2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

# Dense Piecewise Planar RGB-D SLAM for Indoor Environments

Phi-Hung Le and Jana Košecka

*Abstract*— **The paper exploits weak Manhattan constraints to parse the structure of indoor environments from RGB-D video sequences in an online setting. We extend the previous approach for single view parsing of indoor scenes to video sequences and formulate the problem of recovering the floor plan of the environment as an optimal labeling problem solved using dynamic programming. The temporal continuity is enforced in a recursive setting, where labeling from previous frames is used as a prior term in the objective function. In addition to recovery of piecewise planar weak Manhattan structure of the extended environment, the orthogonality constraints are also exploited by visual odometry and pose graph optimization. This yields reliable estimates in the presence of large motions and absence of distinctive features to track. We evaluate our method on several challenging indoors sequences demonstrating accurate SLAM and dense mapping of low texture environments. On existing TUM benchmark [19] we achieve competitive results with the alternative approaches which fail in our environments.**

## I. Introduction

The paper exploits weak Manhattan constraints [15] to parse the structure of indoor environments from RGB-D video sequences. In our setting the structure of the scene is comprised of sets of vertical planes, perpendicular to the floor and grouped to different Manhattan coordinate frames. The problem of geometric scene parsing and the floor plan recovery, involves more than just plane fitting and identification of wall and floor surfaces, which are supported by depth measurements. It requires reasoning about occlusion boundaries, the extent of walls and their intersection in the case of missing or ambiguous depth measurements. Several works tried to infer the Manhattan scene structure using vanishing points and lines [16] and alternative volumetric constraints [8], using RGB images only [23] in a single view setting. In this work we use the Manhattan constraints for scene parsing of video sequences and visual odometry.

We adopt an approach for single view parsing from RGB-D views proposed in [20]. In this work the authors infer the 3D layout of the scenes from a single RGB-D view and pixel level labeling in terms of dominant planar structures aligned with the orientations determined by a Manhattan coordinate frame. The optimal labeling is carried out using dynamic programming over image intervals determined using geometric reasoning about presence of corners and occluding boundaries in the image. In our setting we relax the single Manhattan frame assumption and consider the set of dominant planes perpendicular to the floor, but at varying orientations with respect to each other. We further extend the approach to sequences and show how to formulate the geometric parsing recursively, by updating the single view energy function using previous parsing results. The proposed approach yields better, temporally consistent results in challenging RGB-D sequences. In addition to the estimates of piecewise planar models, we use the Manhattan constraints for estimation of visual odometry in challenging sequences with low texture and large displacements and blur. The compact global models of indoor environments are then obtained by loop closure detection and final pose graph optimization [6] enabling globally consistent models. We carry out extensive experiments to evaluate our approach.

Department of Computer Science, George Mason University, 4400 University Drive MSN 4A5, Fairfax, Virginia 22030, USA {ple13,kosecka}@gmu.edu

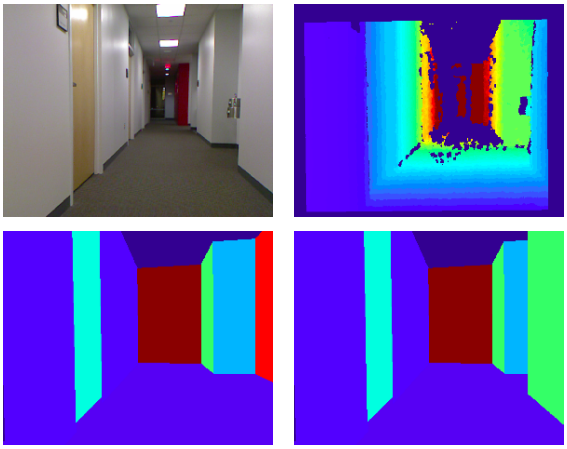

Fig. 1: Top: RGB image and the depth image; Bottom Single view parsing and temporal parsing results. Note the correctly parsed geometric structures in the right results. The colors correspond to different plane labels.

In summary, our contributions are:

- An extension of a geometric parsing approach for a single RGB-D frame to a temporal setting;
- An integration of structures inferred from the parsing step and point features to estimate accurate visual odometry, yielding drift free rotation estimates;
- These two components along with planar RGB-D SLAM, loop closure detection and pose graph optimization enable us to obtain detailed and high quality floor plan including non-dominant planar structures and doors.

