Scene Structure Recovery from a Single Omnidirectional Image

N. D. Özışık, G. López-Nicolás and J. J. Guerrero

Instituto de Investigación en Ingeniería de Aragón - Universidad de Zaragoza, Spain

didemozisik@gmail.com, gonlopez@unizar.es, jguerrer@unizar.es

Abstract

This work tackles the problem of recovering the structure of a scene from a single image. The goal is to interpret automatically the image to obtain the spatial layout of the scene. In essence, the method proposed classifies the environment as floor or walls and their relative positions. Instead of using standard cameras for solving this particular task, our work is novel in using omnidirectional vision, which is advantageous as it captures in a single image the whole surrounding structure. We also consider man-made indoor scenes, where geometric relationships like parallelism and orthogonality are common. Our contribution is a new method for recovering the scene layout by using extracted line segments from a single omnidirectional image. Collection of lines and geometric constraints provide sufficient information to generate a set of possible scene structures. We also create a map of orientations in the image to test these hypotheses and select the one with the best fitting as the resultant structure.

1. Introduction

Visual sensors are powerful tools for perception in many applications. In particular, they provide rich information and high angular accuracy at low cost compared with other sensors. In this work, we tackle the problem of reconstructing the scene layout from visual information. This problem has especial interest in the absence of direct measurements of the scene for the reconstruction. Several images may provide enough information of the environment to allow the reconstruction of the scene layout. A basic work about creating architectural models from conventional images is presented in [15], other methods for 3D reconstruction are [19] and [18], which are based on user-provided coplanarity, perpendicularity and parallelism constraints. However, using less images reduces the computational cost and the prone-error process of matching information across images. Thus, we focus on the problem of scene layout recovery from a single image. A seminar paper in this context is [7], adressing the problem of single view metrology.

Figure 1. Examples of omnidirectional images (top) and standard images (bottom) taken at the same position. These examples illustrate that omnidirectional vision is more powerful for recovering the whole spatial layout of the scene from a single image.

Nowadays, omnidirectional vision is a hot topic research because of the advantages of the higher field of view with respect to standard cameras. In particular an image taken by a standard camera will only captures a small part of the scene whereas omnidirectional vision provides information in a wide field view of 360 degrees in one single image. Two examples obtained from a public database [5] are given in Fig. 1 to illustrate the superiority of omnidirectional vision for structure recovery.

In essence, scene structure recovery classifies the environment as floor or walls and their relative positions. We present a new method for recovering the scene structure by using automatically extracted line segments from a single catadioptric image. Previous works available in this field tend to focus on the use of conventional cameras opposite to omnidirectional ones, but in general, these techniques cannot be directly applied to omnidirectional vision.

We take advantage of the Manhattan World Assumption

[6], that considers that most indoor surfaces are planes following the dominant directions. If the viewer can estimate the three principal directions, then it becomes significantly easier to interpret the scene. In our framework, we assume the existence of geometric relationships such as parallelism and orthogonality are available between the scene surfaces.

An example of scene recognition is given in [16], which is based in a classification of indoor and outdoor scenes that simplifies the scene recognition. Another related work is [10], that considers the presence of clutter, which is a major problem for 3D reconstruction algorithms relying on finding the ground-wall boundary. However, the versatility of the method is reduced by assuming that the global room space can always be modeled with a parametric 3D box. In more complicated situations, [13] presents a scene structure recovery method based on a collection of lines. Several physically valid structure hypotheses are proposed by geometric reasoning and verified to find the best fitting model to line segments. The advantage of this work in comparison with [10] is the ability to recognize several corner types. Thus, [13] is compatible with many indoor scenes. Our proposal is inspired in the work of [13] but using catadioptric images instead of conventional cameras and a different procedure for the scene recognition.

Basically, our goal is to interpret a collection of line segments from the omnidirectional image to recognize the buildings structures. The input of our algorithm is a single indoor image and we assume that the camera calibration parameters are known. The outputs of our work are the orientations of each part in the scene from three dominant directions (two horizontal and one vertical). Thus, we label the parts such as floor or wall. The main disadvantage of omnidirectional vision is that it contains significant deformations due to the geometry of projection. For example, 3D lines in conventional images degenerate into curves in catadioptric images, and then the projection of a 3D line is a conic in the omnidirectional image [1]. For this reason many techniques in image processing become unsuitable if they are applied directly to catadioptric images.

