INDOOR FRAME RECOVERING VIA LINE SEGMENTS REFINEMENT AND VOTING

Jun Chu[1], *Anzheng GuoLu*[1], *Lingfeng Wang*[2], *Chunhong Pan*[2] *and Shiming Xiang*[2]

1. Institute of Computer Vision, Nanchang Hangkong University, {chujun99602, jsjsj_glaz}@163.com
2. NLPR, Institute of Automation CAS, 100190, {lfwang, chpan, smxiang}@nlpr.ia.ac.cn

ABSTRACT

Frame structure estimation from line segments is an important yet challenging problem in understanding indoor scenes. In practice, line segment extraction can be affected by occlusions, illumination variations, and weak object boundaries. To address this problem, an approach for frame structure recovery based on line segment refinement and voting is proposed. We refined line segments by the revising, connecting, and adding operations. We then propose an iterative voting mechanism for selecting refined line segments, where a cross ratio constraint is enforced to build crab-like models. Our algorithm outperforms state-of-the-art approaches, especially when considering complex indoor scenes.

Index Terms— indoor frame recovery; line segment refinement; iterative voting; cross ratio

1. INTRODUCTION

Indoor scenes have been a popular research subject over the past decade. An important aspect of understanding indoor scenes is the recovery of a room frame (Fig. 1). The recovered frame could have applications in indoor scenes, i.e., robot navigation, object recognition, 3D reconstruction, and event detection [1, 2, 3, 4]. However, indoor frame recovery remains a challenging task because of illumination variations, weak boundaries, and partial occlusions. In this work, we address these difficulties by using new line refinement and voting strategies.

Related Work: From the perspective of feature utilization, recent work can be classified into two main groups: texture-based and line segment–based.

Texture-based approaches often over-segment the image into patches first and then label each patch according to two descriptors: the color histogram and the texture histogram. For example, Liu et al. [5] over-segmented an image by using the graph cut and then introduced texture to label the frame. However, the texture descriptors of indoor scenes are sensitive to illumination variations. Moreover, the texture descriptors of ceilings, floors, and walls are often similar to one another in indoor scenes, such that the discrimination of these descriptors is low. Therefore, methods based on texture descriptors usually cannot work well when applied to indoor images. Silberman et al. [6] and Ren et al. [7] introduced depth to determine the category of each pixel based on an RGB-D image. These methods often segment the planes of indoor images according to the depth of each plane. Compared with texture descriptors, the depth descriptor is insensitive to illumination variations. The depth descriptor is a good descriptor for indoor images, but in many real-world applications, it cannot obtain accurate depth information.

This work was supported in part by the National Basic Research Program of China under Grant 2012CB316304, and the National Natural Science Foundation of China under Grants 61263046, 61175025, and 61203277.

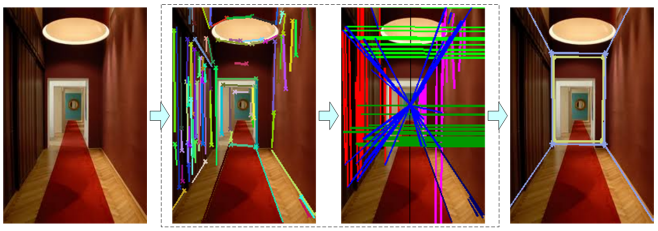

Fig. 1. Overview of our work. (a): Input image of an indoor scene. (b): Extracted line segments. (c): Refined line segments after revising, connecting, adding, and selecting. (d): Output image. It shows the frame recovered from those messy line segments (Fig. 1b), which obtained by fitting our crab-like model to the refined line segments.

Given that texture-based approaches are sensitive to illumination and low discrimination, numerous line segment-based approaches have been proposed [8, 9]. Compared with texture-based algorithms, line segment-based approaches have three advantages: (1) excellent information on the building structure, (2) minimal impact of the distance between the camera and scene on line segments, and (3) line segments robust to illumination variations. Lee et al. [10] proposed 12 corner models to fit the frame through line segments. After them, Hedau et al. [11] generated candidate frames by shooting rays from vanishing points and then selected the best candidate via ranking support vector machines (SVMs). Further more, Flint et al. [12] employed a visual simultaneous localization and mapping system to refine the frame locally. However, these line segment-based algorithms often do not work well when abundant, missing, or incorrect extraction exists in the detected line segments.