## II. Related Work

This work is related to the problem of 3D mapping and motion estimation of the camera from RGB-D sequences. This is a long standing problem, where several existing solutions are applicable to specific settings [11], [12]. Many successful systems have been developed for table top settings or small scale environments at the level of individual

rooms. These environments often have a lot of discriminative structures making the process of data association easier. The camera can often move freely enabling denser sampling of the views, making local matching and estimation of odometry well conditioned. Several approaches and systems have been proposed to tackle these environments and typically differ in the final representation of the 3D model, the means of local motion computation using either just RGB or RGB-D data and the presence or absence of the global alignment step.

For the evaluation of visual odometry approaches only, Freiburg RGB-D benchmark datasets [19] are the de-facto standard. Simultaneous mapping and dense reconstruction of the environments has been successful in smaller workspaces, using a variety of 3D representations, including signed distance functions, meshes or voxel grids [11], [1], [12]. Volumetric representations and an on-line pose recovery using higher quality LIDAR data along with final global refinement were recently proposed [22], with more detailed related work discussion found within. Approaches for outdoor 3D reconstruction and mapping of outdoors environments have been demonstrated in [14].

Another set of works focuses on the use of Manhattan constraints to improve 3D reconstruction either from a single view or multiple registered views as well as 3D structure. In [5] authors focused more detailed geometric parsing into floor, walls and ceiling using stereo, 3D and monocular cues using registered views. In [4] the authors demonstrated an on-line real-time system for semantic parsing into floor and walls using a monocular camera, with the odometry estimated using a Kalman filter. The reconstructed models were locally of high quality, but of smaller extent considering only few frames. In [3] the authors proposed a monocular SLAM framework for low-textured scenes and for the ones with low-parallax camera motions using scene priors. In [2] a dense piecewise monocular planar SLAM framework was proposed. The authors detected planes as homogeneous-color regions segmented using superpixels and integrated them into a standard direct SLAM framework. Additional, purely geometric approaches assumed piecewise planarity [21] and used multiple sensing modalities to reconstruct larger scale environments. The poses and planes were simultaneously globally refined using the final pose graph optimization. These works did not pursue more detailed inference about corners and occlusion boundaries induced by planar structures and estimated the planar structures only where the depth measurements were available.

An attempt to model the world as a mixture of Manhattan frames has been done in [18] where Manhattan mixtures were estimated in the post processing stage to refine the quality of the final model. In our case we handle this in an online setting. In [17] the 3D rotation for visual odometry in an indoor Manhattan world is tracked by projecting directional data (normal vectors) on a hypersphere and by exploiting the sequential nature of the data. An effective approach for single RGB-D view parsing was proposed in [20], where optimal plane labeling was obtained using a dynamic programming approach over a sequence of image width intervals.

The presented work extends single view parsing to video sequences. We show how to change the optimization to include the information from the previous frames and relax the Manhattan assumption, by considering vertical planes perpendicular to the floor. The relative orientation between the frames is estimated from consecutive single view estimates, requiring only single 3D point correspondence to estimate the relative translation between the views.

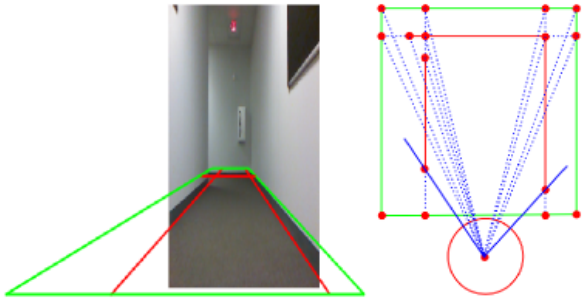

Fig. 2: Right: Bird's eye view of the line endpoints and intersections of all possible lines in the bounding box volume. Left: Intersections superimposed over image. Blue lines: the FOV of the camera. Red line segments: the projection of walls on the ground floor. Green line segments: the projection of the bounding box volume on the ground floor.

