Towards Automated Floorplan Generation

Kunal Chelani
Indian Institute of Science
Bengaluru, India
kunalchelani@iisc.ac.in

Chitturi Sidhartha
Indian Institute of Science
Bengaluru, India
chitturis@iisc.ac.in

Venu Madhav Govindu
Indian Institute of Science
Bengaluru, India
venug@iisc.ac.in

ABSTRACT

In this paper, we propose a pipeline for generating a 2D floorplan using depth cameras. In our pipeline we use an existing approach to recovering the camera motion trajectories from the depth and RGB sequences. Given these motion estimates we construct a full 3D representation of the scanned indoor spaces. For generating a floorplan we need to abstract the large volumes of registered 3D data into a simplified rectilinear representation representation. We evaluate two approaches to solve this problem, i.e. slicing the reconstructed volume at a given height and direct segmentation of the 3D point cloud representation into individual planar segments. We also note that the fidelity of our estimated floorplan crucially depends on the accuracy of the estimation of the ground plane orientation. We examine the comparative accuracies of two ground plane estimation methods for each of the above mentioned approaches to rectilinear abstraction. Given the line drawing abstractions of the individual rooms, we merge them into a consistent floorplan. We present results on a real-world floorplan estimation and demonstrate its accuracy. Additionally, the implications of errors in the individual components of our pipeline are also studied.

CCS CONCEPTS

• **Computing methodologies → Computer vision**.

ACM Reference Format:
Kunal Chelani, Chitturi Sidhartha, and Venu Madhav Govindu. 2018. Towards Automated Floorplan Generation. In *Proceedings of 11th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'18),* Anoop M. Namboodiri, Vineeth Balasubramanian, Amit Roy-Chowdhury, and Guido Gerig (Eds.). ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.475/978_1

1 INTRODUCTION

The availability of cheap depth sensors such as the Kinect has lead to the development of a range of new approaches and applications. While a number of successful approaches to general 3D scene reconstruction have been developed, there has also been interest in solving specific problems in an indoor setting [14, 32]. A problem of interest in this context is that of automatically constructing a floor plan from the depth information obtained from depth sensors. Extracting a floor plan from a 3D reconstruction involves the identification of the appropriate linear structures that correspond to walls, vertical partitions etc. However, any successful pipeline for floor plan extraction needs to solve a number of sub-problems, i.e. accurate 3D registration of the interiors of rooms and spaces, the relative alignment of individual rooms or spaces and abstraction of the large volumes of 3D data into an accurate rectilinear line drawing representation of the physical space. While the planar (or linear) representations of the walls, partitions etc. can be exploited for accurate estimation, there are a number of inherent problems and ambiguities that need to be addressed. In this paper, we develop a complete pipeline for generating an accurate floor plan using scans obtained from a depth sensor.

The rest of the paper is organized as follows: we discuss the related work in Section 2 and the proposed pipeline in detailed in Section 3. In Section 4, we characterize the uncertainty in estimation in individual components of the pipeline. Finally, we present and discuss a floorplan reconstruction using our pipeline in Section 5 and present some conclusions in Section 6.

2 RELATED WORK

While the body of literature on 3D reconstruction using depth cameras is voluminous, in this section, we confine ourselves to a brief review of the literature relevant to our problem of extracting a rectilinear floorplan representation from indoor depth scan sequences.

Camera Motion Estimation: Estimation of the motion trajectory for depth cameras has attracted a lot of attention. KinectFusion [14] does a real-time reconstruction of a scene or an object with a Kinect sensor using enhanced graphics hardware. It fuses the depth data into a surface representation using signed distance functions [7] and tracks the camera poses using the iterative closest point (ICP) algorithm [25]. However, this suffers from the limitation of operating only in small workspaces. A spatially extended KinectFusion namely Kintinuous [31] was proposed by Whelan *et al.* [31] to cater to larger environments using triangular mesh representation of the environment and it uses pose-graph optimisation to minimize the drift. ElasticFusion [32] reconstructs dense surfel-based maps of the environments and deploys surface loop closure optimisations to alleviate the drift problem. In addition to these reconstruction methods, there have been learning based approaches developed in recent years. The most notable of these approaches is Floornet [19] which proposes a 3-branched deep neural network architecture to get a good quality reconstruction and extract the floorplan.

An indoor scene reconstruction pipeline was proposed by Choi *et al.* [6] by registering scene fragments and performing a robust global optimisation using line processes to account for misalignments caused due to erroneous sensors. [1, 27] also attempt to solve

ICVGIP'18, Dec. 2018, Hyderabad, India
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6615-1.
https://doi.org/10.475/978_1

