Planar Segmentation of RGBD Images using Fast Linear Fitting and Markov Chain Monte Carlo

Can Erdogan, Manohar Paluri, Frank Dellaert
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA, 30332
cerdogan@cc.gatech.edu, bpaluri@cc.gatech.edu, dellaert@cc.gatech.edu

***Abstract*—With the advent of affordable RGBD sensors such as the Kinect, the collection of depth and appearance information from a scene has become effortless. However, neither the correct noise model for these sensors, nor a principled methodology for extracting planar segmentations has been developed yet. In this work, we advance the state of art with the following contributions: we correctly model the Kinect sensor data by observing that the data has inherent noise only over the measured disparity values, we formulate plane fitting as a linear least-squares problem that allow us to quickly merge different segments, and we apply an advanced Markov Chain Monte Carlo (MCMC) method, generalized Swendsen-Wang sampling, to efficiently search the space of planar segmentations. We evaluate our plane fitting and surface reconstruction algorithms with simulated and real-world data.**

***Keywords*-Planar Segmentation; Generalized Swendsen-Wang Sampling; Surface Reconstruction; Linear Plane Fitting**

I. Introduction

Interpretation of RGBD images is quickly becoming an important problem in robotics. The advent of affordable RGBD sensing, as exemplified by the Kinect camera has made RGBD data retrieval both easy and affordable. Common techniques for scene reconstruction are either based on calibrated camera systems, monocular and multi-view, or direct sensor systems using time of flight or structured light principles. A wide range of robotics applications now rely on the Kinect, an infrared structured light sensor co-registered with an RGB camera, as a source of RGBD point clouds, and hence the topic of RGBD. Applications include simultaneous localization and mapping, manipulation-aided perception and grasping, and human-robot interaction.

Additional processing is often performed to retrieve semantic information such as fitting planes by the singular value decomposition (SVD), or registering consecutive image frames by the iterative closest point algorithm. However, most of these post-processing approaches made the underlying assumption that the observed data is a 3D point cloud, and that the measurement noise is Gaussian isotropic noise on those 3D points. This is untrue for structured light sensors, where inherently the noise is only on the disparity measurements made by the sensor. In addition, plane fitting approaches often greedily merge planar segments [1], instead of taking a principled, Bayesian inference point of view.

Figure 1. Top left: Original color image. Top right: Oversegmentation. Bottom left: Ground truth segmentation. Bottom right: Most frequent RGBD Swendsen-Wang surface reconstruction after 200 iterations. The black patches are the discarded superpixels that are too small.

In this paper, we make the following two contributions. First, we formulate the noise model of the data captured with a structured-light 3D scanner and demonstrate improved results for the plane fitting problem in comparison to the well-known singular value decomposition approach. Additionally, we establish that the inverse depth parameterization [2] of planes leads to a linear measurement function in the least-squares formulation of the fitting problem. The linearity enables efficient plane fitting operations to an over-segmentation, and subsequently leads to mean and covariance estimates that can be used to quickly merge segments.

Our second contribution is using this fast, linear plane fitting component in a Bayesian inference scheme to reason about the true segmentation of the scene. We use the generalized Swendsen-Wang algorithm [3], an MCMC-based method, to sample from the correct posterior distribution over depth segmentations for an input RGBD image. Our

hypothesis space is the space of partitions of super-pixels, obtained in a pre-processing step using the well known Felzenszwalb & Huttenlocher algorithm [4] that uses depth information in addition to color.

A key realization is that, because the original plane fitting criterion is now linear (as we use disparity), evaluating the merger of different super-pixels is computationally very efficient. This allows us to quickly sample from the posterior distribution over partitions. Having access to a sample from and/or any expected values computed from the true posterior is useful in a number of settings, but in this paper we simply estimate the MAP estimate as the segmentation encountered the most number of times. Figure 1 demonstrates the output of the sampling approach after 200 iterations. Within a small amount of time, most of the planes are reconstructed successfully with the exception of the door and ceiling planes which are still merged with the side wall.

II. Related Work

Surface reconstruction and segmentation are well studied problems and some of the research dates back more than two decades [5]. The problem of surface reconstruction has been tackled for a single image, multiple images, stereo data, laser data, registered color and depth data and in recent times using RGBD data [6]–[9]. Many real world applications use surface reconstruction, which gives the problem a significant importance and continuous attention by the research community. More recently, with the introduction of the Kinect sensor numerous applications have sprung up in a relatively short time span.

Some of the work assume an aligned camera and laser scans and use appearance and geometry cues [7] [8]. In [7] the graph based segmentation proposed in [4] was extended to incorporate both color and depth. The results look very promising although the color and depth cues were combined in a naive manner. In [8] an iterative RANSAC approach is used to find the potential surfaces and the algorithm is halted when there is no more data to fit planes.

