Layouts from Panoramic Images with Geometry and Deep Learning

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

***Abstract*— In this paper, we propose a novel procedure for 3D layout recovery of indoor scenes from single 360 degrees panoramic images. With such images, all scene is seen at once, allowing to recover closed geometries. Our method combines strategically the accuracy provided by geometric reasoning (lines and vanishing points) with the higher level of data abstraction and pattern recognition achieved by deep learning techniques (edge and normal maps). Thus, we extract structural corners from which we generate layout hypotheses of the room assuming Manhattan world. The best layout model is selected, achieving good performance on both simple rooms (box-type) and complex shaped rooms (with more than four walls). Experiments of the proposed approach are conducted within two public datasets, SUN360 and Stanford (2D-3D-S) demonstrating the advantages of estimating layouts by combining geometry and deep learning and the effectiveness of our proposal with respect to the state of the art.**

I. INTRODUCTION

Layout recovery of indoor scenes is an essential step for a wide variety of computer vision tasks and has recently received great attention from several applications like virtual and augmented reality, scene reconstruction or indoor navigation and SLAM [15]. Typical constrains are the limited field of view which conducts to obtain open geometries and simple box assumptions considering rooms to have just four walls. The challenge here is to recover closed geometries without strong shape assumptions (Fig.1).

One of the first approaches dealing with indoor layout reconstructions was [4] which finds floor-wall boundaries by using a Bayesian network model. In contrast, Lee *et al.* [13] use line segments to generate layout hypotheses evaluating with an Orientation Map, that usually gives problems with clutter since no reasoning about the lines is made. Other works [7], [8], [18] try to simplify the problem by assuming that the room is a 3D box, which does not match reality in many cases. These proposals rely on Geometric Context, which improves clutter detection compared with Orientation Map but provides worse results at the higher parts of the scenes. More recently, [19] introduces the concept of integral geometry and pairwise potentials decomposition which results in an efficient structured prediction framework.

A crucial limitation of these works is the fact that they use conventional images with limited field of view (FOV). On the one hand, this prevents the reconstruction of the real closed geometry of the whole room. On the other hand, the ceiling does not usually appear, being nevertheless a useful part to detect the main structure of the room as it usually has much less occluding objects than the others. Therefore, a more recent research direction looks to extend the FOV. Lopez-Nicolas *et al.* in [14] perform the layout recovery using a catadioptric system. In [17], layout hypotheses are made combining fisheye images with depth information that provides scale. But the real impact comes with the omnidirectional 360° images, which nowadays can be easily obtained with camera arrays, special lenses or automatic image stitching algorithms. This type of images allows to acquire the whole scene at once and hence, it is possible to exploit their wide FOV to generate closed room solutions based on the best consensus distributed around the scene. In [10], their method shows the advantages of having a complete scene view over partial views of the same scene [13]. *PanoContext* [24] uses panoramas to recover both the layout, which is also assumed as a simple 3D box, and bounding boxes of the main objects inside the room. Similarly, [21] provides results not limited to simple box shaped rooms but with the limitation of relying on the output of an object detector. In [22] they treat the problem as a graph with lines and superpixels as nodes, solving it with complex geometric constraints instead.

On the other hand, in the last years, the research community started to face layout recovery problems with convolutional neuronal networks (CNN) achieving an outstanding success and providing an unprecedented level of data abstraction and pattern recognition that is inspired by neuronal

Fig. 1: Starting from a single spherical panorama, we exploit the combination of geometry (accurate lines) and deep learning (edge map) to recover the main structure of the room, achieving 3D complex layouts.