Because of the fact that we do not have any measurement of the scene, we exploit prior knowledge of the world. Indoor scenes in human environments have some significant characteristics (e.g. perpendicularity between wall and floor, parallelism between floor and ceiling, types of corners between connected walls etc.). These characteristics are significant and we can group the lines due to dominant directions and imagine a three dimensional scene. Thus, our scene structure recovery approach is based on geometric reasoning and it is independent from color, texture and light information.

Lines in the scene and geometric constraints of catadioptric images may provide sufficient information to create possible scene structures. In order to choose the best hypothesis, we create a map of orientations consisting of the orientations of the different planar regions in which the image is divided. From the possible structures generated, the best fitting one with respect the computed map of orientations is chosen as the resultant structure.

The paper is organized as follows. In section 2 we briefly explain line extraction and vanishing points computation. The hypotheses generation from corners is described in section 3. The following section explains the procedure to generate the orientation map of the image. Section 5 is devoted to the hypotheses evaluation to select the best one. In section 6, the approach proposed is tested with real images.

2. Line Extraction and Vanishing Points

The projections of parallel lines intersect on a point, which is a vanishing point. The first stage of the procedure developed here requires the extraction of lines and vanishing points from the input image. In this section, we briefly describe the method selected for this purpose.

In the context of omnidirectional vision, some works about line extraction are [3] [4] [17] [21]. In the proposal presented in [3], the catadioptric lines are represented by their normal vectors in the sphere space and the proposed algorithm detects the edge chains in the image, projects the chains to the sphere and check if at least 95% of the chain points belong to the great circle, providing as a output the set of vanishing points. A difference between [4] and [3] is the sequence of stages of the procedure for vanishing points estimation and line grouping. In [4], they first compute the vanishing points in the scene and then group the extracted lines according to their vanishing points; whereas in [3], they use the reverse order: first they group the extracted lines according to their normal vectors, and then compute their vanishing points. Besides that, in [4] conics in the images are extracted instead of projection planes in the sphere model as in [3]. In our work, we benefit from the Bazin et al's line extraction algorithm [2],[3]. In particular, we consider central catadioptrics systems. Nevertheless, the extraction of lines for structure computation in non-single viewpoint systems has already been addressed in [20], [9].

3. Hypotheses Generation

In this section, we describe the procedure to generate a set of hypotheses from the extracted lines and vanishing points of the input image. A hypothesis h_i consists of a set of corners and conics defining the structure of the scene. Each hypothesis also classifies the different regions in walls or floor and assigns their orientation. Important elements for generating a hypothesis are the corners and boundaries between walls and floors. Another approach is given in [13], which prefers the ceiling-wall boundary instead of floor-wall boundary in order to avoid the problem
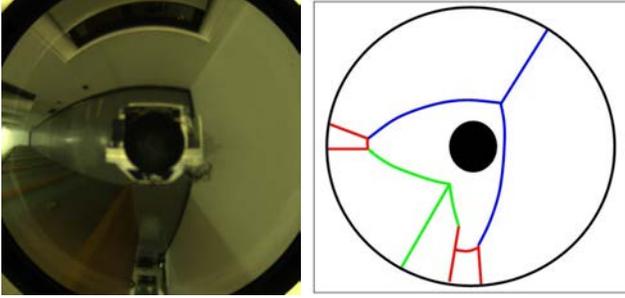
Figure 2. The left image contains three different types of corners. These are depicted in the right image: blue lines define concave corner type, green lines define convex corner type and red lines define occluding corner type.

Figure 3. Example of the extracted corners of the scene using the procedure described in section 3.1.

of occluding objects in the scene. But in general, this is unavailable in catadioptric images, given that we are not able to see the whole ceiling-wall boundary because of the mirror shape. The corners and the floor-wall boundaries of an indoor scene may provide us sufficient information to recognize the layout of the scene. Therefore, the basic step of the scene structure recovery is to detect corners and compute the floor-wall boundaries. To do that, firstly, we define the intersection points of vertical and horizontal lines as corners, and then, we create some hypotheses by pairing corners randomly in order to detect real corners of the room.

