3D Room Reconstruction from A Single Fisheye Image

Mingyang Li
State Key Laboratory of Virtual
Reality Technology and Systems
Beihang University
Beijing, China
sunnyli@buaa.edu.cn

Yi Zhou
Bigview Technology Co. Ltd.
Beijing, China
zy@bigviewcloud.com

Ming Meng
State Key Laboratory of Virtual
Reality Technology and Systems
Beihang University
Beijing, China
mengming@buaa.edu.cn

Yuehua Wang
Department of Computer Science
Texas A&M University-Commerce
Texas, U.S.A 75429
yuehua.wang@tamuc.edu

Zhong Zhou*
State Key Laboratory of Virtual
Reality Technology and Systems
Beihang University
Beijing, China
zz@buaa.edu.cn

***Abstract*—We propose a rapid and accurate approach to recover the layout of a room automatically from a single fisheye image. It decomposes the fisheye image to a set of perspective images and jointly extract line images from the fisheye image and perspective images for geometric information. The semantic information gained from semantic segmentation on a cylinder expansion of the fisheye image are then used for structure line determination. By considering distinct features contained in the perspective images, the invalid hypotheses are filtered effectively and the most accurate structure lines are selected to minimize computational cost. To evaluate the effectiveness of the proposed approach, we construct an annotated fisheye image dataset. Comprehensive experimental evaluation on the dataset illustrate that our proposed approach produces higher quality layout estimations than existing layout reconstruction approaches and being 6 times faster in the reconstruction time.**

I. Introduction

Indoor layout estimation has received a lot of attentions in recent years with the explosive popularity of house selling and renting. It creates a high level of engagement and provides a viable means to view and interact with indoor environments regardless of cost, time, and spatial limitations. Typically, the basic methods for estimating indoor layout from a single image are extracting orthogonal vanishing points based on geometric information by clustering line segments in the scene. They generate a set of candidate box layouts with these line segments [1, 2]. These methods are widely used due to their simplicity and low complexity. However, they rely heavily on the orthogonal lines in the structure, and will get artifacts or wrong layout with non-orthogonal situation. To address this problem, subsequent studies [3–5] work on reducing the cost of computation complexity.

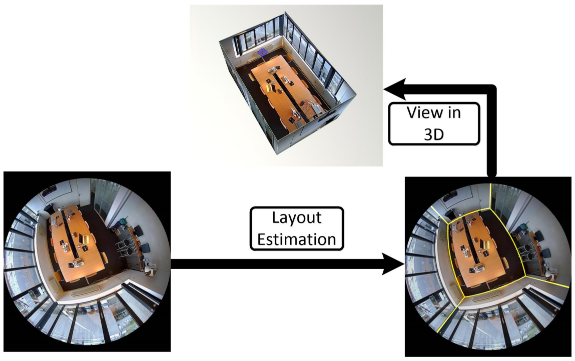

Fig. 1. Our method predicts a cuboid shape room layout from a single fisheye image.

Recent advances in the areas of deep learning have led to the development of layout estimations that build upon deep Convolutional Neural Network(CNN). In order to recover the room layout, Dasgupta et al. [3] use Fully Convolutional Network (FCN) to learn semantic surface labels. Zou et al. [6] use the vanishing point cues precomputed from perspectives, geometric constraints and a corresponding RGB panorama image to get the boundaries and corners with a deep Encoder-Decoder network. The majority of deep learning approaches use the perspective sub-images as inputs and can result in impressive layout reconstruction estimations, but with the cost of high computational complexity or low quality layout reconstruction. Note the field-of-view(FOV) of the perspective image is normally small, the layout of the entire room cannot be obtained easily, or even impossible. We argue that a fisheye

This work is supported by Natural Science Foundation of China under Grant No.61872023 and Grant No.61502020, and the China Postdoctoral Science Foundation under Grant No.2013M540039.

* Corresponding author.