Many existing top-down as well as bottom-up methods lack the ability to find the appropriate number of surfaces in the scene [10] [11]. These methods either use thresholds or fix the number of surfaces arbitrarily in advance. By changing the threshold or by increasing the number of potential surfaces one ends up with different reconstructions, but without a firm basis to choose between them.

In principle, Bayesian model selection affords us to determine, in a principled way, how many planar segments are needed to model the scene. MCMC techniques have been proposed for the problems of segmentation and surface reconstruction in the past [12]–[15]. Most of these fall into the basic Metropolis, Metropolis-Hastings and Gibbs samplers or variants of them, which all work by proposing to flip the label of a pixel or region. However, even when working with over-segmented images, changing the membership of only a single region per iteration makes the process computationally intractable for real-time purposes. In the Markov random field literature, this realization led to the introduction of the so-called Swendsen-Wang sampler [16], which flips the label of multiple sites simultaneously. This idea was used and generalized in computer vision by Barbu et al. [3], who used it to sample over arbitrary graph partitions, where a graph was defined by an over-segmentation of the image and its neighborhood structure. In addition to image segmentation, they applied the same method to region-based stereo, as well. Using the Swendsen-Wang idea, they showed, speeds up the sampling process by a considerable factor and makes the method competitive with other, non-probabilistic methods.

III. Low-level Plane Fitting

In this section we discuss the low-level plane fitting step, where we first over-segment the image and then fit a planar model to each segment. Instead of modeling the noise as Gaussian isotropic on 3D points, we instead use a more natural noise model on the disparity measured by the structured light sensor. Finally, by using an inverse depth formulation for the plane parameters, we are able to efficiently fit the disparities in each segment.

A. Over-Segmentation

To reduce the computation time for subsequent steps, we first *over*-segment the image to a set of super-pixels that are similar in color and continuous in depth. Our overall goal is to, given an RGBD image, output a segmentation where each segment represents a plane in the world. However, doing so at the pixel-level is overly expensive. To obtain an over-segmentation, we use the method proposed by Felzenszwalb and Huttenlocher [4]. They define a graph $G = (V, E)$ where each pixel is a vertex $v_i \in V$, and each edge e_{ij} between pixels v_i and v_j is given a dissimilarity weight w_{ij}. The algorithm then greedily segments the image using an adaptive threshold based on the degree of variability in neighboring regions.

We use a specially tuned weight function as we deal with RGBD images. The weights computed for each edge are a linear combination of difference in color, depth and spatial location of the two pixels. The exact formula is given below:

$$w_{ij} = \alpha * w_C + \beta * w_Z + \gamma * w_S$$

where

$$\begin{cases} w_C = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2 + (z_i - z_j)^2} \\ w_Z = abs(z_i - z_j) \\ w_S = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \end{cases}$$

where (r_i, g_i, b_i) are the color values, z_i is the depth, and (x_i, y_i) are the pixel coordinates of pixel i. The constants

α, β, γ were determined empirically to yield good over-segmentation results. This modified weight as opposed to just using color yields an over-segmentation that not only is continuous in depth, but also mostly satisfies co-planarity in typical man-made scenes. For example, for the Kinect sensor this typically reduces a 640 x 480 image with almost 300,000 pixels to super-pixels of, on average, 200 pixels in size. Combining depth and color in a similar fashion has also been used by other authors [7].

To remove small superpixels that occur due to noise in color values we need to use a low pass filter. This is a preprocessing step to reduce the number of superpixels obtained from over-segmentation. In typical scenarios a Gaussian kernel with a fixed variance is used. But, in our case it is critical that color does not flow across depth boundaries during the smoothing process. So, we use a joint bilateral filter which was proposed in [17]. The joint bilateral filter is an addition to bilateral filter [18], where the weights to smooth the color image come from the depth image. The color image is convolved with a kernel based on both apperance and depth similarities. It is composed of two Gaussians as given below:

$$I(p_c) = \frac{1}{k} \sum g_c(p_c - p_c^{'}) g_d(p_d - p_d^{'})$$

where

$$\begin{aligned} k &= \sum g_c(p_c - p_c^{'}) g_d(p_d - p_d^{'}) \\ g(x : \mu, \sigma) &= exp(-\frac{(x-\mu)^2}{2\sigma^2}) \end{aligned}$$

p_c is the color value and p_d is the depth value. We do the filtering for the three color channels individually. In our experiments we typically use a window size of 5, the variance of the color Gaussian g_c is set to 3 and variance of the depth Gaussian g_d is set to 0.1. A low variance for g_d ensures that the smoothing process does not bleed color across depth boundaries.

