Peripheral Expansion of Depth Information via Layout Estimation with Fisheye Camera

Alejandro Perez-Yus, Gonzalo Lopez-Nicolas, Jose J. Guerrero

Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza
{alperez,gonlopez,josechu.guerrero}@unizar.es

Abstract. Consumer RGB-D cameras have become very useful in the last years, but their field of view is too narrow for certain applications. We propose a new hybrid camera system composed by a conventional RGB-D and a fisheye camera to extend the field of view over 180 degrees. With this system we have a region of the hemispherical image with depth certainty, and color data in the periphery that is used to extend the structural information of the scene. We have developed a new method to generate scaled layout hypotheses from relevant corners, combining the extraction of lines in the fisheye image and the depth information. Experiments with real images from different scenarios validate our layout recovery method and the advantages of this camera system, which is also able to overcome severe occlusions. As a result, we obtain a scaled 3D model expanding the original depth information with the wide scene reconstruction. Our proposal expands successfully the depth map more than eleven times in a single shot.

Keywords: 3D layout estimation · RGB-D · Omnidirectional cameras · Multi-camera systems

1 Introduction

Recent low cost RGB-D cameras have caused a great impact in the fields of computer vision and robotics. These devices usually have a field of view (FoV) too narrow for certain applications, and it is necessary to move the camera in order to capture different views of the scene. However, that is often not easy to achieve or requires to use SLAM algorithms or additional sensors to maintain the system well localized. Recently, some alternatives to extend the FoV of depth cameras using additional elements have been proposed: [1] uses two planar mirrors as a catadioptric extension of the RGB-D device and [2] a consumer set of wide angle lens. Fernandez-Moral et al. [3] proposed a method to calibrate an omnidirectional RGB-D multi-camera rig. While these approaches are interesting, they are either complex to build and calibrate [1,3], or do not provide good enough depth maps [2].

Here, we propose a new hybrid camera system composed by a depth and a fisheye camera. The FoV of the fisheye is over 180°, in contrast with the usual FoV of 43° × 57° of consumer depth cameras (Fig. 1a). Once the cameras are

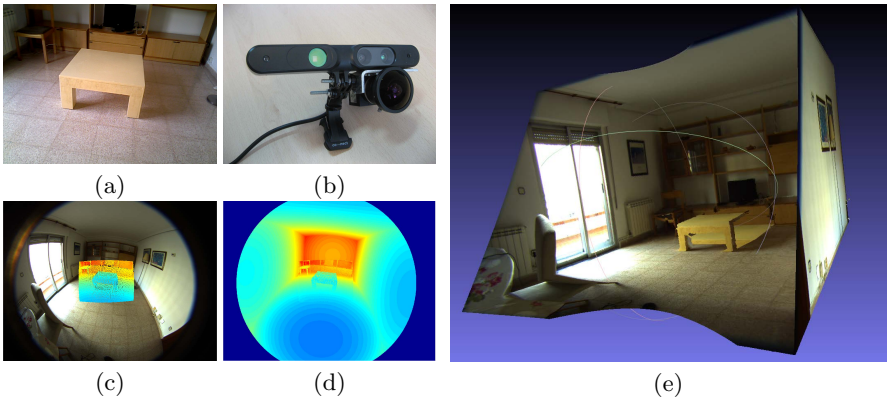


Fig. 1: (a) Image scene as view from a conventional RGB-D camera. (b) Proposed camera system. (c) Depth map projected to the fisheye camera. (d) Expansion of the depth map through spatial layout estimation. (e) From the wide field of view depth map of the scene, we compute a 3D model in a single shot.

calibrated, the system (Fig. 1b) is capable of viewing over a hemisphere of color information where the central part of the image has also depth data (about 8.7% of the total number of pixels, as shown in Fig. 1c). One can think of this configuration inspired in the vision of the human eye, where the central part provides richer information (foveal vision) than the periphery, and the field of view is slightly over 180°. To our knowledge, this is the first time this configuration has been used, although the interest in such sensor pairing is clear in the recent Google's Tango[1]. Notice that, although our work uses a fisheye camera, the approach could be extended to other kinds of omnidirectional systems.

The main contribution of this work is the proposal of a method to extend 3D information from a conventional depth camera to over 180° of field of view in one single shot. The depth camera provides a region of the image with 3D scaled data, from which some basic information about the scene can be recovered. To extend the depth information we propose a spatial layout estimation algorithm based on line segments from the fisheye image, which provides scaled solutions rooted on the seed depth information (Fig. 1d). The line segments from the fisheye image are classified according to the Manhattan directions and projected to a scaled 2D map where the layout hypotheses are built based on physically coherent wall distributions. The algorithm is able to work even under high clutter circumstances due to the combination of lines from both floor and ceiling. The corners of the map are evaluated by our scoring function, and layout hypotheses are proposed by the probability of these corners to occur in the real world. For the evaluation stage we propose three alternative methods.

As a result, a final 3D scene reconstruction is provided. The 3D room layout can be seamlessly merged with the original depth information to generate a

[1] `https://get.google.com/tango/`

3D image with the periphery providing an estimation of the spatial context to the central part of the image, where the depth is known with good certainty (Fig. 1e). The collaboration between cameras is bidirectional, as the extension of the scene layout to the periphery is performed with the fisheye, but the depth information is used both to enhance the layout estimation algorithm and to scale the solution. Experiments using real images show promising results about both the proposed algorithm and the camera configuration.