An orientation map generated from detected line segments that expresses the local belief of region orientations is proposed by Lee et al. [10]. Subsequently, methods based on the orientation map have been proposed. Orientation map-based algorithms often use the orientation map as a supplemental descriptor for texture descriptors because it can describe the local category of regions. For instance, Schwing et al. [13] and Pero et al. [4] introduced the orientation map as well as texture for frame inference. However, approaches based on the orientation map often do not work well because such map is derived from detected line segments, which are sensitive to weak boundaries.

The Method: Following Hedau et al. [11], we propose a new method for frame structure recovery based on line segment refinement. Our algorithm primarily includes the following three procedures. First, the line segments are refined by the revising, connecting, and adding operations. In the revising step, we correct the orientations of misclassified line segments. In the connecting step, line segments corrupted by occlusions and illumination variation-

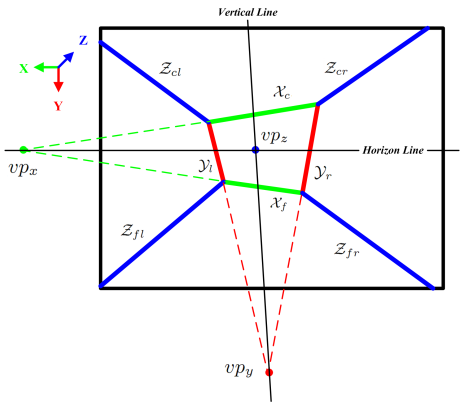

Fig. 2. A sketch for the crab-like model. Under the Manhattan assumption, the model consists of eight lines x_c, x_f, y_l, y_r, z_{cl}, z_{cr}, z_{fr}, and z_{fc} along x, y, z directions respectively.

s are connected to form a complete segment. In the adding step, we shoot rays from the vanishing points to the endpoints of line segments to fit the line segments that are lost on weak boundaries. Line segments extracted from the orthogonal planes of the frame can hardly intersect with those lying on the frame structure. Therefore, we propose an iterative voting algorithm to weigh each line segment and then select those with high scores. In each loop, a cross ratio constraint derived from the Manhattan assumption [14] restricts the hypothetical frames. Compared with state-of-the-art approaches, our algorithm has the following advantages:

1. Given the revising, connecting, and adding procedures in line detection, the proposed method can obtain more accurate line segments. Therefore, our method can estimate line segments from weak boundaries and occluded regions, which may not be detected using other approaches [10, 11]. As a result, the frame can be recovered effectively by our method.
2. As a result of the iterative voting mechanism and cross ratio constraint, we can recover the frame quickly and accurately. Our method can also recover frames better than [10] and [11], especially when weak boundaries, illumination variations, and occlusions exist in the indoor scenes. Moreover, our method is six times faster than [10] and nearly 100 times faster than [11].

2. PROBLEM FORMULATION AND NOTATIONS

Our work aims to obtain a frame from a single indoor scene image. According to the Manhattan assumption,[1] the frame is a crab-like model consisting of eight lines (Fig. 2). The notations for the crab-like model are described as follows:

1. We denote three vanishing points along the mutually orthogonal$x$, y, and z orientations as vp_x, vp_y, and vp_z, respectively. The line passing through vp_y and vp_z is denoted as the vertical line l_{yz}, whereas that through vp_x and vp_z is denoted as the horizontal line l_{xz}. We also denote the line passing through vp_x and vp_y as l_{xy}.

2. The line segments along the x, y, and z directions are partitioned as three sets: $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$. We denote the line segments of the x direction located on the ceiling (floor) as $\mathcal{X}_c$ ($\mathcal{X}_f$). Similarly, for the line segments along the y direction, we denote the ones on the left (right) of l_{yz} as $\mathcal{Y}_l$ ($\mathcal{Y}_r$). The line segments of the z direction on the four corners (from upper left to bottom left in a clockwise direction) are denoted as $\mathcal{Z}_{cl}$, $\mathcal{Z}_{cr}$, $\mathcal{Z}_{fr}$, and $\mathcal{Z}_{fl}$.

