Efficient Reconstruction of Complex 3-D Scenes from Incomplete RGB-D Data

Sergio A. Mota-Gutierrez, Jean-Bernard Hayet, Salvador Ruiz-Correa,
and Rogelio Hasimoto-Beltran

Center for Research in Mathematics, Computer Science Department
Guanajuato, Gto, México
`samota@cimat.mx`

Abstract. In this paper we develop a new approach for reconstructing 3-D scenes from RGB-D data. We use a Markov random field to model appearance relations and geometric cues between different regions of a scene, as a means to provide robustness to noisy and incomplete data often generated by RGB-D devices. A parametric reconstruction of 3-D scenes that enable coherent physical interaction are computed, in near real time, with a standard computer that does not use specialized hardware.

1 Introduction

In recent years, automated reconstruction of 3-D scenes has been focus of active research in the vision community, due to the wide spectrum of applications in areas such as robotics, environment modeling, and augmented reality, among others. Scene reconstruction methods using vision sensors, which often build a representation of a scene from sets of images or a video stream, have gained popularity due to their non-intrusive nature and increased computational capabilities of new sensing technologies [1,2,3,4].

Early reconstruction techniques were designed to build a representation of the scene as a sparse set of 3-D points, which was mainly used for camera localization purposes [5]. The problem of building a visually meaningful representation was addressed later on by densifying a sparse reconstruction [6].

After a decade of advances in monocular techniques for dense reconstruction, the surge of new sensing technologies, able to capture depth maps as well as RGB images, has opened-up new horizons for scene reconstruction, since the depth ambiguity in monocular cameras is no longer a problem. These technologies together with greater processing capabilities, largely due to the popularization of the Graphical Processing Units (GPU's), have allowed significant improvements on the reconstruction quality. However, most approaches are designed to build a non-parametric reconstruction, which are not suitable for applications such as augmented reality, in which user interaction capabilities are required. In order to overcome this limitation, we consider the problem of efficient parametric scene reconstruction using a standard RGB-D camera. For this purpose we develop a

new Markov-random-field-based approach that takes into consideration appearance relationships and geometric clues among different regions of a 3-D scene. The approach is robust to noisy incomplete data often generated by RGB-D sensors, and is able to compute, in near real time and without the use of specialized hardware, a reconstruction that is visually pleasant and can be readily used in applications that require user interaction.

The reminder of the paper is organized as follows. Section 2 describes a brief summary of related work. Our proposed approach is developed in Section 3. The performance of our algorithm is presented in Section 4. A discussion of the results is elaborated in Section 5. Section 6 concludes the paper.

2 Related Work

Early visual scene reconstruction systems consisted of a visual Simultaneous Localization And Mapping (VSLAM) algorithm integrated into a robot that moved within the bound of a planar environment [5]. This approach had to address the fundamental problem of depth information loss due to the projective nature of the monocular camera, which provided the input stream to the VSLAM algorithm. This problem was partially addressed by developing techniques that are capable to find reliable salient points in the acquired image frames in order to perform multiple-view depth inference based on filtering [1] or bundle adjustment [2].

VSLAM systems became more flexible by allowing camera movements with various degrees of freedom [1]; however, the resulting scene reconstructions often resulted in a sparse set of unstructured 3-D points. More recent approaches based on structure from motion (SFM) have been successful in large scale reconstructions of 3-D scenes [7]. These techniques use efficient algorithms for batch processing of thousands of images. Other state-of-the-art methodologies are able to reduce the sparsity of the scene representation while maintaining real-time performance. This is the case of [2], where the localization and mapping tasks, required to compute the camera location and to update the scene reconstruction, were decoupled in order to improve performance.

Densification methods for sparse scene reconstructions has been recently successfully applied in [6]. These techniques are possible due to the increased computational power and availability of graphical processing units (GPU's), which allow efficient execution of computationally intensive algorithms. For instance, in [4], the photometric consistency between the scene representation and the acquired image frames is obtained through the generation of multiple virtual scene views.