978-1-7281-2009-6/$31.00 ©2019 IEEE

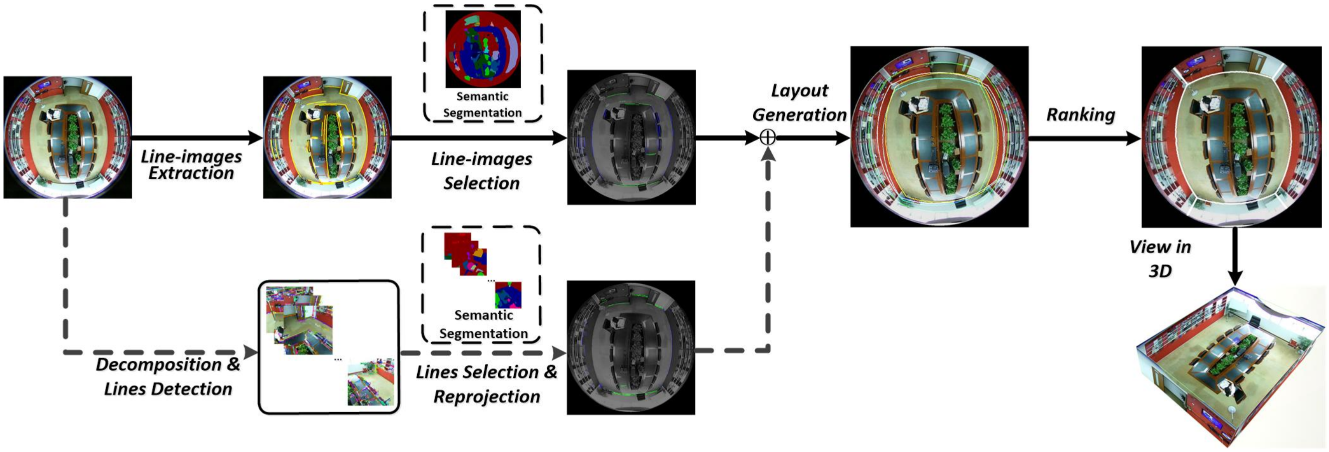

Fig. 2. Overview of our method.

camera with a single lens in reality can keep the natural of large FOV and has relatively lower acquiring cost than a panorama camera.

In this paper, we first investigate how 3D indoor layout can be restored with various cameras and inputs. Without introducing a depth cameras or relying on any depth information, we propose a rapid and accurate approach for 3D indoor layout estimation. It produces high quality layout reconstruction just with a single fisheye image, as shown in Fig. 1. In Fig. 1, a conference room has been well reconstructed in a cuboid shape with our method. Firstly, we extract the curves and get the semantic labels of the fisheye image. Secondly, for each line-image, we determine whether it is related to room structure with an energy function. Line-images that related to room structure are used to generate hypotheses. We then rank the hypotheses constructed with these structure line-images and choose the top one as the optimal layout.

The main contributions of this paper are summarized as follows:

- A novel approach is proposed that directly operate on a single fisheye image to accurately locate semantic indoor structure lines with corners and recover 3D indoor layout.
- Distinct features are always of a special importance in the 3D indoor layout reconstruction, in our approach, an algorithm for structure line determination is derived by fully exploiting the distinct features extracted from perspective images. That is in the fisheye images those features are scarcely noticeable.
- A fisheye image dataset is created by containing over 200 valid images in a cuboid structure with annotated planes, corners, and interesting lines based on available Internet data and the SUN360 dataset [7], which is the largest scale fisheye image dataset for 3D room estimation to now.

We also implement our proposed approach in the created dataset and conduct quantitative and qualitative comparisons of the state-of-the-art 3D indoor layout approaches. The extensive experimental results show that our approach not only has reconstruction time that about almost 6 times faster than PanoContext, but also achieves the best accuracy in both intersection boundary detection and layout estimation.

The rest of this paper is organized as follows. In section II, we provide a brief overview of the existing 3D indoor layout estimation approaches in the literature. Section III provides a detailed explanation of the proposed approach. Section IV demonstrates the effectiveness of the proposed approach using images of the newly created fisheye image dataset comparing state-of-the-art methods, followed by conclusions in Section V.

II. Related Work

The methods of indoor layout estimation by panoramic images have been widely studied in recent years [8–10]. Yang et al. [9] derive an occlusion detection method using a Markov random field (MRF) to select plausible constraints for reconstruction. Zhang el at [10] combine both bottom-up and top-down context information to output 3D bounding boxes of the room and all major objects. They concern a whole-room 3D context model to address the indoor scene understanding problem with a single panorama.

Note that the panoramic images are generally proposed by complex image stitching and inevitably have artifacts, it is rather difficult for users to generate correct models. PanoContext [10] is similar to our method in that it projects the panoramic image into a set of overlapping perspective images and combines the feature maps back into the panoramic image. There are two main problems in PanoContext. The first problem is that the detected lines do not have semantic properties. The number of hypotheses generated by a combination of random selected five lines are huge (more than 20,000) and the run time is really long (over 1 hour). The second problem is that they decompose the panorama image into a series of perspectives with small FOV to acquire structure lines with several unwanted side efforts. It loses unique features that can be abstracted from the panoramic view of the panorama and meaning of using the panorama in a certain sense.

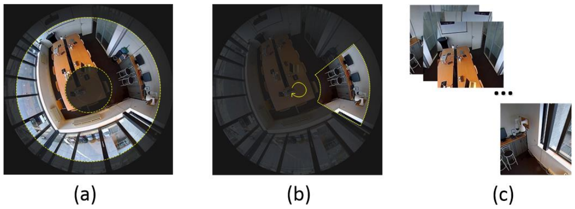

Fig. 3. (a) The highlight part between the two concentric circles is an Annular Area. (b) Slide window can decompose the Annular Area into a series of perspective images. (c) The output after decomposition.