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

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

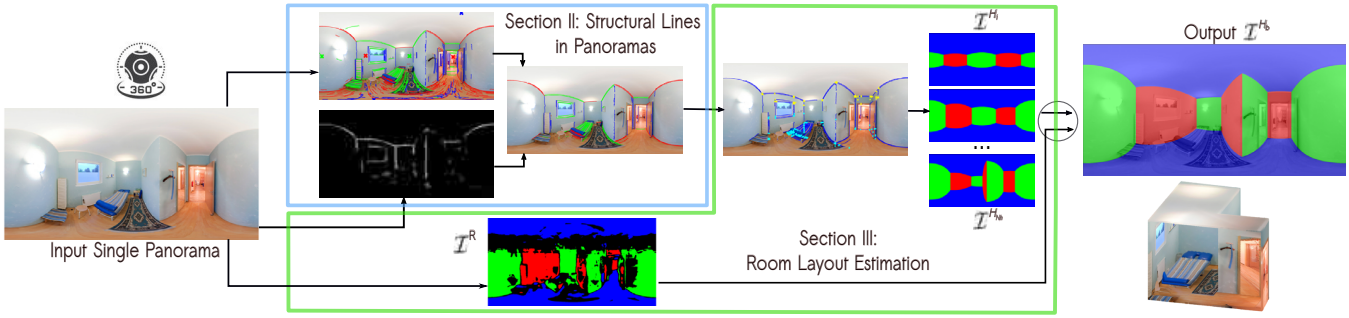


Fig. 2: **Overview**: From a single panorama, the proposed method combines geometric reasoning (lines and vanishing points) and deep learning (edge map [16]) to generate a pruned set of lines belonging to the main structure of the room from which we extract candidate corners. Layout hypotheses are generated from them and those ones satisfying Manhattan world are evaluated, remaining as the final model the one which fits better with a reference map $\mathcal{I}^{\mathcal{R}}$.

processes. For example, [3] provides separate belief maps of the walls, ceiling and floor of the scene. Alternatively, some works use CNNs to extract the informative structural edges of indoor scenes ignoring those edges from clutter [16], [23]. Instead, in [12] they predict the location of the room layout corners. Other deep learning works extract an estimation of the depth or/and surface normals from simple RGB images which also produces an interesting outcome for layout estimation [5], [11]. The main drawback of these CNNs is that they are always focused on traditional images with limited FOV with the consequent limitations we have mentioned before.

In this paper we propose a new entire pipeline which receives as input a 360° full-view panoramic image and returns a closed, 3D reconstruction of the room. Our experimental evaluations in the public databases SUN360 [20] and Stanford (2D-3D-S) [1] show that the proposed pipeline (Fig. 2) results in high accurate reconstructions outperforming quantitatively (↓ pixel error) and qualitatively (greater fidelity to the actual room shapes) the state of the art. The key contributions of the proposed pipeline are the following: 1) The idea of exploiting deep learning combined with geometry to filter non-significant lines. Our proposal allows to work directly with structural lines, and thus structural corners, to create more efficient algorithms that tackle the layout estimation problem with less iterations and more accuracy. 2) We also propose a new evaluation approach, the Normal Map, alternatively to classical and more recent Maps, and demonstrate to achieve a better performance at hypotheses evaluation step. 3) Finally, we are able to handle flexible closed geometries not limited to 4-wall boxes like other works of the state of the art. This point has a high relevance *e.g.* for using our proposal in a real room-navigation system. Users need to be provided the real space and not just a rectangular simplification of it.

II. Structural Lines in Panoramas

In this section we address the initial stage of our proposal, describing how we extract lines and vanishing points (VP) in panoramas, dealing with spherical projection (Section II-A). Then we extract the structural lines as a subset of all the lines that are significant for our task as learned from data with a deep learning approach (Section II-B).

A. *Lines and vanishing points estimation*

In panoramas, a straight line in the world is projected as an arc segment on a great circle onto the sphere and thus it appears as a curved line segment in the image. For this reason, we represent each line by the normal vector n_i of the 3D projective plane that includes the line itself and the camera center. We adopt the Manhattan World assumption whereby there exist three dominant orthogonal directions. Another particularity of this type of projection is that parallel lines in the world intersect in two antipodal VP whereas in conventional images they do in one single VP. In [24] they split the panorama in order to run a specific algorithm that only works with perspective images, warping then all detected line segments back to the panorama, whereas in [2], they solve the problem by a branch-and-bound framework associated with a rotation space search. Here instead, we detect lines and VP by a RANSAC-based algorithm that works directly with panoramas showing entire and unique line segments, avoiding thus duplicate lines coming from different splits and improving the overall efficiency of the method. We achieve really similar results to [24], [2] being also much faster, ∼8s per image in our proposal, ∼67s per image with [2] and ∼42s per image using [24].

