

Planar Modeling of Indoor Scenes using RGBD Images

Shreya Tripathi

IIIT-D-MTech-CS-GEN-14-024

Indraprastha Institute of Information Technology Delhi
New Delhi

Thesis Advisor Dr. Saket Anand

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science

Keywords : RGB-D Kinect Video, 3D Scene Reconstruction

Certificate

This is to certify that the thesis titled “**Planar Modeling of Indoor Scenes using RGBD Images**” submitted by **Shreya Tripathi** for the partial fulfillment of the requirements for the degree of *Master of Technology in Computer Science & Engineering* is a record of the bonafide work carried out by him under our guidance and supervision at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Saket Anand

Indraprastha Institute of Information Technology, Delhi

Abstract

3D registration of scenes is a widely researched topic. Since the availability of low cost sensors like the Kinect RGB-D cameras, they are being extensively used for additional depth information about the scene. Most of the approaches for registration either use point clouds or RGB features to estimate the relative pose between the consecutive views. Our work focuses on using the planar patches of the scenes as primitives for indoor scene registration and modeling. Indoor scenes are structured and can be easily represented using planes. The advantage of using planes is the reduced space complexity from tens of thousands of points to much fewer planes which are present in the scene. Currently ICP is one of the most popular algorithm for scene registration. It uses 3D point clouds for transformation estimation. But the major drawback of the algorithm is that it uses dense points for estimation iteratively which is computationally expensive. Also, ICP requires an initial transform estimate which is used in subsequent iterations which makes it dependent on initial parameters. The proposed algorithm aims to address these issues by using planes for estimation of optimal rotation and translation. General bundle adjustment algorithms use 3D reprojection error between points as the cost function which makes it an expensive step in registration process. We again make use of the planar primitives to reduce the complexity of this step. Finally a 3D model of an indoor scene is obtained from the RGBD images.

Acknowledgments

Nearing the end of the M.tech journey, I would like to thank my parents for their support in every turn. I would like to specially thank my thesis advisor Dr. Saket Anad who helped give direction with his invaluable guidance and suggestions. He helped in every possible way whenever I got stuck or had doubts. His words of encouragement mean a lot. This section can not be complete without a vote of thanks to academic department for their help and never ending support.

Contents

1	Introduction	2
1.1	Overview and Research Motivation	2
1.2	Literature Review	3
2	Preliminaries	6
2.1	Sensor Noise Model	6
2.2	Planar Scene Modeling	7
2.2.1	Normal Computation	8
2.2.2	Model Discovery	8
2.2.3	Model Association	10
2.3	Iterative Closest Point	11
2.3.1	Correspondence Estimation	12
2.3.2	Correspondence Rejection	12
2.3.3	Transformation Estimation	13
2.3.4	Final Algorithm	15
2.4	Bundle Adjustment	15
3	Proposed Algorithm	17
3.1	Boundary Classification	17
3.2	Planar Registration	18
3.2.1	Assumptions	18
3.2.2	Correspondence Estimation	19
3.2.3	Correspondence Rejection	19
3.2.4	Transformation Estimation	20
3.2.5	Final Algorithm	22
3.3	Revised Bundle Adjustment	22
4	Results and Conclusion	24

4.1	Results	24
4.1.1	Boundary Classification	24
4.1.2	Scene Recontruction	25
4.2	Limitations	35
4.3	Conclusion and Future Work	35

List of Figures

2.1	Block diagram [1]	7
2.2	Surfaces before transformation	12
2.3	Tangent distance between point and plane	12
2.4	Bundle adjustment using point data [19]	15
4.1	Boundary Classification Results for NYU dataset	25
4.2	Corner reconstruction of NYU Kitchen	26
4.3	Corner reconstruction of NYU Living Room - 1	27
4.4	Corner reconstruction of NYU Living Room - 2	28
4.5	Corner reconstruction of NYU Living Room - 3	29
4.6	Corner reconstruction of NYU Living Room - 4	30
4.7	Corner reconstruction of NYU Office	31
4.8	Corner reconstruction of NYU Bedroom - 1	32
4.9	Corner reconstruction of NYU Bedroom - 2	33
4.10	Reconstructon of Swarath Lab	34

Chapter 1

Introduction

1.1 Overview and Research Motivation

Planar modeling of a scene is an important area of research with applications like scene understanding, reconstruction, etc. The aim of this work is to use the planar models extracted from RGBD images to address two important problems namely, boundary classification and 3D indoor scene reconstruction.

Object boundaries are important for interpretation of a scene. Generally two boundaries can be classified into two categories : a) occluding boundaries which result from an object occluding other objects like a sofa in front of a wall and b) intersecting boundaries where the objects are intersecting like the intersection of walls. Previous work [17] in the area used color and depth cues as features for a learning algorithm. Such an approach needs prior a training set. We have used the depth information and the noise model of the sensor to classify the edges without the need of a learning algorithm i.e. we only use the geometric cues for the labeling problem.

Automatic 3D indoor scene reconstruction is a long researched problem. Over the course of years, 3D mapping has been confined to registration using point clouds. This can be attributed to the fact that the raw data by most sensors is in the form of point clouds. But they have severe limitations and there is a need for a more robust and efficient way for representation of a model for mapping. The two main limitations of a point based mapping algorithm are the time

and space complexity of the algorithm and the need of an initial transform between the views.

In this work, high level primitives of the scenes are used to model the individual views of the scene. These planes are then merged using the proposed algorithm to get the final scene model.

The key contributions of the research are

- Using the planar information for scene modeling.
- Formulation of heteroscedestic distance function for plane correspondence which takes the sensor noise into account.
- Reducing the time and space complexity of the registration algorithm by using high level representation.
- Revised bundle adjustment to make use of the planar information of the views instead of points.
- Boundary classification without the need of training classifiers using only geometric primitives.

1.2 Literature Review

Major work in using planar patches for registration has been done by Pathak *et al.* [25] [24] [23] [4]. They have worked on extraction and registration of planar segments from range images for unstructured environments. Their proposed algorithm MUMC(Minimally Uncertain Maximal Consensus) [24] determines the unknown plane correspondences by maximizing geometric consistency by minimizing the uncertainty volume in configuration space. The algorithm takes into account information such as size of plane, covariance of normals, etc to get the accurate correspondences. But the drawback of the algorithm is that it needs odometry information for few tests which determine the correspondences.

Many of the proposed algorithms using plane patches have also leveraged the low level RGB features to find the relative pose between the views. Dou *et al.* [7] have used SIFT features

along with planar distances to find the planar correspondences. Relative plane angle and plane appearance similarity are used to formulate the distance measure. Plane appearance is taken into account by joint histogram of hue-saturation calculated from RGB channel. Louren *et al.* [27] use both depth and color information for extracting planes from the scene, and the relative pose estimation between consecutive frames is cast as a plane registration problem. In the absence of the minimum number of required planes, 2D point correspondences are extracted for finding the remaining degrees of freedom.

Weingarten and Siegwart [34] use the planar patches as features for the Extended Kalman Filter to get the relative pose of camera in the global frame. They still use point-cloud based ICP to help in registration and correspondence determination. Trevor *et al.* [29] also use a similar technique. The difference between the two approaches is the use of graph based SLAM and the additional use of 2D laser data instead of RGB images.

Kohlhepp *et al.* [18] [9] use surface representation of outdoor environments. They use neuro fuzzy rules to estimate the correspondences which are then used for calculation of relative transformations. Viejo *et al.* [32] use small local patches to iteratively estimate the transformation between views from semi structured environments. He *et al.* [12] propose the concept of Complete Planar Patches(CPP) which are plane based features according to the visibility of the plane. If the plane is unoccluded, then it is considered as CPP. Then an interpretation tree is created which is used to find the pose estimate.

In case of boundary classification, work done till now uses either just the depth data or depth data combined with the color information as features for learning algorithms. Jia *et al.* [17] use depth and color information to train a Support Vector Regression for local prediction of edges. Then a Conditional Random Field (CRF) is used for a joint inference. They have not used the noise characteristics of the sensors and have instead leveraged the color information for the same. Gupta *et al.* [11] have also used similar approach of using hierarchical learning model using additional features like surface normals and depth gradients.

Hoiem *et al.* [13] [14] have used prior knowledge from the scene such as sky, etc for recovery of occlusion boundaries. But such cues cannot always be used for classification as most of the

scenes are either unstructured or devoid of them. We have used depth and noise model to classify the boundaries without the need of such information.

Chapter 2

Preliminaries

2.1 Sensor Noise Model

For any depth sensor, the noise in the data is dependent on the true measurement of depth i.e. the depth is heteroscedastic. For effectively modeling the distance functions in planar modeling and registration we need an estimation of the noise in the data. In case of Kinect, the depth sensor computes the depth by calculating the disparity between the image points. If d_i is the disparity and z_i is the depth of a point, then the relation between depth and disparity is given by

$$z_i = \frac{f \cdot b}{d_i} \quad (2.1)$$

where f is the focal length of the camera and b is the baseline.