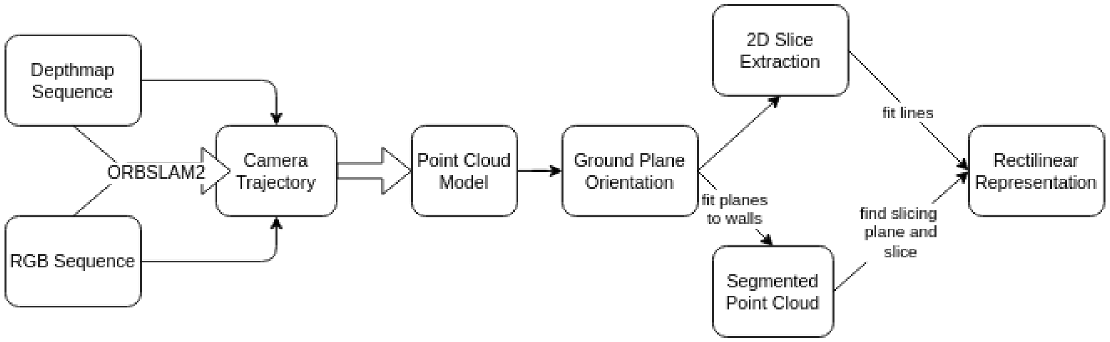

Figure 1: Flow diagram of our floorplan generation pipeline. Please see Section 3 for details of individual components.

indoor reconstruction problem as a global registration problem based on low-rank sparse(LRS) decomposition and by exploiting the loop structures within the point clouds respectively.

There are several Simultaneous Localization and Mapping (SLAM) based approaches to tackle the indoor reconstruction problem. A recent survey is available in [3]. Here, we limit our discussion to RGB-D SLAM approaches. Endres *et al.* [10] proposed a RGB-D SLAM system which uses RANSAC to estimate the motion between the matched features and performs nonlinear pose-graph optimization to generate a map of the environment. The drift error in the reconstruction gets accumulated with every camera and thus for example, in a room environment, one may not find the loop closure resulting in a bad reconstructions. Hence, identifying and correcting for loop closures is crucial to getting a good indoor 3D reconstruction. Many works [6, 13, 28] incorporate loop-closures by adding an edge in the pose-graph and optimizing the whole graph to minimize the global drift. [29] proposed a hierarchical approach for pose-graph optimization to solve the registration problem. It uses a loop-based incremental registration algorithm to refine the edges and then distribute the errors over the global graph. Kerl *et al.* [17] proposed a dense visual SLAM which uses an entropy-based similarity measure to select keyframes and detect loop closures. The pose-graph thus constructed is optimized to give a reduced photometric and depth error. ORB-SLAM2 [23] proposes a versatile SLAM system that works for a whole range of environments. It uses ORB features [26] for tracking, mapping, and place-recognition. The back-end is based on bundle adjustment which helps in accurate camera trajectory estimation and the localization is achieved using either the visual odometry tracks for unmapped regions and map point matches to obtain zero-drift localization.

Planar Segmentation: With respect to planar segmentation based approaches, Gee *et al.* [12] and Chekhlov *et al.* [5] attempted to solve real-time SLAM problem using planes by deploying Kalman-filter-based systems. [21] exploits the information in the keyframes for local tracking and also using the global plane model of the environment to reduce drift by performing global graph optimization. In Dou *et al.* [8], plane correspondences are used along with the traditional features in RGB images and the cost function in the bundle adjustment is modified so as to include planar surface alignment errors in addition to the 3D reprojection errors.

Bhattacharya *et al.* [2] proposed a fast method for 3D Registration by making use of planar structures in the scene. The planar segmentation discussed in this paper exploits the fact that the disparity map and the image points corresponding to a planar region follow an affine relationship. [21] exploits the information in the keyframes for local tracking and also reduces drift by making use of the global plane model of the environment to reduce drift by performing global graph optimization in an EM framework.

Recently, Lee *et al.* [18] proposed a method to jointly estimate scene layout and perform registration to get accurate indoor reconstruction. Initially, they do a partial reconstruction using KinectFusion to get scene fragments and perform global graph optimization for solving the registration problem whose solution may not be accurate. Then they produce plane hypotheses from *supervoxels*, cluster them and extract the dominant planes using hierarchical agglomerative clustering, then use the weak Manhattan world assumption to estimate the layout and finally global registration is solved by formulating an objective function constrained by the layout. The layout estimation and the registration are solved iteratively instead of solving them jointly to reduce the complexity.

3 PROPOSED PIPELINE

In this section, we describe the tasks involved in the proposed pipeline for obtaining a floorplan with dimensions using a Kinect sensor and minimal manual intervention. We also detail the methods used in individual components of the pipeline. Extraction of the rectilinear representation of each room is undertaken separately and later merged to obtain the final floorplan.

3.1 Data Acquisition

We use a Kinect v1 sensor to capture both RGB and depthmap images at the rate of 30 frames per second to maximize our channels of information. We initially point the camera to the ground for reasons explained in 3.3. Being available at low cost and easy to use with commodity graphics hardware, the Kinect depth sensor can be easily used for the task even if the method is deployed for commercial preparation of floorplans or indoor measurements.

3.2 Estimating Camera Motions

To obtain accurate 2D details, such as the shape and measurements from a 3D model of a room, it is essential to first obtain an accurate 3D model and hence accurate camera pose estimation is the most crucial step in the pipeline as it forms the base for other techniques to operate upon. The room scanning can be done in a smooth, controlled motion, providing us access to the sequential information about the frames captured and hence allowing us to pose this as a SLAM problem. A variety of methods have been suggested in the past two decades for solving the camera motion estimation in SLAM.