The development of new RGB-D cameras that provide depth measurements as well as calibrated color images, has solved the depth loss problem inherent to monocular sensors. The use of these devices has resulted in remarkable reconstruction results [3]. Although these reconstructions are visually pleasant, they are composed of a dense set of 3-D points. For this reason, it is difficult to use these type reconstructions in applications that require coherent physical user interaction.

In order to allow such interaction capabilities, a representation where efficient computations can be performed is desirable. In this regard, structured representations are more adequate. For example, an offline approach that successfully computes a structured representation of outdoor scenes was developed by Saxena and collaborators in [8]. In this work, a single monocular camera was used as the input sensor, and a supervised depth learning strategy was followed in order to create a triangular surface mesh of the scene. Saxena’s learning approach assumes that outdoor scenes feature significant depth variations that can be accurately encoded by an appearance descriptor.

Our proposed approach is motivated by Saxena’s work; however, we make use of incomplete depth information provided by an RGB-D camera, as well as appearance relationships and geometric clues among different regions of a 3-D scene in order to create a structured scene representation that is computed in near real time and without the use of specialized hardware.

3 Scene Reconstruction as a Set of Planes

Our scene reconstruction method aims at obtaining a parametric representation of a scene from a single RGB-D image view. The scene is approximated by a set of planes whose extension is determined via image segmentation. Approximating a scene using a set of planar surfaces seems reasonable for man-made indoor scenes. Moreover, objects with complex shapes can still be well approximated by using enough planar surfaces. In fact, planar approximation is the modeling technique most widely used in computer graphics.

3.1 Data Acquisition

We use a Kinect sensor to acquire scene images. Kinect is able to capture a color image I_c as well as a depth map I_d from the scene. The Kinect depth sensor has a measurement range about $[0.4, 8.0]$ meters. Objects out of this range can not be sensed. Moreover, accurate measurements are expected only in the range $[0.4, 3.0]$ meters. In the remaining range, $[3.0, 8.0]$ meters, measurements can be obtained, but their accuracy decays as the depth increases.

An example of the images that can be acquired with this sensor is shown in Figure 1. Note in Figure 1(b) the existence of black regions in the acquired depth map. These regions correspond to invalid depth values; that is, regions where the Kinect sensor could not estimate depth properly. This situation can be due to several reasons, among which the most common are: objects out of the sensing range or objects having specular surfaces. Moreover, note the complete absence of valid depth values on the top and right sides of the depth map. This is because the sensors used to capture the color image I_c and the depth map I_d are not located at the same position, therefore, they sense slightly different parts of the scene.

Fig. 1. Images acquired using the Kinect sensor. (a) Color image (I_c). (b) Depth map (I_d). Lighter colors indicate farther points.

3.2 Planes Parametrization

Our method reconstructs a scene as a set of planes. Planes are parametrized by a vector $\mathbf{n} \in \mathbb{R}^3$: a point $\mathbf{p} \in \mathbb{R}^3$ lies on the plane if and only if $\mathbf{p}^T \mathbf{n} = 1$. A property of this parametrization that will be useful later is that the distance d from a plane with parameters $\mathbf{n}$ to the coordinate frame origin along the direction given by a unitary vector $\mathbf{r} \in \mathbb{R}^3$, also called the ray $\mathbf{r}$, is given by $d = \frac{1}{\mathbf{r}^T \mathbf{n}}$. An illustration of this setup is shown in Figure 2.

3.3 Planes Detection

In order to compute planes that compose a scene, given an RGB-D image, we first over-segment the color image I_c into a set of regions with similar appearance. Hereafter, we refer to these regions as *image segments*. We assume that each image segment lies on a plane in the scene. Our reconstruction problem is then to estimate the parameters ($\mathbf{n}_i$) for the planes where the image segments lie.