As the depth is heteroscedastic, the noise is assumed to be zero mean Gaussian noise with variance being dependent on the depth. So the depth at each point is given by

$$z_i = z_{io} + G(0, \sigma_z(z_{io})) \quad (2.2)$$

where $G(\mu, \sigma)$ is the Gaussian noise with mean μ and σ standard deviation. $\sigma_z(z_{io})$ is the depth

dependent standard deviation and can be written as

$$\sigma_z(z_{io}) = kz^2 \quad (2.3)$$

Here, k is the unknown parameter which is computed by analyzing the residuals after fitting planes to the 3D data. To estimate k , planar segments are extracted from the 3D points using RANSAC. For each pixel lying in the plane, a 3×3 region is selected around it and the mean depth $\hat{z}_i$ is computed. Since the noise is zero mean noise, the $\hat{z}_i$ will be the estimator of actual mean z_{io} . The parameter k is then computed by

$$k = \sqrt{\frac{1}{n} \sum_i \left(\frac{r_i}{\hat{z}_i^2} \right)^2} \quad (2.4)$$

where r_i is the residual given by $r_i = \hat{z}_i - z_i$.

2.2 Planar Scene Modeling

The aim is to find an efficient representation of the scene using planes. The work of Anand [1] is used to extract the planes from a scene and create a 3D model from a single image. The block diagram of the algorithm is shown in Figure 2.1.

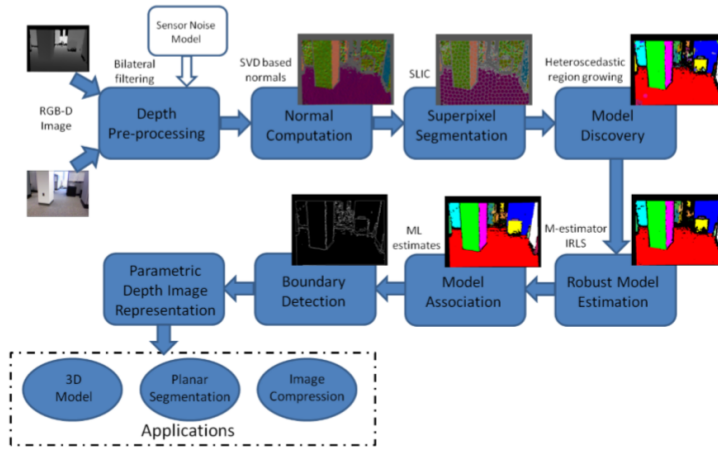


Figure 2.1: Block diagram [1]

The algorithm takes RGB and the depth images as input along with a noise model parameter. The aim of this step is identification of planar surfaces which best describe the scene. The major constraints in planar model is the complexity of scene. Presence of clutter in the scene can lead to wrong estimation of planar segments. In that case, the clutter should be classified as outlier. To take such issues into account, heteroscedastic distance measure is formulated which takes into account the noise model of the scene for model combination. The steps of the algorithm are described below in detail.

2.2.1 Normal Computation

Surface normals are estimated at each pixel using SVD. The SVD of a 9×9 window gives the eigen vectors of the covariance matrix of points. Singular vector corresponding to the smallest singular value gives the surface normal of patch.

Since these normals are estimated over a small neighborhood, they are susceptible to noise at depth discontinuities and at non planar regions. To alleviate this problem superpixel segmentation is done in the next step.

2.2.2 Model Discovery

The aim of this step is the identification of minimal set of planar primitives. This is done in two steps

1. Superpixel segmentation

Superpixels are generated by oversegmenting the surface normal image. The basic assumption here is that the pixels belonging to the same plane will lie in the same superpixel. To generate the superpixels, Simple Linear Iterative Clustering (SLIC) is applied. It performs a k means clustering using the following weighted distance function

$$d_{SLIC} = d_f + w d_{sp} \quad (2.5)$$

Here, d_f is the distance in surface normal space and d_{sp} is the spatial distance in image

plane. The parameter w determines the tradeoff between the feature boundaries and superpixel compactness.

2. Merging of superpixels

The generated superpixels show the small planar regions in the image. In order to reduce the number of models in the scene, superpixels that lie in the same plane are merged. In this step, these superpixels are merged together by using Region Growing algorithm on an undirected graph created from the superpixels based on parameters like centroid, depth dependent covariance, etc.

The vertices of the graph $G_s = (V_s, E_s)$ are the superpixels which are linked to their adjacent superpixels by an edge. These nodes are characterized by the centroid of the superpixel, depth dependent covariance, surface normal of the 3D points and the covariance of the normals. The region growing algorithm groups the superpixels based on heteroscedastic planar distance which takes into account the noise in the data.

Heteroscedastic planar distance is given by

$$d_H(S_i, S_j) = \frac{d_{ij}}{\sigma_{d_{ij}}} \quad (2.6)$$

where d_{ij} is the squared symmetric planar distance and $\sigma_{d_{ij}}$ is the standard deviation of the d_{ij} which is small only if both the superpixels lie on the same plane.

$$d_{ij} = (\hat{\mathbf{n}}_i^T \Delta \hat{\mathbf{p}}_{ij})^2 + (\hat{\mathbf{n}}_j^T \Delta \hat{\mathbf{p}}_{ij})^2 \quad (2.7)$$

where $\Delta \hat{\mathbf{p}}_{ij} = \hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j$. This distance function d_H is less sensitive to noise in depth and orientations of the surface normals.

The result of this step is a set of fragments which are obtained by merging superpixels with high probability of being in the same plane.

2.2.3 Model Association

In the previous step, fragments were formed by merging the superpixels. The processing was done at a coarser level leading to coarse boundaries. In this step, model association is done at a finer level. Pixels are assigned to fragments and overfragmented regions are merged.

To perform pixel level association, maximum likelihood assignment is done by minimizing two distance functions. The heteroscedastic form of distance of point to plane in the direction of incident rays from camera is used instead of directional residual which is more insensitive to orientation of planes. For a point p_k for k^{th} pixel, the distance with respect to i^{th} plane is given by

$$\begin{aligned} r_{ik} &= \frac{\hat{\mathbf{n}}_i^T (\hat{\mathbf{p}}_i - \mathbf{p}_k)}{\cos \theta_{ik}} \\ \sigma_{r_{ik}} &= \sigma_{z_k} \frac{\|\mathbf{p}_k\|}{z_k} \\ \rho_{ik} &= \frac{r_{ik}}{\sigma_{r_{ik}}} \end{aligned} \quad (2.8)$$

To classify the pixels as inliers / outliers with respect to the model, the threshold for the i_{th} fragment is set as

$$\rho_i = \max_{k \in P_i} \rho_{ik} \quad (2.9)$$

where P_i is the set of pixels in fragment F_i .

The other distance function is the heteroscedastic cosine distance between the plane normal $\hat{\mathbf{n}}_i$ having covariance $\mathbf{C}_{\hat{\mathbf{n}}_i}$ and the normal at k_{th} pixel.

$$\phi_{ik} = \frac{1 - \hat{\mathbf{n}}_i^T \mathbf{n}_k}{\sqrt{\mathbf{n}_k^T \mathbf{C}_{\hat{\mathbf{n}}_i} \mathbf{n}_k}} \quad (2.10)$$

The heteroscedastic distance uses the orientation of normal to find the distance while being insensitive to measurement noise. Similarly, the threshold for inlier/outlier classification is given as

$$\phi_i = \max_{k \in P_i} \phi_{ik} \quad (2.11)$$

If any pixel has ρ_{ik} and ϕ_{ik} less than the thresholds calculated in equations 2.9 and 2.11, then it is considered as an inlier. Using this logic, the model association is done as a negative log likelihood by the combination of the two distance functions

$$L_k = \underset{i=1..|F|}{\operatorname{argmin}} \rho_{ik} + \phi_{ik} \quad \forall k \quad (2.12)$$

The noise in the depth is assumed to have a Gaussian distribution with zero mean. A log likelihood of the multivariate normal distribution gives the Mahalanobis distance. The negative square of the this distance gives the likelihood of the pixel $\mathbf{p}_k$ of belonging to a plane P_i . Thus the minimization function in Equation 2.12 is equivalent to the negative log likelihood assignment.

2.3 Iterative Closest Point

The ICP was independently proposed by Besl *et al.* [3], Zhang *et al.* [35] and Chen *et al.* [5] although the name ICP was suggested by the latter. In practice the acronym ICP is now used to describe algorithms that follow an Iterative Corresponding Point framework, rather than the original closest point approach. The Iterative Corresponding Point framework is an iterative two step process where the first step is estimation of closest point correspondences and the second step is computation of transformation based on the point correspondences. The original algorithm had just the two above listed steps. Later on it was modified by to include correspondence rejection to remove the outliers which lead to wrong estimation. Research on the topic continued which led to many variants of ICP each trying to optimize and refine the results. The three main stages in the algorithm are -

- Correspondence Estimation
- Correspondence Rejection
- Transformation Estimation

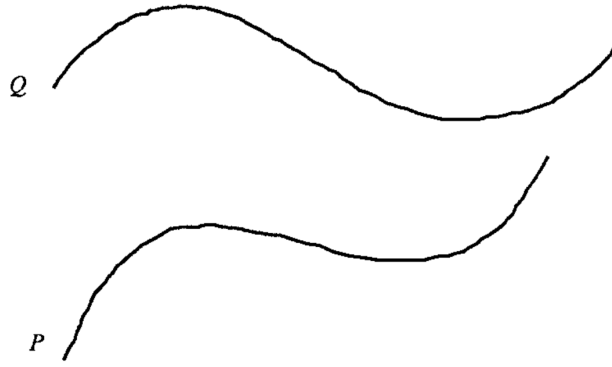


Figure 2.2: Surfaces before transformation

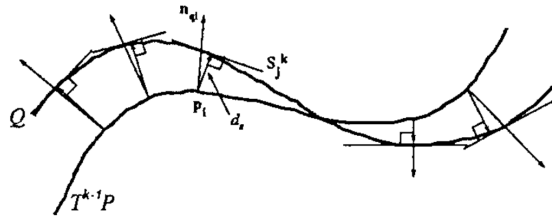


Figure 2.3: Tangent distance between point and plane

2.3.1 Correspondence Estimation

Given a point $p_i \in P$, the goal of this step is to find the corresponding point $q \in Q$. This is not a straightforward problem. If the matching points are found out by taking the distances between all the points in the two views the complexity of the algorithm will be $O(n^2)$ where n is the number of points which is usually in the orders of 10K.