For the purpose of room-sized indoor environments, ElasticFusion by Whelan et al. [33] shows promising results, but the randomized fern encoding based local and global closure detection techniques seem to fail on our data and result in reconstructions with unaligned floor plane and non perpendicular walls. Also, the requirement of extremely slow scanning and the problem of failed reconstructions in case of sparse 3D features prevent us from successfully applying the map optimization based SLAM solver.

To the best of our knowledge, the state-of-the-art RGB-D SLAM system at the time of writing this piece is ORBSLAM2 by Mur-Artal and Tardós [23]. Qualitatively, the point-cloud obtained by back-projecting the pixels to their corresponding depths using the camera trajectory obtained looks natural and the place recognition module based on DBoW2 [11] incorporated in ORBSLAM2 seems to do the task really well by adjusting the global keyframe graph once we scan the entire room and return to the initial position. For these reasons, we rely on ORBSLAM2 as an out of the box module for obtaining camera trajectories. We also characterize the level of uncertainties introduced in physical measurements by using this system in Section 4.3.

3.3 Obtaining Floorspace Measurements

We assume that the walls are perpendicular to the ground and to each other. To obtain the room measurements, the idea is to slice either the 3D pointcloud itself or a cuboid-like representation of it, using a suitable plane and then obtain the rectilinear outline of the room or scanned spaces. We first describe the two methods in Sections 3.3.1 and 3.3.2, assuming that we already have the required slicing plane and then in Section 3.3.3, we discuss the ways of finding that required plane.

3.3.1 Slicing Volumetric Representations. The first method is to slice the reconstruction with a plane parallel to the floor plane, resulting in line-like representation of each wall with the noise level depending on the distance of sensor from the wall while scanning, the material of the wall etc.

To obtain the dimensions, a rectangular representation of the room is required. Various methods such as [9, 15] are available to cluster data drawn from low-dimensional affine subspaces embedded in a high dimensional space. But for our task, these appeared to be an overkill. Also they take time in order of minutes/hours to cluster four lines and the faster approximate versions are inconsistent in terms of the results produced. Robust line fitting using RANSAC with unbiased sampling does the job well enough for our purposes. We fit a minimum area bounding rectangle as suggested in [24] to the intersection points of the fitted lines to obtain the required rectangular representation. Fig 2 shows a strip obtained after sectioning the point cloud representation and Fig 3 shows the corresponding fitted lines.

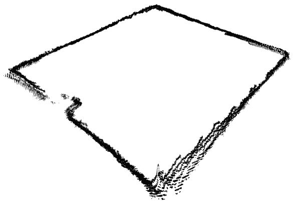

Figure 2: Strip after sectioning

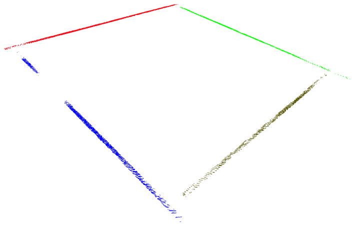

Figure 3: Fitted Lines

3.3.2 Extracting 3D Planar Structures. Unlike the above method of fitting lines to a slice of points, a more intuitive approach would be to fit planes to the vertical walls and instead of obtaining a 2D rectilinear representation, we get a cuboid like volumetric representation. Since the planes that we obtain aren't perfectly perpendicular to each other, we can't directly find distances between them for the room measurements. We either need to approximate the planes as perfectly parallel to obtain measurements, or slice it with a plane parallel to the ground. To fit planes to the walls, we again find RANSAC with a direction prior to perform really well. We provide the floor normal as the prior direction to look for the planes. Fig 6 shows the obtained segmented planes after RANSAC plane fitting.

3.3.3 Finding the slicing plane. To obtain the slicing plane, it is intuitive to look for a plane that is perpendicular to the floor plane. We use the technique suggested by Chatterjee *et al.* in [4] for plane detection from disparity map images to obtain the ground plane which is captured in the first frame while scanning the rooms. This is an extremely cheap method in terms of the compute required and is fairly accurate too. The same thing can also be done by fitting a plane through the actual 3D points belonging to the floor plane (the points need to be obtained manually from points corresponding to intial few frames)

But the drawback with these approaches is that we have about 8000 frames per room and keyframes in order of 100s. The ORB-SLAM2 technique, although performs well on loop closue to avoid additive errors but it isn't perfect and even a slight misalignment of the floorplanes on loop completion, can cause the slicing plane to shift considerably. In the case of approach mentioned in Section