2 Related Work

Probably the first attempt to recover 3D reconstructions of indoor environments with single images was [4], which uses a Bayesian network model to find the floor-wall boundary. In contrast, Lee et al. [5] use line segments to generate layout hypotheses evaluating their fitness to an orientation map. Using lines has the advantage of producing results without relying on scene-specific properties such as colors and image gradients. However, while some lines can actually include structural information of the environment (e.g. intersections wall-wall or wall-floor), usually most of them belong to clutter or are useless and misleading.

To help with this problem, some assumptions are made, and consequently some set of rules are proposed based on physical coherence. Usually the main assumptions are that all structures in indoor environments are composed by planar surfaces and that these surfaces are oriented according to three orthogonal directions (known as Manhattan World assumption [6]). This assumption holds for most indoor environments, and it is widely used in the literature [7,8]. Other works try to simplify the problem by making assumptions about the structure, e.g. assuming that the room is a 3D box [9,10,11,12]. In [13,14,15,16,17] they used this kind of reasoning to simultaneously perform object detection. Other approaches make use of video sequences instead of single images [18,19].

These methods have in common that all of them use images from conventional cameras. As opposed to that, some recent works use omnidirectional cameras such as catadioptric systems or fisheye cameras. Having greater field of view has many advantages for this task:

- Cameras are able to capture more portion of the room at the same time, which provides complete room reconstructions.
- Line segments appear entirely, so the reasoning is based on more evidence.
- Larger field of view (FoV) provides better view of the ceiling, which may help in cluttered scenes assuming structural floor-ceiling symmetry.

Some related works taking advantage of omnidirectional cameras are [20,21,22,23]. In [20], they use a fisheye camera to perform layout retrieval, essentially extending the work from [5]. Lopez-Nicolas et al. in [21] perform the layout recovery using a catadioptric system mounted in a helmet. Jia and Li [22] use 360° panorama full-view images, which allows them to recover the layout of the whole scene at once. Similarly, [23] uses the same type of images to perform layout retrieval along with a whole-room context model in 3D including bounding boxes of the main objects inside the room.

One common feature of all these approaches mentioned above is that the recovered 3D layout is obtained up to a scale. Our proposal combines the advantages of omnidirectional cameras (recover wider information) and depth cameras (provide 3D certainty and scale) with an easy to reproduce camera system.

3 Depth and Fisheye Images Processing

Before addressing the layout recovery, in this section we present the essential computer vision procedures in our approach, including calibration of the system (Section 3.1), line extraction (Section 3.2), estimation of Vanishing Points (Section 3.3) and classification of lines and points (Section 3.4).

3.1 System Calibration

To map world points $\mathbf{X}$ from the depth camera reference frame D to the fisheye camera reference frame F, it is necessary to calibrate the extrinsic parameters $(\mathbf{R}, \mathbf{t})$ and the intrinsic parameters of both cameras. The extrinsic calibration of range sensors to cameras is not a new issue, but most related works require manual selection of correspondences or do not support omnidirectional cameras [24,25,26]. To obtain the intrinsic parameters of the fisheye camera, we need a specific method with an appropriate camera model [27]. We adapted the proposal from [28], substituting the color camera model to the one from [29]. The intrinsic parameters of the depth camera are also computed as defined in [28] to improve the default parameters of the system. The depth images as captured by the sensor are transformed to point clouds using these parameters, and then they are rotated and translated to the fisheye camera reference frame, following $\mathbf{X}_F = \mathbf{R} \cdot \mathbf{X}_D + \mathbf{t}$. From now on, every computation is done in that frame. A more detailed analysis of our calibration procedure is presented in [30].

3.2 Line Extraction in the Fisheye Image

For the line extraction in the fisheye camera we choose the work from [31], which includes a method for self-calibration and line extraction for central catadioptric and dioptric systems with revolution symmetry. It uses the sphere camera model [32] to describe the point projection in dioptric systems (Fig. 2a). Every 3D world point $\mathbf{X}_i$ is first projected in a point $\mathbf{x}_i$ onto a unitary sphere around the viewpoint of the system. In the case of our equiangular fisheye lens this point is projected to $\hat{\mathbf{x}}_i$ in the image plane by using the projection function $\hat{r} = f\phi$, where $\hat{\mathbf{x}} = (\hat{r}, \hat{\theta})$ is expressed in polar coordinates with respect to the image center, ϕ is the elevation angle in the sphere and f is the main calibration parameter.

Unlike conventional cameras, 3D lines in space do not appear as straight lines in the omnidirectional images, but they are projected to curves called line-images. In the schematic scene from Fig. 2a we can see highlighted a vertical line on the sphere model and its projection in the fisheye image. The shape of these line-images changes with the type of omnidirectional camera and its specific

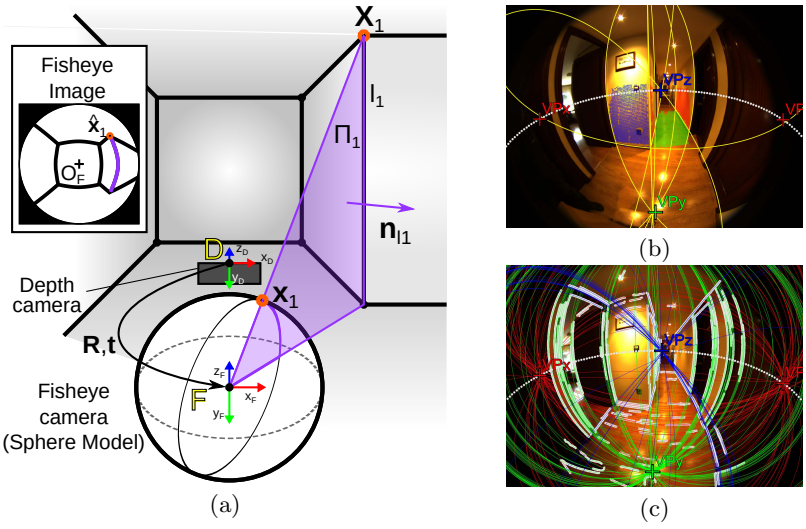


Fig. 2: (a) Scheme of the system in a 3D world scene and the correspondent fisheye image. (b) Depth planes classified according to the Manhattan directions (red in $\mathbf{m}_x$, green in $\mathbf{m}_y$, blue in $\mathbf{m}_z$), initial extracted vanishing points, horizon line (white dotted line), and 3D intersections in yellow lines. (c) Line-images classified with their contours in white and the vanishing points after the second optimization.