In this work, we have used a modified version of the graph-cut based algorithm presented in [9] to perform the image segmentation. The original algorithm builds a graph whose vertices are the pixels in the image and its edges have weights corresponding to a measure of the dissimilarity between each pixel and its neighbours. The modified algorithm also incorporates the available depth data (when

Fig. 2. Illustration of some properties of the plane parametrization

available) into the segmentation, so that the dissimilarity $\Delta_{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2}$ between two neighbouring pixels $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$ is computed by

$$\Delta_{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2} = \alpha_c \Delta_{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2}^c + \alpha_d \Delta_{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2}^d,$$

where $\Delta_{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2}^c$ and $\Delta_{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2}^d$ are the dissimilarity in the color image and the depth map, respectively. The values α_c and α_d are parameters that control the importance of each data source in the segmentation.

Color dissimilarity is computed as the L_2 norm between pixels chrominance,

$$\Delta_{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2}^c = \|I_c(\hat{\mathbf{x}}_1) - I_c(\hat{\mathbf{x}}_2)\|_2,$$

where $I(\hat{\mathbf{x}})$ is the intensity of pixel $\hat{\mathbf{x}}$ in the image I .

To handle invalid measures in I_d , the depth dissimilarity is computed by

$$\Delta_{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2}^d = \begin{cases} |I_d(\hat{\mathbf{x}}_1) - I_d(\hat{\mathbf{x}}_2)| & \text{if } \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2 \in \mathcal{D} \\ 0 & \text{if } \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2 \notin \mathcal{D} \\ \Delta_0^d & \text{otherwise,} \end{cases}$$

where $\mathcal{D}$ is the set of pixels in I_d having a valid depth value, and $\Delta_0^d > 0$ a constant value. This constant value is used because, for the Kinect sensor, invalid depth values often occur along the boundaries between objects (see Figure 1(b)) which we want to be assigned to different image segments.

The inclusion of depth data leads to better segmentation results in regions presenting depth discontinuities, as shown in Figure 3. Note in the figure the improved detection of objects boundaries (Figure 3(c)), compared with the segmentation using color data only (Figure 3(b)).

Our reconstruction approach makes use of a neighbourhood between image segments. For an image segment i , its neighbourhood $\mathcal{N}_i$ is defined as the set of all image segments adjacent to i , as illustrated in Figure 3(d).

3.4 Scene Observations

In order to estimate the plane parameters ($\mathbf{n}_i$), we get scene points locations samples. These samples are easily obtained from the depth map I_d as

$$\mathbf{x} = z\mathbf{K}^{-1}\hat{\mathbf{x}}, \quad (1)$$

Fig. 3. Segmentation results. (a) Original image. (b) Segmentation using color data only. Each color represents a region in the image with uniform appearance. (c) Segmentation using both color and depth data. (d) Illustration of the neighbourhoods, i.e. all the image segments adjacent to one image segment. (Best viewed in color)

where $\mathbf{x}$ is the location of a scene point in the direction of pixel $\hat{\mathbf{x}}$, z the depth of the scene point, and $\mathbf{K}$ the camera intrinsic parameters matrix (assumed known). This way, we get the set of scene points $\mathcal{P}_i$ associated to each image segment i .

For an RGB camera, the calibration procedure has been widely studied and the calibration can be performed straightforwardly using well known methods [10]. However, the calibration of RGB-D cameras, as the one we use, is still subject of active research. This calibration involves finding the intrinsic parameters for both color and depth cameras as well as the rigid transformation between them. The Kinect sensor is calibrated during its manufacturing process, but we have found this calibration inaccurate for our purposes. Therefore we use the method proposed by [11] to calibrate the Kinect camera pair.

3.5 Plane Parameters Estimation

As a means to estimate the parameters of the planes, we look for planes that explain the observed scene points. Due to the noise in the observations and to the existence of regions without valid observations, using only the observed data leads to noisy reconstructions. In order to improve the reconstruction quality, we establish relations between planes corresponding to neighbouring image segments. These relations are intended to selectively enforce connectivity and co-planarity. Our goal is to obtain visually pleasant reconstructions, so that we use both appearance and geometric data to compute indices of the co-planarity and connectivity between the planes. These indices are used to weight the effect of the relations. For two adjacent image segments, we enforce:

1. connectivity, by penalizing the distance between points on the boundary of the planes, and
2. co-planarity, by penalizing the distance between points lying in the intersection of the planes and rays passing by the interior of the image segments.