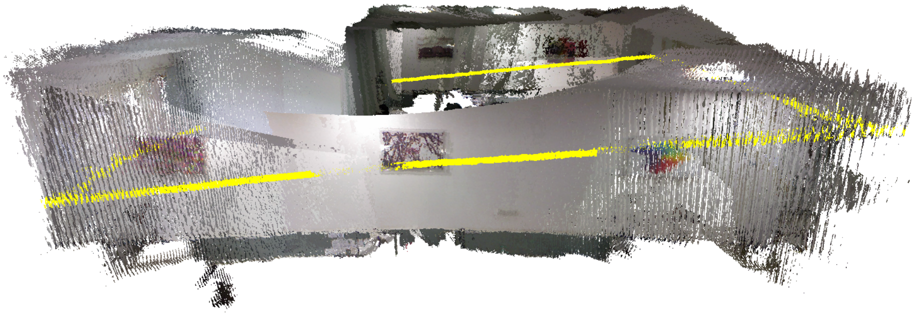

Figure 4: Incorrect ground plane

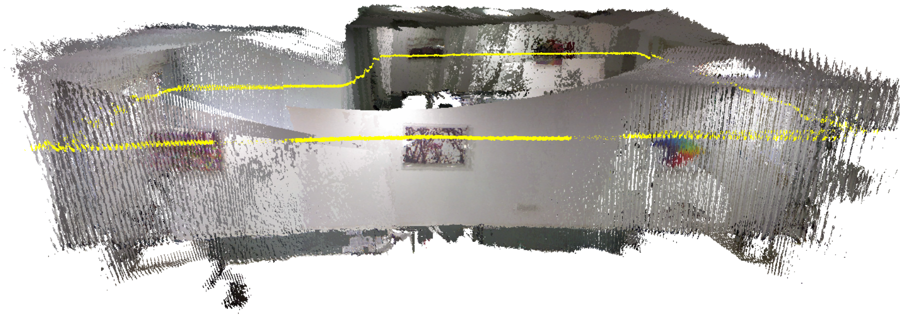

Figure 5: Proper ground plane

3.3.1, we do not have much sense of direction as we have to directly slice the 3D pointcloud and hence we rely on the approach of using the floor plane obtained by using the method in [4].

Let $aX+bY+cZ+d = 0$ be the representation of the floor plane in the first frame (also the global frame of reference). Then any plane parallel to this can be written as $aX + bY + cZ + d' = 0$ where the distance between this and the ground plane is $\frac{d'-d}{\sqrt{a^2+b^2+c^2}}$. We need the representation of a plane at a height h, parallel to the ground. Since the camera is above the ground, we can say that out of the two planes at distance h from the ground plane we want the one that is closer to the origin. In the representation we use, the equation is normalized so as to make $d = 1$. Hence $d' = 1 - h * \sqrt{a^2 + b^2 + c^2}$.

In case of Section3.3.2, we have more information. We actually have the planar representation of the walls and we can have a more principled way of finding the right plane to slice the planes fitted to the walls. In theory we would want to find a plane that is perpendicular to each of the obtained wall-planes, but that won't necessarily be true due to noise. Hence as an approximation, we calculate the vector that is most perpendicular to all the wall normals in some sense. Let N_1 through N_4 be the unit normal vectors to the walls W_1 through W_4. Now, we would want a vector N that minimizes:

$$f = (N^T N_1)^2 + (N^T N_2)^2 + (N^T N_3)^2 + (N^T N_4)^2 \qquad (1)$$

Hence, we use the least singular vector of $(N_1N_1^T + N_2N_2^T + N_3N_3^T + N_4N_4^T)$ as the normal for the required slicing plane. We can also consider other pairs of walls and if there's much difference we can take a mean of these vectors, in the sense that the mean vector should minimize the sum of squared angles with each of the participating vectors.

3.4 Door Detection

Door detection in rooms using a rgb images or an RGB-D reconstruction has its own challenges. [16] propose a technique for doorway detection, which works for only open doors and and for most doors in our data, the doors were auto-close and could not held open for an extended duration of scanning. Since using just RGB information doesn't help much in door detection where doors are painted with same color as the room, we manually label doors for rooms where we could not keep the doors open. For other rooms, we just use the slice obtained using 3.3.1 to look for discontinuities. Appropriate height is chosen for such slicing and a min threshold is chosen for the width of discontinuity to prevent false positives.

3.5 Room Arrangement and Floorplan Generation

An obvious limitation of our approach is that a continuous scan of the entire floorspace would not work well. This is because of the lack of connecting features as we move from one room to the other resulting in disconnected components of the entire scene. This limitation results in the need to take individual scans of rooms, extract rectilinear outline of each room (called a rectangular box from here on) as explained in Section 3.3, and arrange the rooms so

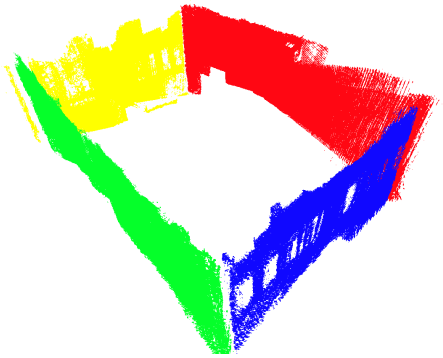

Figure 6: Segmented Planes

as to create an actual floorplan. This problem can be now posed as a two dimensional bin packing problem. For a survey of the problem and proposed solutions, please refer to [20] and [22]. However, there is an inherent ambiguity in the solution to the problem, where the rooms can be swapped in a cyclic fashion, resulting in multiple optimal solutions.