[1]The Manhattan assumption describes the frame of the indoor scene as a cube. The six planes of the frame lie in three mutually orthogonal orientations.

3. LINE SEGMENT REFINEMENT

In our method, line segments are initialized by the detector proposed in [15]. By using the vanishing point estimation algorithm proposed in [16], we obtain the three vanishing points vp_x, vp_y, and vp_z and then divide the initial line segments into three sets: $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$. However, the frame can hardly be recovered directly from the three line segment sets because these sets often have the following problems:

Problem 1: According to [16], the line segments close to l_{xz} are often misclassified.

Problem 2: Given the illumination variations and occlusions, long line segments are often divided into many parts.

Problem 3: Edges on weak boundaries often cannot be detected as line segments. That is, some important line segments are often lost.

To address these difficulties, we propose a new line segment refinement method via the following three operations: revising, connecting, and adding.

3.1. Revising

To address **Problem 1**, we propose a revising algorithm:

Step_1: For each line segment l in $\mathcal{X}$ and $\mathcal{Z}$, we calculate the angle θ between l and l_{xz}.

Step_2: If $\theta < \tau_\theta$, Step 3 is performed. Otherwise, Step 1 is repeated. τ_θ is the threshold, which we set to 20°.

Step_3: We use nearest neighbor assignment to reclassify the line segment l. That is, we find the line segment l' closest to line segment l and classify l with l'.

3.2. Connecting

To address **Problem 2**, we propose an approach to connecting line segments. The main idea of this method is to check whether two line segments are collinear.

Step_1: Two line segments l_i and l_j are selected from $\mathcal{X}$ ($\mathcal{Y}$ or $\mathcal{Z}$).

Step_2: The total length of the two line segments is calculated as $length = \text{len}(l_i) + \text{len}(l_j)$, where $\text{len}(\cdot)$ is the length function at the pixel level. The longest distance $longDis$ and shortest distance $shortDis$ between the two line segments are also calculated.

Step_3: The distance error is computed as,

$$e = |longDis - shortDis - length|, \tag{1}$$

If $e < \tau_e$, we connect the two line segments l_i and l_j. Otherwise, the entire procedure is repeated. τ_e is the threshold, which we set to 0.3.

3.3. Adding

To add line segments, we divide $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ into eight subsets: $\mathcal{X}_c$, $\mathcal{X}_f$, $\mathcal{Y}_l$, $\mathcal{Y}_r$, $\mathcal{Z}_{cl}$, $\mathcal{Z}_{cr}$, $\mathcal{Z}_{fr}$, and $\mathcal{Z}_{fl}$. This process has two stages: constructing the coordinate by l_{xz} and l_{yz} and then assigning each line segment to a quadrant according to its midpoint.

Line segments along each direction may be lost (**Problem 3**). Under the Manhattan assumption, each line segment should pass through one vanishing point. To fit the missing line segments $\mathcal{L}_{\text{mis}}$ in the z (x or y) direction, we use the concept of *neighbor sets*. For example, the *neighbor sets* of $\mathcal{Z}_{cl}$ are $\mathcal{X}_c$ and $\mathcal{Y}_l$, and those of $\mathcal{X}_c$ are $\mathcal{Z}_{cl}$, $\mathcal{Y}_l$, $\mathcal{Z}_{cr}$, and $\mathcal{Y}_r$. We then shoot rays from the vanishing point vp_z (vp_x or vp_y) to the endpoints near l_{mis} that belong to the *neighbor sets* of l_{mis}. As a result of the three procedures (i.e., revising, connecting, and adding), the final line segment set $\mathcal{L}$ consists of two parts:

$$\mathcal{L} = \bar{\mathcal{L}} + \hat{\mathcal{L}}, \quad \mathcal{L} = \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}, \tag{2}$$

where $\bar{\mathcal{L}}$ is the detected line segment after the revision and connection, and $\hat{\mathcal{L}}$ is the added line segment.