The approach is to then approximate Q using its tangent plane S_j at q_j .

$$S_j = s|\mathbf{n}_{q_j}^k \cdot (q_j - s)| = 0 \quad (2.13)$$

2.3.2 Correspondence Rejection

The purpose of this step is outlier rejection. It refines the pairs to get more accurate transformation.

Some of the methods that have been suggested are -

1. Rejection of corresponding pairs which are more than a distance apart.
2. Rejection of worst n pairs based on some metric. This is usually the distance between points [26].
3. Rejection of pairs whose point to point distance is larger than multiple of standard deviation [20].
4. Rejection of pairs containing points on boundaries [30].
5. Rejection of pairs that are not consistent with neighboring pairs. This follows the assumption that surfaces move rigidly. Under this scheme, two correspondences $\mathbf{p}_1, \mathbf{q}_1$ and p_2, q_2 are consistent iff

$$\text{Dist}(\mathbf{p}_1, \mathbf{p}_2) - \text{Dist}(\mathbf{q}_1, \mathbf{q}_2)$$

is greater than some threshold. In the paper by Dorai et al [6], the following threshold is used

$$\tau = 0.1 * \max(\text{Dist}(\mathbf{p}_1, \mathbf{p}_2), \text{Dist}(\mathbf{q}_1, \mathbf{q}_2))$$

2.3.3 Transformation Estimation

Given a point $\mathbf{p}_i \in P$ and $\mathbf{q}_i \in Q$, the transformation $\mathbf{T}$ can be found out by minimization of the error function

$$e = \sum_{i=1}^N \|\mathbf{T}^k \mathbf{p}_i - \mathbf{q}_i\|^2 \quad (2.14)$$

The minimization of this error function is in itself a research problem with several least squares solutions being proposed. The primary difference between all of them is the representation of the transformation $\mathbf{T}$. Arun *et al* [2], Horn *et al* [16], Schonemann *et al* [28] and Umeyama *et al* [31] used matrix representation of $\mathbf{T}$ and then solved the error function using SVD. Quaternions were made popular by Horn *et al* [15] and was later used by Walker *et al* [33] and Faugeras *et al* [8].

Both SVD and quaternion based approaches start by decoupling the rotation $\mathbf{R}$ and translation $\mathbf{t}$ and by shifting the original points to get zero mean points $\mathbf{p}'_i = \mathbf{p}_i - \boldsymbol{\mu}_p$ and $\mathbf{q}'_i = \mathbf{q}_i - \boldsymbol{\mu}_q$.

The least squares formulation is then written as

$$e = \sum_{i=1}^N \|\mathbf{R}^k \mathbf{p}'_i - \mathbf{q}'_i - \mathbf{t}\|^2 \quad (2.15)$$

In case of matrix representation, the optimal rotation is found by using Singular Value Decomposition on the correlation matrix

$$\mathbf{C} = \sum_{i=1 \text{ to } N} \mathbf{p}'_i \mathbf{q}'_i{}^T = \begin{bmatrix} C_{xx} & C_{xy} & C_{xz} \\ C_{yx} & C_{yy} & C_{yz} \\ C_{zx} & C_{zy} & C_{zz} \end{bmatrix}$$

Singular Value Decomposition of $\mathbf{C}$ gives

$$SVD(\mathbf{C}) = \mathbf{U} \mathbf{S} \mathbf{V}^T$$

The final rotation is given by

$$\mathbf{R} = \mathbf{U} \begin{bmatrix} 1 & & \\ & 1 & \\ & & det(\mathbf{U} \mathbf{V}^T) \end{bmatrix} \mathbf{V}^T \quad (2.16)$$

The quaternion based solution is given by the eigenvector corresponding to the maximal eigenvalue of the matrix

$$N = \begin{bmatrix} C_{xx} + C_{yy} + C_{zz} & C_{yz} - C_{xy} & C_{zx} - C_{xz} & C_{xy} - C_{yx} \\ C_{yz} - C_{zy} & C_{xx} - C_{yy} - C_{zz} & C_{xy} + C_{yx} & C_{zx} + C_{xz} \\ C_{zx} - C_{xz} & C_{xy} + C_{yx} & C_{yy} - C_{xx} - C_{zz} & C_{yz} + C_{zy} \\ C_{xy} - C_{yx} & C_{zx} - C_{xz} & C_{yz} + C_{zy} & C_{zz} - C_{yy} - C_{xx} \end{bmatrix}$$

The translation in both the methods is found by aligning the means of the reference and the rotated point clouds.

$$\mathbf{t} = \boldsymbol{\mu}_q - \mathbf{R} \boldsymbol{\mu}_p \quad (2.17)$$

Besl *et al* [3] use the unit quaternion method by Horn *et al* [15] whereas Zhang *et al* [35] use the method suggested by Walker *et al* [33].

2.3.4 Final Algorithm

Assuming we have two surfaces P and Q and an initial transformation $\mathbf{T}_0$, the minimization function is

$$e_k = \sum_{i=1}^N d_s^2(\mathbf{T}\mathbf{T}_{k-1}p_i, S_i^k) \quad (2.18)$$

The algorithm finds the transformation $\mathbf{T}$ which minimizes e_k with a least squares method iteratively.

2.4 Bundle Adjustment

Due to the presence of noise in measurements there are errors in the relative transformations obtained from the registration algorithms. In that case a global optimization step is needed to minimize the overall error in registration.

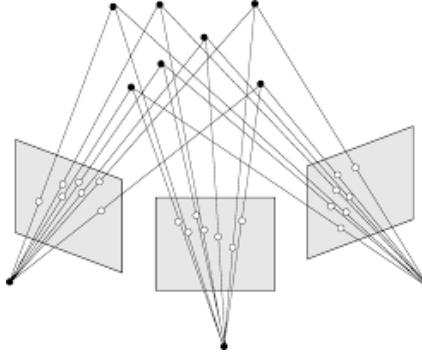


Figure 2.4: Bundle adjustment using point data [19]

In case of point clouds, bundle adjustment is done by minimizing the reprojection error between the observed point and the predicted point. Let X_i be a 3D point observed from multiple views. The aim is to minimize the difference between the point and the reprojected image points. So

the cost function becomes

$$\min_{V_j, X_i} \sum_{i=1}^m \sum_{j=1}^n \|f(V_j, X_i) - x_{ji}\|^2 \quad (2.19)$$

The measurement function $f(V_j, X_i)$ estimates the predicted projection of X_i on image corresponding to view V_j . The minimization function can be solved by applying the Levenberg-Marquardt algorithm [21] which uses a combination of Gauss Newton and Gradient Descent for obtaining the minima.

Chapter 3

Proposed Algorithm

3.1 Boundary Classification

The algorithm uses the planes extracted using the algorithm described in Section 2.2 to classify the boundaries as occluding or intersecting. It uses the noise model parameter to set the threshold for the distance between the actual boundary and the geometric boundary. The basic assumption is that the deviation of the boundary with respect to the actual boundary should not be greater than the standard deviation of depth which is given by Equation 2.3.

The first step is finding the edges between the planes. This is done by creating a graph between the planes which are connected by edges only when they are adjacent i.e. they have a shared boundary between them. The dual of the graph gives the nodes as boundaries and the planes as edges between them.

The algorithm is given below

Algorithm 1: Boundary Classification

Input: Graph $G = (V, E)$ where $V \in \text{Edges in the scene}$ and $E \in \text{Planes}$

Output: A labeling (0,1) for each node in V where 0 = occluding and 1 = intersecting

```

foreach  $v \in V$  do
    Extract the edges  $e_1, e_2$  corresponding to the planes  $pl_1, pl_2$  corresponding to that
    boundary. Find the geometric line  $l_1$  of intersection of the planes using the plane
    equations given by
    
$$\mathbf{n}_1^T \mathbf{p}_1 = d_1$$


$$\mathbf{n}_2^T \mathbf{p}_2 = d_2$$

    Find the actual boundary line  $l_2$  from the extracted planes in 3D

    if  $\text{dist}(l_1, l_2 < \sigma_z)$  then
        |  $v = 1$ 
    end

    else
        |  $v = 0$ 
    end
end

```

3.2 Planar Registration

The aim of the algorithm is to find the rigid body transform between two 3D models which are defined by their planar segments. These planar segments are represented in terms of the normal, offset and the covariance. It uses the Iterative Closest Point framework and iteratively estimates the transformation. One of the key points is that it does not require an estimate of the initial relative transform.