B. Disparity-based Noise Model

Instead of modeling the noise as Gaussian isotropic on 3D points, we instead use a more natural noise model on the disparity measured by the structured light sensor. A body of work has focused on the reverse engineering and the accuracy analysis of the structured-light 3D scanners like the Kinect, which we use here as an example. Given the two onboard cameras, the infrared and the color camera, and the infrared projector, the Kinect firmware first decrypts the projected light pattern to gain depth information and then, registers the depth measurements with the observed color data. Figure 2 depicts the basic structure of the Kinect hardware with a baseline of 2.5 cm between the IR and RGB cameras, and a distance of approximately 10.0 cm between the IR camera and the light projector.

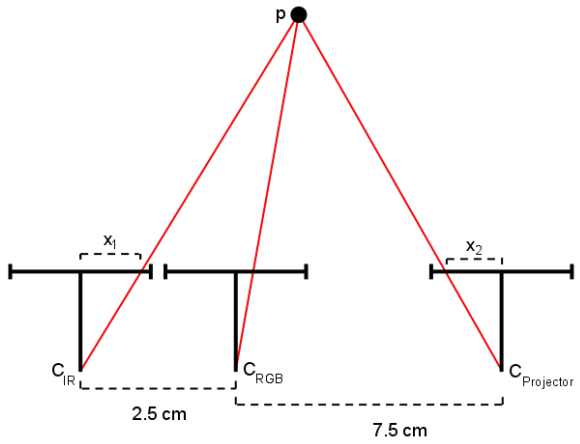

Figure 2. The Kinect geometry.

The structured-light principle that drives the Kinect depth measurements is based on the measure of disparity between the observed pattern points and the expected counterparts in the database. Therefore, the correct formulation of the noise model underlying the Kinect output is a 1D noise on the observed disparity values, rather than a 3D noise on the deduced world point locations. This observation leads to the inverse depth parameterization of the planes and critical performance gains, as shown in the results, since the coupling of the noise on the three dimensions of the data is overlooked by algorithms such as SVD plane fitting.

C. Optimal Estimation of Planes

We define the set of observations $Z = \{u_i, v_i, \delta_i \,|\, 0 \leq i \leq wh\}$ where (w, h) is the resolution camera and for each pixel (u_i, v_i), we make a disparity measurement δ_i. Let $\mathbf{V}$ be the set of super-pixels obtained from over-segmenting the image and let V_i represent the i^{th} super-pixel. An image can be seen as a graph $G = (\mathbf{V}, \mathbf{E})$ where the nodes are the super-pixels and an edge E_{ij} between super-pixels V_i and V_j exists if the two super-pixels are neighbors.

The RGBD segmentation of the image is then a partition of the graph into segments S where a segment S_i with m super-pixels is $S_i = \{V_1^i, ..., V_m^i\}$. The goal is to find a partition S^* that is the *maximum a posterior* estimate of the right partition given the data Z:

$$P(S|Z) = \sum_{S_j \in S} \sum_{V_i \in S_j} \sum_{(u,v,\delta) \in V_i} \left\| h_{k_j}(\theta_j, u, v) - \delta \right\|^2 \quad (1)$$

where the measurement function $h_{k_j}(\theta_j, u, v)$ returns a predicted disparity value for the pixel (u, v) whose segment S_j is fitted with the plane parameters θ_j of model k_j. Next, we explain the inverse depth parameterization of planes which yields a linear measurement function.

D. A Linear Measurement Function

The measurement function $h_k(\theta, u, v)$ is nonlinear in the general plane parameters $\theta = (a, b, c, d)$, with

$$ax + by + cz = d \tag{2}$$

where (a, b, c) is the plane normal and d is the distance from the origin. If we multiple both sides of Equation 2 by $f\beta/dz$ we obtain

$$\frac{af\beta}{d}\frac{x}{z} + \frac{bf\beta}{d}\frac{y}{z} + \frac{cf\beta}{d} = \frac{f\beta}{z}$$

Substituting in the definitions of the image coordinates $u = x/z$, $v = y/z$ and of the disparity $\delta = f\beta/z$, we obtain the following *linear* measurement function for the disparity δ:

$$\delta = h_{linear}(a_l, b_l, c_l) = a_l u + b_l v + c_l$$

where the parameters $a_l = \frac{\beta a}{d}$, $b_l = \frac{\beta b}{d}$ and $c_l = \frac{f\beta c}{d}$ represent the plane and the subscript l stands for the "linear" parameterization $\theta_l = \{a_l, b_l, c_l\}$. The same parameterization was also derived in [2].

The implicit constraint is $d \neq 0$, i.e., we will assume the plane does not pass through the camera origin. Since in this case the plane would only be visible as a line in the image, this is not a practical case of interest.