In our approach, we extract the lines directly in the fisheye image and filter out the lines that are not in the ground-wall intersection area according to semantic segmentation. Different from PanoContext that randomly samples five non-degenerative lines to form a room layout hypothesis, in our approach, up to four lines are needed to determine a hypothesis. The number of hypotheses is greatly reduced. In [8], the layout of the whole scene is recovered by the combination of fisheye camera and depth camera. Perez-Yus et al. [8] combine depth information with the large view of fisheye camera to get the structure of room. Although this fisheye image-based layout estimation method produces better reconstruction results, the fisheye camera is needed to be calibrated along with a depth camera, which costs extra efforts and consequently reduces the methods' feasibility. This is in contrast to our work which performs indoor layout reconstruction merely with a single fisheye image. To our knowledge, this is the first work to tackle the single fisheye-based layout reconstruction in the field.

III. FISHEYE LAYOUT ESTIMATION

From a single fisheye image I, the proposed method estimates the layout based on three modules: preprocessing, structure line-images selection and layout determination. The preprocessing module includes line-images extraction and classification. Line-images are detected from the fisheye image and perspective images respectively and then classified according to two indicators. Structure line-images selection module selects structure-related line-images based on semantic segmentation. When the number of structurally related line-images detected in the fisheye image is less than the threshold, we will supplement it with the structural lines detected in the perspectives as shown by the dotted line in Fig. 2. Finally, layout determination module generates hypotheses and chooses the optimal one as the final estimation. The above process is shown in Fig. 2.

A. Preprocessing

1) Line-images Extraction: For fisheye image, we choose the work from [11] for line-images extraction. 3D lines in space are shown as straight lines in perspectives, but curved lines in omnidirectional images which are called line-images. The shapes of these line-images vary alone with the change of camera types. Each 3D line L_i forms a plane α_i with the principal point of the camera. For every point P lying on 3D line L_i, the projection of P must satisfy the condition $n_{\alpha_i}^T \cdot p = 0$, where n_{α_i} denote the normal of the plane α_i. The constraint for points on the line projection in image coordinates is as follows:

$$n_x \cdot \hat{x} + n_y \cdot \hat{y} + n_z \cdot \hat{r} \cot \frac{\hat{r}}{f} = 0 \qquad (1)$$

where $\hat{x}$ and $\hat{y}$ refer to the image coordinates of $\hat{p} = (\hat{x}, \hat{y})$, $\hat{r}$ represents the polar of $\hat{p}$ in polar coordinates and $n_{\alpha_i} = (n_x, n_y, n_z)^T$. We extract f and normal n_α of every 3D line as main calibration parameters. We notice that the normal n_α can measure the similarity of every line-image extracted from fisheye image, and can be used to merge similar line-images and simplify calculation.

For perspective images, we apply the line segment detection (LSD) algorithm [12] to extract lines on perspective images. Different from [10], we only project a specific Annular Area into a series of perspective images instead of the entire image. Annular Area, as shown in Fig. 3(a), is the area where corners and boundaries appear frequently. In order to obtain the location of corners and boundaries more accurately, we use the slide window to decompose the Annular Area into a series of specified-sized (320 $*$ 320) perspectives with overlapping areas as shown in Fig. 3(c). Since the perspective is enlarged after the decomposition process, the details that are not obviously noticeable in the source fisheye image would be clearly displayed in the perspectives.

2) Line-images Classification: We project the lines obtained from the perspectives back into the fisheye image and classify these line-images based on two indicators: orientation and position. Orientation refers to the direction of the vanishing point to which the current line-image belongs. Position refers to the spatial position of the surface where the current line-image is located. In the process of hypotheses generation, we randomly select one line-image from each category to form a closed area.

Orientation: Every line-image in 3D space corresponds to a part of great circle on the unitary sphere and appears as a curve in fisheye image. For each line-image l_i, we use n_{α_i} to denote where it lies on. The vanishing direction $V_{p_i} = (e_{1_i}, e_{2_i}, e_{3_i})$ associated with the line l_i should be perpendicular to n_{α_i}. We use a RANSAC-based algorithm to determine the location of three vanishing points and then mark the orientation label for each line-image according to the vanishing point that it belongs to.

Position: Another important projection property of sphere camera model is that the great circles of 3D parallel lines intersect in two antipodal points $p^{'}$ and $p^{''}$ in sphere. The connection of these two points is recorded as $l_{p^{'}p^{''}}$. We can divide the line-image belonging to the same vanishing point into two parts according to the relative positions of the line-image itself and their corresponding $l_{p^{'}p^{''}}$.

Classification: We assume that after rotation, the vanishing point in the horizontal direction is v_2, and the vanishing point in the vertical direction is v_3. The line-images belonging to the same vanishing point can be divided into two parts according

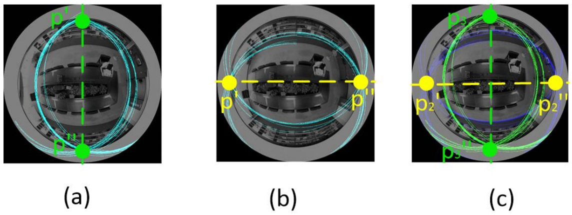

Fig. 4. (a) Line-images belong to v_3 . (b) Line-images belong to v_2 . (c) Line-images classification: The set of green line-images on the left side of the green dotted line belong to S_l and right side belong to S_r. The set of blue lines on the up side of the light yellow dotted line belong to S_f and below side belong to S_b.