3.2.1 Assumptions

The algorithm uses the geometric properties of a scene and makes the following assumptions

- The scene has a structured environment

- Each frame in the scene has planes in three linearly independent directions.
- Each frame has atleast three non parallel planes.

As discussed in Section 2.3, the ICP framework iterates through the following steps until the error converges.

1. Correspondence Estimation
2. Correspondence Rejection
3. Transformation Estimation

3.2.2 Correspondence Estimation

Given a set of planes $p_l \in F_l$ and $p_r \in F_r$, where F_r is the reference frame and F_l is the frame to be registered. The aim of this step is to find the matching set of planes which can be used to estimate the transform correctly. This is done by minimizing a function of distances between the planes in two views. The measure used is heteroscedastic planar distance which is given by

$$dist(P_i, P_j) = \frac{1}{2} \left(\frac{d_i - d_j}{\sigma_i} + \frac{d_i - d_j}{\sigma_j} \right) + d_{cos}(\mathbf{n}_i, \hat{\mathbf{n}}_j) + d_{cos}(\hat{\mathbf{n}}_i, \mathbf{n}_j) \quad (3.1)$$

where d_{cos} is the heteroscedastic cosine distance given by

$$d_{cos}(\mathbf{n}_i, \mathbf{n}_j) = \frac{1 - \mathbf{n}_i^T \mathbf{n}_j}{\sqrt{\mathbf{n}_i^T \mathbf{C}_{\mathbf{n}_i} \mathbf{n}_j}} \quad (3.2)$$

where $\mathbf{C}_{\mathbf{n}_i}$ is the covariance of plane normal $\mathbf{n}_i$.

This distance measure is used for estimation as it uses the uncertainty in the measurements and gives more weight to the planar correspondences which have less noise.

3.2.3 Correspondence Rejection

In two consecutive views, there are planes which might not be present in both the frames. The correspondence estimation phase finds the best matching plane in second view for each plane in

first view. But the aim is to find the planes which are true correspondences and reject outliers. The pairs of planes are rejected based on a threshold on heteroscedastic planar distance between them. We have followed the thresholding by Masuda *et al.* [20] and taken the threshold as the standard deviation in the planar distances of pairs.

3.2.4 Transformation Estimation

3.2.4.1 Decoupling of Rotation and Translation

Consider the frames F_l and F_r with the corresponding planes pl_l and pl_r with normals $\mathbf{n}_l$ and $\mathbf{n}_r$.

The planes in the two frames can be given by

$$\begin{aligned}\mathbf{n}_l^T \mathbf{p}_l &= d_l \\ \mathbf{n}_r^T \mathbf{p}_r &= d_r\end{aligned}\tag{3.3}$$

If the camera moves from left to right and observes the same physical point as $\mathbf{p}_l$ and $\mathbf{p}_r$ in the two frames, then the coordinates are related by

$$\mathbf{p}_l = \mathbf{R}\mathbf{p}_r + \mathbf{t}\tag{3.4}$$

Substituting 3.4 in 3.3 and comparing the coefficients, we get

$$\mathbf{n}_l = \mathbf{R}\mathbf{n}_r\tag{3.5}$$

$$\mathbf{n}_l^T \mathbf{t} = d_l - d_r\tag{3.6}$$

The registration problem is now of estimating the $\mathbf{R}$ and $\mathbf{t}$ from the planar correspondences.

3.2.4.2 Least Squares Estimate of Rotation

The rotation can be estimated by aligning the normals of matching planes by minimizing the SSD

$$\min ||\mathbf{R}\mathbf{n}_l - \mathbf{n}_r||^2 \quad (3.7)$$

The above minimization function can be solved by using SVD by replacing points with normals in the solution by Horn *et al.* [15]. The estimation requires planes in three linearly independent directions.

3.2.4.3 Least Squares Estimate of Translation

By stacking the Equations 3.6, the translation should satisfy

$$\mathbf{N}\mathbf{t} = \mathbf{D} \quad (3.8)$$

$$\text{where } \mathbf{N} = \begin{Bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \dots \\ \mathbf{n}_m \end{Bmatrix} \text{ and } \mathbf{D} = \begin{Bmatrix} d_{l1} - d_{r1} \\ d_{l2} - d_{r2} \\ \dots \\ d_{lm} - d_{rm} \end{Bmatrix}$$

This is solved using Total Least Squares method (TLS) method as suggested by Golub *et al.* [10] using SVD. But unlike rotation, here we need more than three non parallel planes for correct estimation of translation.

3.2.5 Final Algorithm

Algorithm 2: Planar Registration

Input: Planes pl_l and pl_r from frames F_l and F_r

Result: Relative Transform $\mathbf{T}$

Initialization:

```

(a) Find corresponding planes using heteroscedastic planar distance while the error does not converge do
    (a) Compute incremental transformation using the current correspondences using least squares solution
    (b) Update cumulative transformation and apply to planes.
    (c) Find corresponding planes and compute distance.
    (d) if number of iteration > maximum allowable iterations then
        | terminate
    end
end

```

3.3 Revised Bundle Adjustment

When the pairwise alignment of the views is complete, it can be observed that individual pairwise errors are accumulated and hence introduce drift in the model. To reduce this error a cost function has to be formulated which minimizes the error between frames.

The registration algorithm also returns the planar correspondences which can be used in bundle adjustment step to minimize the distance between the planes in consecutive views. Let pl_1 and pl_2 be corresponding planes in views V_1 and V_2 respectively which have a relative transformation $\mathbf{T}$. The equations of the planes are

$$\begin{aligned} \mathbf{n}_1^T \mathbf{p}_1 &= d_1 \\ \mathbf{n}_2^T \mathbf{p}_2 &= d_2 \end{aligned} \tag{3.9}$$

Then the cost function to be minimized for estimation of transformation $\mathbf{T}$ is given by

$$\mathbf{T} = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{\forall V} \sum_{\forall pl} \|\mathbf{n}_1^T (\mathbf{p}_1 - \mathbf{T} \mathbf{p}_2)\|^2 \quad (3.10)$$

where $\mathbf{p}_1$ and $\mathbf{p}_2$ are the centroids of planes pl_1 and pl_2 respectively.

The cost function aims to minimize the dot product of the normal of the plane and the vector of difference between the centroids. If the planes are matching planes then both the centroids should lie on the plane and hence the difference vector will in an ideal case have a dot product zero with the normal of the plane.

This cost function can be solved by the Levenberg Marquardt Algorithm as discussed Section 2.4. Once the global transformations are obtained by the optimization, it is applied to the frames to get the final model.

Chapter 4

Results and Conclusion

4.1 Results

4.1.1 Boundary Classification

NYU Depth Planar dataset [22] is used to test the boundary classification algorithm. The dataset has RGBD images taken from the Kinect sensor of indoor scenes. The dataset contains 480×640 RGB and depth images taken from different indoor areas like the living room, kitchen, etc. Data collection was done with fps varying between 20 and 30.

In the results, the occluding boundaries are represented using blue lines and intersecting ones are drawn using red lines.