The number of actual scene corners is unknown a priori. Thus, the algorithm will test hypotheses considering four, five, six... corners. For the generation of one hypothesis, the algorithm select four (or five, six...) corners from the extracted corner's list of the scene, and create a hypothesis from these corners. Given the set of corner of an image,

$$c = \{c_1, c_2, \cdots, c_n\} \qquad (1)$$

where n is the number of detected corners. The set of different hypothesis are denoted as

$$h = \{h_1, h_2, \cdots, h_k\} \qquad (2)$$

where k is the number of the possible hypotheses and it is given by the combination's equation:

$$k = \left(\begin{array}{c} n \\ r \end{array} \right) = \frac{n!}{r!(n-r)!} \qquad (3)$$

where r is the number of corners considered for each hypothesis (i.e. $r = 4, 5, 6...$).

3.1. Find Corners

A general classification for types of vertices and edges in the scene is explained in [11] and [13]. In essence, there are three types of corners for scene structure: convex, concave and occluding corners. A convex or a concave corner is formed when two walls meet at one place in 3D space

and an occluding corner is formed when one wall is in front of another wall but appears to be adjacent in the image. A convex or a concave corner is obtained by the intersection of one vertical and two horizontal lines. An occluding corner is obtained by the intersection of one vertical and one horizontal line (Fig. 2). In the 3D world, the intersection point of three orthogonal lines, i.e. in different directions, is defined as a corner of the room. But sometimes we are not able to see all these three lines due to the viewpoint or occluding objects. For this reason, we extend the corner searching criterion in order to find more corners in the scene, and then, the intersection point of a vertical and a horizontal line is also considered as a corner. This intersection of lines process is carried out in the projective space.

We assume that the consideration of corners from the intersection of at least one horizontal and one vertical line allows us to detect and recognize all corners in the scene. However, we detect not only the real corners of the room, but we also detect spurious corners which are not the corners of the room: corner of a window, corner of a table frame on the wall, etc. Therefore, we need to define additional constraints to filter these spurious corners. The first condition for the filtering step is that one vertical line in the world plane can contain only one corner. In order to define the elimination criterion, we take advantage of the manmade environment assumption. In indoor scenes, when the optical axis of the camera is approximately perpendicular to the ground plane, the vertical vanishing point is close to the principal point which is also close to the image center. So, the highest point on a vertical line in the 3D world has the longest distance from the principal point of the image, and the lowest point, which will be on the ground plane, has the shortest distance. If we assume that the wall in the scene contains several horizontal lines such as a fence, the real corner will be the closest one from the intersection points. Therefore we compute the distances between the intersection points and the principal point of the image. Then, we eliminate the intersection points with the longer distances.

The second criterion of the filter eliminates the corners which are too close to each others in the horizontal direc-

tion. In this criterion, we aim to avoid inaccurate vertical lines of the corners. For example, the vertical borders of a door intersect with the floor-wall boundary and cause erroneous corners. In order to filter these corners, we use a threshold value which is the minimum angle between two possible real corners, and we use the orientation map as a guide to select the corner which will be eliminated. The computation procedure of the orientation map will be detailed in section 4.

In the 3D world, the corners are between adjacent walls with two different orientations. So, the neighbors of a real corner must have different orientations. We check the neighbor regions of the corner in the orientation map, and we eliminate the corner whose neighbor regions have the same orientation with each other. The number of detected corners is very important in our algorithm. The reason is that each corner in the scene create several hypotheses by the possible combinations (3). Therefore, each eliminated erroneous corner reduce the execution time of the algorithm. Figure 3 shows an example of the extracted corners in the scene after applying the previous constraints.

3.2. Floor-Wall Boundaries

In order to create hypotheses of the floor and wall regions, we need to recognize the floor-wall boundaries. Floor-wall boundaries is a useful visual cue widely used in the literature, for instance, in the segmentation of indoor corridor floors [14]. The start and end points of each boundary are its two corresponding corners of the scene. We can compute this boundary in standard images by drawing a straight line between these corners. But, as previously mentioned, the lines in the 3D world are seen as curves in the omnidirectional images. So, the goal of this step is the computation of the conics that joins the corners of each boundary. First, the corner points on the image are projected to the unitary sphere, and then, we compute the normal plane by using the rays on the sphere. Finally, we compute the great circle with the known normal vector.