camera configuration. We have used the implementation from [31] to extract the main calibration parameter f and optical center from the images.

The projection of a line l_i in the 3D space can be represented by the normal of the plane $\mathbf{\Pi}_i$ defined by the line itself and the viewpoint of the system, with normal $\mathbf{n}_{l_i} = (n_x, n_y, n_z)^\top$. The points $\mathbf{X}$ lying on a 3D line l are projected to points $\mathbf{x}$ satisfying the condition $\mathbf{n_l}^\top \mathbf{x} = 0$. From [31], the constraint for points on the line projection in image coordinates for equiangular dioptric systems with symmetry of revolution is:

$$n_x\hat{x} + n_y\hat{y} + n_z\hat{r}\,\cot(\hat{r}/f) = 0 \tag{1}$$

where $\hat{x}$ and $\hat{y}$ are the image coordinates centered in the principal point, $\hat{\mathbf{x}} = (\hat{x}, \hat{y})$. The line-images are non-polynomial and do not have conic shape. To extract them is necessary to solve a minimization problem [31].

3.3 Extraction of the Vanishing Points

In this work we assume the scenes satisfy the Manhattan World assumption, i.e. the world is organized according to three orthogonal directions $(\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z)$. Parallel lines in the 3D world intersect in one single point in perspective images, called Vanishing Point (VP). In omnidirectional images, line projections result in curved line-images, and parallel lines intersect in two VPs. We estimate the VPs to classify lines and planar surfaces from the depth information according to the three Manhattan directions.

There are previous approaches to obtain the VPs from omnidirectional images [33]. However, we propose a method to extract the VPs taking advantage of both cameras with a two step optimization problem. Depth information is more robust, but less accurate than RGB information. Using fisheye images typically obtain a more accurate VP solution, but the problem may be unable to converge if the initial solution is not good enough. Besides that, a joint optimization is problematic as it needs to weight both terms appropriately. Experiments showed that our two-stage optimization procedure performs well with no extra cost.

The initial solution of the Manhattan direction is set as a trivial three orthogonal vector base ($I = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$). The variables to optimize are the roll-pitch-yaw angles (α, β and γ) that form the rotation matrix $R_{\alpha,\beta,\gamma}$ that after the optimization process should orient the vector base according to the Manhattan directions, $[\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z] = R_{\alpha,\beta,\gamma} \cdot I = R_{\alpha,\beta,\gamma}$. The two stages are:

1. The vector base is rotated until the angle between the normals of as many points from the point cloud as possible and one of the three vectors from the base is minimized. The normals $\mathbf{n}_{\mathbf{X}_i}$ of every point $\mathbf{X}_i$ can be estimated using the method from [34]. To reduce computation time, the cloud can be previously downsampled (e.g. with a voxel grid filter).
2. The vector base is rotated until the angle between the normals of as many lines as possible and one of the three vectors from the base is as close of being orthogonal as possible, using the initial solution provided by stage 1. This is based in that, by definition, the normal $\mathbf{n}_l$ of every line l_i is orthogonal to the direction of the line in the 3D world, and therefore, if a line follows the Manhattan direction $\mathbf{m}_j$, then $\mathbf{n}_{l_i}^\top \cdot \mathbf{m}_j = 0$.

The columns of the final rotation matrix $R_{\alpha,\beta,\gamma}$ are the three Manhattan directions. Our convention is to denote $\mathbf{m}_y$ the column whose vector is closest to the gravity vector given an intuition of how the camera is posed (pointing to the front, slightly downwards). We choose $\mathbf{m}_z$ to be the column pointing to the front and leaving $\mathbf{m}_x$ orthogonal to the previous two. The VPs are the points in the image that result of projecting rays following the Manhattan directions according to the sphere model.

3.4 Classification of Lines and Points

The points from the point cloud $\mathbf{X}$ are classified depending on the orientation of their normals $\mathbf{n}_\mathbf{X}$ in the three orthogonal classes. A 3D clustering is then performed for each class to recover the planes P in the image (Fig. 2b). For those with normal $\mathbf{n}_P = \mathbf{m}_y$, the lowest one is chosen as *floor plane* (P_{floor}). Those lines l_i whose minimum angular distance to their closest Manhattan direction $\mathbf{m}_j$ is below a threshold are classified as lines in that direction L_j, where $j = x, y, z$ (Fig. 2c). The *horizon line* is the line-image l_H corresponding to the normal $\mathbf{n}_{l_H} = \mathbf{m}_y$ (drawn in dotted white line in Fig. 2c). Lines oriented in $\mathbf{m}_x$ and $\mathbf{m}_z$ are classified as *upper lines* ($\overline{L}$) when they are above horizon, and *lower lines* ($\underline{L}$) when they are below. Lines oriented in $\mathbf{m}_y$ (L_y) are classified as *long lines* when they have contour points above and below the horizon.