E. Fitting Planes Linearly to Disparities

Plane fitting with this representation is reduced to minimizing the following least square function:

$$\left\| \begin{bmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots \\ u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} a_l \\ b_l \\ c_l \end{bmatrix} - \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_v \end{bmatrix} \right\|^2 \tag{3}$$

where the input data with n observations is $Z = \{(u_1, v_1, \delta_1), \ldots, (u_n, v_n, \delta_n)\}$. Note that we can always recover the 3D plane normal $(a, b, c) = (\frac{a_l}{N}, \frac{b_l}{N}, \frac{c_l}{N})$ and the distance $d = \frac{\beta}{N}$ where

$$N = \sqrt{a_l^2 + b_l^2 + (\frac{c_l}{f})^2}$$

is a normalization constant, and hence we recover the original parameterization $\theta = \{a, b, c, d\}$.

IV. A Markov Chain Monte Carlo Approach to Surface Reconstruction

In this section we describe our use of the generalized Swendsen-Wang algorithm [3], an MCMC-based method, to sample from the correct posterior distribution over depth segmentations for an input RGBD image. Because, as explained in Section III-E, the super-pixel plane fitting criterion is now linear, evaluating the merger of different super-pixels is computationally very efficient. This allows us to quickly sample from the posterior distribution over partitions, which is needed in the Swendsen-Wang sampler.

A. Markov Chain Monte Carlo

Markov Chain Monte Carlo or MCMC [19] is a principled way of exploring a high-dimensional probability distribution. This class of algorithms has become a standard tool in both the statistics and machine learning literature, and is gaining in popularity in computer vision as well (the Marr prize at ICCV 2003 went to an MCMC paper [20]). The challenge lies in making these methods efficient and making sure that they are not trapped in local minima. This can be done by exploiting the structure of the problem in devising appropriate MCMC moves.

At present, we are interested in obtaining samples from the posterior probability $P(S|Z)$ over segmentations S given the observations Z, i.e., the RGBD image. A super-pixel segmentation $S = \{S_1, \ldots, S_{|S|}\}$ is defined as a partition over the super-pixels, where each of the subsets is assumed to be part of a single planar segment with parameters θ_i. Note that $P(S|Z)$ is a discrete probability distribution and is a special case of Bayesian model selection: we are interested how many planes there are and what super-pixels they are made of. In this, we will treat the continuous plane parameters for each segment (made out of several super-pixels) as nuisance parameters to be integrated out. This is described below.

B. Rao-Blackwellization

To overcome the challenge of sampling over the joint probability of both high-dimensional plane parameters Θ *and* segmentations S, we marginalize out the plane parameters,

$$\begin{aligned} P(S|Z) &\propto P(S)P(Z|S) \\ &= P(S)\int_{\Theta} P(\Theta|S)P(Z|S,\Theta) \end{aligned} \tag{4}$$

where $P(S)$ is a prior over the possible segmentations S, and $P(\Theta|S)$ is a prior on the continuous plane parameters Θ given a segmentation S. Note that, in the results, we use uniform priors for both. We assume that, given a segmentation S, the measurements Z_i for each segment S_i are conditionally independent, hence we can factor Equation 4 as:

$$P(S|Z) \propto P(S)\prod_{S_i \in S}\int_{\theta_i} P(\theta_i|S_i)P(Z_i|S_i,\theta_i) \tag{5}$$

The integral above is in general intractable, but since the measurement model is linear and the measurement noise is assumed normally distributed, the non-normalized posterior $P(\theta_i|S_i)P(Z_i|S_i,\theta_i)$ will also be normal, as long as the prior is normal (or improper). Hence, given the *maximum a priori* estimate θ_i^* for segment S_i, we can analytically compute it,

$$\int_{\theta_i} P(\theta_i|S_i)P(Z_i|S_i,\theta_i) = \exp\{-\tfrac{1}{2}E(\theta_i^*; S_i, Z_i)\}\sqrt{|2\pi\Sigma_i|} \tag{6}$$

where $E(\theta_i^*; S_i, Z_i)$ is the residual at θ_i^*, and Σ_i is the posterior covariance. It is interesting to note the balance between the error terms $\exp\left\{-\frac{1}{2}E(\theta_i^*; S_i, Z_i)\right\}$ and the information terms $\sqrt{|2\pi\Sigma_i|}$, which avoids over-fitting the data while also maintaining the desired accuracy.