4. CRAB-LIKE MODEL CONSTRUCTION

The crab-like model consists of x_c, x_f, y_l, y_r, z_{cl}, z_{cr}, z_{fr}, and z_{fc}. The following subsections describe the two processes that shorten running time. We first introduce a cross ratio constraint based on the Manhattan assumption.

4.1. Cross Ratio Constraint

According to the Manhattan assumption, a property of the frame in Fig. 2 is easily proven. Specifically,

$$(x_c, l_{xz} : x_f, l_{xy}) = (z_{cl}, l_{xz} : z_{fl}, l_{yz}) = (z_{cr}, l_{xz} : z_{fr}, l_{yz}), \tag{3}$$

The case for the vertical direction is similar. Therefore, the cross ratio constraint is formulated as,

$$2c_{(c,f)} - c_{(cl,fl)} - c_{(cr,fr)} < \varepsilon,\ 2c_{(l,r)} - c_{(fl,fr)} - c_{(cl,cr)} < \varepsilon, \tag{4}$$

where $c_{(c,f)}$ denotes $(x_c, l_{xz} : x_f, l_{xy})$. The other denotations are similar. ε is a small constant ranging from 0.1 to 0.15.

4.2. Iterative Voting Mechanism

Under the Manhattan assumption, each edge of the frame structure is the intersection of two orthogonal planes, and the line segments extracted from the orthogonal planes of the frame could not intersect with those on the frame structure. According to the property, we design an iterative voting algorithm to select the top n line segments in $\mathcal{X}_c$, $\mathcal{X}_f$, $\mathcal{Y}_l$, $\mathcal{Y}_r$, $\mathcal{Z}_{cl}$, $\mathcal{Z}_{cr}$, $\mathcal{Z}_{fr}$, and $\mathcal{Z}_{fl}$. The weight for each line segment l_i can be described as,

$$w_i^{k+1} = w_i^k + v_i^k, k \in N, \tag{5}$$

where $(k+1)$ is the duration of the loop, w_i^k is the weight of l_i after the k-th loop, and v_i^k is the voting weight of l_i in the $(k+1)$-th loop. To describe the denotations in Equation 5, we define two functions: $\varphi(\cdot)$, a normalization function, and $\eta(\cdot)$, a reverse normalization function.

$$\psi_{(x_i)} = \frac{x_i}{\sum\{x_i\}}, \eta_{(x_i)} = \frac{\max(\{x_i\}) - x_i}{\sum(\max(\{x_i\}) - x_i)}, x_i \in X. \tag{6}$$

The initial weight w_i^0 can be evaluated from three aspects:

$$w_{\text{len}} = \psi_{(\text{len}(l_i))}; \quad w_{\text{ang}} = \eta_{(\text{ang}(l_i))}; \quad w_{\text{dis}} = \eta_{(\text{dis}(l_i))}, \tag{7}$$

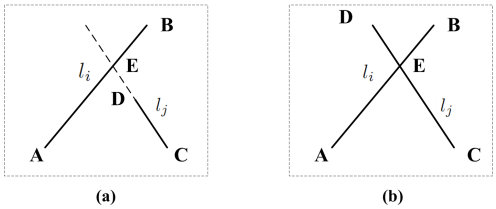

Fig. 3. This is a sketch to describe line segment l_j (i.e., $\overline{CD}$) voting for line segment l_i (i.e., $\overline{AB}$). (a) l_i and l_j don't intersect, yet the extension line of l_j intersect with l_i at point E. (b) l_i and l_j intersect at point E.

where ang$(\cdot)$ denotes the angle between l_i and the line across the corresponding vanishing point and the middle point of l_i; dis$(\cdot)$ is the distance between l_i and vp_z; and l_i belongs to $\mathcal{X}_c$, $\mathcal{X}_f$, $\mathcal{Y}_l$, $\mathcal{Y}_r$, $\mathcal{Z}_{cl}$, $\mathcal{Z}_{cr}$, $\mathcal{Z}_{fr}$, and $\mathcal{Z}_{fl}$. Thus, w_i^0 can be formulated as,

$$w_i^0 = w_{\text{len}} \cdot \xi_{\text{len}} + w_{\text{ang}} \cdot \xi_{\text{ang}} + w_{\text{dis}} \cdot \xi_{\text{dis}}, \tag{8}$$

where $\xi_{\text{len}} + \xi_{\text{ang}} + \xi_{\text{dis}} = 1$. However, the following should be noted.