First, we run a Canny edge detector on the panorama and cluster contiguous edge points in edge groups. Each point of the edge group i is projected into the 3D space as a spatial ray $r_{ij}, \forall j = \{1..N_{pts}\}$. Iteratively, two points of each group are randomly selected (r_{i1}, r_{i2}) and thus we get a possible normal direction for the edge group $n_i = (r_{i1} \times r_{i2})$. The number of *inliers* is evaluated, *i.e.* how many rays fulfill the condition of perpendicularity with the normal under an angular threshold of $\pm 0.5°$, $|arccos(n_i \cdot r_{ij}) - \frac{\pi}{2}| \leq \theta_{th}$. After a certain number of iterations the process outputs, for each edge group, the model leading to the highest number of inliers giving the n_i that fits the line best.

We obtain the three orthogonal VP directions (vp_k) with another RANSAC algorithm, considering $vp_k = n_a \times n_b$ where n_a and n_b are the normal vectors of two world parallel

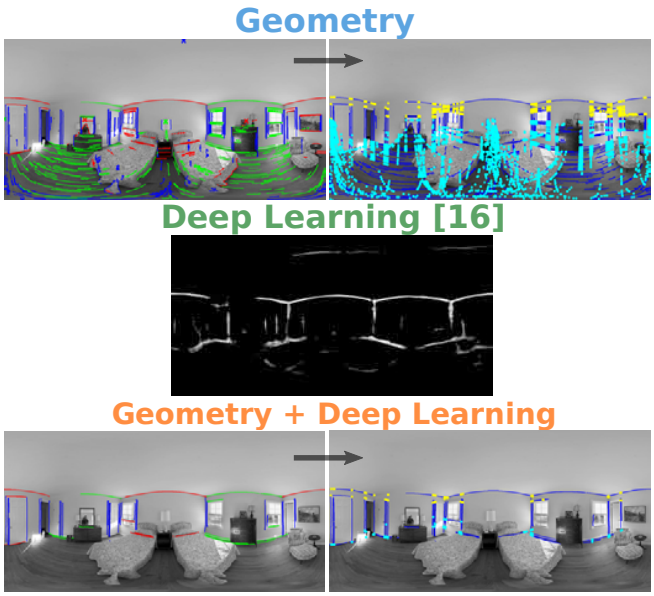


Fig. 3: **Top**: Oriented lines and corners extracted just with geometric reasoning. **Center**: Edge Map obtained through [16]. **Bottom**: Resulting structural lines and corners after combining geometry an deep learning. A large reduction of them is shown, while those more significant for the main structure remain. Corners become good candidates for the hypotheses generation.

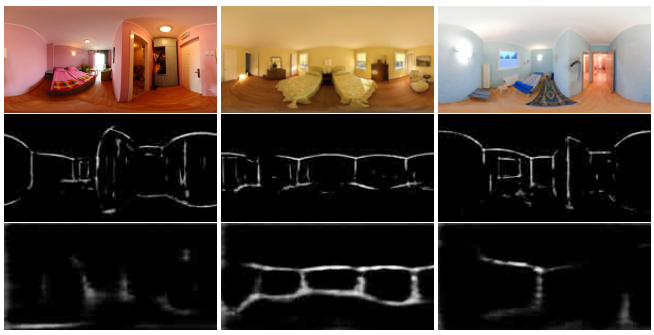


Fig. 4: Comparison of edge maps obtained applying [16] through the proposed discretization of the sphere (**Center**) and directly on the panorama (**Bottom**).

lines. Eventually we select the three VPs (vp_x, vp_y, vp_z) that have the most number of inlier lines, exploiting that normal vectors n_i must be orthogonal to the main directions $|arccos(n_i \cdot vp_k) - \frac{\pi}{2}| \leq \theta_{th}$ where $k = x, y, z$. Inlier lines are classified according to the VP, whereas the other lines are discarded (those whose normals are not perpendicular to any of the main directions). The lines with the same Manhattan direction are shown in identical color in Fig. 3 (top-left). Once VP are computed, we rotate the panorama in a way that it is pointed perpendicularly to one of the room walls.