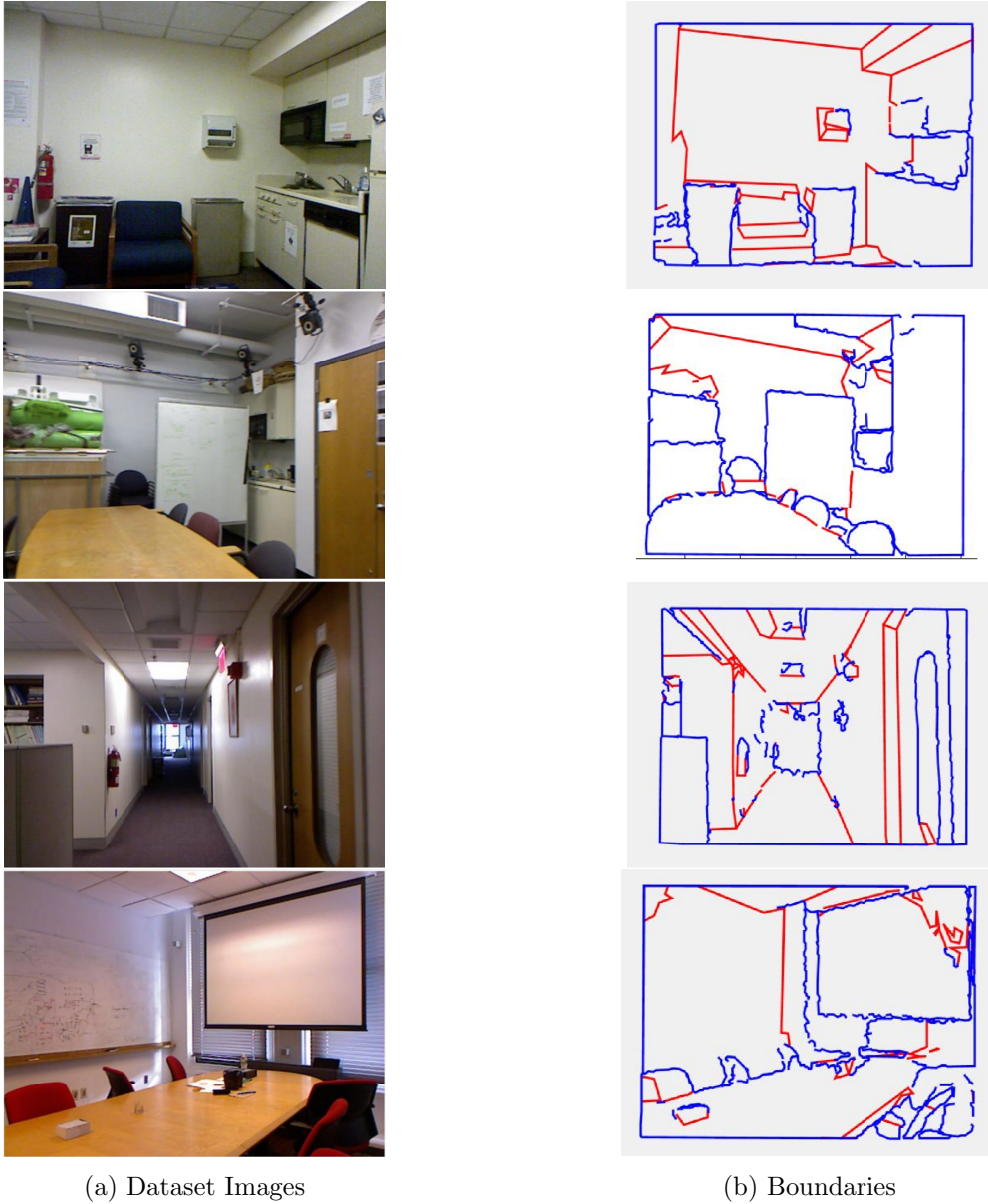


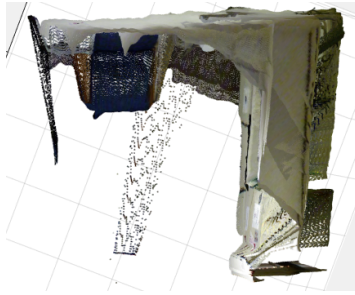
Figure 4.1: Boundary Classification Results for NYU dataset

4.1.2 Scene Reconstruction

4.1.2.1 NYU dataset

Since NYU dataset does not have data for full reconstruction of a scene, results for a part of the room are shown in Figures 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, and 4.9.

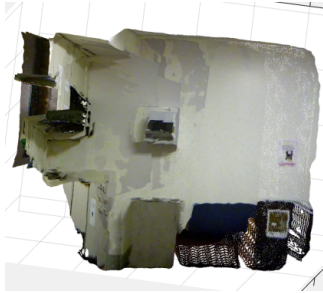
4.1.2.2 NYU Kitchen



(a) Top View



(b) Front View



(c) Back View



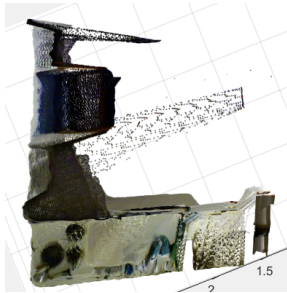
(d) Back View



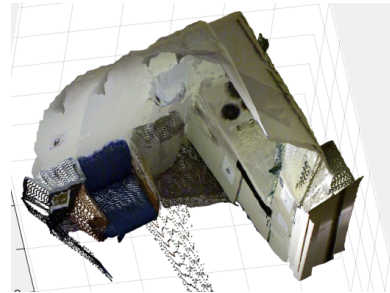
(e) Side View



(f) Side View



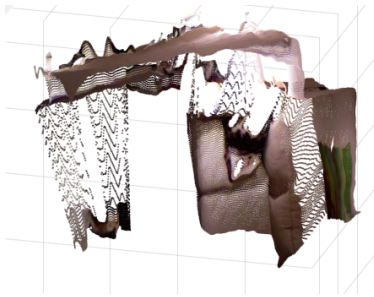
(g) Bottom View



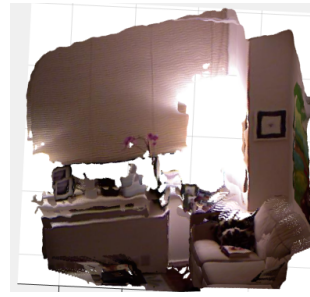
(h) Tilted View

Figure 4.2: Corner reconstruction of NYU Kitchen

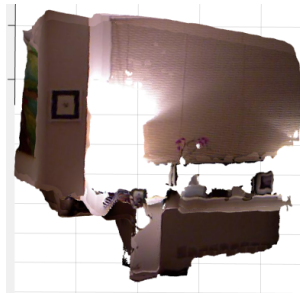
4.1.2.3 NYU Living Room 1



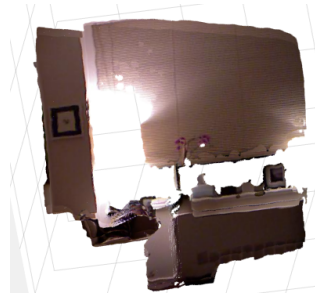
(a) Top View



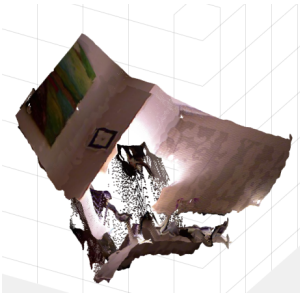
(b) Front View



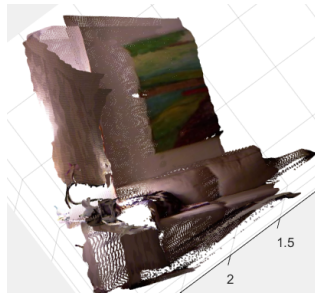
(c) Back View



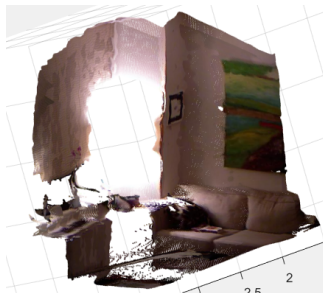
(d) Back View



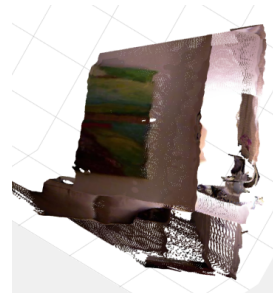
(e) Side View



(f) Side View



(g) Front View



(h) Back View

Figure 4.3: Corner reconstruction of NYU Living Room - 1

4.1.2.4 NYU Living Room 2

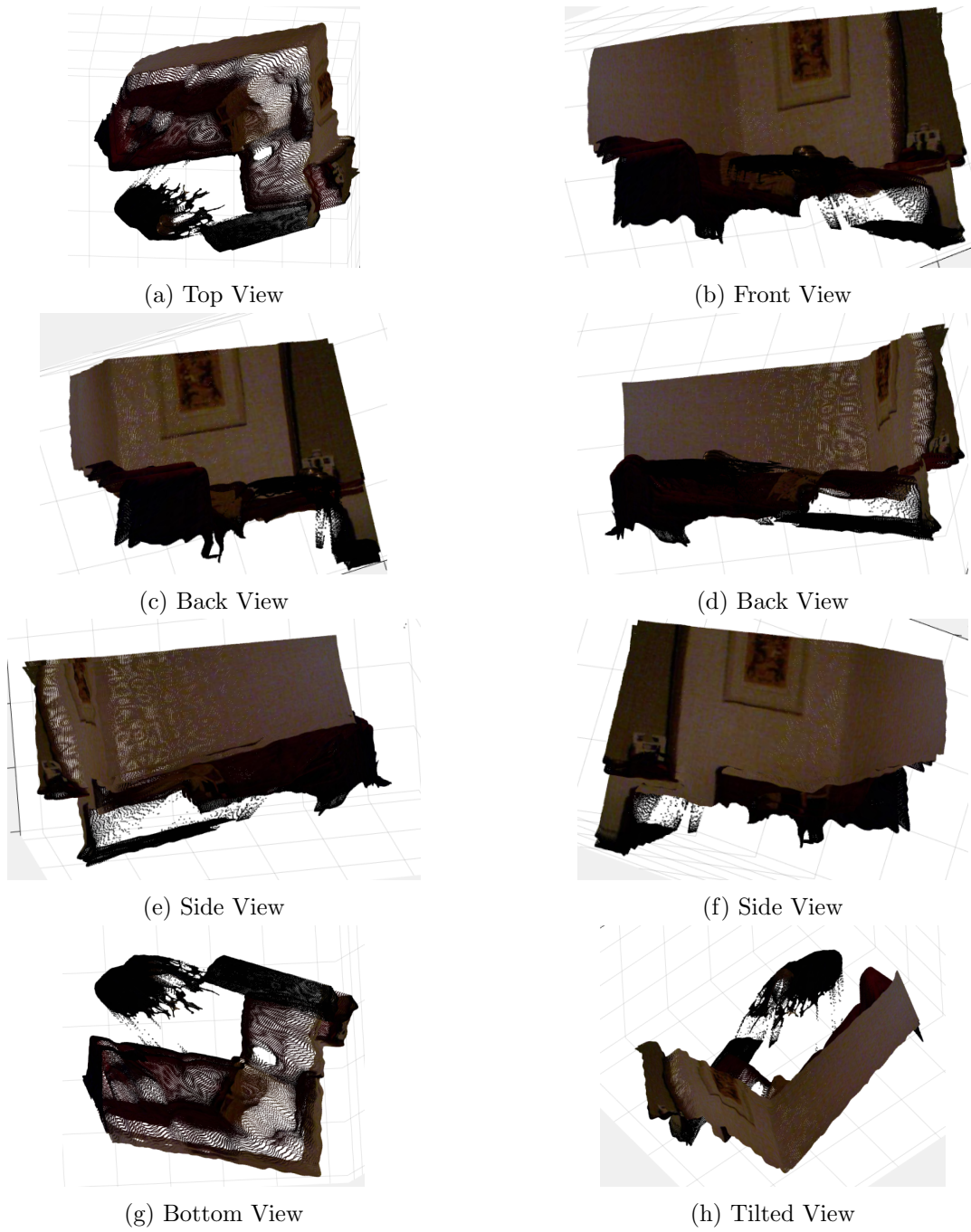
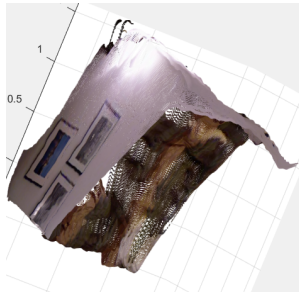