In this work, we follow a different approach to this problem where we take information about the adjacency relationships of the rooms as an additional input to resolve the aforementioned ambiguities. After obtaining the 2-D rectangular boxes from Section 3.3 each of which are in different frames of reference, we rotate them appropriately such that they align with the canonical X, Y axes in a global frame of reference. Now, the arrangement problem could be thought of as translating the centers of each room appropriately to generate the floorplan. This could be formulated as the following optimization problem

$$\begin{aligned} &\text{minimize} \sum_{(i,j,p)\in\mathcal{E}} ||C_i - C_j||^2 \; + \lambda(var(Y_U) \; + var(Y_L)) \\ &\text{subject to} \quad (C_i - C_j)\cdot p \geq (d_{ip} + d_{jp})/2 \quad \text{for all } (i,j,p) \in \mathcal{E} \end{aligned}$$

where $\mathcal{E}$ is the set of adjacency relationships of the form (i, j, p) where i, j are the indices of the rooms that are adjacent along the axis p ($p = (1,0)$ and $(0,1)$ for adjacency along X-axis and Y-axis respectively), C_i is the center of the i^{th} room, d_{ip} is the dimension of the i^{th} room along the axis p, λ is the regularization factor, Y_U and Y_L are the two sets of y-coordinates of the walls of the rooms that share the two common edges of the floor. For example, rooms 1, 2, 3, 4, 5, 9, 12, 13, 14 share the common top edge, and rooms 1, 2, 3, 4, 5, 6, 7, 10, 11, 13, 14 share the common bottom edge of the floor in Figure 9. $\lambda = 420$ gives the result as shown in Figure 10. The set of transformations thus obtained to bring rectangular boxes from individual scans to the result in Figure 10 are applied to the strips obtained in Section 3.3 to yield the result as shown in Figure 11.

4 CHARACTERIZING UNCERTAINTIES

Before we present our results, we wish to assess the uncertainties in estimates at different stages in the pipeline which affect the resulting dimensions of the rooms.

4.1 Kinect Error Characterization

The Kinect's uncertainty is characterized as explained in [4]. The relation between depth (Z) and disparity (D) is given as

$$\frac{\partial Z}{\partial D} = -\frac{Z^2}{fB}$$

where f is the focal length of the depth sensor in Kinect, B is the baseline. We assume that the average distance (depth) we measure using a Kinect for our reconstruction is 5 metres. Using the above relation for $Z = 5$ and the fact that Kinect's uncertainty in computing the disparity map is $1/8$ th of a pixel, we calculate the overall uncertainty as

$$\frac{\partial Z}{\partial D} = -\frac{5000^2}{75 * 587} = 567mm$$

$$\partial Z = 567/8 = 70.875mm$$

4.2 Plane Estimation Algorithm

In order to test the validity of the ground plane representation obtained using the method of [4], we take a depth map containing a plane and perform Bootstrapping *i.e,*, pick a subset of pixels randomly corresponding to the same plane as outputted by our Plane segmentation algorithm and estimate the plane parameters corresponding to those pixels. This experiment is performed several times and the average of the errors in angle between each normal and the mean normal is calculated and we observed that the average and the standard deviation were calculated to be 0.66° and 0.37°

4.3 ORBSLAM2 uncertainty characterization

An experiment was conducted to calculate the uncertainty introduced by ORB-SLAM. In this experiment, a sequence of frames containing different regions of the same plane are captured and using the Planar Segmentation idea of Chatterjee *et al.* [4], the normal to the plane was calculated for all the frames. The normals in the first frame and the last frame capturing the same plane were brought to the same coordinate system by applying suitable transformations with the relative motion corresponding to each camera obtained from the ORBSLAM2 reconstruction. Then the angle between the two normals (θ) is the error produced due to ORBSLAM2 and is calculated to be 66.2°. The error could also be introduced due to improper plane parameter estimation or improper plane segmentation but we verified that our plane parameter estimation is correct through the experiment in Section 4.2 and manually checked that the segmented planes were correct. The mean of the deviation (δ) between normal in each frame and the mean normal is calculated to be 10.85° and the standard deviation to be 3.58°. The distribution of the normals is shown in Figure 7.

5 RESULTS

Having listed the various sources of uncertainty in our pipeline, we show the results of our pipeline on a real world dataset. For our experiment, we build a complete floorplan for the floor wing in which our lab is located. We collected the data for all 14 rooms and enclosures in our floor wing. For evaluating the accuracy of our floorplan reconstruction, we obtain the ground truth by careful measurement using a measuring tape.

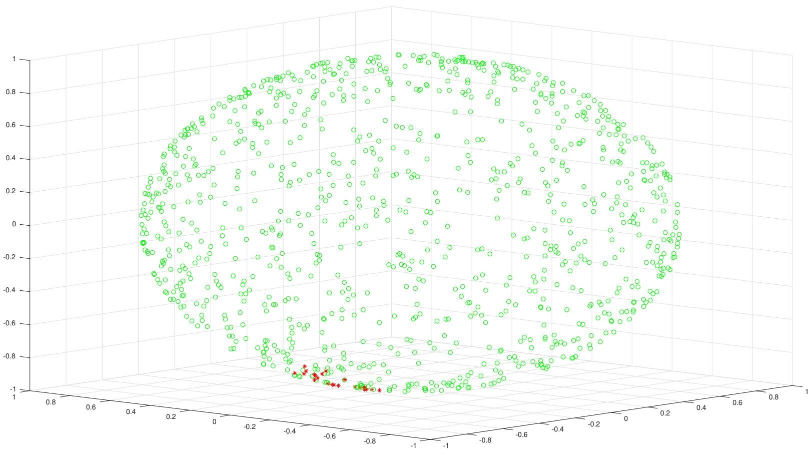

Figure 7: Normals of the plane(shown in red, points in green correspond to the unit sphere) obtained using the experiment conducted in Section 4.3

