, P., Zeng, G., Wonka, P., Gool, L.V., 2007. Image-based Procedural Modeling of Facades. ACM Transactions on Graphics 26(85).
Pulli, K., 1999. Multiview registration for large data sets, 3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on, pp. 160-168.
Rosser, J., Morley, J., Jackson, M., 2012. Crowdsourcing of building interior models in: Gensel, J., Josselin, D., Vandenbroucke, D. (Eds.), International Conference on Geographic Information Science AGILE / Multidisciplinary Research on Geographical Information in Europe and Beyond, Avignon, pp. 130-134.
Stiny, G., Gips, J., 1971. Shape Grammars and the Generative Specification of Painting and Sculpture, The Best Computer Papers of 1971 O.R. Petrocell ed, Auerbach, Philadelphia, pp. 125-135.
Thrun, S., Martin, C., Yufeng, L., Hahnel, D., Emery-Montemerlo, R., Chakrabarti, D., Burgard, W., 2004. A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. Robotics and Automation, IEEE Transactions on 20(3), 433-443.
Vanegas, C.A., Aliaga, D.G., Benes, x, B., 2010. Building reconstruction using manhattan-world grammars, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 358-365.
Vosselman, G., 2009. Advanced point cloud processing, in: Fritsch, D. (Ed.), Photogrammetric Week 2009. Institute for Photogrammetry, University of Stuttgart, Institute for Photogrammetry, Stuttgart, pp. 137-146.
Xiao, J., Furukawa, Y., 2012. Reconstructing the World’s Museums, in: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (Eds.), Computer Vision – ECCV 2012. Springer Berlin Heidelberg, pp. 668-681.
Yang, M.Y., Förstner, W., 2010. Plane Detection in Point Cloud Data, Technical Report, TR-IGG-P-2010-01. Department of Photogrammetry, Institute of Geodesy and Geoinformation, University of Bonn, Germany.
Zhang, L., Wang, Y., Shi, H., Zhang, L., 2012. Modeling and analyzing 3D complex building interiors for effective evacuation simulations. Fire Safety Journal 53(0), 1-12.