Closest to our approach is the work in [23]. The authors developed a real-time monocular plane SLAM incorporating single view scene layout understanding for low texture structural environments. They also integrated planes with point-based SLAM to provide photometric odometry constraints as planar SLAM can be easily unconstrained. Our single view parsing attains higher quality 3D models (including doors) and is tightly integrated with pose optimization and loop closure detection.

## III. Approach

### A. Single View Parsing

This paper extends the work of [20] in which authors proposed a dynamic programming solution for single view parsing of RGB-D images to video sequences. In this section, we briefly summarize their method and demonstrate its extensions for the parsing of video sequences. The method takes as an input a single RGB-D view and proceeds in the following steps. The RGB image is first over-segmented into superpixels which respect the straight line boundaries. RANSAC based plane fitting estimates the dominant planes and the associated Manhattan coordinate frame of the current view, determined by one or two vertical planes perpendicular to the floor. The intersection of the vertical planes with the floor plane defines an infinite line; a wall may contain more than one disjointed planar segment with the same normal vector and offset. The end points of the lines segments together with the intersection between pairs of perpendicular infinite lines are found. These points then determine the hypothesized wall intersections and occluding boundaries. The projections of these hypothesized intersections and occluding boundaries onto an image determine the boundaries

between the intervals and can be seen in Figure 2. The labels assigned to intervals are the identities of the dominant planes $\mathbf{l} = \{l_1, l_2, \ldots, l_k\}$, where $l_i = (n_i, d_i)$ is the plane normal and offset for one of the infinite dominant planes; $\mathbf{x} = \{x_1, x_2, \ldots, x_k\}$ is the set of intervals, with $\mathbf{x_i} = (p_i, p_{i+1})$ is a segment of a field of view. We seek to assign the most likely assignment of plane labels to the set of intervals $P(\mathbf{x}|\mathbf{z})$ given the depth measurements $\mathbf{z}$. Maximization of the probability can be rewritten as minimization of the following energy function

$$E(\mathbf{x}) = \sum_{i=1}^{n} (f_i(x_i, \mathbf{z}) + e_i(x_i, x_{i-1}, \mathbf{z}))$$

where $f_i(x_i, \mathbf{z})$ is the cost to assign label $l_i$ to the $i^{th}$ interval, and $e_i(x_i, x_{i-1}, \mathbf{z})$ is the pairwise cost of assigning to the $(i-1)^{th}$ and $i^{th}$ intervals labels $l_{i-1}$ and $l_i$, respectively. Given the set of dominant planes defining the labels, each depth measurement is assigned the most likely plane. See Figure 6 (upper left corner image) for each example. Figure 3 shows the cost computation on an interval $f_i(x_i = l_i)$ defined as the fraction of all pixels with available depth measurements inside the quadrilateral with the best label $l_i$, divided by the total number of pixels in the quadrilateral $c_1(x_i, l_i) = 1 - \frac{labelCount}{totalCount}$ ranging between 0 to 1. The cost for the intervals corresponding to virtual planes which are not supported by depth measurements and are defined by the bounding box volume is $c_1(x_i = l_i) = 0.5$. The virtual planes are color coded in red in Figure 6. The final label cost is the plane support cost weighted by the fraction of the total FOV the interval subtends $f_i(x_i = l_i) = w_i.c_1(x_i, l_i)$. The pairwise cost $e_i(x_i = l_j, x_{i-1} = l_k)$ penalizes the discontinuity when two consecutive intervals are assigned different labels. When the optimal labeling is achieved, consecutive intervals with the same label are merged. The final result is a compact wall layout of the scene. See Figure 6.

Fig. 3: Left: Best wall for each pixel. Middle: A projected quadrilateral from an interval to a wall. Right: support pixels for the wall being considered.

### B. Dynamic Programming for Sequences

In a video sequence, the local structure of the scene changes very little between two consecutive frames, yet if all the frames are parsed independently, it is easy to obtain parses which are inconsistent. This is due to low quality of the depth measurements, a large amount of missing data due to reflective structures or glass, or too-oblique angles of planar structures. The brittle nature of the raw depth data further affects the process of estimating the dominant planes, determining the intervals and the labeling. Next we describe how to introduce some temporal consistency into the parsing process and obtain a locally consistent 3D layout. We will do this by incorporating the result of the previous labeling in the optimization $P(\mathbf{x}|\mathbf{x}', \mathbf{z})$ given the depth measurements $\mathbf{z}$, with $\mathbf{x}$ and $\mathbf{x}'$ denoting the set of intervals and their labeling in the current and previous frame.