Finally, combining Equations 5 and 6, we can write the target distribution $P(S|Z)$ over segmentations S as

$$P(S|Z) \propto P(S) \prod_{S_i \in S} \exp\left\{-\frac{1}{2}E(\theta_i^*; S_i, Z_i)\right\} \sqrt{|2\pi\Sigma_i|}$$

C. Swendsen-Wang Sampling

The key idea that distinguishes the Swendsen-Wang algorithm from other MCMC methods is the use of more powerful proposal densities that alter the labels of multiple sites simultaneously. In the RGBD scene reconstruction problem, we will propose to flip the label of multiple super-pixels together. Below we follow mostly the exposition from [3].

Let $C_j = \{C_j^0, \ldots, C_j^n\}$ be the set of components of segment V_j where a component C_j^k is a set of super-pixels $S_i \in V_j$. At each iteration, for each segment V_j, a set of components C_j is created stochastically based on the normalized probability $q_{i,i+1}$ for the similarity of the two super-pixels S_i and S_{i+1}:

$$q_{i,i+1} = \exp\{-w_1 acos(n_i n_{i+1}) - w_2|d_1 - d_2|\} \quad (7)$$

where (n_i, d_i) represents the normal and the distance from the origin for the best fit plane for super-pixel S_i and (w_1, w_2) are two calibrated constants, both set to 4.0, in the implementation.

Once the set of components $\mathbf{C}$ are created from the segmentation S, a component C^* is chosen randomly and either merged with a neighboring subgraph or transformed into a new subgraph. We define the proposal density of a component C^* to be merged with a segment V_j in a segmentation S as:

$$q(V_j|C^k, S) = \begin{cases} a & \text{if } V_j \text{ is adjacent to } C^* \\ b & \text{otherwise} \end{cases} \quad (8)$$

where the constants (a, b) are chosen to be $a = 0.5$ and $b = 0.5$. Note that the random choices of the component, the operation on the component and the merged subgraph can also be based on the best-fit plane parameters.

Let S^B be the new segmentation retrieved from S^A by splitting the component C^* from segment $V_1 \in S^A$, and merging it with or transforming into segment $V_2 \in S^B$. The acceptance ratio is defined as

$$\alpha(S^A \to S^B) = \frac{\prod_{e \in E(C^*, V_1 - C^*)}(1 - q_e)\, q(V_1|C^*, S^B)\, p(S^B|Z)}{\prod_{e \in E(C^*, V_2 - C^*)}(1 - q_e)\, q(V_2|C^*, S^A)\, p(S^A|Z)} \quad (9)$$

where $E(C^*, A)$ is the set of cut edges between the super-pixel sets C^* and A, and q_e for the edge $e = \{S_i, S_{i+1}\}$ is the edge probability q_{ij} defined in Equation 7.

D. Hierarchical Fitting

A key computation above is the evaluation of the target density when multiple super-pixels are merged. Once the local plane parameters for the super-pixels are known, the goal is to fit planes to different combinations of super-pixels and determine the most probable plane parameters as shown in Equation 1. Given two super-pixels, V_i and V_{i+1} with the disparity data Z_i and Z_{i+1}, let θ_1 and θ_2 be the best-fit plane parameters with the common plane model k. The goal is to find the plane parameters θ with model k for the combined set $Z = Z_i \cup Z_{i+1}$ to decide whether the super-pixels should be merged.

Let A_1 be the input to the measurement function and b_1 be the measured disparity values for the data Z_1 such that $\theta_1 = (A_1^T A_1)^{-1} A_1^T b_1$ as the solution to the least square function in Equation 3. Similarly, the second plane parameters $\theta_2 = (A_2^T A_2)^{-1} A_2^T b_2$ where A_2 and b_2 are retrieved from Z_2. The goal is to compute $\theta = (A^T A)^{-1} A^T b$ where $A = [A_1, A_2]$ is the concatenation of the measurement inputs and $b = [b_1, b_2]$ is the concatenation of the measurements.

The inverse depth parameterization and the subsequent linear measurement function $h(\theta) = A\theta$ opens the possibility of reusing the latent variables in the computations of θ_1 and θ_2 to quickly compute θ. Note that the multiplications $A_1^T A_1$ and $A_1^T b_1$ take considerable time with thousands of pixels. However, these values can be reused as $A^T A = A_1^T A_1 + A_2^T A_2$ and $A^T b = A_1^T b_1 + A_2^T b_2$, leading to significant time savings over a nonlinear measurement function.

In general, the plane parameters of model k for a segment S with n super-pixels $\{V_1, \ldots, V_n\}$ can be computed in $O(n)$ linear time as:

$$\theta = (\sum_i A_i^T A_i)^{-1} \sum_i A_i^T b_i \quad (10)$$

where $A_i^T A_i$ and $A_i^T b_i$ are latent variables from previously fitting k-model planes to the n super-pixels.

V. Experiments

We evaluate our plane fitting and surface reconstruction algorithms in the following two sections with simulated and real-world data.