Without loss of generality, for generating the floor hypotheses, we assume that the omnidirectional camera system axis is approximately vertical with respect to the floor. Notice that this is not a hard constraint, as it is only required to detect automatically the floor orientation. Most mobile robots consist of wheel-based systems. For this reason, we can clearly assume that the central region of the catadioptric images indicates the floor. Thus, we can consider the area from the image center through the floor-wall boundary as a floor hypothesis. In particular, we consider the floor hypotheses like a cake slice. Note that, once the vanishing points have been estimated, this assumption allows to determine the vanishing point of the vertical direction. Therefore, the assumption of verticality for the omnidirectional system is not a handicap for the algorithm presented.
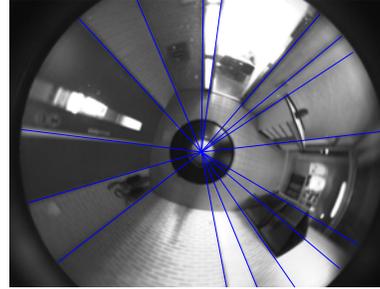
Figure 4. Example of the radii generated on the omnidirectional image in the first step of the orientation map generation.

On the other hand, for generating the wall hypotheses, we assume that walls in the scene contain two corners and each wall hypothesis is independent from the others. Two corners on the same line direction create a wall hypothesis by a floor-wall boundary. The vertical line segments of the corners are vertical boundaries of the wall hypothesis. We denote two corners of the wall as $c_1=(l_{x_i}, l_{y_j}, l_{z_k})$ and $c_2=(l_{x_i}, l_{y_m}, l_{z_n})$, where l_x, l_y, l_z are line segments on directions x, y and z, respectively. The parameter l_{x_i} indicates the i'th line on direction x. Line segments on direction x of c_1 and c_2 are the same, denoting that this wall is aligned with the x direction. In order to define a wall, we need to recognize the borders. Corners include sufficient information to create two vertical putative borders of the wall. The third border of the wall is the floor-wall boundary, which was created with the procedure described above.

At this point, a set of floor and wall hypotheses has been generated. However, the orientations of the hypotheses are still undefined. These orientations will be assigned in the evaluate hypotheses section by similarity comparison with the orientation map.

4. Orientation Map Generation

The goal of generating the orientation map is to assign the orientations of each region. For this purpose, we intend to divide the projected scene on the image into small regions, as many and small as possible. In the evaluating hypotheses step, these regions vote the different hypothesis. So, smallness of regions provides more robust results. The procedure for the orientation map generation consists of four different steps: i) Creating radii. ii) Searching interception line segments. iii) Generating regions. iv) Orientation map definition. Each one of these steps is described in the following.

Due to the fact that omnidirectional images have circular shapes, we start to create regions from the vertical vanishing point (approximately the image center in our case) like a cake slice. A cake slice is defined as the angular section between two radii. We denote as radii the vertical lines of the walls projected onto the image. If the camera system
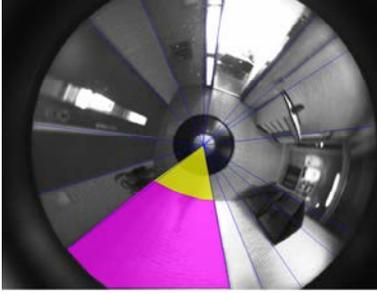
Figure 5. Example of two regions generated in an angular sector. The yellow polygon is the first region and the magenta polygon is the second region.

Figure 6. The orientation map. Red color represents orientation z (i.e. the floor), green color represents orientation y and blue color represents orientation x.

axis is vertical, these are radial lines, i.e. radii. Given that we require only approximated vertical alignment, vertical lines may be projected into conics, but anyway, we will still refer to them as radii. Human eye can distinguish regions in a scene trivially. However, this is a very complicated problem when we consider the automatization of this task. So, we need to consider the procedure to distinguish them via image processing algorithms. Next, we aim to obtain smaller regions by grouping each significant line. We start by defining a set of radii as radial lines that start from the center of the image and continue through each vertical line in the scene and finish on the limit circle of the catadioptric image (Fig. 4).