Some lines correspond to intersections of 3D planes extracted from the depth image. In order to detect such correspondences, we compute the 3D intersection lines of wall planes with the floor plane and between walls. When there are two consecutive wall planes of the same orientation, the line of the border is computed instead. An example is shown in Fig. 2b, where the 3D lines have been drawn in yellow. These 3D intersection lines can be projected to the fisheye image and have its line normal computed. To perform the association, we evaluate the angular distance between their normals, and choose the closest if the angular distance is below a small threshold. Those lines supported by 3D evidence have more relevance when generating layout hypotheses.

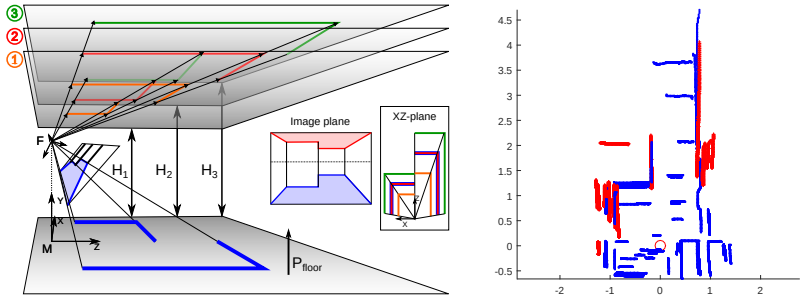
4 Layout Estimation

To extend the depth information to the periphery, we look for features in the fisheye image that allow us to draw coherent layout hypotheses. We choose *corners*, i.e. points of intersection of three alternatively oriented planes in the 3D world, manifested in the image as intersections of line-images. The intersections between lines in a general case with a highly populated scene can be endless, so we consider only those segments close to each other as more likely to actually intersect. To determine the proximity between segments, instead of using pixel distances, we reason in the 3D world with metric distances. Pixel distance is misleading, as it is affected by how far the points are from the camera, the perspective and the heavy distortion of the fisheye camera. Note that it is also difficult to deal with distances in the 3D world when there is no scale information available. With our system we can integrate scale information in the process.

We assume that, from previous steps, we have the 3D location of at least one structural plane from the depth data. Without loss of generality, we use the floor plane, which is the most probable plane to be in every scene. We use that plane to project all the lower lines and place them in a scaled 2D floor plan of the scene, we call *XZ-plane*. Similarly, the upper lines are projected to the *ceiling plane* (P_{ceil}), as explained in Section 4.1. Both floor and ceiling are going to be considered unique and symmetric. Together with the Manhattan World assumption, the 3D layouts we estimate from the images are a sequence of vertical walls alternatively oriented in $\mathbf{m}_x$ and $\mathbf{m}_z$, closed by the floor and ceiling planes. One of the main advantages of using a 2D projection is that the layout hypotheses can be generated using corners from floor or ceiling indistinctly, potentially overcoming hidden corners. In Section 4.2 we describe how the corners are detected and scored. The generation of layout hypotheses is explained in Section 4.3, to finally deal with the evaluation process in Section 4.4.

4.1 Extraction of the Ceiling Plane for Horizontal Line Scaling

We can get the ray emanating from the optical center to every contour point of every lower line in X and Z and intersect it with the P_{floor} in 3D (Fig. 3a). With the floor plane equation and the Manhattan directions, we can transform



(a) (b)

Fig. 3: (a) Projection of the lower line segments in the image to the P_{floor} and the upper segments to three virtual ceiling planes at different H_{ceil}. The chosen H_{ceil} is the one with most overlapping in the XZ-plane (H_2 in the example). (b) Real example of contour projection of lower lines (blue) and upper lines (red) to the XZ-plane. The small circle represents the position of the camera system.

these points from the camera reference frame F to a new reference axis M, with the Manhattan directions and origin at the floor level. If we plot the transformed points in the axis XZ, we can get a 2D scaled floor plan of the contours with scale (the *XZ-plane* in Fig. 3a).

As for the upper lines, the rays traced from the optical center to the contour points must be intersected with the P_{ceil} instead. As we assume structural symmetry, we know that the normal of the ceiling plane will be the same as the floor normal, but the distance to the origin is still unknown. To estimate H_{ceil} we use the fact that, in the XZ-plane view, wall-floor ($\underline{l_j}$) and wall-ceiling ($\overline{l_i}$) intersections must be coincident. We can generate a P_{ceil} at an arbitrary height, compute the 3D intersections of the projection rays and evaluate how well the contours from upper segments $\overline{C}_i$ coincide with contours from lower segments $\underline{C}_j$ in the XZ-plane. In Fig. 3a there is a visual example with three different H_{ceil}. H_1 is too small and H_3 too big, so the segments of the floor do not match the segments of the ceiling in the XZ-plane. H_2 is the best one as contours from both planes match perfectly. Mathematically, we propose the following optimization problem:

$$\arg\max_{H_{ceil}} \sum_{i=1}^{N_{\overline{L}}} card(\overline{C}_i) \cdot \delta \tag{2}$$

where the function $card(C_i)$ means the number of contour points of the line l_i and δ is a binary variable of value 1 when $\overline{C}_i \cap \underline{C}_j$ in the XZ-plane for at least one $\underline{l}_j$ in the set. One of the advantages of working with scaled distances is that we can set reasonable valid ranges of heights to constraint the values of H_{ceil} (e.g. 2-3 meters for indoor environments). If the problem has no solution between the valid range it could be due to clutter, undetected lines or absence of ceiling in the image. Then the algorithm goes on without considering ceiling lines in the layout retrieval. In Fig. 3b the XZ-plane with the contours of both lower and upper lines from the case from Fig. 2c is shown.