Consider $\mathbf{n}$ as a matrix holding the parameters for *all* planes in a scene as rows, $\mathbf{p}$ as a matrix holding all observed scene points, and $\mathbf{q}$ (resp. $\mathbf{v}$) as a matrix holding appearance (resp. depth) descriptors of the planes. To capture the relations between plane parameters and the mentioned properties, we formulate a MRF as

$$P(\mathbf{n}|\mathbf{p}, \mathbf{q}, \mathbf{v}) = \frac{1}{Z} \prod_i f_1(\mathbf{n}_i|\mathbf{p}_i) \prod_{i,j} f_2(\mathbf{n}_i, \mathbf{n}_j|\mathbf{q}_i, \mathbf{q}_j, \mathbf{v}_i, \mathbf{v}_j) \quad (2)$$

where $\mathbf{p}_i$ is a matrix containing all scene points in the image segment i , i.e., $\{\mathbf{p}_{i,k}|\mathbf{p}_{i,k} \in \mathcal{P}_i\}$, and $\mathbf{q}_i$ and $\mathbf{v}_i$ are matrices containing appearance and depth descriptors of the plane i .

The first term $f_1(\cdot)$ in Equation 2 explains the observed scene points by penalizing the deviation of the scene points locations from the plane, that is

$$f_1(\mathbf{n}_i|\mathbf{p}_i) = \exp \left\{ -\frac{1}{|\mathcal{P}_i|} \sum_{\mathbf{p}_{i,k} \in \mathcal{P}_i} |\mathbf{p}_{i,k}^T \mathbf{n}_i - 1| \right\}.$$

The second term $f_2(\cdot)$ in Equation 2 models the relations between the plane parameters for neighbouring image segments. This term is defined as

$$f_2(\cdot) = \prod_{\{i,j\}|j \in \mathcal{N}_i} g_{i,j}^{\text{con}}(\cdot) g_{i,j}^{\text{cop}}(\cdot),$$

with $g_{i,j}^{\text{con}}(\cdot)$ and $g_{i,j}^{\text{cop}}(\cdot)$ selectively enforcing connectivity and co-planarity, respectively. Both terms are defined as

$$g_{i,j}^*(\mathbf{n}_i, \mathbf{n}_j | \mathbf{q}_i, \mathbf{q}_j, \mathbf{v}_i, \mathbf{v}_j) = \exp \left\{ - \sum_{j \in \mathcal{N}_i} \alpha_{i,j}^*(\mathbf{q}_i, \mathbf{q}_j, \mathbf{v}_i, \mathbf{v}_j) \sum_{\mathbf{p}_k \in \mathcal{P}_{i,j}^*} \left| \mathbf{p}_k^T \mathbf{n}_i - \mathbf{p}_k^T \mathbf{n}_j \right| \right\}, \quad (3)$$

where $\alpha_{i,j}^*(\cdot)$ measures the connectivity (or co-planarity) of the image segments i and j , and $\mathcal{P}_{i,j}^*$ is a set of pixels lying on the image segments i and j . Connectivity is enforced when the pixels in $\mathcal{P}_{i,j}^*$ are chosen close to the common boundary of the image segments. Co-planarity is enforced when the pixels in $\mathcal{P}_{i,j}^*$ are chosen from the interior of the image segments. In detail, the used parametrization ensures that $\tilde{\mathbf{p}}^T \mathbf{n}_i = \frac{1}{d_i}$, where d_i is the distance from the camera to the plane i in the direction of the ray $\tilde{\mathbf{p}}$. Then the term between absolute value in Equation 3 penalizes the fractional distances between points in the common boundary or the interior of the planes according to the selection of $\mathcal{P}_{i,j}^*$.