Although we have divided the scene such as cake slices, we need smaller regions than cake slices for providing more rich information. Then, searching *intercept line segments* is the second iteration for the image labeling. The line segments, which are extracted in the procedure described in section 2, are used to identify the orientation of the cake slices. Each consecutive two radii create a slice, and we call the line segments inside this slice as *intercept line segments*.

Let us consider that s_i is a slice, and s_i is created by two radii r_s and r_e. The line segments found in this step inside s_i are denoted as

$$s_i = \{l_1, l_2, l_3, \cdots\} . \tag{4}$$

Once the set of slices has been obtained (4), the next step is the generation of the regions by means of these slices. Regarding to the definition of regions in indoor scenes, we have generally that the first level is the floor plane and the second level is the wall plane. Therefore, in this method we consider that there are two levels in a slice (Fig. 5).

Apart from walls and floor, the third kind of plane, the ceiling, cannot be seen in general because of the point of view and the mirror shape. Thus, the first *intercept line segment* in a slice creates the first region for this slice. We can define the first region as follows

$$g_1 = (s_i, l_1) . \tag{5}$$
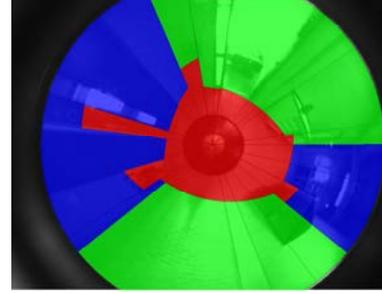
where s_i is the corresponding angular sector and l_1 is the first *intercept line segment*. The second region of this slice consists of the other line segments inside the area of the slice, which are denoted as

$$g_2 = (s_i, l_2, l_3, \cdots) . \tag{6}$$

In the last step for the orientation map generation, the orientations of each region of the omnidirectional image is defined. In general, the more we have small regions, the more we obtain better results. In the *Hypotheses Evaluation* step, described in the next section, the orientation map will be used as a guide. So, a fatal region in the orientation map may affect the result. All line segments inside the region affect the computation of the orientation of that region. Thus, each line segment contributes to the count of the possible orientations of the region. In particular, the orientation with the higher line segment count is chosen as the resultant orientation of the region. Then, we can define the ratios r_x, r_y, and r_z of the x, y, and z directions, respectively, as follows

$$r_x = \frac{L_x}{L} , \; r_y = \frac{L_y}{L} , \; r_z = \frac{L_z}{L} , \tag{7}$$

where L_x, L_y, and L_z are the number of lines inside the region on the x, y, and z directions, and L is the number of all the lines inside the region.

Figure 6 presents an example of a computed orientation map. The three different directions are drawn with different colors. It can be seen that not all the regions have their orientations correctly estimated. For example, some parts of the walls are detected with orientation in direction z, i.e. a floor region. This is usual as we cannot assume perfect data for the input set of extracted lines, and its quality affects directly to the orientation map. Given that these orientations are not coherent with the real scene, the application of the geometric reasoning via the hypotheses will produce a coherent result with the geometric constraints of the real scene. This evaluation follows in the next section.
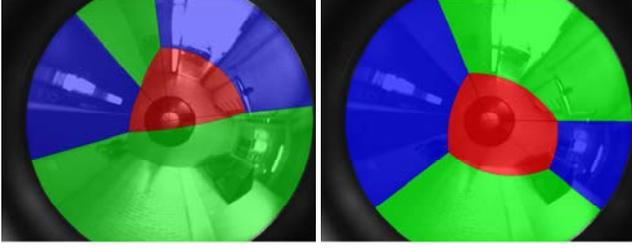
Figure 7. Examples of two different hypotheses for an image. The left one is a bad hypothesis, whereas the right one is a good hypothesis obtained as the best fitting result with respect the orientation map in Fig. 6.