Figure 4.4: Corner reconstruction of NYU Living Room - 2

4.1.2.5 NYU Living Room 3



(a) Top View



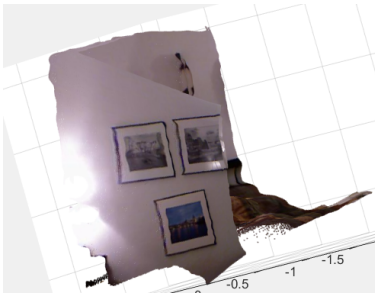
(b) Front View



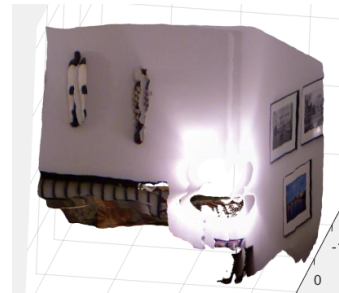
(c) Back View



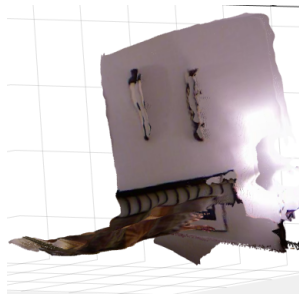
(d) Back View



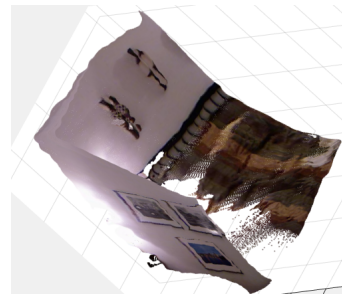
(e) Side View



(f) Side View



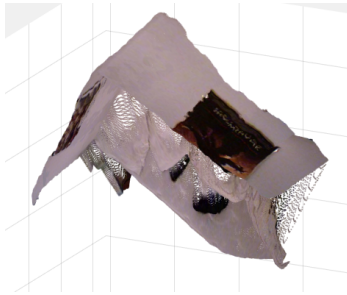
(g) Side View



(h) Tilted View

Figure 4.5: Corner reconstruction of NYU Living Room - 3

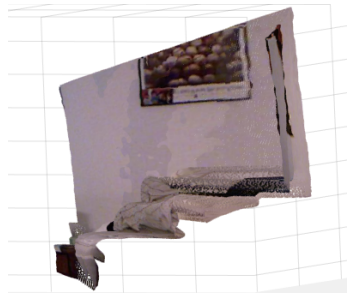
4.1.2.6 NYU Living Room 4



(a) Top View



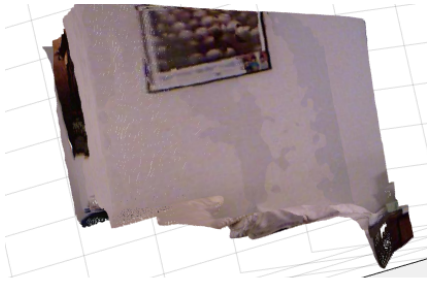
(b) Front View



(c) Back View



(d) Back View



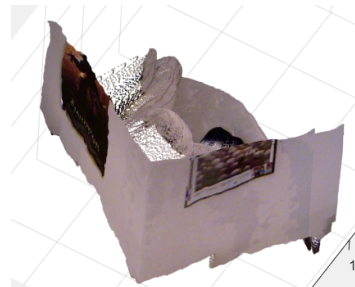
(e) Side View



(f) Side View



(g) Side View



(h) Tilted View

Figure 4.6: Corner reconstruction of NYU Living Room - 4