Room	Ground Truth				Ours - using Section 3.3.1				Ours - using Section 3.3.2			
	Length	Width	Aspect Ratio	Area	Length	Width	Aspect Ratio	Area	Length	Width	Aspect Ratio	Area
1.	14.15	9.85	1.43	139.38	13.97	10.09	1.38	140.96	14.06	9.99	1.41	140.46
2.	6.85	9.95	0.69	68.15	7	10.01	0.70	70.07	6.99	9.93	0.70	69.41
3.	6.85	9.95	0.69	68.15	6.9	9.94	0.69	68.59	6.94	9.98	0.70	69.26
4.	3.31	9.95	0.33	32.93	3.3	9.86	0.33	32.54	3.11	9.72	0.32	30.23
5.	3.5	9.95	0.35	33.33	3.54	9.96	0.36	35.26	3.42	9.9	0.35	33.86
6.	3.45	3.48	0.99	12.00	3.41	3.5	0.97	11.94	3.11	3.41	0.91	10.61
7.	3.43	8.21	0.42	28.16	3.42	8.29	0.41	28.35	3.53	8.23	0.43	29.05
8.	3.45	2.95	1.16	10.18	3.42	2.91	1.18	9.95	3.38	2.92	1.16	9.87
9.	3.45	3.32	1.04	11.45	3.41	3.23	1.06	11.01	3.37	3.33	1.01	11.22
10.	7.05	5.85	1.20	41.24	7.03	5.75	1.22	40.42	6.97	5.73	1.22	39.94
11.	7.2	6.7	1.07	48.24	7.12	6.65	1.07	47.35	7.09	6.71	1.06	47.57
12.	7.2	3.25	2.21	23.4	7.24	3.29	2.20	23.82	7.14	3.13	2.28	22.35
13.	7.025	9.95	0.70	69.9	6.84	10.01	0.68	68.47	6.83	9.95	0.69	67.96
14.	3.55	9.95	0.35	35.32	3.54	9.83	0.36	34.80	3.46	9.87	0.35	34.15

Table 1: Dimensions - Ours and Ground Truth. The room number has been mentioned as per the floorplan figure

We found out that the dimensions we obtained for our rooms were scaled by a factor in the range of 1.02 – 1.04. Our observation is consistent with that reported in [30]. In their experiments, they found that the depth estimates obtained from Kinect were scaled by a factor of 1.03. Therefore, we re-scale the results obtained by a constant value 1.03.

5.1 Error Analysis

Table 1 shows the ground truth and the results obtained using the two methods mentioned in Sections 3.3.1 and 3.3.2. The sectioning plane used for obtaining the results of Section 3.3.1 is obtained using the method suggested in [4]. In some cases, we notice that the ground plane, although locally correct, does not take into account the overall drift resulting from the ORBSLAM2 camera motion estimation and is a poor estimate of what would be a correct sectioning plane for the corresponding room. For this reason and to check the effects of obtaining the correct sectioning plane, we manually choose points in the pointcloud that would correspond to a correct sectioning plane. For such a choice we obtain results as per Table 2 which are much more consistent and accurate. Hence we notice that the sectioning plane selection is important for accurate measurements (Fig 4 and 5).

Hence we expect the results in Table 1 corresponding to Section 3.3.2 to be quite accurate since the sectioning plane is chosen to

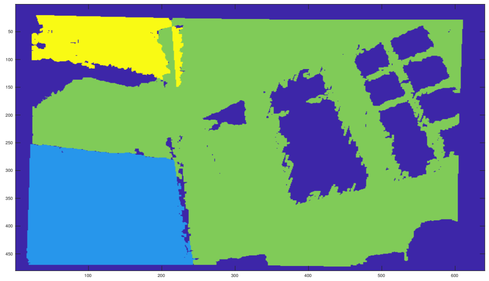

Figure 8: Planes obtained from depth map

Table 2: Dimensions - Ours using manual ground plane selection. The room number has been mentioned as per the floorplan figure

Room	Ours - Manual Ground Plane Selection			
	Length	Width	Aspect Ratio	Area
1.	14.21	9.88	1.43	140.39
2.	7	9.9	0.70	69.3
3.	7.05	9.95	0.70	70.14
4.	3.34	9.87	0.34	32.97
5.	3.5	9.97	0.35	34.89
6.	3.45	3.48	0.99	12.00
7.	3.45	8.24	0.42	28.42
8.	3.44	2.92	1.17	10.04
9.	3.41	3.25	1.05	11.08
10.	7.03	5.77	1.21	40.56
11.	7.18	6.6	1.08	47.39
12.	7.17	3.26	2.2	23.37
13.	6.83	10.05	0.68	68.64
14. 2	3.48	9.82	0.35	34.17

be perpendicular to the 4 four wall normals. However, for rooms 6 and 12, we notice the values to be far from the ground truth. On investigation, we find that the wall normals obtained for these rooms are not vertical for some walls due to extreme clutter and very less visible wall area. Hence, we observe that for this method to perform well, we must obtain well conditioned planes that fit the wall structures.