A. Plane fitting comparison

In this section we compare our method of fitting planes with inverse depth parameterization to SVD fitting. Since the comparison requires accurate ground-truth we chose to use simulated data for this purpose. As discussed in the previous sections the noise of the data acquired from a time of flight sensor is in the direction of the incident ray. In our simulation we use the same principle and noise is added along the viewing direction of each point. We show how this noise affects the 3D points in Figure 3.

Figure 3. A vertical plane with simulated noise. The noise levels from right to left are 0.1, 0.5 and 1.0.

The figure shows an example of a vertical plane viewed from the side with different noise levels. It is quite evident that the points disperse in the viewing direction as the noise variance is increased. We use these noisy measurements to fit a plane using our formulation . The same noisy measurements are also converted to 3D points and we use SVD to find the best fit plane parameters for the 3D points. We use the Kinect camera configuration to convert the disparity values to 3D points to simulate conditions as close to real world data as possible.

We do a quantitative comparison of the two techniques and show that for time of flight sensor data or stereo data IDP formulation is better than SVD. In our simulation we chose a random plane and a stereo configuration similar to the Kinect and we add noise to the disparities. We compare the plane parameters obtained from both methods with the ground-truth. We define two error measures for comparison, one measure for normal and another for distance. The two metrics are defined as follows:

$$error_{Normal} = acos(N_1.N_2)$$

$$error_{distance} = abs(\delta_1 - \delta_2)$$

where $N_1$and N_2 are the unit normals of the two planes to be compared and $\delta_1$and δ_2 are the distances to the planes from the origin. We compare the two methods for various noise levels and show that the average error is notably less for our method(IDP) compared to SVD. Figure 4 shows the errors computed for both methods as we increase the noise variance. Even though the noise is increased IDP is very robust whereas SVD starts diverging relatively quickly.

B. Sampling

For our experiments we used a turtlebot [21] which comes with the kinect device mounted. With the help of Robot Operating System(ROS) and OpenNI drivers we were able to capture data and use our formulation to get reconstructions. Although our formulation uses disparity values, the present OpenNI driver does not give access to the raw disparity values. So, we take the depth values from OpenNI and convert back to disparities. While collecting the data our motivation was to pick various scenarios that will challenge the sampling algorithm. The collected data consists of scenarios such as: hallways/corridors, cubicle, meeting lounge, blocks on a table and general office spaces. This set of choices define scenarios with varying number of surfaces, types of surfaces and their relationship with each other, varying illumination conditions and also surfaces at different orientations and depths. We wanted to show that our approach works well on these typical yet challenging scenarios. In all of the results demonstrated below we included four images. In the top row we have the original input image followed by the over-segmented image using the over-segmentation approach described in the previous section. In the bottom row we have the corresponding over-segmented point cloud and the final sampler output respectively.

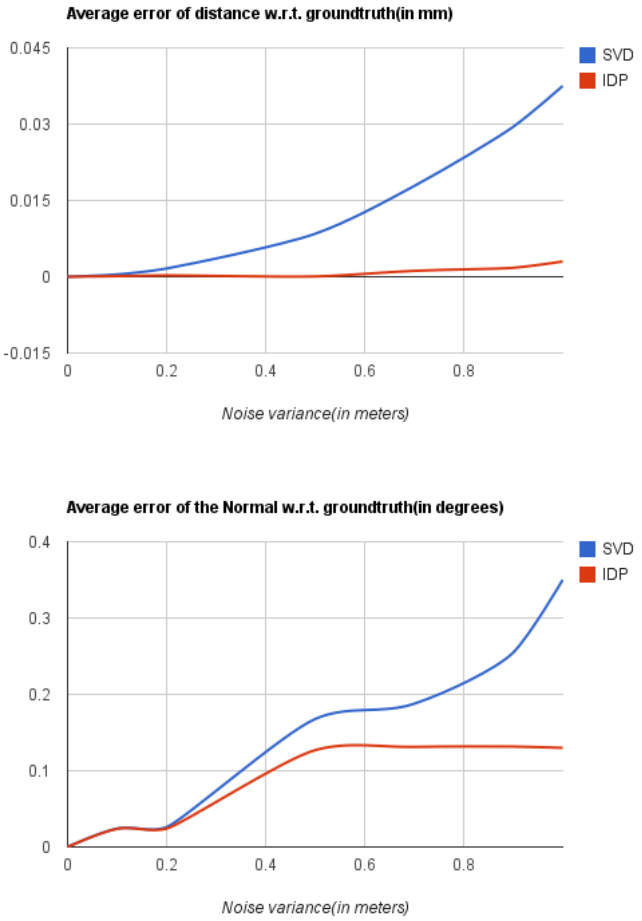

Figure 4. Comparison with SVD

For over-segmentation we have four free paramters, α, β, γ and the threshold τ used in the Felzenszwalb segmentation algorithm. By varying these paramters we can control the number of superpixels in the over-segmentation. The bigger the threshold τ the fewer super-pixels and the faster the sampling algorithm. But, the tresholds need to be carefully picked as there is a possibility of superpixel containing points from multiple planes. For our experiments we chose the values $\alpha = 1$,$\beta = 10$, and $\gamma = 5$ and $\tau = 100$. These values gave us optimal number of super-pixels at the same time making sure that coplanarity is preserved.