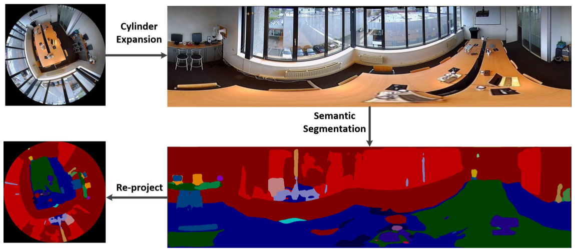

Fig. 5. The process of generate semantic labels for fisheye image.

to their two antipodal points as shown in Fig. 4. The line-images belong to v_2 and lie in the upper part of line $l_{p'p''}$ are labeled as front and denoted as S_f. The lower part are labeled as back denoted as S_b. Similarly, the line-images belong to the vanishing point v_3 can be divided into two parts according to their two antipodal points, and the left part are labeled as left and denoted as S_l and the right part are labeled as right and denoted as S_r.

Therefore, all of the line-images, including that generated by back-projected to the fisheye image from perspectives, are classified into four sets: S_f, S_b, S_l and S_r. So we can get the following equation:

$$L_{fish} \cup L_{pers} = S_f \cup S_b \cup S_l \cup S_r \quad (2)$$

where L_{fish} respects the line-images detected in the fisheye image and L_{pers} denotes the lines extracted from the perspective images.

Because the wall-wall boundaries are perpendicular to floor in Manhattan world, we can gain the layout estimation by determine the wall-floor boundaries and corners. So, after classification, we randomly sample four line-images with semantic limitation from S_f, S_b, S_l and S_r to form a closed curved quadrilateral as hypothesis.

B. Structure Lines Determination

1) Semantic Segmentation: We use semantic segmentation results as input for structure line selection. Since the kernel of the deep neural network is rectangular, we first convert the fisheye image to a rectangular image, and use RefineNet [13] to get suitable semantic label for each pixel. With rich contextual information in fisheye images, semantic label for each pixel is more accurate than that in perspective image with small FOV. By training the model on ADE20K [14], we can obtain semantic labels on the cylindrical expansion of the fisheye image. Then we re-project the semantic labels back to fisheye image. For perspective images, we perform semantic segmentation with RefineNet. The process of generating semantic labels for fisheye image is shown in Fig. 5.

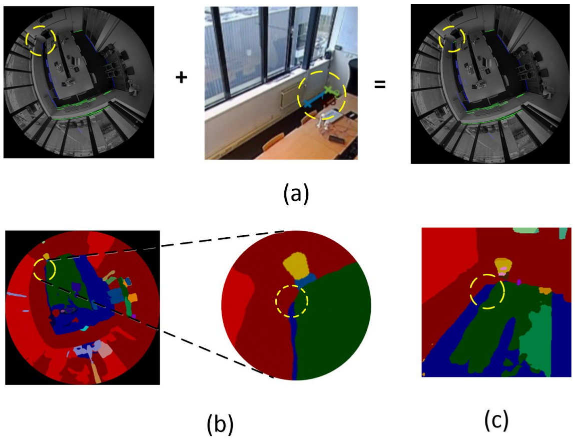

Fig. 6. (a) Before and after supplementation. (b) Semantic labels for source image. Dark red for wall and blue for ground. (c) Sematic labels on perspective image.

2) Structure Line-images Determination: We determine the initial set of structural lines by selecting ling-images. It is then supplemented with line-images extracted from the perspective images.

Structure Line-images Selection: Ideally, the line lies on wall-floor boundaries extracted from the RGB image should also be the boundary line for semantic labels of wall and ground in the semantic segmentation map. But in fact, the accuracy of semantic segmentation is not perfect, these lines are not always aligned. So, we design a strategy to preserve the lines with high probability of separating the walls and floor.

We design an energy function E_{l_i} to score every line-image in L_{fish} or every line in L_{pers} according to its respective semantic labels:

$$E_{l_i} = \sum_{p \in l_i} w \cdot f_E(p, \sigma_{win}) \quad (3)$$

where l_i stands for a line-image in L_{fish} or a line in L_{pers}, p represents pixel on l_i. w is the weight item and its default value is 1. When the scenes are cluttered with complex geometry, we increase the w of L_{pers} to get more details as a supplement. $f_E(p, \sigma_{win})$ is a measurement of the distribution of semantic labels within range σ_{win} of pixel p. The score E_{l_i} is higher when the degree of mixing of wall label and the floor label on either side of line l_i is lower. After determining the energy of each line, we keep the lines whose energy is higher than τ.

