# Surface Layout Estimation Using Multiple Segmentation Methods and 3D Reasoning

Michael Hödlmoser[1] and Branislav Micusik[2]

[1] CVL, Vienna University of Technology
hoedl@caa.tuwien.ac.at.edu
[2] AIT Austrian Institute of Technology
branislav.micusik@ait.ac.at

**Abstract.** In this paper we present a novel algorithm to estimate the surface layout of an indoor scene, which can serve as a visual cue for many different applications, e.g. 3D tracking, or localization in visual odometry. The main contribution of this work lies in combining multiple superpixel segmentation methods in order to obtain semantically meaningful regions. For each segmentation method, we combine 3D reasoning with semantic reasoning to generate multiple surface layout label hypotheses for each pixel. We then get the final label for each pixel within a Markov Random Field (MRF) by combining all hypothesis and by enforcing spatial consistency between neighboring pixels. Experimental results on complex indoor scenes show that our proposed method outperforms state-of-the-art methods.

**Keywords:** Semantic labeling, MRF, Superpixel labeling

## 1 Introduction

When humans are looking on an image, they can immediately interpret the scene since they are able to capture the semantic and geometric context. Consider the image shown in Fig. 1. Even when looking at it the first time, a human brain does not have any troubles to assume the 3D layout of the scene without having any further information. Humans are able to roughly gather the position of the viewpoint where the image was taken, to estimate the ground plane and ceiling orientations, to find vertical wall segments and even to distinguish between inside and outside the building although there is a reflexive door surface in the middle of the image. As can be seen, obtaining the 3D layout, detecting occluded objects and even gathering the 3D relationship between objects in the scene is something which is obviously beyond the visible 2D scene.

If humans would look on a single pixel of an image without having any other kind of information, they would not be able to do so. In computer vision, images are often described by features, which are mostly pixelwise, meant to be a low level description and which do not tell us anything about the semantic context. Semantically, it would be the most meaningful representation to use the occuring objects and their geometric relationships in the scene. To obtain such

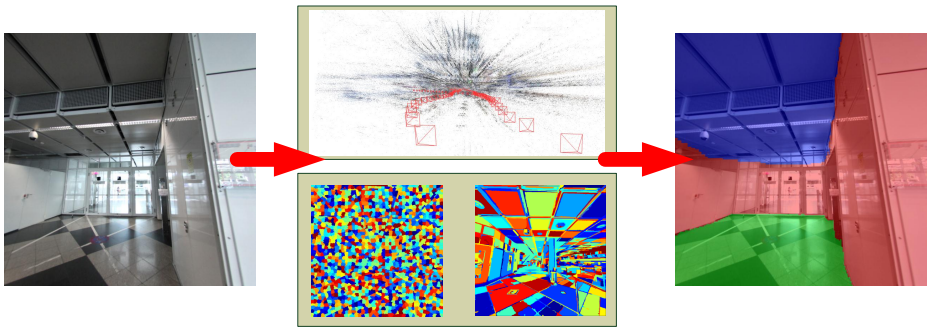


Fig. 1: Proposed scene layout estimation by combining multiple superpixel segmentation methods and 3D reasoning.

a representation, one can use superpixels, which is a representation between low level features and high level objects. Superpixels treat semantically connected pixels as a single patch. The problem is that there are many interpretations about what defines a semantic meaningful patch. As there is no superpixel generation method which works for all kind of different scenarios and environments, it can result in a wrong segmentation, which would mean that the application on top of it would be based on a wrong initialization and cannot recover anymore.

In this paper, we therefore try to overcome this problem and propose a novel method to estimate the 3D scene layout of a scene. As a prerequisite, we assume that each superpixel segment in a 2D image can be represented by a planar patch in its corresponding 3D environment. The contribution of this work is two-fold. (i) Yet, to the best of our knowledge, we are the first to combine the strengths of several superpixel segmentation methods to build a stronger classifier for pixelwise labeling of an image in *vertical plane*, *ground plane* and *ceiling* (see Fig. 1). (ii) We combine semantic reasoning with geometric reasoning for improving the labeling accuracy. We are using the method described in [1] for multiple segmentations, to which we refer to as semantic reasoning. We then generate a sparse point cloud using Structure from Motion (SfM) in a first step and calculate geometric features for each superpixel in a second step. We then use the segmentations as a soft constraint and a Markov Random Field (MRF) to do a pixelwise classification. The workflow can be seen in Fig. 2.