The relative pose between two consecutive frames is estimated using visual odometry, which will be discussed in Section 4.1. Two walls are associated if they have the same orientation, and if their offset difference is below a threshold of 0.05m. For a plane of the previous frame that does not have any associations, it is added to the set of labels of the current frame.

In the current frame, after all the intervals have been identified using the method described in Section 3.1, a set S of interval $\{p_0, ..., p_m\}$ is found. The layout produced for the previous scene yields a collection of intervals and endpoints, which is projected to the current frame, obtaining another set S' of endpoints. Let S' be $\{p'_0, ..., p'_n\}$ and $l'_i$ is the assigned label for the $i^{th}$ interval, $[p'_{i-1}, p'_i]$, taken from the previous layout. Now, the intervals formed by a union of the end points in S and S' and a new set of labels is given by the union of the previous and current labels after plane association. Given a new set of endpoints and the intervals they induce, we now formulate the modified label costs taking into account the results of the optimal label assignment from the previous frame.

Lets denote a new set of endpoints on the circle $\{s_0, ..., s_k\}$. When assigning label costs to a particular interval $[s_{j-1}, s_j]$, we need to consider several scenarios. First that there is an interval $[p'_{i-1}, p'_i]$ in the previous frame that completely covers it, with the previously assigned label $l_i$. The cost of assigning this label again should be lower, reflecting the increased confidence in the presence of the label in the current frame, given the previous frame. When parsing the video sequence we modify the cost function by introducing two additional costs; the fitting cost $c_2(x_i = l_j, \mathbf{z})$ and the temporal cost $c_3(x_i = l_j)$. The fitting cost $c_2(x_i = l_j, \mathbf{z})$ is the average residual for the depth measurements that lie inside the projected quadrilateral that has its best label as $l_j$. Let $\{X_1, ..., X_k\}$ be the set of these 3D points, and $l_j$ is the wall label characterized by parameters $(n_j, d_j)$.

$$c_2(x_i = l_j, \mathbf{z}) = \begin{cases} \min(\frac{\sum_i d(X_i, l_j)}{k}, 0.15) & \text{if } l_j \text{ real wall} \\ 0.5 & \text{virtual wall} \end{cases}$$

where $d(X_i, l_j)$ is the 3D point to plane distance. This cost models the scenario where there may be more then one suitable plane model for the interval, but the plane fitting process has omitted the plane selection due to missing data or ambiguities. This plane label in question was however successfully detected and labeled in the previous frame and hence it is a good candidate for explaining the depth values in the interval. The suitability of the plane is measured by the average residual error. For the temporal cost if $l_j$ is not the preferred label, a cost $c_3(x_i = l_j)$ of 0.1 is added, otherwise there is no penalty. The total cost of assignment of the label

$l_j$ to the $i^{th}$ interval is:

$$f_i(x_i = l_j, \mathbf{z}) = c_1(x_i = l_j) + c_2(x_i = l_j, \mathbf{z}) + c_3(x_i = l_j).$$

Similarly as in the single view case, the final label cost of the interval is $f_i(x_i = l_j, \mathbf{z})$ is weighted by the fraction of the FOV the interval $x_i$ subtends.

The pairwise cost is also modified to accommodate the temporal constraint. In the case that the proposed labeling introduces discontinuity of depth at the junction between the two intervals, the following penalty is applied:

$$e_i(x_i = l_j, x_{i-1} = l_k, \mathbf{z}) = \begin{cases} \delta, & \text{if } l_j \text{ is not preferred} \\ \frac{\delta}{3}, & \text{otherwise.} \end{cases}$$

If there is no discontinuity induced from the proposed labeling, then $e_i(x_i = l_j, x_{i-1} = l_k, \mathbf{z}) = 0$. We used a discontinuity cost of $\delta = 0.03$ in our experiments. Given that our state space $\mathbf{x}$ is a linear 1D chain of intervals, the optimal labeling problem can now be solved using dynamic programming as described in [20]. The recursive formula for temporal dynamic programming is similar to that in the case of single view, except for the extra temporal constraint. The results of the optimal scene parsing using single view and temporal constraints is described in more detail in the experiments.