Plane parameters estimation being a L_1 minimization problem, we re-write as a linear problem (LP) and use an off-the-shelf LP solver [12] to solve it.

3.6 Co-planarity and Connectivity Estimation

Terms $\alpha_{i,j}^*(\cdot)$ in Equation 3 weight the co-planarity and connectivity relations between planes in the reconstruction. These terms are computed based on appearance similarity of the image segments and simple geometric estimations computed from the sensor raw data, as described hereafter.

Appearance data terms. We enforce both co-planarity and connectivity using appearance data motivated by the following intuitive reasoning: two adjacent planes with similar appearance are very likely to be “close” in the scene. Then, a measure of the similarity between the image segments can be a useful cue when building relations between image segments.

In order to define a similarity measure we compute feature vectors for the image segments. For image segment i , the feature vector $\mathbf{q}_i$ is computed by concatenating this segment filtering responses to a filter bank $\mathcal{F} = \{f_1, f_2, \dots, f_k\}$.

Motivated by the work in [8], we use the filter bank to capture color and texture information. Hence, the filter bank is composed by 9 masks, computed as all the product combinations of the following Laws’ masks [13]:

$$L_3 = [1 \ 2 \ 1]^T, \quad E_3 = [-1 \ 0 \ -1]^T, \quad S_3 = [-1 \ 2 \ -1]^T,$$

used to compute local averages and detect edges and spots, respectively.

The averaging filter obtained from $L_3L_3^T$ is applied to each channel of the color image I_c . All filters are applied to a gray scale image computed from I_c .

To form $\mathbf{q}_i$, the filter responses are averaged over the image segment i ,

$$\mathbf{q}_{i,m} = \frac{1}{|i|} \sum_{\hat{\mathbf{p}}_k \in i} |I(\hat{\mathbf{p}}_k) * f_m|,$$

where $\mathbf{q}_{i,m}$ is the m -th entry of $\mathbf{q}_i$, and $|i|$ the number of pixels in segment i .

We then compute an appearance difference measure $\Delta_{i,j}^{app}$ between two image segments i and j as the L_2 norm between the segment descriptors. The appearance similarity between the image segments is computed as

$$s_{i,j}^{app} = \exp \left(-\sigma_{app} \frac{\Delta_{i,j}^{app}}{\Delta_{max}^{app}} \right), \quad (4)$$

where Δ_{max}^{app} is the maximum appearance difference possible between two image segments, and $\sigma_{app} > 0$ is a control parameter.

The appearance similarity measure is bounded to the range $[\exp(-\sigma_{app}), 1]$. It increases as the image segments responses to the bank filter are more similar.

Geometric data. While the appearance data allows us to establish relations between image segments with similar appearance, it suffers from some drawbacks in regions not coplanar nor connected, but with similar appearance (illustrations of this situation are shown in Figure 4). In these regions, an approach based only on visual appearance would fail to capture the scene geometry.

The failure examples described above can be solved by using the depth data available. We are interested in determining whether two image segments are connected, coplanar or occluded. Consider two image segments. If they are coplanar, we expect their depth distributions to be similar. If they are connected, we expect the depth distributions of the pixels on their common boundary to be similar.

Based on these observations, we propose to adjust depth distributions to the image segments and use them to compute connectivity and co-planarity indices.