For $l_i \in \mathcal{Z}$: First, a tangent relation exists between w_{ang} and w_{dis}. Therefore, we set ξ_{dis} to 0. Second, the added line segment l_i in $\hat{\mathcal{L}}$ does not come from the original image but is shot from the vanishing point vp_z. Therefore, w_{len} and w_{dis} are meaningless, and we set ξ_{ang} to 1.

For $l_i \in \{\mathcal{X}, \mathcal{Y}\}$: For the added line segment l_i in $\hat{\mathcal{L}}$, both w_{len} and w_{ang} are meaningless. Thus, we set ξ_{len} and ξ_{ang} to 0. The parameters ξ_{len} and ξ_{ang} used in the experiments are both fixed as 0.5 for $l_i \in \mathcal{Z} \cap \bar{\mathcal{L}}$. ξ_{len}, ξ_{ang}, and ξ_{dis} are respectively fixed as (1.3/6), (0.7/6) and (4/6) for $l_i \in \{\mathcal{X}, \mathcal{Y}\} \cap \bar{\mathcal{L}}$. The voting weight for each line segments can be described as,

$$v_i^k = \sum_j (-1)^{\text{label}} \cdot \lambda_j \cdot w_j^k, \tag{9}$$

where l_j is from the *neighbor sets* of l_i. For example, if $l_i \in \mathcal{Z}_{cl}$, $l_j \in \{\mathcal{X}_c, \mathcal{Y}_l\}$, or if $l_i \in \mathcal{X}_f$, $l_j \in \{\mathcal{Y}_l, \mathcal{Y}_r, \mathcal{Z}_{fl}, \mathcal{Z}_{fr}\}$. Assuming that l_j is voting for l_i (Fig. 3), if l_i and l_j intersect, the label is equal to 1; otherwise, it is equal to 0. The line segments used for voting should belong to $\bar{\mathcal{L}}$. Finally, the voting weight of l_j to l_i(i.e., λ_j) is formulated as,

$$\lambda = \eta(\frac{\min(\text{len}(\overline{EC}), \text{len}(\overline{ED}))}{\text{len}(\overline{CD})}), \tag{10}$$

Thus, in each loop, the score of the crab-like model can be formulated as the sum of the weight of each line segment in $\mathcal{X}_c$, $\mathcal{X}_f$, $\mathcal{Y}_l$, $\mathcal{Y}_r$, $\mathcal{Z}_{cl}$, $\mathcal{Z}_{cr}$, $\mathcal{Z}_{fr}$, and $\mathcal{Z}_{fl}$. We calculate only the score of candidate frames that satisfy the cross ratio constraint.

According to the iterative voting mechanism, even when the initial weight of each line segment is set to 0, the weights of the line segments on the frame increase along the iteration time, whereas those of line segments away from the frame decrease rapidly. Therefore, the top n line segments in each group ($\mathcal{X}_c$, $\mathcal{X}_f$, $\mathcal{Y}_l$, $\mathcal{Y}_r$, $\mathcal{Z}_{cl}$, $\mathcal{Z}_{cr}$, $\mathcal{Z}_{fr}$, and $\mathcal{Z}_{fl}$) finally converge. Moreover, given the cross ratio constraint, the candidate frames selected from the top $8 * n$ line segments also converge. Thus, we compute the score of each crab-like model in each iteration step and then rank the models according to their scores. We break the iteration until the rank of the models no longer changes.

5. EXPERIMENTS AND RESULTS

In this section, the proposed method is compared with the state-of-the-art approaches by Lee et al. [10] and Hedau et al. [11]. The test images are downloaded from the Internet. The ground truths of all images are manually labeled. Our method as well as those of Lee et al. [10] and Hedau et al. [11], utilizes the algorithm proposed in [15] to initialize line segments.