## IV. Visual Odometry

With single view parsing we estimate the rotation of the camera with respect to the world coordinate frame $R_i^{cw}$. We omit the subscript $cw$ for clarity. Relative rotation between consecutive frames is estimated as $R_{i-1,i} = R_{i-1}^T R_i$. The relative translation is estimated using SIFT matching and RANSAC requiring only a single 3D point correspondence. In this work we assume that the environment can have multiple local Manhattan frames. This corresponds to the settings where corridors are not orthogonal to each other [15]. We develop a simple but effective mechanism to detect these new frames in an online setting and adjust the process of estimation of the relative rotation $R_{i-1,i}$ accordingly. We assume that the first view of the sequence determines the initial word reference Manhattan frame $R_w$. In subsequent frames the single view rotation estimates are composed together to yield the rotation estimate of the camera pose with respect to the world reference frame. In the case when the single RGB-D frame has multiple vertical walls which are not perpendicular to each other, we get several estimates of the local Manhattan frame for that view, lets denote them $R_i$ and $R_i'$. To determine the rotation which will yield the correct relative rotation $R_{i-1,i} = R_{i-1}^T R_i$ and $R_{i-1,i} = R_{i-1}^T R_i'$, we choose the one which yields smaller relative rotation as the motion between consecutive frames is small. We also store the angle between $R_i$ and $R_i'$ representing the alignment between two different Manhattan frames.

### A. Graph SLAM and Loop Closure Detection

The visual odometry technique described above yields very good rotation estimates even in the absence of features in the environment. When aligning the sequences for longer trajectories the system accumulates a small drift requiring global alignment step. We exploit the structures detected from single view reconstruction, such as walls, corners (the intersection between two walls) for the global alignment. We use GRAPH SLAM [6] optimization approach. Since the height of the camera is fixed and we can estimate the single view rotation, we can always assume that camera motion is planar. The pose of each frame is a node in the graph. An edge is added between any two consecutive nodes and filled with information provided by the relative translation estimated using visual odometry. Between consecutive frames, walls are associated using the joint compatibility branch and bound test [10]. The dominant walls that align with the local Manhattan frame are used to enhance the consistency between two consecutive poses. With the tracking of the planes, the enhanced constraint for the graph optimization is: $((\mathbf{t_i} - \mathbf{t_{i-1}})^T \mathbf{u})\mathbf{u} + ((\mathbf{t_i} - \mathbf{h_i}) - (\mathbf{t_{i-1}} - \mathbf{h_{i-1}}))$, where $\mathbf{v}$ is the normal vector of plane $p_i$, $\mathbf{u} = \mathbf{n} \times \mathbf{v}$ with $\mathbf{n}$ be the normal vector of the floor. $\mathbf{h_{i-1}}$ and $\mathbf{h_i}$ are the orthogonal projection of $\mathbf{t_{i-1}}$ and $\mathbf{t_i}$ on $p_{i-1}$ and $p_i$ respectively. Intuitively, this pairwise constraint enforces the co-planarity between associated walls between two consecutive frames when loop closure is run.

*Loop Closure Detection:* In our work, the loop closure detection is done by using GIST features [13] at places marked as geometric signatures such as connected pair of walls that are orthogonal to each other. Geometric signatures are found at intersections, T-junctions, and corner turns. When such keyframe is detected it is matched with previously found key frames. The association match is successful if the relative rotation between the two pairs is less than $10^o$, the distance between their locations is less than 5 meters, and finally the GIST score between the two scenes is less than 0.025.

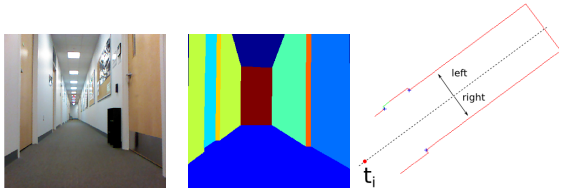


Fig. 4: RGB image and the corresponding single view reconstruction; right is the projection of the walls onto the ground floor.