Fig. 4. Regions where wrong relations based on appearance only are likely to occur

We adjust two Gaussian distributions to each image segment. The first Gaussian $f(d; \mu_i, \sigma_i)$ is adjusted using the observed depth of all pixels in the image segment. The second Gaussian $f(d; \mu_{i,j}, \sigma_{i,j})$ is adjusted using the observed depth of pixels near to the common boundary of the neighbouring image segments i and j , lying in the image segment i . Then, for two image segments i and j , we compute the co-planarity index $\hat{\alpha}_{i,j}^{\text{cop}}$ as

$$\hat{\alpha}_{i,j}^{\text{cop}} = \exp \{ -\sigma_{\text{cop}} \Psi(f(d; \mu_i, \sigma_i), f(d; \mu_{i,j}, \sigma_{i,j})) \}, \quad (5)$$

where $\Psi(\cdot)$ is a distance between the distributions and σ_{cop} a control parameter. Similarly, $\hat{\alpha}_{i,j}^{\text{con}}$ is computed as a function of the difference between their depth distributions along the common boundary with σ_{con} a control parameter,

$$\hat{\alpha}_{i,j}^{\text{con}} = \exp \{ -\sigma_{\text{con}} \Psi(f(d; \mu_{i,j}, \sigma_{i,j}), f(d; \mu_{j,i}, \sigma_{j,i})) \}. \quad (6)$$

Information fusion. In order to take into account both appearance and geometric information sources, we weight the connectivity and co-planarity indices using the appearance similarity index as $\alpha_{i,j}^{\text{cop}} = s_{i,j} \hat{\alpha}_{i,j}^{\text{cop}}$ and $\alpha_{i,j}^{\text{con}} = s_{i,j} \hat{\alpha}_{i,j}^{\text{con}}$.

Perspective correction. In previous paragraphs, we have assumed that if two image segments are coplanar their depth distributions will be similar. However, this assumption only holds for planes seen from a front-parallel point of view. In order to satisfy our assumption regardless the viewpoint, we adjust a plane to one of the planes of interest and compute a transformation so that such plane is seen from a front-parallel perspective. Then we apply the computed transformation to the scene observations related to both planes.

4 Results

The presented approach was tested using 210 RGB-D pairs acquired in man-made indoor scenes using a Kinect sensor. As expected for man-made indoor scenes, tested scenes are mainly composed by planar surfaces; however, a significant amount of non-planar objects also appear. In order to visualize the reconstructions, we compute the convex hull for each image segment in the acquired color image. Then, for each image segment, the convex hull is projected from the image plane to its corresponding estimated plane. The plane is then coloured with the mean color intensity observed in the image segment.

Examples of reconstructions obtained using our approach are shown in Figure 5. The acquired color images are shown in the first row. Reconstructions obtained from sensor raw data are shown in the second row. These reconstructions are obtained by estimating the parameters for each plane using only observations related to its associated image segment. Several wrongly estimated planes can be observed, which pollute the reconstruction visual appearance. Such wrong estimations are mainly due to noisy observations, lack of valid depth data and sensor calibration errors. Besides, note that under our assumption that an image segment lies on a plane in the scene, a quasi-linear image segment corresponds

Fig. 5. Examples of reconstructions. (First row) Acquired color image. Reconstruction using: (Second) raw data. (Third) relations between planes based on appearance, and (Fourth) relations between planes based on appearance and depth distributions.

to a scene line. Therefore, the parameters estimation for such planes is ill-posed. Reconstructions obtained using plane relations based on appearance similarity only are shown in the third row. Observe that the established relations lead to smoothness on the reconstruction. However, such relations are wrongly established in regions with similar appearance, leading to errors in the estimated scene geometry. Finally, reconstructions obtained using our full MRF are shown in the fourth row. As in the previous case, the reconstructions exhibit smoothness and less wrongly estimated planes are observed.

In order to conduct a quantitative evaluation of the reconstruction quality, we test our approach on a scene mainly composed by three mutually orthogonal planes (first image in Figure 5). We reconstruct the scene from several viewpoints and evaluate the angles between dominant planes normals. Since the reconstruction

is not guaranteed to detect a single plane for each dominant plane, we cluster the estimated planes and compute a mean normal vector for each group. The larger error, with a magnitude of $3,3^\circ$, was estimated between the desk and the wall planes, with very similar appearance. Results are summarised in Table 1.

Table 1. Measured angles in degrees (mean - variance) for a scene composed by three mutually orthogonal planes. Plane order correspond to the white desk, the white wall and the black CPU, respectively.