5.1. Evaluation of Our Algorithm

In the first experiment, we select three indoor images with background clutter, illumination variations, and weak boundaries to evaluate the proposed method.

A large number of incorrect extractions are found among the initial line segments (Fig. 4b). In the first image, some line segments are not detected in the ceiling. After revision, connection, addition, and selection, the refined line segments are significantly better than the initially detected line segments (Fig. 4c). Some corrupted line segments on the walls are connected to form a complete line segment. Moreover, as a result of the adding procedure, we are able to add some line segments that cannot be detected by the algorithm proposed in [15] (refer to the line segments on the ceiling and floor). Thus, the frames recovered by our method are similar to the ground truth.

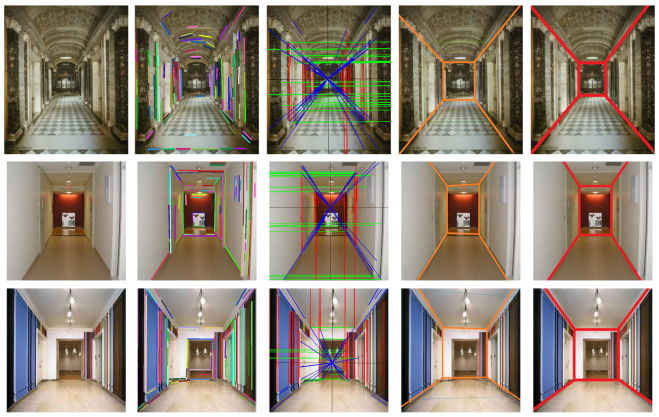

Fig. 4. Our results on three images with difficulties from background clutter, illumination variations and weak boundaries. (a) The input indoor images. (b) The initial line segments obtained by the algorithm proposed in [15]. (c) The line segments after refining. (d) The detected frames, where the candidate best fitting the ground truth is labeled with a heavier line. (e) The ground truth.

5.2. Comparisons with State-of-the-Art Approaches

In this experiment, we compare our method with those of Lee et al. [10] and Hedau et al. [11] when applied to five challenging indoor scenes with weak boundaries, occlusions, and illumination variations. The methods are compared by running their executable codes with default parameters. The best candidate frames are labeled with a heavier line. The results of the comparison are illustrated in Fig. 5. According to this figure, our method outperforms the state-of-the-art approaches in terms of frame recovery. The advantages of our method are derived mainly from the following strategies:

1. The refinement procedures, especially addition, facilitate the detection of line segments on weak boundaries or in regions with illumination variations and occlusion. Without this line refinement strategy, the frames obtained by [10] and [11] are shapeless and twisted.

2. The voting process and cross ratio constraint facilitate the selection of superior frames. Compared with [11], which utilizes sampling and SVM ranking methods, our method can obtain better frames at a lower computational cost. Comparisons of the computational cost are presented below.

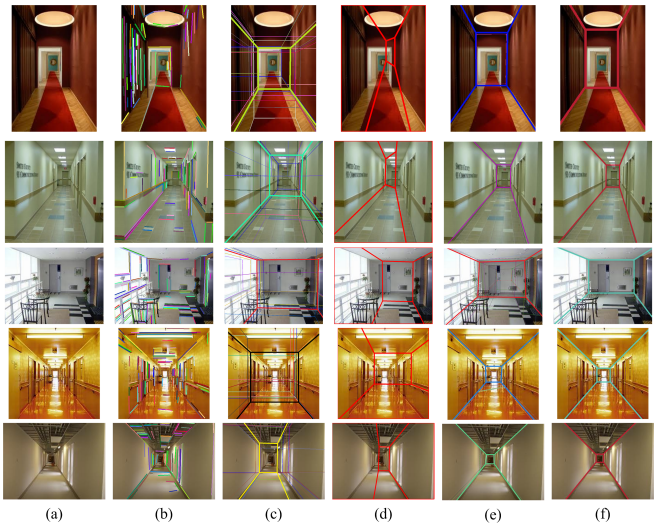

Fig. 5. Comparisons with [10, 11]. (a) The input image. (b) The initial line segments. (c) The results obtained by [10]. (d) The results obtained by [11]. (e) Our results. (f) The ground truth.