### B. Final Global Map Generation

When the globally refined poses are found, the locations of walls in each frame are updated. At the end of the sequence, walls are merged frame after frame to generate the global wall maps. First walls with large extent are merged to generate a coarse map. In Figures 7 and 8, the coarse map consists of the red lines that are 1 meter or longer.

*Door detection:* To detect doors, we keep track of a set of corners detected in all the frames. These corners are the ends of the innermost walls in the left and the right of the camera, as shown in Figure 4. A wall is a door candidate if it is not wider than 1 meter (we assume that door has a

width of about 0.825 meters) and has at least 2 corners near each end (within 0.25 meters). For the small walls that do not pass the door test, we merge them with the big walls in the coarse map.

## V. Experiments

We evaluate our algorithm on several RGB-D sequences of indoor scenes with minimal texture, that satisfy Manhattan or weak Manhattan constraints. One of the sequences is from the TUM RGB-D dataset [19], *fr3/structure-notexture-far*, which is publicly available and comes with ground truth. Besides this, we collected two other sequences of large scale office corridors to test our approach.

### A. Temporal Parsing

Our experiments demonstrate that temporal parsing produced incrementally better results in scenarios where depth data was missing or noisy due to sensor limitations, or due to the difficult nature of data such as glass doors and glass walls. Our algorithm also consistently detected door planes once they had been picked up. Qualitative results of temporal parsing for five different scenarios, each consisting of three consecutive frames, are shown in Figure 6. For each frame, the top row shows the RGB image on the left, and walls aligned with the dominant Manhattan frame on the right. The bottom row shows the result of single view parsing on the left, and that of temporal parsing on the right. The first frame of each scenario is the starting frame, so the results for the single view and temporal parsing are the same. Scenario 1 demonstrates that temporal parsing consistently picked up doors while single view parsing failed. Scenarios 2 and 5 shows that temporal parsing produced better results for small walls in complex scene. An enclave area was correctly parsed in Scenario 3. In Scenario 4, after picking up the first frame, temporal parsing could infer a plane for the glass area in consecutive frames, while single view parsing assigned a virtual plane for it.

### B. Graph SLAM for weak Manhattan Indoor Environments

We show next that under the weak Manhattan assumption, graph SLAM optimization could be carried out on the positions of the cameras only, since the estimated rotations were good and drift free. For each test sequence, a global map was generated. See Figures 7 and 8. We compare our method with DVO-RGBD SLAM [7] and ORB-RGBD SLAM [9] showing superior or comparable results on several sequences.

*TUM SLAM dataset* Our algorithm focuses on SLAM for scenes with Manhattan/weak Manhattan structure without features. The sequence *fr3/snf* with available ground truth is a top candidate for our approach. We ran DVO-RGBD, ORB-RGBD, and our algorithm five times to obtain the average root mean squared error (RMSE) and the deviation. The comparison is shown in Table I. We also include the result of Pop-up Plane SLAM, taken directly from [23], which is not a standard RGB-D SLAM framework but it is relevant. The comparison is shown in Table I.

*Large Scale Indoor Office Scenes:* We do not have the ground truth for these sequences, thus only qualitative comparison is possible. As the source code for Pop-up Plane SLAM is not available, we only compare our algorithm with DVO-RGBD SLAM and ORB-RGBD SLAM. DVO-RGBD SLAM produced a meaningless trajectory for both sequences, while ORB-RGBD SLAM kept losing the point tracks and did not produce a complete trajectory for the sequences. Figure 5 shows the result of our algorithm versus DVO-RGBD SLAM. Figures 7 and 8 show the point cloud reconstruction of the two sequences using our estimated poses.

TABLE I: Results for TUM RGB-D Dataset

| | RMSE (m) |
|---|---|
| Pop-up Plane SLAM | $0.18 \pm 0.07$ |
| DVO-RGBD SLAM | $0.097 \pm 0.000$ |
| ORB-RGBD SLAM | $0.016 \pm 0.002$ |
| Ours | $0.043 \pm 0.001$ |

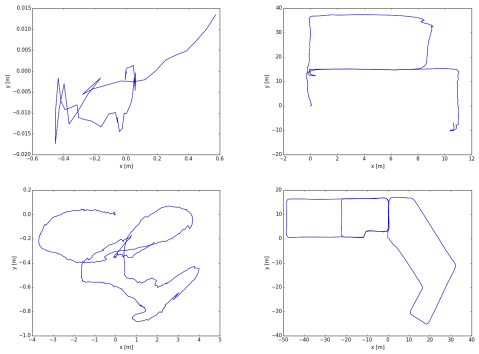

Fig. 5: Left column: DVO SLAM trajectory for the sequences. Right column: our results.