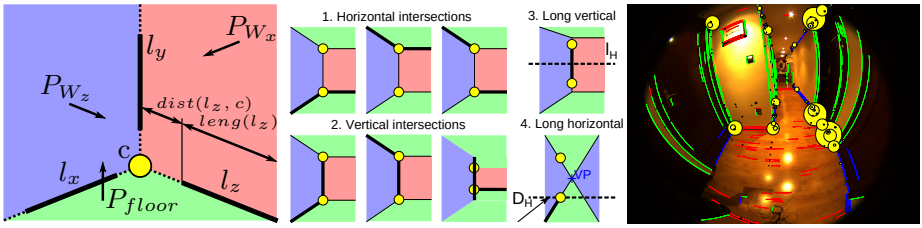


Fig. 4: (a) Graphical definition of a corner c, its line segments (l_x, l_y, l_z) and the *dist* and *leng* functions used in our scoring method. (b) Different types of corners we consider. (c) Example of a subset of the 100 most scored corners plotted as a yellow circles of diameter proportional to their score projected to the image in floor and ceiling.

4.2 Detection and Scoring of Corners

Line segments are the main piece of information we use to create layout hypotheses. However, we do not know whether they come from actual wall-ceiling or wall-wall intersections, or from other elements of the scene. In the literature there are many approaches to tackle this problem. For instance, [5] defines a corner when a minimal set of three or four line segments in certain orientations are detected. This requires having clear environments where most line segments can be perfectly detected. However, in the real world, occlusions or bad lighting conditions may cause some contours to remain undetected. In a Manhattan World, two line segments are enough to define a corner. Other works such as [21,22] tend to give more emphasis to vertical lines and the extension of their segments in their corner definition, which may be problematic for the same reason as before.

We propose to use more relaxed requirements to define corners, using just one or two line segments, and then use a scoring function to select the most salient ones and favor their appearance in the layout hypotheses generation. A corner c is then defined by a set of N_l line segments, and its score depends on the number of lines and their score value, which is defined by:

$$S_{l_i} = (leng(l_i) - dist(l_i, c)) \cdot \lambda \tag{3}$$

where *leng* measures the length of the line segment in meters, *dist* measures the distance of the closest point of the segment to the actual intersection point c in meters and λ is a multiplier of value 2 when the line is associated to a 3D line obtained from the depth camera and 1 when it is not (Fig. 4a). The score of a corner is the sum of line scores multiplied by the N_l to increase the score of corners supported by more lines:

$$S_{c_j} = N_{l_j} \cdot \sum_{i=1}^{N_{l_j}} S_{l_i} \tag{4}$$

We consider four different cases of corners to be retrieved depending on the classification of the lines involved according to our criterion (Fig. 4b):

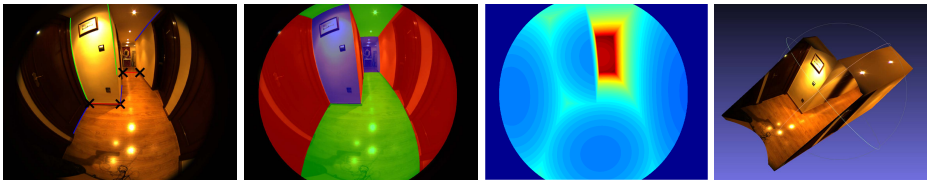
1. **Horizontal intersections ($l_x - l_z$)**: If there is a l_y passing through the intersection point c, the contour points of l_y are scaled by assuming they share the same wall as l_x (i.e. 3D plane P_{W_x}) or l_z (P_{W_z}). If the scaled 3D contours of l_y have heights between 0 and H_{ceil} the vertical is included to improve the score of the corner.
2. **Vertical intersections ($l_x - l_y$ or $l_y - l_z$)**: As with the previous case, the segment l_y is scaled to verify plausability. The case of occluding walls is also considered in this type of intersection.
3. **Long vertical lines (l_y)**: Only the ones crossing the horizon are considered as they are more likely to be wall to wall intersections instead of clutter. The projection of their topmost contour point to the P_{ceil} and the bottommost one to the P_{floor} are considered as two separate corners.
4. **Long horizontal lines (l_x or l_z)**: Sometimes there is no visible or detected corner at the farther end of a corridor or a big room. We consider the possibility of long horizontal lines to intersect with the horizon. To keep layouts of reasonable size we restrict the distance to a maximum of D_H (in particular we set $D_H = 10$ m). The line score for this case is modified:

$$S_{l_i} = length(l_i) \cdot \lambda \cdot max(D_H - dist(l_i, c), 0) \tag{5}$$

After the extraction of corners we keep those with $S_c > 0$. To avoid redundant corners we merge those which are horizontally close to each other, summing the score of all corners involved to increase the likelihood of the resulting corner to appear in the hypotheses generation process. For this merging it is necessary to watch that the direction of the contours is compatible to the physical coherence (e.g. a corner supported by line segments directed towards $\mathbf{m}_x$ and $\mathbf{m}_z$ cannot be merged with another corner whose line segments are directed towards $-\mathbf{m}_x$ and $-\mathbf{m}_z$). In Fig. 4c there is an example of the 100 most scored corners after the merging.