3D Reconstruction is usually done by generating a sparse point cloud obtained by triangulation followed by a densification [2]. In case of challenging environments, this is not possible anymore because there are wrong matches between corresponding camera views due to similar features obtained from flat and textureless surfaces (e.g. walls, floors). To overcome this problem, Hoiem *et al.* [1] came up with a segmentation-based combination of 2D cues, which is trained on multiple still images using boosted decision trees, which enables 3D reasoning using segments instead of point features. A similar approach based on segments was presented by Saxena *et al.* in [3]. Single image 3D labeling and 3D

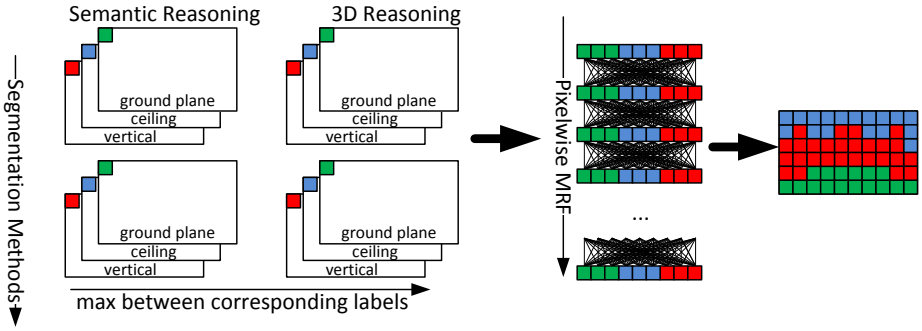



Fig. 2: Workflow of the proposed algorithm. For each segmentation method, semantic and geometric features are calculated in order to obtain a likelihood for each segment being labeled *ground plane*, *ceiling*, or *vertical*. The final label for each pixel is then obtained by using a pixelwise MRF.

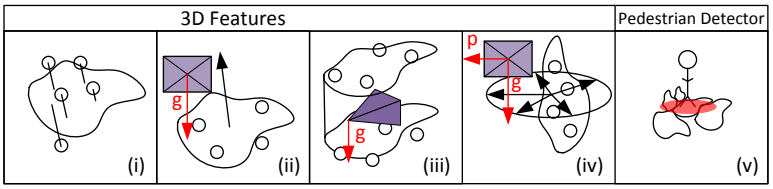
reconstruction from line representations instead of feature points was proposed by [4–8]. Surface labeling is also obtained by using a point cloud, as proposed in [9, 8, 10]. Different to all existing approaches, the 3D labeling result is gathered by combining multiple cues coming from both (i) a variety of different shaped segments, obtained from various segmentation methods and from (ii) combining conventional feature based matches with semantic patch-based 2D information.

## 2 Proposed Approach

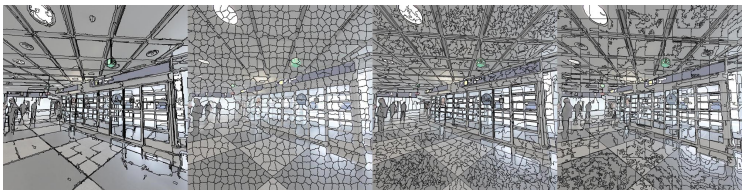
First, multiple images are taken to obtain a sparse point cloud and corresponding 3D camera positions using a conventional Structure from Motion (SfM) pipeline [11]. Each image is then segmented into semantically meaningful parts using multiple superpixel segmentation methods [12–15], where the outcome of the first three methods can be seen in Fig. 3b. This variety is exploited to obtain more accurate results by combining the strengths of different methods. By applying Hoiem's approach [1], the likelihood for each segment having an orientation label $l \in \mathcal{L} = \{\text{ground plane}, \text{ceiling}, \text{vertical}\}$ is obtained. The likelihood $u$ for a pixel $y_i$ within this segment and at image location $i$, is given by $u(y_i) = P(l|y_i)$.