*Door detection:* We have also evaluated the door detection accuracy in two indoor office sequences. In the first environment 32 doors were detected, with 17 correct detections and 7 misses and in the second environment 86 doors were detected with correct 77 detections and 7 misses.

Note that DVO-RGBD and ORB-RGBD work well for the TUM sequence, which does not have texture, but fails on our sequences. There is a major difference between the TUM sequence and ours. Despite the blank walls in the TUM sequence many reliable point features corresponding to small wall irregularities can still be detected and tracked in high resolution images. For our indoor office sequences, the scene often consists just of blank walls and a few distinct feature points. For our case, as rotation is reliably estimated from the scene structure, only one matching point feature is needed to estimate the translation, which is not the case for DVO-RGBD and ORB-RGBD.

## VI. Conclusions

We have presented a temporal parsing algorithm that yields better, temporally consistent results in challenging RGB-D sequences. The algorithm consistently and correctly parses meaningful structures in the scene such as door planes. This

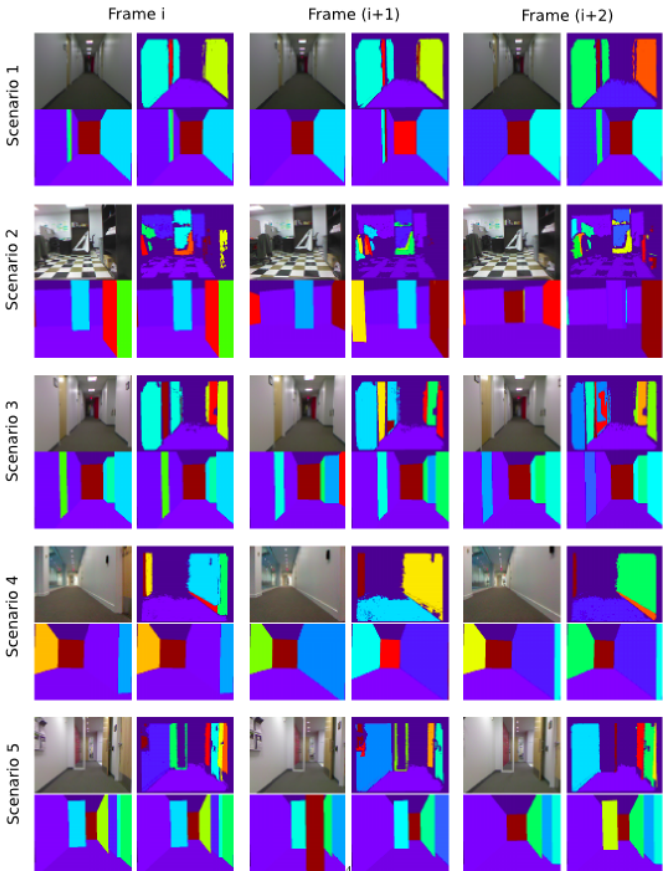


Fig. 6: Each row has three blocks showing the result of three consecutive frames. For each block, the top left is the RGB image, the top right are the walls that align with the dominant Manhattan frame, the bottom left is the result of single view parsing, and the bottom right is the result of temporal parsing.

enables an efficient on-line method to detect doors which were not propagated to the final global map. We have also introduced an efficient visual odometry algorithm that works without rotation drift in a weak Manhattan world setting. Finally, pose optimization based on the locations of the camera and the constraints obtained by matching geometric signatures between key frames provides global refinement of poses. At the end of the pipeline, a global map for the sequence is generated.