Plane	Plane 1	Plane 2	Plane 3
Plane 1	—	(86,7 - 3,4)	(91,3 - 4,2)
Plane 2	—	—	(92,6 - 2,8)
Plane 3	—	—	—

Since the plane parameters estimation problem is posed as a linear program, processing time is polynomial on the image segments. For the test set an average of 378,6 image segments per image were found, and the average processing time for the reconstruction was 1682,9 ms on a i3 CPU (4GB RAM). Approximately 46% of the processing time was spent on the image segmentation and 51% on the actual estimation of the plane parameters.

5 Discussion

Our results suggest that the reduced computational complexity of our proposed approach enables real-time applications. Moreover, since the method does not requires special hardware it is potentially useful for exploration tasks. Our technique uses image segmentation to detect planes in the scene. Since we over-segment the image, the reconstruction does not depend critically in the segmentation method as long as the segmentation is “fine” enough.

For most of the image pairs on our test set, the algorithm produces visually pleasant reconstructions. Errors on the reconstructions mostly occur on object boundaries. These errors can be attributed to inaccurate sensor calibration. Calibration deficiencies result in imperfect mappings between color and depth images, making some depth observations to be assigned to wrong image segments. Although this is not a common situation, it is important because leads to planes attached to depth gaps polluting the reconstruction.

6 Conclusions and Future Work

This work developed an approach for structured scene reconstruction using RGB-D data. Although our RGB-D device outputs noisy observations and missing depth information for some regions in a scene, our proposed approach is able to compute reconstructions that are visually pleasant. The reconstruction is conducted in near real-time on a standard computer without using specialized

hardware. This, in conjunction with the parametric nature of the reconstruction, makes our technique suitable for tasks where interaction is required.

A step towards full scene reconstruction is to embed our approach into an incremental reconstruction method. Since it is not likely to have coherent segmentations for a scene along multiple viewpoints, it is necessary to establish relations between planes detected on multiple images. Moreover, in order to reduce the computational complexity of the reconstruction, a strategy to merge coplanar planes should be developed.

References

1. Davison, A.J., Reid, I., Molton, N., Stasse, O.: MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29(6), 1052–1067 (2007)
2. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: *Proc. of the IEEE and ACM Int. Symp. on Mixed and Augmented Reality* (2007)
3. Newcombe, R., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-Time Dense Surface Mapping and Tracking. In: *Proc. of the IEEE Int. Symp. on Mixed and Augmented Reality* (2011)
4. Newcombe, R., Lovegrove, S., Davison, A.J.: DTAM: Dense Tracking and Mapping in Real-Time. In: *Proc. of the IEEE Int. Conf. on Computer Vision* (2011)
5. Karlsson, N., Di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P., Munich, M.E.: The vSLAM Algorithm for Robust Localization and Mapping. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 24–29 (2005)
6. Hödlmoser, M., Micusik, B., Kampel, M.: Sparse Point Cloud Densification by Using Redundant Semantic Information. In: *Proc. of the Int. Conf. on 3D Vision* (2013)
7. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a Day. In: *Proc. of IEEE Int. Conf. on Computer Vision* (2009)
8. Saxena, A., Sun, M., Ng, A.: Make3D: Learning 3D Scene Structure from a Single Still image. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 30(5), 824–840 (2009)
9. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. *Int. Journal of Computer Vision* 59(2), 167–181 (2004)
10. Zhang, Z.: A Flexible New Technique for Camera Calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22(11), 1330–1334 (2000)
11. Daniel Herrera, C., Kannala, J., Heikkilä, J.: Joint depth and color camera calibration with distortion correction. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 34(10), 2058–2064 (2012)
12. Inc. Free Software Foundation. GLPK (GNU Linear Programming Kit) (2012)
13. Laws, K.: Rapid Texture Identification. In: *Proc. SPIE Image Processing for Missile Guidance*, pp. 376–381 (1980)