5. Hypotheses Evaluation

In order to identify the correctness of the floor and wall hypotheses, we use the orientation map. The image of the hypotheses and the image of the orientation map are created. We show an example of an orientation map image in Fig. 6, and two examples of hypothesis images are given in Fig. 7. The common pixels in both images give us the similarity rate,

$$Similarity\ rate = \frac{Number\ of\ common\ pixels}{Number\ of\ total\ pixels}\ . \quad (8)$$

The most similar hypothesis to the orientation map is chosen as the best hypothesis and assigned as the resultant structure of the scene. For example, the hypothesis with the biggest similarity rate is chosen as the resultant structure in Fig. 7 right, while an example of a bad hypothesis, with lower similarity rate, is shown in Fig. 7 left.

The difference between the resultant images of the orientation map generation part and the hypotheses evaluation part is that, in orientation maps, colored regions are created by each vertical line in the scene, and their orientations are computed by horizontal lines inside the region. Thus, they give local information for the orientation regions. On the other hand, in floor-wall hypothesis evaluation step, regions are given by the area between two sequential corners, and their orientations are computed from the orientations of regions in the orientation map. Therefore, the region orientations are globally computed from local information. In this way, local errors in the orientation map can be avoided in the hypotheses evaluation step.

6. Experiments

In this section, we present the experimental evaluation of the proposed algorithm. Experiments with real images taken with different cameras in different environments are presented to evaluate the performance of the approach. In our experiments, the camera calibration parameters are known and we use the efficient algorithm proposed in [3] to extract lines from single catadioptric images.

We have taken part of the images tested from the omnidirectional image database from the University of Amsterdam [22], which is available via web. These omnidirectional images were taken by a camera with a hyperbolic mirror on a mobile robot driving through an indoor environment. The image size is 1024×768 pixels. The calibration of the omnidirectional camera and data from laser range scanner, sonar range scanners and odometry sensors are also available [22]. We also tested images publicly available taken from [5]. They provide sensor data sets for the testing of systems on real-world environmental data. In particular, the sensor used is an omnidirectional camera, consisting of a Prosilica GC1020C color GigE Vision camera fitted with an off-the-shelf hyperbolic mirror by Vstone [5]. In this case, the image size is 640×640 pixels.

One of the most important difficulties of omnidirectional image processing algorithms is the deformation on the image, and it is common to come across broken edges because of the illumination, blurry scenes or a shadow in the scene, etc. Therefore, we extended the algorithm to recognize the structure even with broken edges as previously explained. So, our work is able to recognize the structure with partially extracted lines. However, the accuracy rate of the resultant structure still depends on the accuracy of the directions of the lines. This is because our work is based on the corners in the scene, and the corners are created by using these extracted lines and their directions.

The execution time of the algorithm depends on the scene and the number of extracted corners. If the parameters in the corner extraction step are adjusted with a wide range, the accuracy of the resultant structure increases. But, in this manner, the execution time also increases. So, in order to obtain faster results, the parameters of the corner extraction step must be adjusted carefully. However, by using potent computers, there is no need to this parameter adjustment.