4.3 Layout Hypotheses Generation

Our algorithm for layout hypotheses generation starts by picking up a set of corners from the set and sorting them clockwise in the XZ-plane. We proportionally increase the probability of choosing corners rewarding those with higher score. In order, we generate 2D corner distributions where every wall in direction $\mathbf{m}_x$ is followed by a wall in $\mathbf{m}_z$. We look for closed layout hypotheses. As the fisheye can only view a hemisphere, if there is no corners from behind the camera, the layout would be generated so it ends at the rear VPz (or by our design at D_H). We extract the layout only of the room where the camera is, so all hypotheses which are closed leaving out the camera position are discarded. As we have the contour points that define the lines of the corners, the hypothetic walls must not contradict their location in the map, otherwise the hypothesis is discarded as well. One of the keypoints of our method is that, taking advantage of the Manhattan assumption, we can integrate in the layout undetected corners and form a closed solution: when consecutive corners do not generate walls in Manhattan directions, intermediate virtual corners are created in order to generate



(a) (b) (c) (d)

Fig. 5: (a) Sample hypothesis with the selected corners and the corresponding contours drawn. (b) Overlay of the planes generated by the layout colored depending on their orientation. (c) Extended depth map of the scene. (d) Colored point cloud of the layout obtained from the depth map.

two Manhattan-oriented walls. A more detailed description of the generation of layout hypotheses with special cases can be found in the supplementary material.

In Fig. 5 there is an example of a layout hypothesis with the contours and corners drawn in (a) and the resulting wall distribution colored in (b). As the XZ-plane is scaled and the H_{ceil} has been estimated we can generate a 3D depth map of the scene (Fig. 5c). We compare the result with the depth map provided by the depth camera, and use that to discard contradictory hypotheses (i.e. there cannot be walls in front of the given depth map). Finally, the depth map can be used to recover the 3D point cloud of the complete layout (Fig. 5d).

4.4 Evaluation of the Hypotheses

We have described a method to find the relevant corners in the image and to generate layout hypotheses. Next, we propose three methods to select the best hypothesis:

- **Sum of Scores (SS):** We define the score of an hypothesis as the sum of scores of the corners that have been used to generate it. The additional corners defined to generate Manhattan layouts have a score of zero.
- **Sum of Edges (SE):** The polygon defined by the corners of the hypotheses as vertices can be drawn on the XZ-plane in order to choose the hypotheses which overlaps the most with the observed contours.
- **Orientation Map (OM):** It requires to build a reference image called *orientation map* [5], which is an image whose pixels encode the believed orientation given the line segments for perspective cameras. To build that image we create a set of overlapping perspective images from the fisheye image, apply the orientation map algorithm from [5] in each one of them and finally stitch them back together to form an omnidirectional orientation map. The evaluation consists in selecting that layout hypotheses which have better fitness between pixels with the same orientation.

In general, the fastest method is to score the hypotheses, which does not require any extra computation. Using the orientation map usually provides the best results, but it requires the previous computation of the map itself, which can be time consuming.

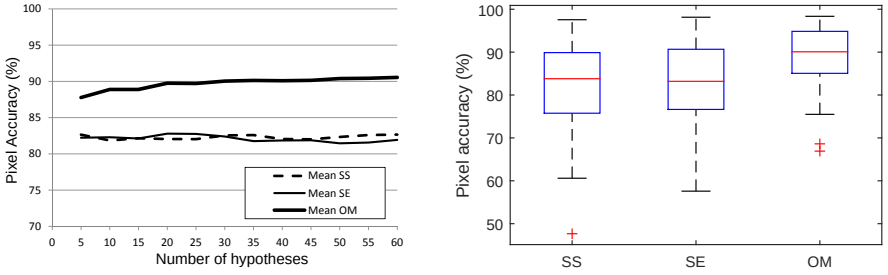


Fig. 6: (a) Pixel accuracy over the number of hypotheses generated. (b) Boxplot of the results using the three evaluation methods with the set of 70 images.

5 Experiments

In this work, we use a novel camera system with fisheye and depth image. Many datasets for indoor layout retrieval are usually based on conventional images, but not so many on omni-images, and none combining them with depth. For the experimental evaluation we have collected a set of 70 captures of indoor scenarios ourselves, including 23 from corridors/entrances, 15 from four different bedrooms, 4 from a bathroom, 12 from two living rooms, 4 from a kitchen and 12 from two small cluttered rooms. In order to perform a quantitative evaluation we have manually labelled these images to provide a per pixel tag between the three main classes (walls in X or Z and floor/ceiling). The measure employed is the percentage of pixels correctly tagged over the totality of pixels from the ground truth, which we call *pixel accuracy* (PA).

To choose the best fixed number of geometrically valid hypotheses to draw, we have tried with different quantities and observed the pixel accuracy for the three evaluation methods (Fig. 6a). Evaluating with Orientation Map (OM) seems to slightly improve the score when the number of hypotheses is increased. However, using Sum of Scores (SS) or Sum of Edges (SE) does not seem to improve the pixel accuracy at all. This is a consequence of a the good detection and scoring of corners in our method, which affects the probability of good corners to be selected when drawing hypotheses. As a result, only a few valid hypotheses are necessary to get one with more than 80% of PA. For our experiments, we choose a number of hypotheses of 20, substantially less than other similar works [23].