Figure 6 demonstrates the segmentation outputs for two different scenarios where the first two columns are on a blocks world experiment with three blocks and the second two are on a real world data set. It is evident that the the reconstruction gets horizontal, vertical and most of the general planes in the scene. In few cases it gets some spurious planes and also combines multiple planes. We observe that as the number of iterations increase, the detail of planar segmentation also increases. With a small number of iterations, our algorithm captures the coarse geometry of the scene and higher levels of detail can be achieved with more iterations.

VI. Conclusion and Future Work

This paper makes three contributions to the field of scene reconstruction with RGBD sensors First, we use the correct formulation of the 1D noise model of a structured light sensor. Second, by fitting planes to super-pixels linearly, we lay the basis for an efficient, MCMC-driven inference over planar segmentations. And third, we apply an advanced MCMC method, the generalized Swendsen-Wang sampling algorithm from [3], to quickly sample over partitions of super-pixels, flipping multiple super-pixels simultaneously for faster convergence.

The current sampler demonstrates what is possible but is still too slow. In future work, we would like to dramatically speed up the SW sampler to achieve real-time performance, mainly by devising smarter, data-driven proposals, and by fine-tuning the implementation. In addition, MCMC sampling also lends itself very well to sampling over segmentations that include different surface categories, such as vertical, horizontal, and planes in general orientation, but also other surface-types such as cylinders, spheres, and general b-splines. Our long term goal is to perform multi-robot simultaneous localization and mapping where the extracted surface parameters will be the main source of communication and registration.

References

[1] M. Bleyer and M. Gelautz, "Graph-based surface reconstruction from stereo pairs using image segmentation," in *Proc. SPIE*, vol. 5665. Citeseer, 2005, pp. 288–299.

[2] T. Tang, W. Lui, and W. Li, "A lightweight approach to 6-dof plane-based egomotion estimation using inverse depth," in *ACRA*, 2011.

[3] A. Barbu and S.-C. Zhu, "Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities," *PAMI*, vol. 27, no. 8, pp. 1239–1253, August 2005.

[4] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *Intl. J. of Computer Vision*, vol. 59, pp. 167–181, 2004.

[5] R. Bolle and B. Vemuri, "On three-dimensional surface reconstruction methods," vol. 13, no. 1, pp. 1–13, 1991.

[6] G. Storvik, "Bayesian surface reconstruction from noisy images," *Interface*, vol. 96, 1996.

[7] J. Strom, A. Richardson, and E. Olson, "Graph-based segmentation for colored 3d laser point clouds," in *IROS*. IEEE, 2010, pp. 2131–2136.

[8] C. J. Taylor and A. Cowley, "Fast scene analysis using image and range data," in *ICRA*, may 2011, pp. 3562 –3567.

[9] F. Han and S. Zhu, "Bayesian reconstruction of 3d shapes and scenes from a single image," in *Higher-Level Knowledge in 3D Modeling and Motion Analysis, 2003. HLK 2003. First IEEE International Workshop on*. IEEE, 2003, pp. 12–20.

[10] D. Terzopoulos, "Multilevel computational processes for visual surface reconstruction," *Computer Vision, Graphics, and Image Processing*, vol. 24, no. 1, pp. 52–96, 1983.

[11] P. Kovesi, "Shapelets correlated with surface normals produce surfaces," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 994–1001.

[12] A. Dick, P. Torr, and R. Cipolla, "A Bayesian estimation of building shape using MCMC," in *ECCV*, 2002, pp. 852–866.

[13] P. Qiu, "Jump-preserving surface reconstruction from noisy data," *Annals of the Institute of Statistical Mathematics*, vol. 61, no. 3, pp. 715–751, 2009.

[14] M. Kaess, R. Zboinski, and F. Dellaert, "MCMC-based multiview reconstruction of piecewise smooth subdivision curves with a variable number of control points," in *ECCV*, vol. 3023. Springer, 2004, pp. 329–341.

[15] Z. Tu, S.-C. Zhu, and H.-Y. Shum, "Image segmentation by data driven markov chain monte carlo," in *ICCV*, 2001.