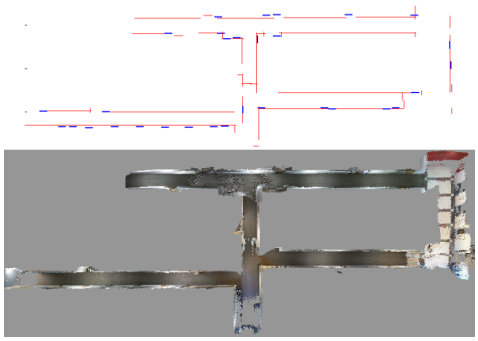

Fig. 7: Global maps and reconstructed point cloud of the first sequence (scales are not the same).

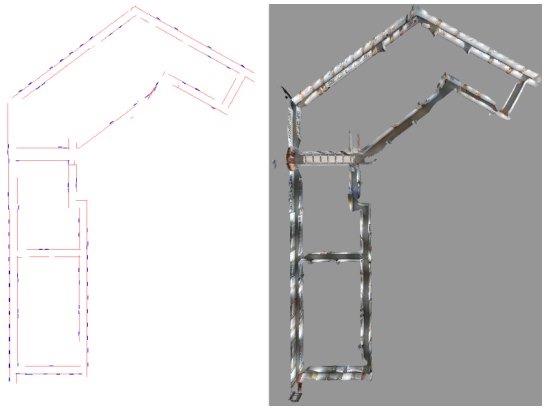

Fig. 8: Global maps and reconstructed point cloud of the second sequence (scales are not the same).

## References


[1] S. Choi, Q. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. CVPR, 2015.

[2] A. Concha and J. Civera. DPPTAM: Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence. In *IROS*, 2012.

[3] A. Concha, W. Hussain, L. Montano, and J. Civera. Incorporating Scene Priors to Dense Monocular Mapping. In *Autonomous Robots, vol. 39(3), pp. 279-292*, 2015.

[4] A. Flint, C. Mei, I. D. Reid, and D. W. Murray. Growing semantically meaningful models for visual SLAM. In *CVPR*, 2010.

[5] A. Flint, D. Murray, and I. Reid. Manhattan Scene Understanding Using Monocular, Stereo and 3D features. In *ICCV*, 2011.

[6] G. Grisetti, R. Kuemmerle, C. Stachniss, and W. Burgard. A Tutorial on Graph-based SLAM. In *ITS*, 2010.

[7] C. Kerl, J. Sturm, and D. Cremers. Dense Visual SLAM for RGB-D Cameras. In *IROS*, 2013.

[8] D. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009.

[9] R. Mur-Artal and J. Tardos. ORB-SLAM2: an OpenSource SLAM System for Monocular, Stereo and RGB-D Cameras. In *arXiv:1610.06475*, 2016.

[10] J. Neira and J. Tardos. Data association in stochastic mapping using the joint compatibility test. In *CVPR*, 2001.

[11] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. CVPR, 2015.

[12] M. Niesner, M. Zollhofer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. TOG, 2013.

[13] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001.

[14] M. Pollefeys, D. Nister, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. Kim, P. Merrell, and et al. Detailed real-time urban 3d reconstruction from video. *IJCV*, 2013.

[15] O. Saurer, F. Fraundorfer, and M. Pollefeys. Homography based visual odometry with known vertical direction and weak manhattan world assumption. In *ViCoMoR*, 2012.

[16] A. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. CVPR, 2012.

[17] J. Straub, T. Campbell, J. How, and J. F. III. Small-variance nonparametric clustering on the hypersphere. In *CVPR*, 2015.

[18] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher III. A Mixture of Manhattan Frames: Beyond the Manhattan World. In *CVPR*, June 2014.

[19] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers, and R. Siegwart. Towards a benchmark for RGB-D SLAM evaluation. In *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at RSS*, 2011.

[20] C. J. Taylor and A. Cowley. Parsing Indoor Scenes Using RGB-D Imagery. Robotics: Science and Systems, 2012.

[21] A. J. B. Trevor, J. G. Rogers III, and H. I. Christensen. Planar Surface SLAM with 3D and 2D Sensors. In *ICRA*, 2012.

[22] H. Wang, J. Wang, and W. Liang. Online Reconstruction of Indoor Scenes From RGB-D Streams. In *CVPR*, June 2016.

[23] S. Yang, Y. Song, M. Kaess, and S. Scherer. Pop-up SLAM: Semantic Monocular Plane SLAM for low-texture environments. In *IROS*, 2016.