Algorithm	Processing Time (s)
Ours	5.26
Lee et al. [10]	32.61
Hedau et al. [11]	505.79

Table 1. Running time comparison

5.3. Running Speed Evaluation

We also compare the running speed of our algorithm with [11] and [10]. All algorithms, including ours, are performed with MATLAB on a Windows system with a 3.2 GHz CPU and 2.0 GB RAM. We used images with 208×343 resolution as the test images. The results are illustrated in Table 1. Our method has a large advantage over [10] and [11]. In particular, the proposed method is nearly 100 times faster than [11].

6. CONCLUSIONS

In this paper, we proposed a frame recovery method with line segment refinement, a voting mechanism, and a cross ratio constraint. The experimental results prove the excellent performance of our algorithm. As a line segment-based method, our algorithm depends on the detection accuracies of three vanishing points, which are always obtained from a number of line segments. In the future, we will incorporate the optimization of vanishing points into the framework for line segment refinement. With the refined line segments, we can obtain an accurate orientation map, which we will consider integrating into our method.

7. REFERENCES

[1] Varsha Hedau, Derek Hoiem, and David A. Forsyth, "Recovering free space of indoor scenes from a single image.," in *CVPR*. IEEE, 2012, pp. 2807–2814.

[2] David F. Fouhey, Vincent Delaitre, Abhinav Gupta, Alexei A. Efros, Ivan Laptev, and Josef Sivic, "People watching: Human actions as a cue for single view geometry," in *ECCV (5)*, 2012, pp. 732–745.

[3] Min Sun, Sid Ying-Ze Bao, and Silvio Savarese, "Object detection using geometrical context feedback," *International Journal of Computer Vision*, vol. 100, no. 2, pp. 154–169, 2012.

[4] Luca Del Pero, Joshua Bowdish, Daniel Fried, Bonnie Kermgard, Emily Hartley, and Kobus Barnard, "Bayesian geometric modeling of indoor scenes," in *CVPR*, 2012, pp. 2719–2726.

[5] Xiaoqing Liu, Olga Veksler, and Jagath Samarabandu, "Graph cut with ordering constraints on labels and its applications," in *CVPR*, 2008.

[6] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV (5)*, 2012, pp. 746–760.

[7] Xiaofeng Ren, Liefeng Bo, and Dieter Fox, "Rgb-(d) scene labeling: Features and algorithms," in *CVPR*, 2012, pp. 2759–2766.

[8] Luca Del Pero, Jinyan Guan, Ernesto Brau, Joseph Schlecht, and Kobus Barnard, "Sampling bedrooms," in *CVPR*, 2011, pp. 2009–2016.

[9] Huayan Wang, Stephen Gould, and Daphne Koller, "Discriminative learning with latent variables for cluttered indoor scene understanding," in *ECCV (2)*, 2010, pp. 435–449.

[10] David C. Lee, Martial Hebert, and Takeo Kanade, "Geometric reasoning for single image structure recovery," in *CVPR*, 2009, pp. 2136–2143.

[11] Varsha Hedau, Derek Hoiem, and David A. Forsyth, "Recovering the spatial layout of cluttered rooms," in *ICCV*, 2009, pp. 1849–1856.

[12] Alexander Flint, Christopher Mei, Ian D. Reid, and David W. Murray, "Growing semantically meaningful models for visual slam," in *CVPR*, 2010, pp. 467–474.

[13] Alexander G. Schwing, Tamir Hazan, Marc Pollefeys, and Raquel Urtasun, "Efficient structured prediction for 3d indoor scene understanding," in *CVPR*, 2012, pp. 2815–2822.

[14] James M. Coughlan and Alan L. Yuille, "Manhattan world: Orientation and outlier detection by bayesian inference," *Neural Computation*, vol. 15, no. 5, pp. 1063–1088, 2003.

[15] Peter Kovesi, "Phase congruency detects corners and edges," in *DICTA*, 2003, pp. 309–318.

[16] Jean-Philippe Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *ICCV*, 2009, pp. 1250–1257.