6 FUTURE WORK AND CONCLUSIONS

Commodity depth cameras have been playing a huge role on revolutionizing robotics and automating related tasks. Floorplan generation is one more such area that we believe can be tackled using these sensors. In this paper, we propose a pipeline for generating floorplans using RGB-D sensors, analyze the uncertainties involved with the methods used and finally show that the pipeline produces extremely consistent and accurate results as far as the dimensions are concerned. Although this approach tries to move towards an automated pipeline for generating floorplans, certain steps of the pipeline require human intervention. For example, the height of the slicing plane used in 3.3.1 is to be decided based on the room and the clutter, requiring manual choosing. Method proposed in 3.3.2 requires the walls to contribute heavily to become a dominant plane in the specified direction. In certain cases, clutter such as cupboards and other structures having same planar inclinations as the walls add up to the noise in this regard. The issue of room alignment arises if the observations are made one room at a time and any solution is bound to be ambiguous. Hence, human intervention is again required at this step in the form of providing adjacency relationship between rooms. A fully automatic system should be capable of operating on data from scratch and producing a floorplan as an output. This requires the addressing of the problem of obtaining accurate camera trajectories as we move from one room (enclosure) to the other, possibly, by using information from sensors other than cameras. We hope that our work can be built upon by us and others, in future, to develop more efficient and robust systems for floorplan production.

REFERENCES

[1] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. 2016. Global Registration of 3D Point Sets via LRS Decomposition. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 489–504.

[2] Uttaran Bhattacharya, Sumit Veerawal, and Venu Madhav Govindu. 2017. Fast Multiview 3D Scan Registration Using Planar Structures. In *3D Vision (3DV), 2017 International Conference on*. IEEE, 548–556.

[3] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics* 32, 6 (2016), 1309–1332.

[4] Avishek Chatterjee and Venu Madhav Govindu. 2015. Noise in Structured-Light Stereo Depth Cameras: Modeling and its Applications. *CoRR* abs/1505.01936 (2015). arXiv:1505.01936 http://arxiv.org/abs/1505.01936

[5] Denis Chekhlov, Andrew P Gee, Andrew Calway, and Walterio Mayol-Cuevas. 2007. Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 1–4.

[6] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. 2015. Robust Reconstruction of Indoor Scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 303–312.

[8] Mingsong Dou, Li Guan, Jan-Michael Frahm, and Henry Fuchs. 2012. Exploring high-level plane primitives for indoor 3D reconstruction with a hand-held RGB-D camera. In *Asian Conference on Computer Vision*. Springer, 94–108.

[9] E. Elhamifar and R. Vidal. 2009. Sparse subspace clustering. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2790–2797. https://doi.org/10.1109/CVPR.2009.5206547

[10] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 2014. 3-D mapping with an RGB-D camera. *IEEE Transactions on Robotics* 30, 1 (2014), 177–187.

[11] Dorian Gálvez-López and J. D. Tardós. 2012. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics* 28, 5 (October 2012), 1188–1197. https://doi.org/10.1109/TRO.2012.2197158

[12] Andrew P Gee, Denis Chekhlov, Walterio W Mayol-Cuevas, and Andrew Calway. 2007. Discovering Planes and Collapsing the State Space in Visual SLAM.. In *BMVC*. 1–10.

[13] Peter Henry, Dieter Fox, Achintya Bhowmik, and Rajiv Mongia. 2013. Patch volumes: Segmentation-based consistent mapping with RGB-D cameras. In *3D Vision-3DV 2013, 2013 International Conference on*. IEEE, 398–405.

[14] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 559–568. https://doi.org/10.1145/2047196.2047270

[15] S. Jain and V. M. Govindu. 2013. Efficient Higher-Order Clustering on the Grassmann Manifold. In *2013 IEEE International Conference on Computer Vision*. 3511–3518. https://doi.org/10.1109/ICCV.2013.436

[16] B. Kakillioglu, K. Ozcan, and S. Velipasalar. 2016. Doorway detection for autonomous indoor navigation of unmanned vehicles. In *2016 IEEE International*