Figure 8 shows experimental results of our algorithm. The images in the first and second rows are taken from the omnidirectional database of University of Amsterdam [22]. The images in the third, fourth and fifth rows are taken from the database in [5]. Each column in the in Fig. 8, illustrates a step of the algorithm. In particular, the first column shows the result of the line extraction. The second column illustrates the orientation map generation. The third column shows the corners eventually extracted and the last column, shows the final result, obtained as the best fitting hypothesis given by the hypotheses evaluation step. The first example corresponds to an image of a room with four walls, with a big window. The second example corresponds to a long corridor with convex and concave corner types. The third row image consists of occluding corner types. The image in the fourth row includes convex and concave corner types. The last row shows an example in which the algorithm fails. When the corners of the room cannot be seen because of the
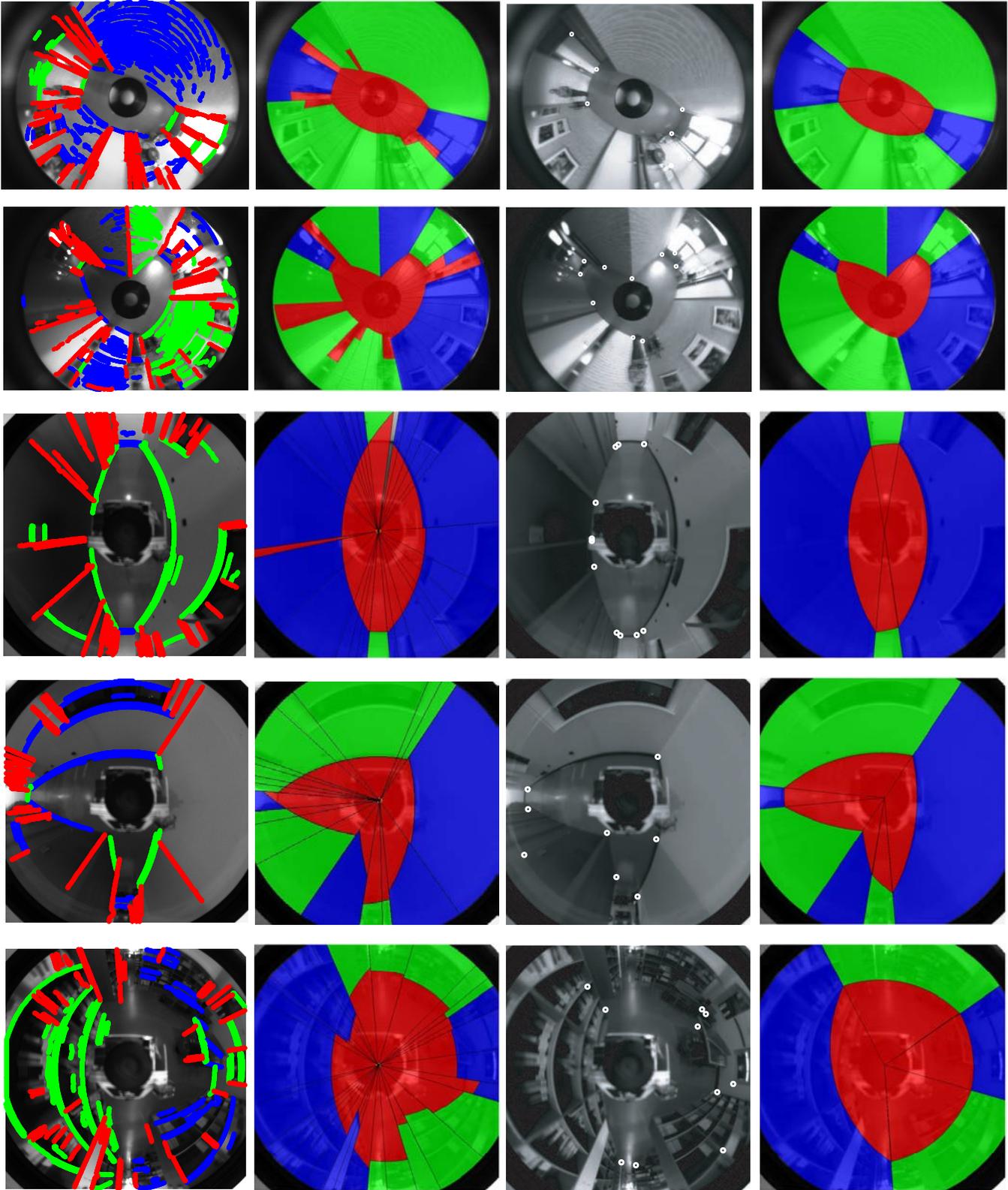
Figure 8. Different examples with images taken from public databases [22] [5]. Each row represents one example illustrating the different steps of the procedure. The first column shows the extracted lines grouped by colors according to their vanishing points. The second column depicts the orientation maps. The third column shows the set of putative corners, and the last column shows the best hypothesis given as a result of the algorithm. The four first rows are successful examples, whereas the last row is an example of failure.

image deformation or an occluding object, the structure of the scene cannot be detected correctly. For example, some parts of the bookshelf in the left side are recognized with different orientations from the real one, and the table in the right side is detected as floor plane.

7. Conclusions

In this work, we present a method to make sense of a set of line collections in the scene. In essence, we define some hypotheses about the scene by using prior knowledge of man-made environments. These hypotheses are compared with a map of orientations, which is created according to the 3D directions of the line segments in the image. Then, the best fitting hypothesis is chosen, and we recover the scene structure over the image provided.