Structure Line-images Supplementation: The scenes are usually cluttered by objects severely. In this case, most of the intersection of the walls and the ground is blocked by the object. Therefore, the number of line-images that extracted on the fisheye image while satisfy the semantic limitation may be too small. At this point, as shown in Fig. 6, the lines detected on the perspective images should be used as a supplement to increase the robustness of the system.

Each line-image l_i in sets S_f, S_b, S_l and S_r includes two values: one is label L_{fish} or L_{pers}, where L_{fish} means l_i has been extract from fisheye and L_{pers} means l_i has been extract from perspectives. The other is energy E_l calculated from Eq.3. We divide each set into three subsets with different priorities. Finally, we use the highest priority subset of each set to generate hypotheses. The specific method is shown in Algorithm 1.

```
Algorithm 1: Structure Line-images Determination
Input: Four line-image sets $S_f$, $S_b$, $S_l$ and $S_r$
Output: $S_{res} = \{S_f^1, S_b^1, S_l^1, S_r^1\}$
$S_{res} = \phi$
for $S \leftarrow S_f, S_b, S_l, S_r$ do
    $S^1 = \phi; S^2 = \phi; S^3 = \phi$;
    for $l_i \in S$ do
        if $Label(l_i) == L_{fish} \wedge E_{l_i} > \tau$ then
            $S^1 \leftarrow S^1 + l_i$ by adding the line-image $l_i$ into $S^1$;
        else if $Label(l_i) == L_{pers} \wedge E_{l_i} > \tau$ then
            $S^2 \leftarrow S^2 + l_i$ by adding the line-image $l_i$ into $S^2$;
        else
            $S^3 \leftarrow S^3 + l_i$ by adding the line-image $l_i$ into $S^3$;
    set $priority(S^1) > priority(S^2) > priority(S^3)$;
    if $|S^1| < \tau_c$ then
        if $S^2 \neq 0$ then
            $S^1 = S^1 \bigcup S^2$;
        else
            $S^1 = S^1 \bigcup S^3$;
    $S_{res} = S_{res} \bigcup S^1$
return $S_{res}$;
```

C. Layout Generation and Ranking

Room layout hypotheses in fisheye image can be generated by connecting line-images to create room corners. Once the hypotheses are generated, a joint inference is applied to select the most appropriate combination of them as an interpretation. The number of hypotheses depending on the trade-off between accuracy and speed is always very large, i.e. 20,000 in [10]. In this section, we will illustrate how to generate and rank layout from line-images of fisheye image.

1) Similar Line-images Merger: Similar line-images often generate similar hypotheses. In order to reduce this cost, we decide to merge similar line-images in spatial position. For every two line-images l_i and l_j in each set S_f, S_b, S_l and S_r, we make decision based on line-image's normal vector n_α. If the angle between n_{α_i} and n_{α_j} is less than ∂_m while the starting point of one line-image is located on another line-image, we will merge them. The normal vector n_α of the new merged line-image is equal to the longer one. The starting and ending points of the new merged line-image are same as the minimum starting point and the maximum ending point of the two line-images.

2) Hypotheses Generation: Our hypotheses generation is based on corners produced by four structural lines' intersection. In Manhattan World assumption, two lines are enough to define a corner. We iteratively choose one line-image in every structure set $(S_f^1, S_b^1, S_l^1, S_r^1)$ to form a group of corners G_{cor}. These corners are clockwise arranged in the XY–plane. Each group of corners generates a hypothesis. We start with S_l^1, for example, to select a line-image l_i, then we extend it in two directions alone with the track of the big circle which it lies on. Next, in a clockwise direction in the XY–plane, a line-image l_j in S_f^1 is randomly selected and prolonged to find its intersection with l_i. We will choose another one $l_j^{'}$ in S_f^1 if there is no intersection between l_j and l_i after extending the specified length σ_{len}. According to this process, we can gain a closed Manhattan layout for every group of line-images. We record the id for each line-image in every group and then score each hypothesis.

3) Ranking: In the evaluation process, we determine which one is the best from all layout hypotheses H_i generated in the last stage. The ranking function consists of three items, one measures the score of corners, the other two measure the fitness of hypothesis and the orientation map as well as semantic segmentation.

Our ranking formulation is:

$$\begin{aligned} h^* = \arg\max_h \sum_{h_i} [& w_1 \cdot AccCor(h_i) \\ & + w_2 \cdot AccOM(O^{h_i}, O^{world}) \\ & + w_3 \cdot ErrSem(R^{h_i}, S^{world})] \end{aligned} \tag{4}$$

where h_i is one of hypotheses in H_i. We use *AccCor* to determine the score of corners for each hypothesis, *AccOM* to measure the difference of orientation between hypothesis and ground truth, *ErrSem* to measure the proportion of inaccurate semantic labels between the hypothesis and the scene in real world. Weight w_1, w_2 and w_3 determine the weight of the corresponding item. We will explain the details of each item in the following sections.