### 2.1 3D Reasoning

Geometric features are then calculated for each segment using the sparse point cloud. As the goal is to classify each pixel according to the label set $\mathcal{L}$, it is assumed that each segment in the image can be represented by a planar patch. All geometric features are obtained relatively to the camera's orientation. The camera's gravity vector $\mathbf{g}$ is therefore determined from the vertical vanishing point. It is further known that the ground plane's and ceiling's surface normal




(a)



(b)

Fig. 3: (a) Geometric features with respect to the camera's gravity vector **g** and a random perpendicular vector **p** used for 3D reasoning. (i) Planarity, (ii) ground plane / ceiling orientation, (iii) ground plane position, (iv) vertical plane orientation, (v) segments where footpoints of pedestrians are located are labeled as ground plane. (b) From left to right: Outcome of segmentation methods [12],[13],[14] and [15].

are aligned with this gravity vector. Surface normals of vertical patches must be perpendicular to the gravity vector. Only those segments, where four or more reprojected 3D points are located, are labeled. Using RANSAC, a plane is fitted to the 3D points. Fig. 3a shows the features used for labeling the segments after fitting the plane. In the following, the surface normal of a patch is denoted as **n**.

– **Planarity likelihood** $P_{\text{plan}}$(Fig. 3a(i)): By calculating the Euclidean distances **d** between the 3D points and the plane, the planarity likelihood $P_{\text{plan}}$ is calculated by

$$P_{\text{plan}} = \frac{Q(\mathbf{d})_{25} + Q(\mathbf{d})_{75}}{Q(\mathbf{d})_{50}}, \tag{1}$$

where $Q(\mathbf{d})$ are the quantiles of **d**.

– **Horizontality likelihood** $P_{\text{hor}}$ (Fig. 3a(ii)): By determining the surface normal difference between the normal of the plane in question **n** and the gravity vector **g**, the likelihood for the plane being horizontal is obtained by

$$\begin{aligned} \alpha &= \min(\cos^{-1}(\mathbf{g} \cdot \mathbf{n}), \pi - \cos^{-1}(\mathbf{g} \cdot \mathbf{n})) \\ P_{\text{hor}} &= exp(-|\alpha|\ \pi/180). \end{aligned} \tag{2}$$

– **Ground plane / ceiling likelihood** $P_{\text{gp}}/P_{\text{cei}}$ (Fig. 3a(iii)): This feature helps determining if the patch in question is more likely to be located on the ground or

on the ceiling. Given the center point of the plane in question **a** and the camera center **b**, the likelihood for the segment being located on the ground plane is then given by

$$P_{\text{gp}} = \begin{cases} 1 & \text{if } \mathbf{a} \text{ below } \mathbf{b} \\ 0.2 & \text{else} \end{cases}, \tag{3}$$

$P_{\text{cei}}$ is set up vice versa.

– **Verticality likelihood** $P_{\text{ver}}$(Fig. 3a(iv)): The patch's surface normal $\mathbf{n}$ is rotated around the camera's gravity vector $\mathbf{g}$ with a stepsize of $r = 0° \dots 5° \dots 360°$ to obtain $\mathbf{n}_r$. By defining $\beta = [\beta_0 \dots \beta_{360}]$, the verticality likelihood is then obtained by

$$\begin{aligned} \beta_r &= \min(\cos^{-1}(\mathbf{p} \cdot \mathbf{n}_r), \pi - \cos^{-1}(\mathbf{p} \cdot \mathbf{n}_r)) \\ P_{\text{ver}} &= exp(-|\beta|\ \pi/180), \end{aligned} \tag{4}$$

where $\mathbf{p}$ is any random perpendicular vector to the camera's gravity vector.

– **Pedestrian Likelihood** $P_{\mathbf{ped}}$(Fig. 3a(v)): This feature helps distinguishing between ground plane and ceiling. Pedestrians are detected using [16]. It is assumed that the lower boundary of the bounding box is a person's foot point $\mathbf{f}$. To gain robustness, an ellipse (height 5 pixels, width 10 pixels) is defined which also takes neighboring segments into account. The likelihood that a pedestrian is located on a given segment $s$ is defined by

$$P_{\text{ped}} = \begin{cases} \lambda & \text{if } \mathbf{f} \in s \\ \lambda/2 & \text{else} \end{cases}, \tag{5}$$

where $\lambda$ is a multiplier constant in order to increase the likelihood for the segment to be located on the ground plane.