In Fig. 6b there is a boxplot showing the distribution of pixel accuracy in the 70 images with the three evaluation methods. We can see how the SS and SE evaluation methods are able to tag correctly about 83% of the pixels in the image (83.8% and 83.2% respectively). Using the orientation map reaches the 90% and has less variance. However, the need for the previous computation of the orientation map makes this approach the least suitable for real-time applications: In our current implementation it takes around 23 s just to generate the map. The rest of the computations, including depth and fisheye preprocessing, corner extraction, generation of hypotheses and evaluation in our current imple-

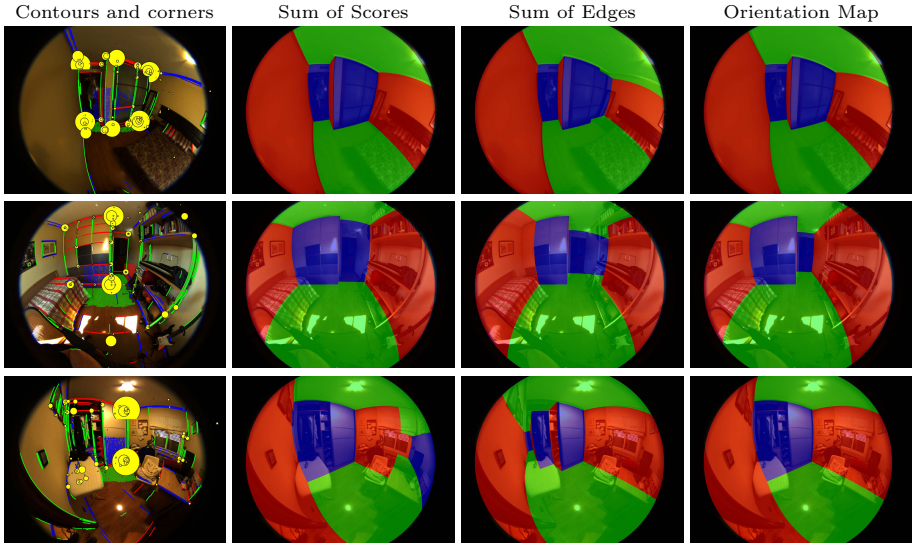


Fig. 7: Three examples of results from images from our set with best layout proposal for each method. More examples in the supplementary material.

mentation take an average sum of 33 s. Some examples of scenes from our set and their best results are shown in Fig. 7.

There is no fair comparison to be made with other works, because no other related work known to the authors use similar type of image combination. However, we explore the benefits of using such camera system compared to merely using a fisheye camera. We repeated the experiments aforementioned removing all depth information throughout the system in order to numerically observe how the results are affected. The first thing to notice is that the scale of the system is now arbitrary. Also, the absence of depth affects the computation of the VPs, the scoring of lines, the retrieval of the H_{ceil} and the elimination of contradictory hypotheses. A comparison of mean results with and without depth information is shown in Table 1, where it can be observed how the mean pixel accuracy decreases about 10%.

A breakdown of the results depending on the type of room is provided in Table 2. In general we have experienced better performance in uncluttered environments (corridors), where structural lines can be easily seen. Compared to conventional perspective images, using omnidirectional cameras are better suited to overcome highly cluttered scenes, as the probability of encounter important lines either from the wall-floor or wall-ceiling intersections is higher. Sometimes, the problem does not come from the lines being occluded, but on the difficulty of them being detected. When the walls and the ceiling have similar color or the lightning conditions are poor, the line extraction algorithm might fail to detect important lines. This is a common problem to all similar methods in the literature. However, the depth information remains invariant to such problems

Table 1: Mean pixel accuracy of the system with and without depth information (%).

Method	With depth	No depth
SS	82.7	71.7
SE	82.5	72.2
OM	89.2	80.2

Table 2: Mean pixel accuracy depending on the type of scene tested (%).

Room	SS	SE	OM
Corridor	86.5	84.3	90.3
Bedroom	78.6	79.6	87.7
Bathroom	75.0	69.6	84.9
Living Room	85.1	87.5	91.2
Kitchen	83.2	76.9	85.3
Other	81.6	84.5	90.3

in indoor environments, so the hybrid camera system we propose for this task always has some safe zone where we know for certain the shape of what is in front of the camera, even when the scene visual conditions are far from perfect. In normal conditions, with our camera configuration we are able to estimate the omnidirectional scaled layouts in one single shot, where the information from the depth camera can be integrated seamlessly and be used to detect objects inside the room or generate complete 3D models as the one shown in Fig. 1e.

Additional experiments detailing failure cases, as well as a summary video, can be found in the supplementary material.

6 Conclusion

In this work we have developed a new method to extend the 3D information of a depth camera to a field of view of over 180 degrees. To do that, we propose a novel spatial layout retrieval algorithm, whose main novelty is combining a fisheye and a depth camera. The large field of view helps to use information from both the ceiling and the floor, which is helpful when there is clutter in the scene. To take advantage of that, we look for relevant corners by using the line segments extracted from the fisheye image and the depth information to assign probability of being actual corners in the scene. A heuristic procedure is then used to generate layout hypotheses from these corners, which are then put through an evaluation procedure for which we propose three methods. Experimental evaluation with real images of indoor environments shows great results in terms of accuracy, improving the state of the art in functionality: our method has less layout shape restrictions, needs less hypotheses and provide 3D models of the scene with scale in a single shot.

Acknowledgments. This work was supported by Projects DPI2014-61792-EXP and DPI2015-65962-R (MINECO/FEDER, UE) and grant BES-2013-065834 (MINECO).