Corner Evaluation: Each hypothesis contains four corners and each corner defined by two line-images. Inspired by the work of [8], the accuracy of each corner depends on two factors: a) the length of corresponding line-images l_i and l_j that intersect at the corner; b) the distance between the corner and two nearest endpoints of l_i and l_j.

$$\begin{aligned} AccCor(h_i) = \sum_L [& f_{leng}(l_i, l_j) \\ & + \frac{1}{f_{dist}(l_i, l_j, f_{intsec}(l_i, l_j))}] \end{aligned} \tag{5}$$

where *L* is the collection of lines in h_i and f_{intsec} denotes the point of intersection of two line-images. f_{leng} measures the length of two corresponding line-images and f_{dist} measures the distance between two line-images' endpoints and the corner.

Orientation Map Evaluation: We use orientation map to evaluate the fitness of the orientation between hypothesis and the surfaces in real world. Orientation map, introduced by [2], is an image whose pixels encode the believed orientation according to the line segments and the vanishing points. For each hypothesis h_i, we generate a labeled image O^{h_i}, in which each pixel encodes the orientation of the surface. We divide O^{h_i} in three regions $O^{h_i}_{floor}$, $O^{h_i}_{LRwall}$, $O^{h_i}_{FBwall}$ representing the region of floor, the region of left and right wall and the region of front and back wall. For the orientation map O^{world} about the surface in the world, we also divide it in three parts as O^{world}_x, O^{world}_y, O^{world}_z according to the vanishing points. The AccOM is given by:

$$AccOM(O^{h_i}, O^{world}) = \sum_{ch} \frac{O^{h_i}_{ch} \bigcap O^{world}_{ch}}{O^{h_i}_{ch} \bigcup O^{world}_{ch}} \tag{6}$$

where ch represents the number of channels. $O^{h_i}_{ch}$ denotes the ch channel of O_{h_i}.

Semantic Evaluation: The value of $ErrSem(R^{h_i}, S^{world})$ is used to measure the proportion of inaccuracy between semantic label in the world S_{world} and semantic region in the hypothesis R^{h_i}. We divide h_i in two regions: floor and walls. A place outside the closed area encircled by the line-images in h_i is belongs to walls $R^{h_i}_{wall}$ and the other is seen as floor $R^{h_i}_{floor}$. The pixels with wall semantic tags in S^{world} belong to S^{world}_{wall}, and the pixels with the ground semantic tags in S^{world} belong to S^{world}_{floor}. The ErrSem is given by:

$$\begin{aligned} ErrSem(R_{h_i}, S_{world}) = -\frac{1}{|P|} \cdot (R^{h_i}_{wall} \bigcap S^{world}_{floor} \\ + \lambda \cdot R^{h_i}_{floor} \bigcap S^{world}_{wall}) \end{aligned} \tag{7}$$

where $|P|$ is the total number of pixels in the fisheye image. λ determined by the clutter of room. When there are many objects in the room, the accuracy of semantic segmentation about floor will decrease. Consequently, the proportion of score for walls should be larger.

IV. EXPERIMENT

We conduct our experiments on a newly created fisheye image dataset crawl from Internet and SUN360 dataset [7], and perform quantitative and qualitative evaluations by comparing to two representative indoor layout reconstruction approaches LayoutNet [6], PanoContext [10]. We test our method and panoContext on Linux machine with Intel Xeon 3.5G Hz in CPU mode and a single NVIDIA Titan X GPU for LayoutNet.

A. Dataset

While the research on 3D indoor layout reconstruction has been receiving intensive attention, the public datasets to date are still the most scare resource in field. We have studies all relevant image datasets that can be partially or possibly used for 3D indoor layout reconstruction and argued that the majority of datasets are small (e.g., up to 70 images) and very specific (e.g., containing perspective or panoramic images). To substantially evaluate our approach and provide a new means of 3D indoor layout reconstruction, we collect fisheye images and create a new large dataset after essential preprocessing like determine the effective domain of the fisheye image which is currently available to the scientific community. The dataset consists of 200 fisheye images mainly collected by reprojecting from SUN360 dataset, and from the Internet. Given the retrieved fisheye images, those without a cuboid structure, are moved manually. For every fisheye image, we marked the corners and the surface of walls. Similar to PanoContext, we recover cuboid shape layout from a single fisheye image. Fig. 7 shows representative fisheye images of different indoor scenes in our dataset. As is often the case, object occlusions and cluttering commonly appears in images. We highlighted its importance by collecting 200 images from different scenes. Our dataset contains 70 images acquired with a fisheye camera and 130 images re-projected from the panorama in SUN360 dataset, including 40 offices, 90 bedroom rooms and 70 living rooms. To annotate these fisheye image, we design a Matlab annotation tool to mark the corners, structure lines and the orientation of each surface.