The final likelihoods for the labels are calculated by

$$v(y_i) = P(l|y_i) = \begin{cases} P_{\text{plan}} \cdot P_{\text{hor}} \cdot P_{\text{gp}} \cdot P_{\text{ped}} & \text{if } l = \text{ground plane} \\ P_{\text{plan}} \cdot P_{\text{hor}} \cdot P_{\text{cei}} & \text{if } l = \text{ceiling} \\ P_{\text{plan}} \cdot P_{\text{ver}} & \text{else} \end{cases}. \tag{6}$$

### 2.2 Pixelwise Labeling

To get a spatial consistent result for the whole image, the label for each pixel is determined independently of the segments of an image. The solution to this problem corresponds to finding the configuration of a Gibbs distribution with maximal probability, which is equivalent to finding the maximum posterior (MAP) configuration of an MRF. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph described by vertices $\mathcal{V}$, which in this case are represented by the $N$ pixels of the image, and edges $\mathcal{E}$. When having a set of random variables $\mathbf{X} = \{X_1, X_2, \dots X_N\}$ and a label configuration $\mathbf{x} = \{x_1, x_2, \dots x_N\}$ which can take values from the discrete set of labels $\mathcal{L}$, the energy term $E$ of the pairwise MRF is defined by

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}(i)} \psi_{i,j}(x_i, x_j), \tag{7}$$

| *Airport* Dataset | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | BASE | BASE+MRF | Proposed 4SP1N | Proposed 4SP1Y | Proposed 4SP2N | Proposed 4SP2Y | Proposed 4SP3N | Proposed 4SP3Y |
| ground plane | 63.57 | 76.77 | 80.58 | 80.40 | 82.83 | 82.40 | 82.83 | **82.92** |
| ceiling | 36.98 | 58.20 | 61.61 | 64.91 | 62.54 | 66.25 | 62.63 | **66.82** |
| vertical | 17.98 | 23.07 | 25.37 | 25.04 | 25.84 | 25.62 | **25.94** | 25.64 |
| global | 54.36 | 70.49 | 73.42 | 74.85 | 74.64 | 76.22 | 74.67 | **76.48** |
| average | 39.51 | 52.68 | 55.85 | 56.78 | 57.07 | 58.09 | 57.13 | **58.46** |

| *Rooms* Dataset [5] | | | | | |
|---|---|---|---|---|---|
| | BASE | BASE+MRF | Proposed 4SP1N | Proposed 4SP2N | Proposed 4SP3N |
| ground plane | 63.54 | 71.60 | 73.06 | 74.07 | **74.91** |
| ceiling | 37.35 | **46.12** | 42.60 | 43.19 | 43.53 |
| vertical | **80.77** | 80.63 | 79.53 | 80.10 | 80.10 |
| global | 84.87 | 85.00 | 85.05 | 85.81 | **85.92** |
| average | 60.55 | 66.11 | 65.06 | 65.78 | **66.18** |

Table 1: Percentage of correctly classified pixels for *Airport* and *Rooms* [5] dataset.

where $\mathcal{N}(i)$ is the neighborhood of node $i$, $\psi_i$ is the unary potential in the graph and $\psi_{i,j}$ is the pairwise potential, or smoothness term, between neighboring pixels. These terms are defined to be

$$\begin{aligned} \psi_i(x_i) &= 1 \; - \; \max(u(y_i), v(y_i)) \\ \psi_{i,j}(x_i, x_j) &= \begin{cases} 0.5 & \text{if } x_i = x_j, \\ 1 & \text{if } x_i \neq x_j \end{cases} \end{aligned} \tag{8}$$

The MAP configuration $\hat{\mathbf{x}}$ is then found by $\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} E(\mathbf{x})$.

## 3 Experiments

We evaluate our algorithm on two datasets. The *Airport* dataset provides an image sequence of 100 images having a resolution of 1001x1001 pixels, where a screenshot of the reconstructed scene can be seen in Fig. 4b. We manually segmented all the images in the dataset. Objects which cannot be classified by the method are not considered in this evaluation (e.g. people, columns) and are marked as a black region. The second dataset was taken from [5] to which we refer to as the *Rooms* dataset. It provides 314 images with varying resolutions showing cluttered indoor scenes. Since this dataset does only provide still images, 3D reasoning cannot be applied.

We use [12–15], where each of those methods is performed multiple times using different parameter sets to obtain varying segmentation results. We compare the base method of Hoiem [1] (BASE), which is using [12] for segmentation to the same method using a pixelwise MRF (BASE+MRF) and to our proposed method using all superpixel methods (4SP) 1,2, or 3 parameter sets (P1/2/3) and optionally 3D reasoning (Y/N). The labels for the MRF outcome of all methods are *ground plane*, *ceiling* and *vertical*.