References

1. Endres, F., Sprunk, C., Kummerle, R., Burgard, W.: A catadioptric extension for RGB-D cameras. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2014) 466–471
2. Tomari, R., Kobayashi, Y., Kuno, Y.: Wide field of view Kinect undistortion for social navigation implementation. In: Advances in Visual Computing. Springer (2012) 526–535
3. Fernandez-Moral, E., González-Jiménez, J., Rives, P., Arévalo, V.: Extrinsic calibration of a set of range cameras in 5 seconds without pattern. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2014) 429–435
4. Delage, E., Lee, H., Ng, A.Y.: A dynamic bayesian network model for autonomous 3D reconstruction from a single indoor image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (2006) 2418–2428
5. Lee, D.C., Hebert, M., Kanade, T.: Geometric reasoning for single image structure recovery. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2009) 2136–2143
6. Coughlan, J.M., Yuille, A.L.: Manhattan world: Compass direction from a single image by bayesian inference. In: IEEE International Conference on Computer Vision (ICCV). Volume 2. (1999) 941–947
7. Del Pero, L., Bowdish, J., Fried, D., Kermgard, B., Hartley, E., Barnard, K.: Bayesian geometric modeling of indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2012) 2719–2726
8. Chang, H.C., Huang, S.H., Lai, S.H.: Using line consistency to estimate 3D indoor Manhattan scene layout from a single image. In: IEEE International Conference on Image Processing (ICIP). (2015) 4723–4727
9. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: IEEE International Conference on Computer Vision (ICCV). (2009) 1849–1856
10. Schwing, A.G., Hazan, T., Pollefeys, M., Urtasun, R.: Efficient structured prediction for 3D indoor scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2012) 2815–2822
11. Ramalingam, S., Pillai, J., Jain, A., Taguchi, Y.: Manhattan junction catalogue for spatial reasoning of indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2013) 3065–3072
12. Mallya, A., Lazebnik, S.: Learning informative edge maps for indoor scene layout prediction. In: IEEE International Conference on Computer Vision (ICCV). (2015) 936–944
13. Hedau, V., Hoiem, D., Forsyth, D.: Thinking inside the box: Using appearance models and context based on room geometry. In: European Conference on Computer Vision (ECCV), Springer (2010) 224–237
14. Gupta, A., Hebert, M., Kanade, T., Blei, D.M.: Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In: Advances in Neural Information Processing Systems 23. Curran Associates, Inc. (2010) 1288–1296
15. Del Pero, L., Guan, J., Brau, E., Schlecht, J., Barnard, K.: Sampling bedrooms. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2011) 2009–2016
16. Choi, W., Chao, Y.W., Pantofaru, C., Savarese, S.: Understanding indoor scenes using 3D geometric phrases. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2013) 33–40

17. Schwing, A.G., Fidler, S., Pollefeys, M., Urtasun, R.: Box in the box: Joint 3D layout and object reasoning from single images. In: IEEE International Conference on Computer Vision (ICCV). (2013)
18. Flint, A., Murray, D., Reid, I.: Manhattan scene understanding using monocular, stereo, and 3D features. In: IEEE International Conference on Computer Vision (ICCV). (2011) 2228–2235
19. Furlan, A., Miller, S.D., Sorrenti, D.G., Li, F.F., Savarese, S.: Free your camera: 3D indoor scene understanding from arbitrary camera motion. In: British Machine Vision Conference (BMVC). (2013)
20. Jia, H., Li, S.: Estimating the structure of rooms from a single fisheye image. In: IAPR Asian Conference on Pattern Recognition (ACPR). (2013) 818–822
21. López-Nicolás, G., Omedes, J., Guerrero, J.J.: Spatial layout recovery from a single omnidirectional image and its matching-free sequential propagation. Robotics and Autonomous Systems **62**(9) (2014) 1271–1281
22. Jia, H., Li, S.: Estimating structure of indoor scene from a single full-view image. In: IEEE International Conference on Robotics and Automation (ICRA). (2015) 4851–4858
23. Zhang, Y., Song, S., Tan, P., Xiao, J.: Panocontext: A whole-room 3d context model for panoramic scene understanding. In: European Conference on Computer Vision (ECCV). Springer (2014) 668–686
24. Zhang, Q., Pless, R.: Extrinsic calibration of a camera and laser range finder (improves camera calibration). In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2004) 2301–2306
25. Scaramuzza, D., Harati, A., Siegwart, R.: Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2007) 4164–4169
26. Geiger, A., Moosmann, F., Car, O., Schuster, B.: Automatic camera and range sensor calibration using a single shot. In: IEEE International Conference on Robotics and Automation (ICRA). (2012) 3936–3943
27. Puig, L., Bermudez-Cameo, J., Sturm, P., Guerrero, J.J.: Calibration of omnidirectional cameras in practice: A comparison of methods. Computer Vision and Image Understanding **116**(1) (2012) 120–137
28. Herrera C, D., Kannala, J., Heikkilä, J.: Joint depth and color camera calibration with distortion correction. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(10) (2012) 2058–2064
29. Scaramuzza, D., Martinelli, A., Siegwart, R.: A toolbox for easily calibrating omnidirectional cameras. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2006) 5695–5701
30. Perez-Yus, A., Lopez-Nicolas, G., Guerrero, J.J.: A novel hybrid camera system with depth and fisheye cameras. In: IAPR International Conference on Pattern Recognition (ICPR). (2016)
31. Bermudez-Cameo, J., Lopez-Nicolas, G., Guerrero, J.J.: Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry. International Journal of Computer Vision **114**(1) (2015) 16–37
32. Geyer, C., Daniilidis, K.: A unifying theory for central panoramic systems and practical implications. In: European Conference on Computer Vision (ECCV). Springer (2000) 445–461
33. Bazin, J.C., Kweon, I., Demonceaux, C., Vasseur, P.: A robust top-down approach for rotation estimation and vanishing points extraction by catadioptric vision in urban environment. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2008) 346–353

34. Rusu, R.B., Cousins, S.: 3D is here: Point cloud library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). (2011)