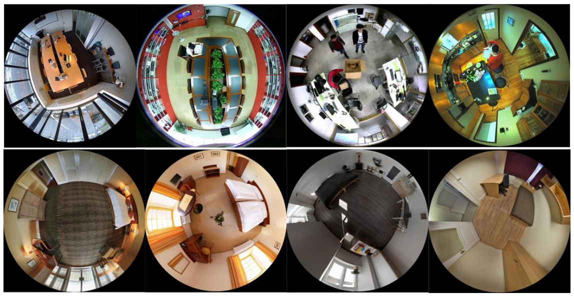

Fig. 7. Representative fisheye image selected in different scenes in our dataset.

B. Quantitative Evaluation

To find the proper weights in the scoring function (4), we test the impact of each sub-item in the scoring function. We first calculate the accuracy of corner estimation by employing keypoint Error (KE) and Pixel Accuracy (PA) as evaluation metrics. Keypoint Error (KE) refers to the distance of pixels between the estimated points and the annotated points normalized by the diagonal length of fisheye image's cylindrical expansion. Pixel Accuracy (PA) represents pixelwise error between the predicted surface labels and ground truth labels.

Table I shows the quantitative comparison among the proposed scoring functions. Our approach $AccCor + AccOM + ErrSem(4)$ outperforms other individual functions in terms of pixel accuracy and keypoint errors. It comes from a good tradeoff among the accuracy of corner detecting, the accuracy of orientation map and the error of semantic map. In contrast, for those three scoring functions, $AccOM(6)$ achieves higher

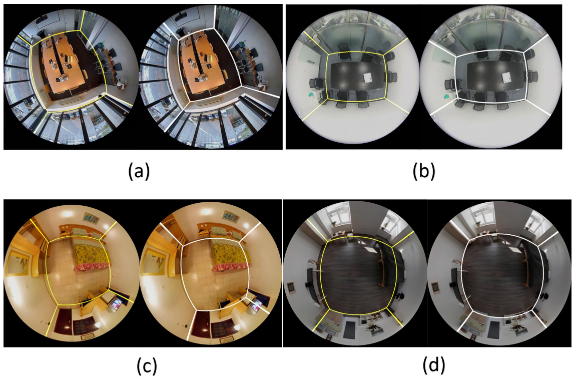

Fig. 8. Qualitative comparison results with LayoutNet. Yellow lines are the results of LayoutNet and white lines are our result. For ease of comparison, we re-project the panoramic images of LayoutNet into fisheye images to display.

TABLE I
RESULTS ON OUR FISHEYE DATASET

Method	*Keypoint Error(%)*	*Pixel Accuracy(%)*
AccCor(5)	2.63	85.53
AccOM(6)	1.70	93.71
ErrSem(7)	3.57	82.76
AccCor+AccOM+ErrSem(4)	1.08	95.80

accuracy by selecting matching degree between orientation of each surface in hypotheses and in real scene. When the position of a corner in the hypothesis is slightly inaccurate, two or more surfaces would be affected. This will magnify the minor errors in the hypothesis and differentiate hypotheses scores. In this way, the best candidate can be selected. $ErrSem(7)$ uses the difference between the distribution of ground and wall in hypotheses and in real scene as filter condition. The accuracy is the lowest when we use it independently, as shown in the third row of Table I. That is because in a real scene, most of the ground is cluttered by objects. The semantic label of the blocked part is object, which reduces the pixel proportion of ground and wall in the picture. Thus, when there is a tiny difference in the corner, semantic image is not sensitive enough to recognize it and filter ability is relatively weak. $AccCor(5)$ uses the length of curves and the distance between curves and intersections to rank hypotheses. When using $AccCor(5)$, independently, result accuracy can be easily affected by the degree of clutter of objects in the room. When there are fewer objects in the room, multiple long curves appear. At this moment, $AccCor(5)$ will have better filter ability. Therefore, we combine application of these three scoring functions together, as shown in the fourth row of Table I, to handle different indoor scenes and achieve the best results.

C. Qualitative Evaluation

We compared our method with a neural network based method LayoutNet and a geometric information based method PanoContext.

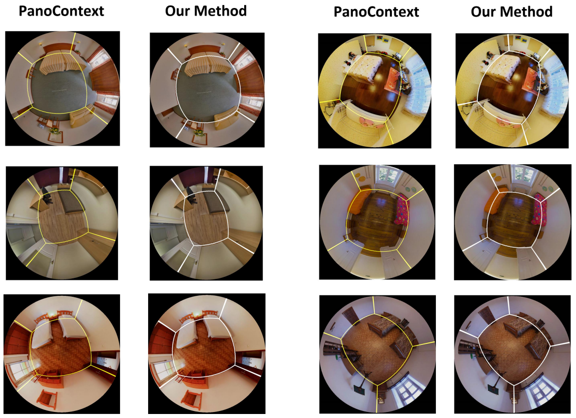

Fig. 9. Qualitative comparison between PanoContext and our approach for cuboid layout prediction on SUN360 dataset. Yellow lines are the results of PanoContext and white lines are that of ours. Similar to Fig.8, the panoramic images are reprojected to fisheye images.