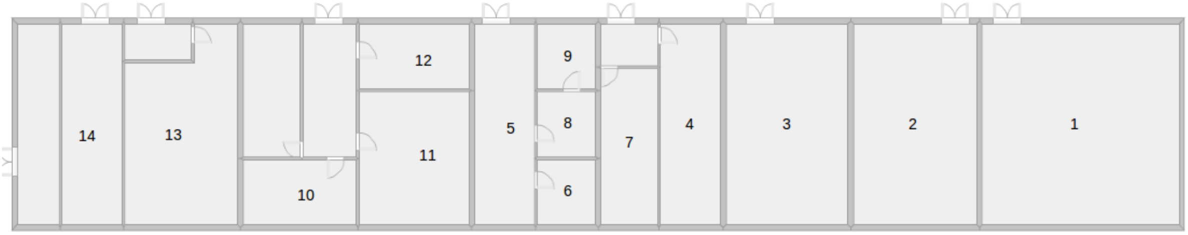

Figure 9: Floorplan obtained from architectural drawing

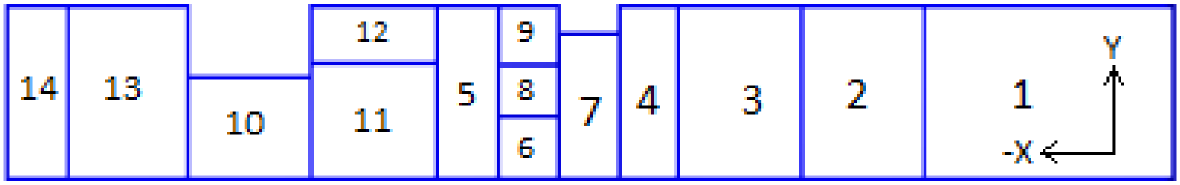

Figure 10: Room arrangement obtained by solving the optimization in Section 3.5

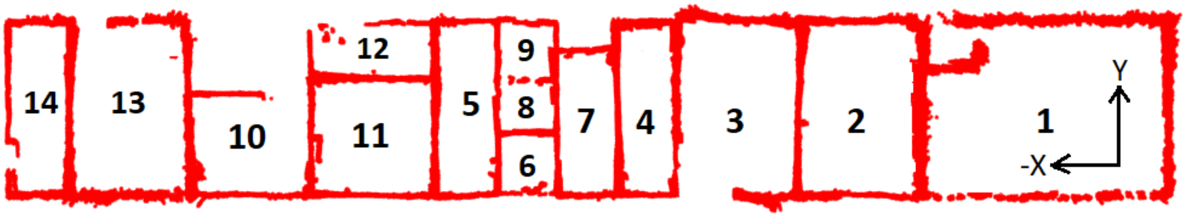

Figure 11: Floorplan constructed using the obtained dimensions.

Conference on Image Processing (ICIP). 3837–3841. https://doi.org/10.1109/ICIP.2016.7533078

[17] Christian Kerl, Jurgen Sturm, and Daniel Cremers. 2013. Dense visual SLAM for RGB-D cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. Citeseer, 2100–2106.

[18] Jeong-Kyun Lee, Jae-Won Yea, Min-Gyu Park, and Kuk-Jin Yoon. 2017. Joint Layout Estimation and Global Multi-View Registration for Indoor Reconstruction. *arXiv preprint arXiv:1704.07632* (2017).

[19] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. 2018. FloorNet: A Unified Framework for Floorplan Reconstruction from 3D Scans. *CoRR* abs/1804.00090 (2018). arXiv:1804.00090 http://arxiv.org/abs/1804.00090

[20] Andrea Lodi, Silvano Martello, and Michele Monaci. 2002. Two-dimensional packing problems: A survey. *European Journal of Operational Research* 141 (2002), 241–252.

[21] Lingni Ma, Christian Kerl, Jörg Stückler, and Daniel Cremers. 2016. Cpa-slam: Consistent plane-model alignment for direct rgb-d slam. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 1285–1291.

[22] N. Ma and Z. Zhou. 2017. Mixed-Integer Programming Model for Two-Dimensional Non-Guillotine Bin Packing Problem with Free Rotation. In *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. 456–460. https://doi.org/10.1109/ICISCE.2017.102

[23] Raul Mur-Artal and Juan D. Tardós. 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics* 33 (2017), 1255–1262.

[24] Joseph O'Rourke. 1985. Finding minimal enclosing boxes. *International Journal of Computer & Information Sciences* 14, 3 (01 Jun 1985), 183–199. https://doi.org/10.1007/BF00991005

[25] Neil D. McKay Paul J. Besl. 1992. Method for registration of 3-D shapes. (1992), 1611 - 1611 - 21 pages. https://doi.org/10.1117/12.57955

[26] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, 2564–2571.

[27] T. Shiratori, J. Berclaz, M. Harville, C. Shah, T. Li, Y. Matsushita, and S. Shiller. 2015. Efficient Large-Scale Point Cloud Registration Using Loop Closures. In *2015 International Conference on 3D Vision*. 232–240. https://doi.org/10.1109/3DV.2015.33

[28] Frank Steinbrucker, Christian Kerl, and Daniel Cremers. 2013. Large-scale multi-resolution surface reconstruction from RGB-D sequences. In *Proceedings of the IEEE International Conference on Computer Vision*. 3264–3271.

[29] Yizhi Tang and Jieqing Feng. 2015. Hierarchical multiview rigid registration. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 77–87.

[30] TUM. [n. d.]. https://vision.in.tum.de/data/datasets/rgbd-dataset/file_formats. ([n. d.]). Access = 15/08/2018.

[31] KinectFusion Thomas Whelan, Michael Kaess, Maurice F. Fallon, Hordur Johannsson, John J. Leonard, and John McDonald. 2012. Kintinuous : Spatially Extended KinectFusion.

[32] Thomas Whelan, Stefan Leutenegger, Renato Salas Moreno, Ben Glocker, and Andrew Davison. 2015. ElasticFusion: Dense SLAM Without A Pose Graph. (07 2015).

[33] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. 2016. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research* 35, 14 (2016), 1697–1716. https://doi.org/10.1177/0278364916669237