[16] R.H.Swendsen and J.S.Wang, "Nonuniversal critical dynamics in Monte Carlo simulations," *Physics Review Letters*, vol. 58, no. 2, pp. 86–88, 1987.

[17] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," in *ACM Transactions on Graphics*, vol. 23, no. 3. ACM, 2004, pp. 664–672.

[18] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.

[19] W. Gilks, S. Richardson, and D. Spiegelhalter, Eds., *Markov chain Monte Carlo in practice*. Chapman and Hall, 1996.

[20] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu, "Image parsing: Unifying segmentation, detection and recognition," in *ICCV*, 2003, pp. 18–25.

[21] B. Gerkey and K. Conley, "Robot developer kits [ros topics]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 3, pp. 16–16, 2011.

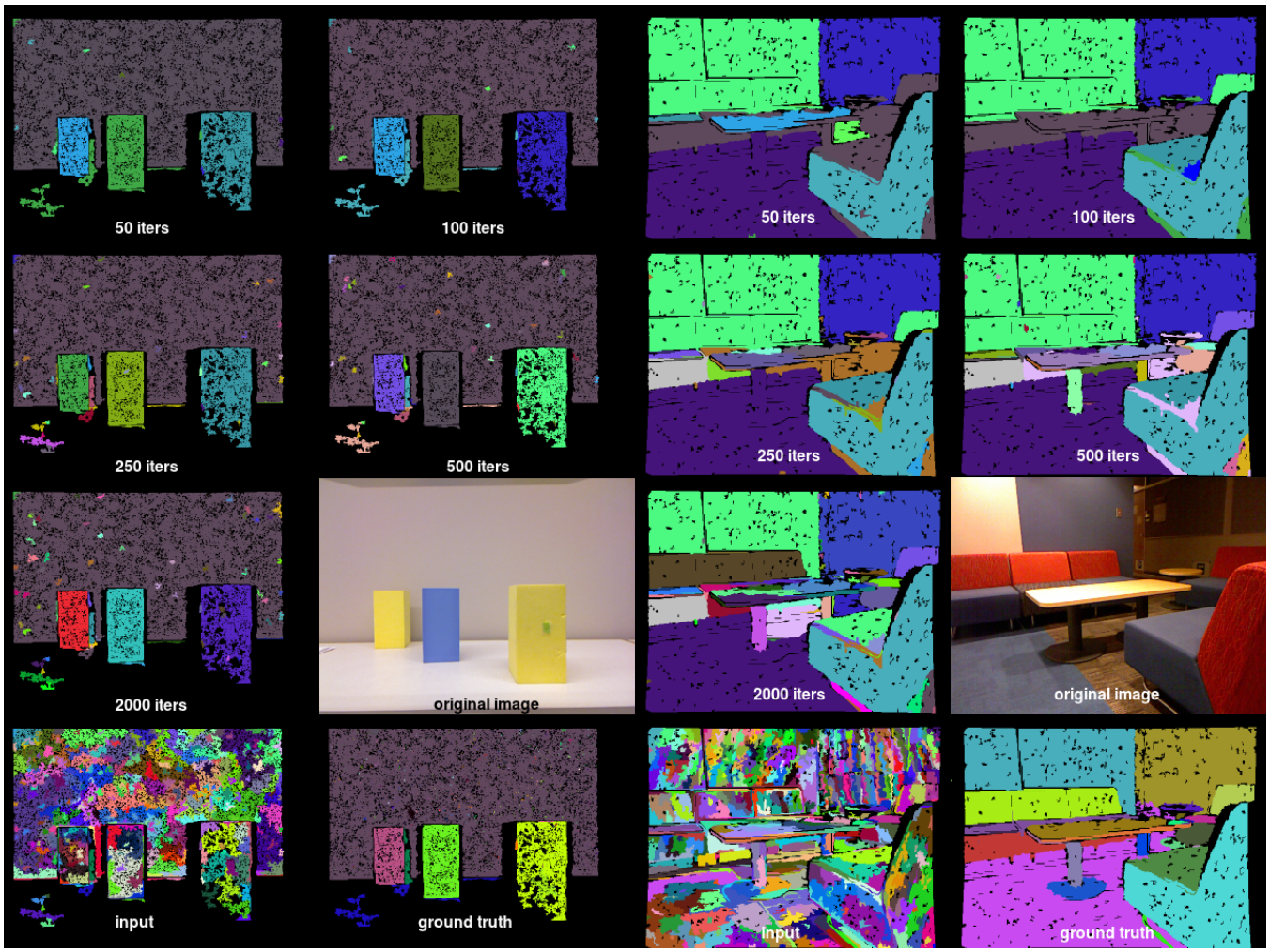

Figure 5. The two real world data sets and the effect of number of iterations in the captured level of detail.