**Quantitative Experiments:** As our method is calculating a corresponding label for each pixel, we compare the correctly classified pixels to the ground truth

images. For both the *Airport* dataset and the dataset of [5], pixels which are left blank in the ground truth image are not taken into account for the comparison. Having a ground truth labeled image $\mathcal{G}$ and a resulting image $\mathcal{R}$, the accuracy for label $l$ is determined by $\frac{|\mathcal{G}_l|\cap|\mathcal{R}_l|}{|\mathcal{G}_l|\cup|\mathcal{R}_l|}$, where $|\mathcal{G}_l|$ refers to the number of pixels of label $l$ in image $\mathcal{G}$. The percentage of correctly classified pixels for each label can be seen in Table 1. As can be seen, the proposed approach performs better on horizontal patches than on vertical ones when applied on the *Airport* dataset. This happens since 3D points recovered from 2D feature points obtained from vertical structures are noisier than points belonging to horizontal patches. This noise occurs due to the facts that (i) there are much more orientation variations of vertical walls than on the ceiling or the ground plane and (ii) reflexive surfaces (e.g. mirrors, glasses) tend to be attached to vertical structures.

***Airport* Dataset:** As can be seen, there is an improvement of approximately 20% between using [1] and the same method using an MRF for pixel labeling. It can also be seen that there is an improvement between using only the superpixel method proposed in [12] and using all described methods having multiple parameter sets. The difference between incorporating geometric reasoning and not incorporating is up to 4%, no matter if a segmentation algorithm is applied once or multiple times. The improvement is obviously higher between using a variety of superpixel methods than between a variety of different parameter sets for each method, regardless of incorporating 3D reasoning or not.

***Rooms* Dataset:** There is also an improvement between using a single superpixel segmentation method and multiple ones for this dataset. Since the scenes shown in the images are not as complex as the ones shown in the images of the *Airport* dataset, the BASE method delivers better results and the improvement when processing the frame using 4SP1N, 4SP2N, or 4SP3N is not as obvious as for complex scenes. Nevertheless, an average accuracy improvement of almost 6% can be obtained when using multiple segmentation methods. For the *ground plane* label, an improvement of over 10% is reached.

**Qualitative Experiments:** Fig. 4a shows qualitative results of our method using different parameter settings and a comparison to the state-of-the-art method proposed in [1]. Each row shows a different image of the scene. The first image of each row shows the manually labeled ground truth data, where black regions indicate objects which are not taken into account in the evaluation. The following columns show the results for BASE, BASE+MRF, 4SP1N, 4SP1Y, 4SP2N, 4SP2Y, 4SP3N, 4SP3Y. The labels *ground plane*, *ceiling* and *vertical* are indicated by the colors green, blue, and red, respectively. As can be seen, there is clearly an improvement between only considering a single segmentation method and incorporating multiple ones. It is also visible that there is in improvement in labeling the surfaces by incorporating 3D reasoning about the scene.

## 4 Conclusion

We presented a framework for estimating the 3D scene layout of a scene. A semantic meaningful patch can be obtained by using different cues. Depending

on these cues, the resulting patches of different segmentation methods vary in shape and size. By combining the strengthnesses of several superpixel segmentation methods, we are able to obtain a stronger classifier for labeling each pixel's surface orientation. The labeling accuracy is improved by incorporating geometric features, obtained from 3D point clouds of the scene. The most likely label is then obtained by exploiting an MRF. Experiments on novel and existing datasets show superior results of our approach compared to state-of-the-art methods.

**Acknowledgments:** The authors would like to thank the reviewers for their valuable comments. This work was partly supported by the Austrian Research Promotion Agency's (FFG) FIT-IT projects 835916 (PAMON), 830042 (CAPRI) and CogVis Ltd.