LayoutNet utilizes panoramic images and deep neural network to capture the structural layout of the room. It performs better in terms of time consumed and accuracy compared to other existing work on panoramas. We first conduct a comparison with LayoutNet on the fisheye images collected from the Internet in our dataset. In particular, the fisheye images and theirs cylinder expansion are used in our approach and LayoutNet, respectively. Fig. 8(a) and (b) show the detected intersecting lines between planes in white with our approach and that of LayoutNet in yellow. Since there is less clutter in the upper part of the walls in panorama, the intersections between walls can make a clear judgment based on the orientation of lines. However, due to the lack of semantic constrains, the intersections between the ground and the wall are often missed or mis-delineated, which significantly degrades the accuracy and quality of 3D indoor layout reconstruction, especially for the rooms with object occlusions and clutters. In our approach, we consider the semantic segmentation information, leading to more accurate structure lines between wall and ground. We make another comparison with LayoutNet on the SUN360 dataset. The deep neural network infers the position of the structural line by acquiring features within the image to obtain room layout. Low-level texture information gradually misses due to convolution and pooling operations. Therefore, deep neural networks can only infer the approximate position in some cases and cannot restore the original position of the true structural line. By combining lowlevel texture information with high-level semantic information, we can get the original position of the structure line. This makes our layout estimation more accurate, are shown in Fig. 8(c) and (d). The results demonstrate our proposed algorithm is very effective in detecting intersecting lines between planes, and can efficiently apply to different scenes with various object occlusion and clutters.

PanoContext projects the panoramic image into a set of overlapping perspective images, and then combines with feature maps to yield a panoramic image with more accurate structural layout. For comparability, we reproject the panoramic images captured in cuboid shape rooms to fisheye images and add to our dataset. The inputs for PanoContext are original pictures in SUN360 dataset and for ours are reprojected fisheye images. The experimental results show that the running time of the panoContext is more than 1 hour, and ours is no more than 10 minutes. Regarding the room layout effect, PanoContext is based on a large number of hypotheses, and the optimal layout estimate by the scoring function is greatly affected by the degree of room confusion. Although the final layout hypothesis is also generated based on low-level texture information and jointly optimized with objects, its accuracy in estimating the position of the structure line is quite low. We observe that some segments detected on the wall and ground are treated mistakenly as structural lines to form a layout hypothesis due to the lack of semantic label constraints. In our method, there are only no more than 2000 hypotheses, making the computing time really short. Under the premise of ensuring efficiency, our method can accurately locate the position of the structural line through the semantic labeling, and then infer the position of the corner and obtain a reasonable indoor layout estimation as shown in Fig. 9.

V. Conclusion

In this paper, we have presented an approach that enables to recover a cuboid shape room with the Manhattan World assumption by directly operating a single fisheye image. Our approach highlights the impact of semantic information and distinct structural features on the results of the layout reconstructions in the presence of object occlusions and clutters. We newly create a fisheye dataset for approach evaluations and the experimental results provide a strong evidence for the effectiveness and feasibility of our proposed approach. As a part of our future work, we will investigate a deep neural network for fisheye images to generate layout estimation and combine with the specific semantic lines to handle the non-cuboid layout.

References

[1] D. C. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[2] V. Hedau, D. Hoiem, and D. Forsyth, "Recovering the spatial layout of cluttered rooms," in *IEEE International Conference on Computer Vision*, 2010.

[3] S. Dasgupta, K. Fang, K. Chen, and S. Savarese, "Delay: Robust spatial layout estimation for cluttered indoor scenes," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] L. Del Pero, J. Bowdish, B. Kermgard, E. Hartley, and K. Barnard, "Understanding bayesian rooms using composite 3d object models," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[5] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun, "Efficient structured prediction for 3d indoor scene understanding," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2815–2822.

[6] C. Zou, A. Colburn, Q. Shan, and D. Hoiem, "Layoutnet: Reconstructing the 3d room layout from a single rgb image," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2051–2059.

[7] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE conference on Computer vision and pattern recognition (CVPR)*. IEEE, 2010, pp. 3485–3492.

[8] A. Perez-Yus, G. Lopez-Nicolas, and J. J. Guerrero, "Peripheral expansion of depth information via layout estimation with fisheye camera," in *European Conference on Computer Vision*. Springer, 2016, pp. 396–412.

[9] H. Yang and H. Zhang, "Efficient 3d room shape recovery from a single panorama," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5422–5430.

[10] Y. Zhang, S. Song, P. Tan, and J. Xiao, "Panocontext: A whole-room 3d context model for panoramic scene understanding," in *European Conference on Computer Vision*. Springer, 2014, pp. 668–686.

[11] J. Bermudez-Cameo, G. Lopez-Nicolas, and J. J. Guerrero, "Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry," *International Journal of Computer Vision*, vol. 114, no. 1, pp. 16–37, 2015.

[12] G. V. G. Rafael, J. Jérémie, M. Jean-Michel, and R. Gregory, "Lsd: a fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[13] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multipath refinement networks for high-resolution semantic segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5168–5177.

[14] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 2, 2017.