4.1.2.7 NYU Office

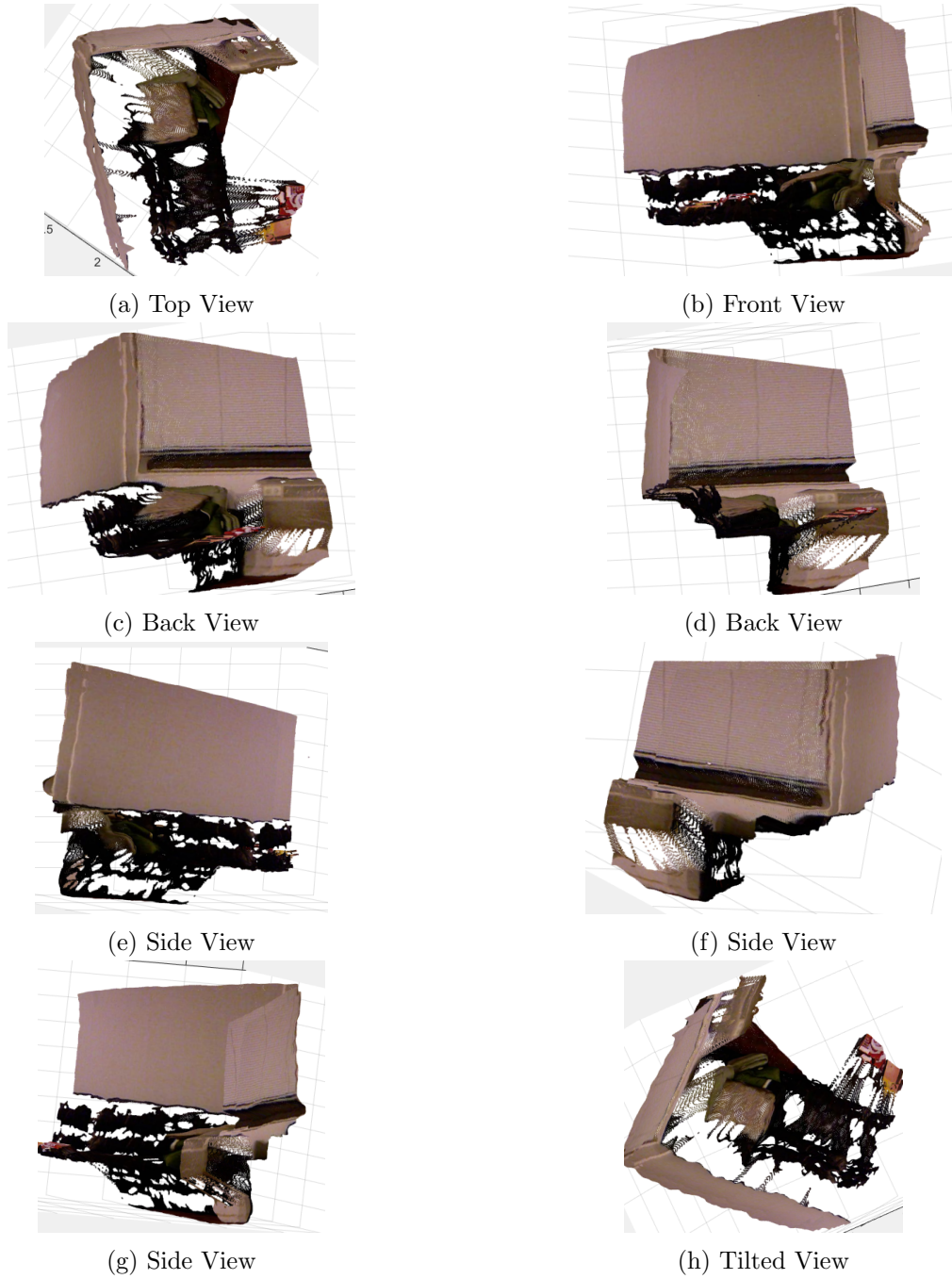


Figure 4.7: Corner reconstruction of NYU Office

4.1.2.8 NYU Bedroom 1

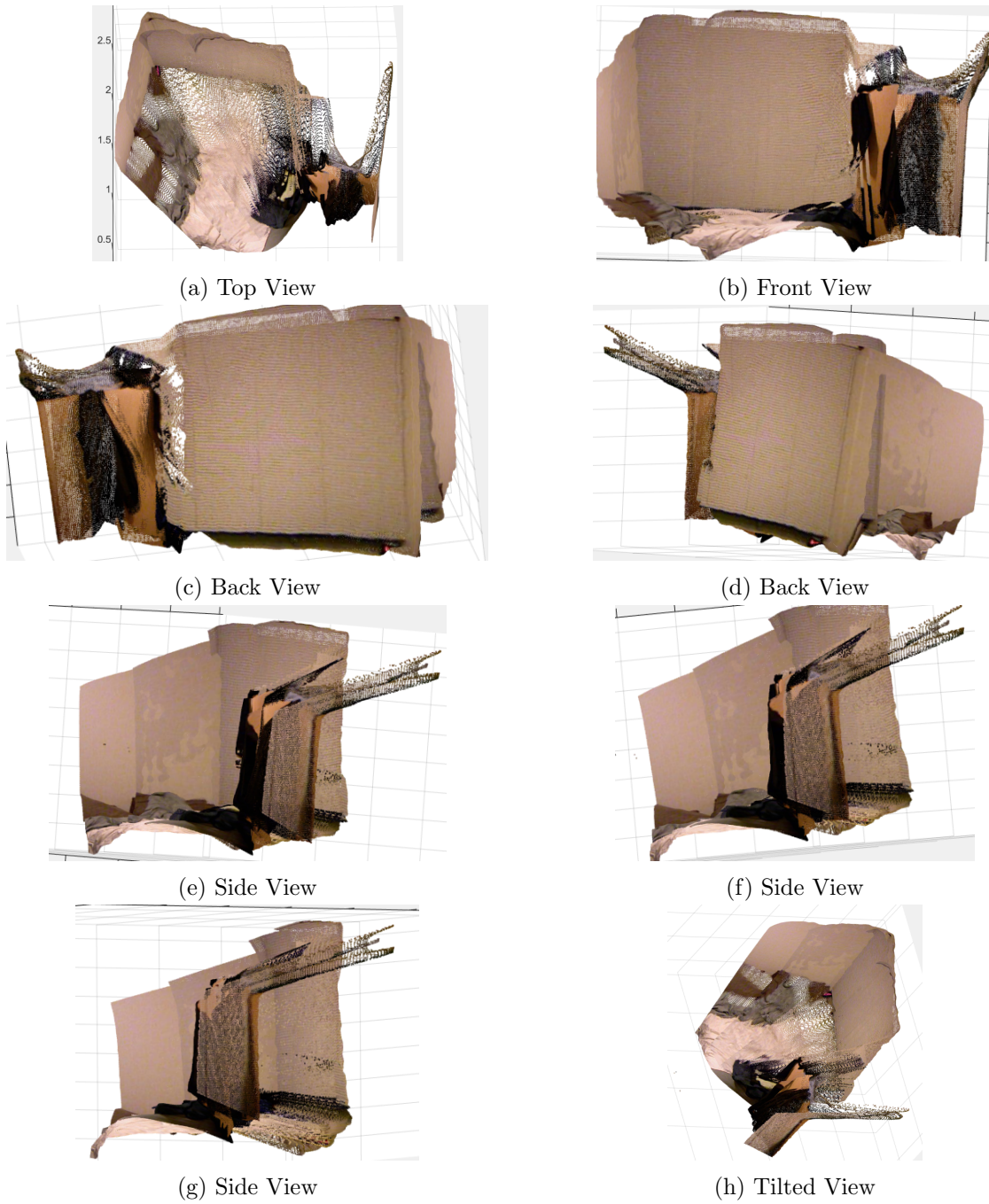


Figure 4.8: Corner reconstruction of NYU Bedroom - 1

4.1.2.9 NYU Bedroom 2

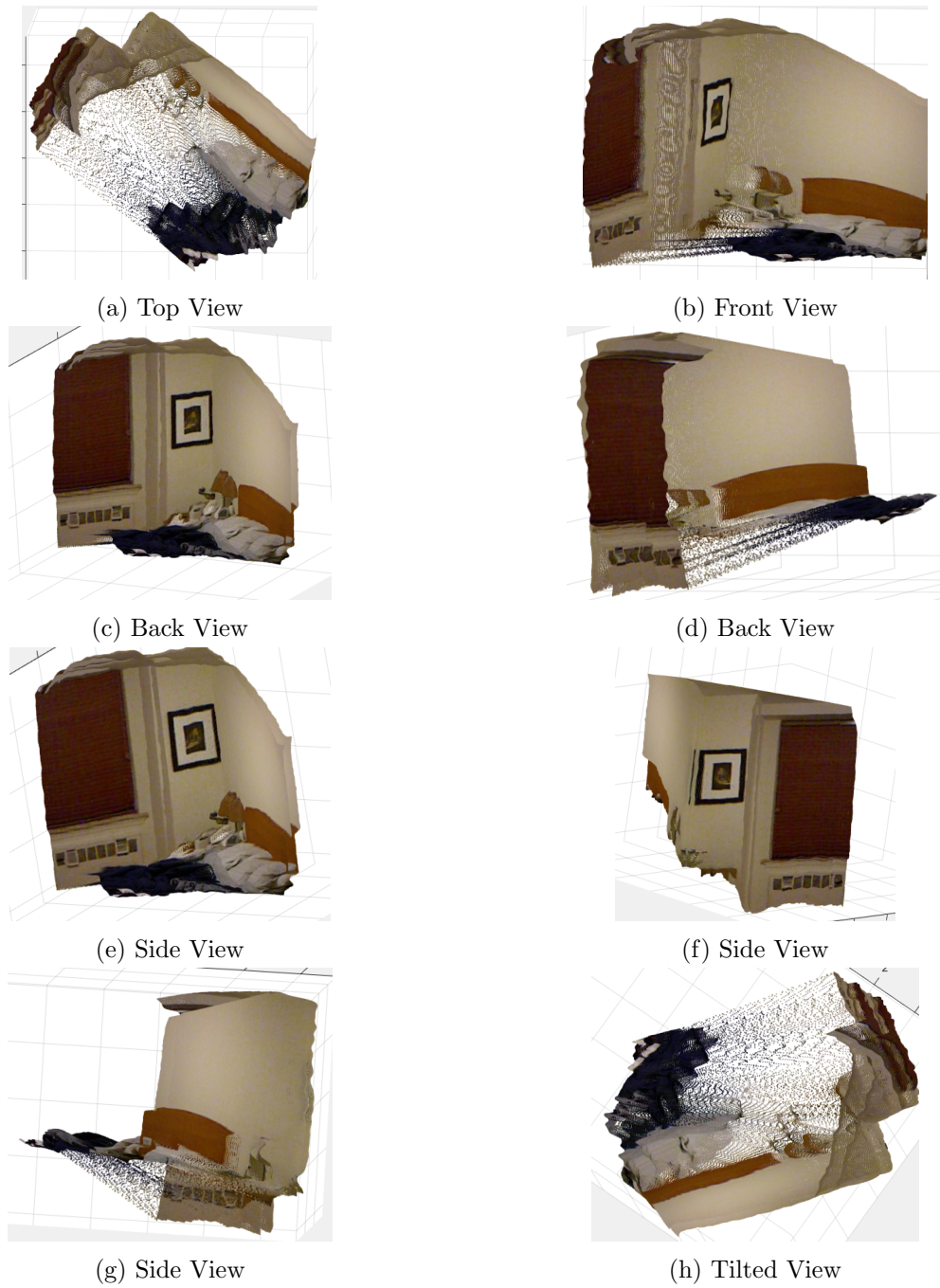


Figure 4.9: Corner reconstruction of NYU Bedroom - 2

4.1.2.10 Swarath Lab

Data was collected using Kinect v2 for one scene. The results are in Figure 4.10. As can be seen, the plain walls have transformation errors. This is due to the absence of three linearly independent direction planes in the walls which lead to erroneous estimation of transformation. The registration at the junctions is done accurately because three directions are obtained owing to the presence of walls and table.