## References

1. Hoiem, D., Efros, A., Hebert, M.: Recovering surface layout from an image. IJCV **75**(1) (2007)
2. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. In: CVPR. (2007)
3. Saxena, A., Sun, M., Ng, A.: Make3d: Learning 3d scene structure from a single still image. PAMI **31**(5) (2009) 824–840
4. Lee, D., Hebert, M., Kanade, T.: Geometric reasoning for single image structure recovery. In: CVPR. (2009)
5. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: ICCV. (2009)
6. Del Pero, L., Guan, J., Brau, E., Schlecht, J., Barnard, K.: Sampling bedrooms. In: CVPR. (2011) 2009–2016
7. Del Pero, L., Bowdish, J., Fried, D., Kermgard, B., Hartley, E., Barnard, K.: Bayesian geometric modeling of indoor scenes. In: CVPR. (2012)
8. Flint, A., Murray, D., Reid, I.: Manhattan scene understanding using monocular, stereo, and 3d features. In: ICCV. (2011)
9. Tsai, G., Xu, C., Liu, J., Kuipers, B.: Real-time indoor scene understanding using bayesian filtering with motion cues. In: ICCV. (2011) 121–128
10. Brostow, G., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: ECCV. (2008) 44–57
11. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. In: ACM TRANSACTIONS ON GRAPHICS. (2006) 835–846
12. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV **59**(2) (2004) 167–181
13. Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. PAMI **31**(12) (2009) 2290–2297
14. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. In: CVPR. (2011) 2097–2104
15. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Suesstrunk, S.: SLIC Superpixels Compared to State-of-the-art Superpixel Methods. PAMI (2012)
16. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI **32** (2010) 1627–1645

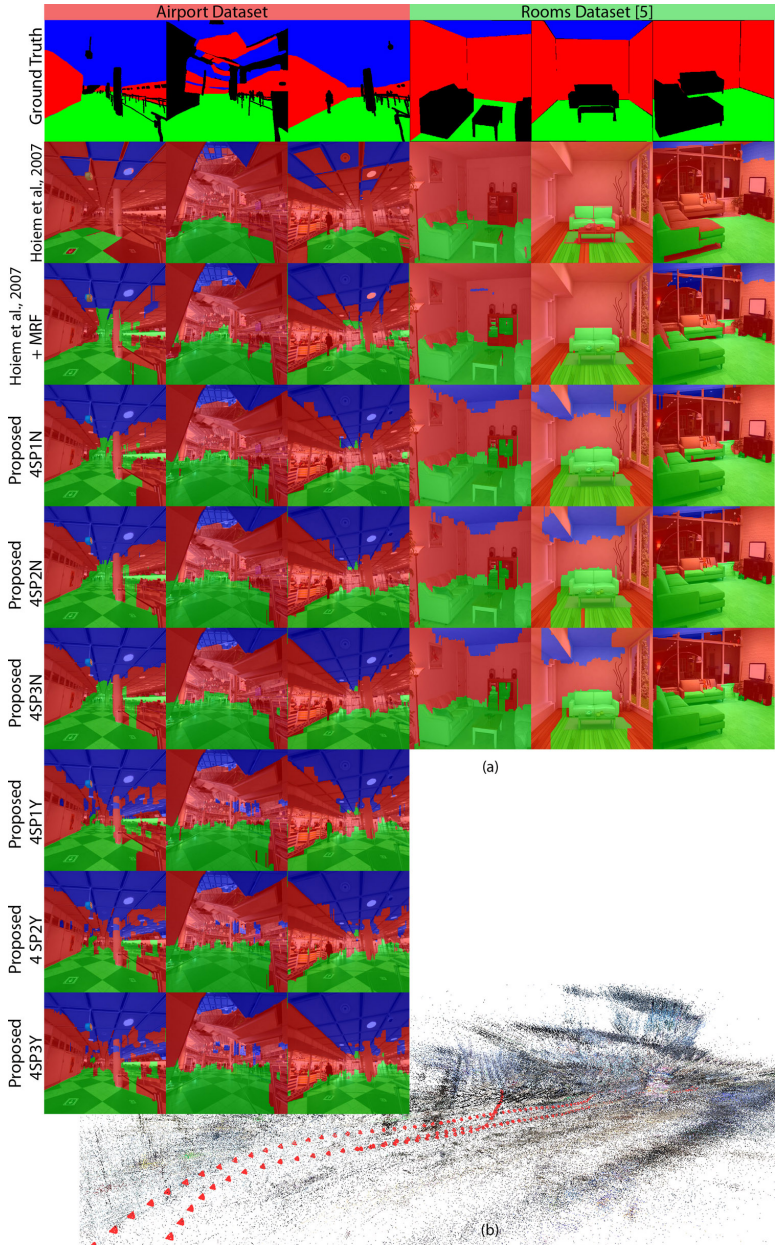



Fig. 4: (a) Qualitative results. First image of each row shows ground truth (excluded objects are marked black), following columns show results for BASE, BASE+MRF, 4SP1N, 4SP1Y, 4SP2N, 4SP2Y, 4SP3N, 4SP3Y. Green= ground plane, blue=ceiling, red=vertical. (b) Screenshot of sparse 3D point cloud and camera positions from *Airport* dataset.