A main advantage of this work is that using single images allows us to avoid the costly and prone-error process of matching. Our proposal is simple to implement and versatile and, as shown in the experiments, our method provides high accuracy rates for images even with broken edges, which is another advantage of our method. Nevertheless, future work is devoted to the evaluation of the possible advantages of using multiple views. Moreover, if multiple images are available, visual SLAM [12] or reconstruction methodologies [8] could be used to improve the performance of the proposal. A possible application of the work is the detection of obstacles in the guide task of impaired people. Another research line for future work is that general structure of a building with different rooms may be recovered by adding structures consecutively.

Acknowledgment

References

[1] J. P. Barreto and H. Araujo. Geometric properties of central catadioptric line images and their application in calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1327–1333, 2005.

[2] J. C. Bazin, Y. Jeong, P. Y. Laffont, I. S. Kweon, C. Demonceaux, and P. Vasseur. An original approach for automatic plane extraction by omnidirectional vision. In *IEEE/RSJ Int. Conf. on Int. Robots and Systems*, pages 752–758, 2010.

[3] J. C. Bazin, I. Kweon, C. Demonceaux, and P. Vasseur. 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*, pages 346–353, 2008.

[4] J. Bermudez, L. Puig, and J. J. Guerrero. Line extraction in central hyper-catadioptric systems. In *OMNIVIS*, 2010.

[5] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, and P. Taddei. Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, 27(4):353–371, 2009.

[6] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Int. Conf. on Computer Vision*, pages 941–947, 1999.

[7] A. Criminisi, I. D. Reid, and A. Zisserman. Single view metrology. In *International Conference on Computer Vision*, pages 434–441, 1999.

[8] S. Fleck, F. Busch, P. Biber, W. Strasser, and H. Andreasson. Omnidirectional 3d modeling on a mobile robot using graph cuts. In *IEEE International Conference on Robotics and Automation*, pages 1748 – 1754, 2005.

[9] S. Gasparini and V. Caglioti. Line localization from single catadioptric images. *International Journal of Computer Vision*, 94:361–374, 2011.

[10] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *IEEE International Conference on Computer Vision*, pages 1849–1856, 2009.

[11] D. A. Huffman. Impossible objects as nonsense sentences. In *Machine Intelligence*, volume 6, pages 295–323, 1971.

[12] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. Munich. The vSLAM algorithm for robust localization and mapping. In *IEEE International Conference on Robotics and Automation*, pages 24–29, 2005.

[13] D. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2009.

[14] Y. Li and S. Birchfield. Image-based segmentation of indoor corridor floors for a mobile robot. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 837 –843, 2010.

[15] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Annual Conference of the European Association for Computer Graphics (Eurographics)*, volume 18, pages 39–50, 1999.

[16] V. Nedovic, A. W. M. Smeulders, A. Redert, and J. M. Geusebroek. Depth information by stage classification. In *IEEE Int. Conf. on Computer Vision*, pages 1–8, 2007.

[17] L. Puig, J. Bermúdez, and J. J. Guerrero. Self-orientation of a hand-held catadioptric system in man-made environments. In *IEEE Int. Conf. on Robotics and Automation*, pages 2549–2555, 2010.

[18] P. Sturm. A method for 3D reconstruction of piecewise planar objects from single panoramic images. In *IEEE Workshop on Omnidirectional Vision, OMNIVIS*, pages 119–126, 2000.

[19] P. Sturm and S. Maybank. A method for interactive 3D reconstruction of piecewise planar objects from single images. In *British Machine Vision Conf.*, pages 265–274, 1999.

[20] R. Swaminathan, A. Wu, and D. H. Depth from distortions. In *ECCV-OMNIVIS*, 2008.

[21] X. Ying and Z. Hu. Catadioptric line features detection using Hough transform. In *International Conference on Pattern Recognition*, volume 4, pages 839–842, 2004.

[22] Z. Zivkovic, O. Booij, and B. Krose. From images to rooms. *Robotics and Autonomous Systems*, 55(5):411–418, 2007.