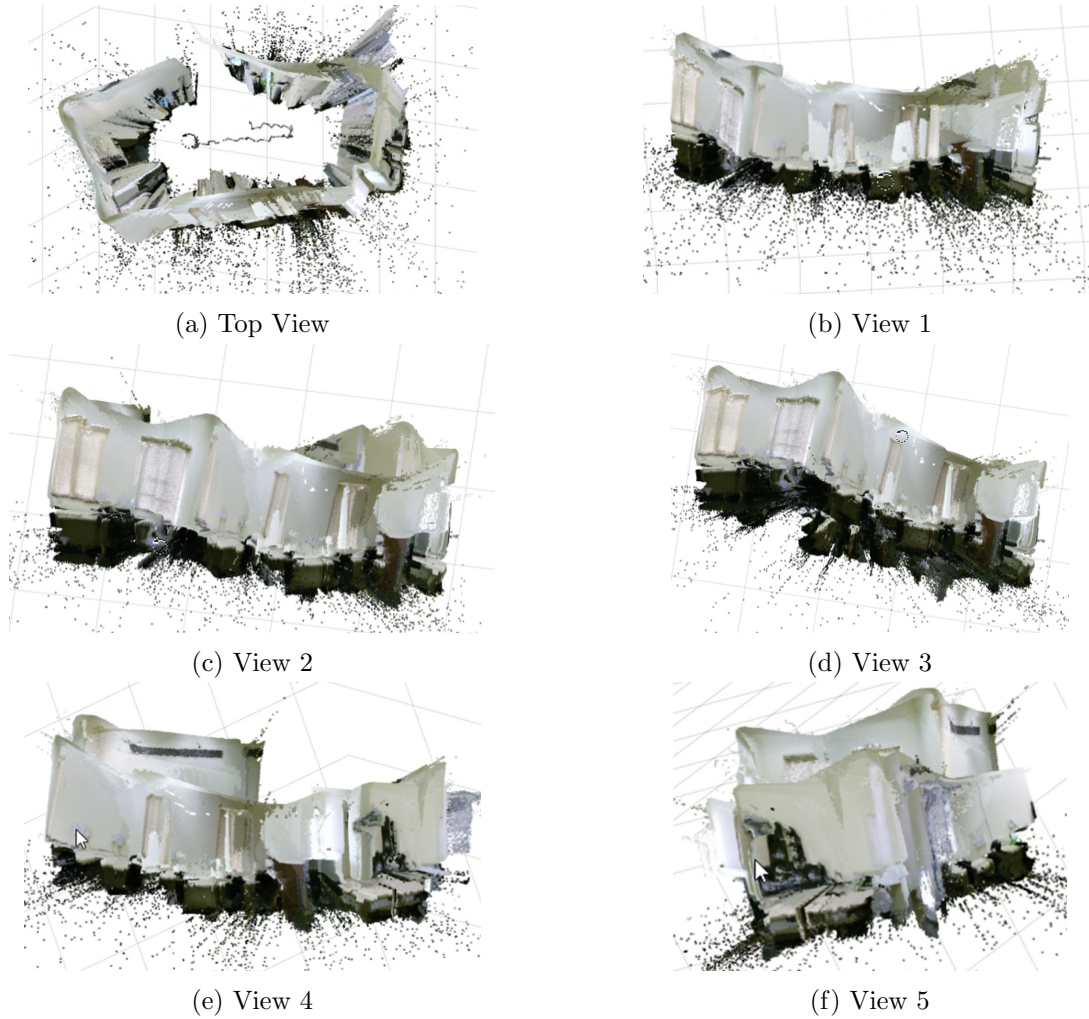


Figure 4.10: Reconstructon of Swarath Lab

4.2 Limitations

- Since the estimation of rotation uses SVD, it requires the presence of planes in three linearly independent directions. This cannot always be possible. In such a case the estimation leads to transformation errors
- The estimation of translation also require the presence of three non parallel planes. If they are not available, the translation estimates are erroneous.
- The aforementioned constraints on presence of requisite number of planes in different directions puts a limitation on the scenes that can be reconstructed using the algorithm.

4.3 Conclusion and Future Work

As seen from the results, planar primitives can be used for efficient reconstruction subject to some constraints. To eliminate such issues the RGB features such as SIFT can also be merged into the feature space to get accurate transformation estimates. Planar information can be used for loop closing algorithm which further leads to better reconstruction of indoor scenes by keeping track of the starting scene and making adjustments when the starting point has been reached again.

Bibliography

- [1] ANAND, S. *Robust Methods for Multiple Model Discovery in Structured and Unstructured Data*. PhD dissertation, Rutgers, The State University of New Jersey, 2013.
- [2] ARUN, K. S., HUANG, T. S., AND BLOSTEIN, S. D. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, 5 (1987), 698–700.
- [3] BESL, P. J., AND MCKAY, N. D. Method for registration of 3-d shapes. In *Robotics-DL tentative* (1992), International Society for Optics and Photonics, pp. 586–606.
- [4] BIRK, A., PATHAK, K., VASKEVICIUS, N., PFINGSTHORN, M., POPPINGA, J., AND SCHWERTFEGGER, S. Surface representations for 3d mapping. *KI-Künstliche Intelligenz* 24, 3 (2010), 249–254.
- [5] CHEN, Y., AND MEDIONI, G. Object modelling by registration of multiple range images. *Image and vision computing* 10, 3 (1992), 145–155.
- [6] DORAI, C., WANG, G., JAIN, A. K., AND MERCER, C. Registration and integration of multiple object views for 3d model construction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 1 (1998), 83–89.
- [7] DOU, M., GUAN, L., FRAHM, J.-M., AND FUCHS, H. Exploring high-level plane primitives for indoor 3d reconstruction with a hand-held rgb-d camera. In *Asian Conference on Computer Vision* (2012), Springer, pp. 94–108.
- [8] FAUGERAS, O. D., AND HEBERT, M. The representation, recognition, and locating of 3-d objects. *The international journal of robotics research* 5, 3 (1986), 27–52.
- [9] FISCHER, D., AND KOHLHEPP, P. 3d geometry reconstruction from multiple segmented surface descriptions using neuro-fuzzy similarity measures. *Journal of Intelligent and Robotic Systems* 29, 4 (2000), 389–431.

- [10] GOLUB, G. H., AND VAN LOAN, C. F. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis* 17, 6 (1980), 883–893.
- [11] GUPTA, S., ARBELAEZ, P., AND MALIK, J. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 564–571.
- [12] HE, W., MA, W., AND ZHA, H. Automatic registration of range images based on correspondence of complete plane patches. In *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)* (2005), IEEE, pp. 470–475.
- [13] HOIEM, D., EFROS, A. A., AND HEBERT, M. Geometric context from a single image. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (2005), vol. 1, IEEE, pp. 654–661.
- [14] HOIEM, D., STEIN, A. N., EFROS, A. A., AND HEBERT, M. Recovering occlusion boundaries from a single image. In *2007 IEEE 11th International Conference on Computer Vision* (2007), IEEE, pp. 1–8.
- [15] HORN, B. K. Closed-form solution of absolute orientation using unit quaternions. *JOSA A* 4, 4 (1987), 629–642.
- [16] HORN, B. K., HILDEN, H. M., AND NEGAHDARIPOUR, S. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A* 5, 7 (1988), 1127–1135.
- [17] JIA, Z., GALLAGHER, A., AND CHEN, T. Learning boundaries with color and depth. In *2013 IEEE International Conference on Image Processing* (2013), IEEE, pp. 3049–3053.
- [18] KOHLHEPP, P., BRETTHAUER, G., WALTHER, M., AND DILLMANN, R. Using orthogonal surface directions for autonomous 3d-exploration of indoor environments. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006), IEEE, pp. 3086–3092.
- [19] LOURAKIS, M. I. A., AND ARGYROS, A. A. Sba: a software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software* (2009), 1–30.
- [20] MASUDA, T., SAKAUE, K., AND YOKOYA, N. Registration and integration of multiple range images for 3-d model construction. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on* (1996), vol. 1, IEEE, pp. 879–883.
- [21] MORÉ, J. J. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*. Springer, 1978, pp. 105–116.

- [22] NATHAN SILBERMAN, DEREK HOIEM, P. K., AND FERGUS, R. Indoor segmentation and support inference from rgbd images. In *ECCV* (2012).
- [23] PATHAK, K., BIRK, A., VASKEVICIUS, N., PFINGSTHORN, M., SCHWERTFEGER, S., AND POPPINGA, J. Online three-dimensional slam by registration of large planar surface segments and closed-form pose-graph relaxation. *Journal of Field Robotics* 27, 1 (2010), 52–84.
- [24] PATHAK, K., BIRK, A., VASKEVICIUS, N., AND POPPINGA, J. Fast registration based on noisy planes with unknown correspondences for 3-d mapping. *IEEE Transactions on Robotics* 26, 3 (2010), 424–441.
- [25] PATHAK, K., BORRMANN, D., ELSEBERG, J., VASKEVICIUS, N., BIRK, A., AND NÜCHTER, A. Evaluation of the robustness of planar-patches based 3d-registration using marker-based ground-truth in an outdoor urban scenario. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on* (2010), IEEE, pp. 5725–5730.
- [26] PULLI, K., AND SHAPIRO, L. G. Surface reconstruction and display from range and color data. *Graphical Models* 62, 3 (2000), 165–201.
- [27] RAPOSO, C., LOURENÇO, M., ANTUNES, M., AND BARRETO, J. P. Plane-based odometry using an rgb-d camera. In *BMVC* (2013).
- [28] SCHÖNEMANN, P. H. A generalized solution of the orthogonal procrustes problem. *Psychometrika* 31, 1 (1966), 1–10.
- [29] TREVOR, A. J., ROGERS, J. G., AND CHRISTENSEN, H. I. Planar surface slam with 3d and 2d sensors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (2012), IEEE, pp. 3041–3048.
- [30] TURK, G., AND LEVOY, M. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (1994), ACM, pp. 311–318.
- [31] UMEYAMA, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence* 13, 4 (1991), 376–380.
- [32] VIEJO, D., AND CAZORLA, M. 3d plane-based egomotion for slam on semi-structured environment. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007), IEEE, pp. 2761–2766.
- [33] WALKER, M. W., SHAO, L., AND VOLZ, R. A. Estimating 3-d location parameters using dual number quaternions. *CVGIP: image understanding* 54, 3 (1991), 358–367.

- [34] WEINGARTEN, J., AND SIEGWART, R. 3d slam using planar segments. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006), IEEE, pp. 3062–3067.
- [35] ZHANG, Z. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision* 13, 2 (1994), 119–152.