B. Structural lines introducing deep learning

The main piece of information we use to create layout hypotheses are lines. However, in cluttered scenes is very difficult to know whether they come from actual wall intersections or from other elements of the scene. Proceeding with all the lines leads to an intractable number of hypotheses. In order to tackle this problem, we propose to evaluate the extracted lines on the panoramic image introducing deep learning. CNNs have been successfully applied to extract complex features such as corners [12] or structural edges [16]. However, they have not been trained to deal with omnidirectional images and then, they are very inaccurate when used directly on panoramas. Besides that, it does not exist any dataset collecting panoramic images with enough amount and variety of labeled data required to train a deep neural network. Thus, we do not directly train an end-to-end CNN and decide, instead, to adapt an existing CNN to our image geometry. Here, we adapt the Fully Convolutional Network (FCN) proposed by Mallya and Lazebnik [16]. This network was trained to estimate probability maps representing the room edges of the projected 3D box that fits the room better, even in the presence of clutter and occlusions. Our proposal is to combine such rough yet meaningful information with more accurate geometric cues such as lines.

To apply the FCN, we split the panoramas into a set of overlapping perspective images with a FOV similar to conventional images ($\sim 70^{\circ}$) and planar projection. We run the algorithms in each of them separately to obtain local results and finally stitch them all back to the panorama as in [20], [24], [21]. For the discretization of the sphere, instead of selecting spherical coordinates from uniform distributions $\theta \in (-\pi/2, \pi/2)$ and $\phi \in (-\pi, \pi)$ (which is not adequate since the density increases as we get closer to the poles), we use an algorithm based on the golden section spiral [6]. For any given number of points, it results in an evenly distribution with bins covering areas of similar size equally distant from their closest neighbor. We experimentally choose 60 points, *i.e.* 60 perspective images.

To improve the edge maps, we avoid noise by removing low probability pixel values below a certain threshold (0.2 out of 1). When the virtual perspective images are stitched back to the panorama, there are some overlapping regions that we solve by choosing the maximum value of probability to not lose information. In Fig 4 we show that the accuracy of the edge map substantially improves when we split the panorama, specially in those cases where the result of applying directly the FCN on the panorama is completely uninformative (first and third columns). Once we have the edge map of the panorama given by the FCN [16], we give each extracted line a score calculated as the sum of the corresponding probability values to the pixels it occupies in the edge map. In this way, we remove those lines whose score is below a certain threshold (the 10% of their length), while the others are classified as structural lines. An example of this process can be observed in Fig. 3. It shows clearly the advantage of merging both approaches, where those lines that belong to clutter such as those from the parquet, the tables and even many windows, pictures and doors have been removed, but most relevant lines to recover the structure of the room remain for further stages. With this operation the

number of lines may be reduced to one-third or even a quarter depending on the scene.

III. Room layout estimation

Our goal is to extract the main structure of an indoor environment *i.e.* the distribution of floor, ceiling and walls, abstracting all objects within rooms. For this purpose we have developed a method to generate layout hypotheses from corners found with the already filtered significant lines. Our algorithm is divided in three stages:

A. *Candidate corners extraction*

Our layout generation process is based on corners, *i.e.* structural intersections between two walls and ceiling or floor. In a Manhattan World, two line segments are enough to define a corner, so we intersect all the significant lines in different directions (x, y, z) among themselves in pairs as long as they do not cross each other. The direction vector of the corner point is computed with a cross product of the lines intersecting in that corner, $c_{ac} = (n_a \times n_c)$. The previous selection process of structural lines with learning makes these extracted corners already good candidates. Fig. 3 shows the large difference between obtaining corners with first line extraction (top) and with structural lines (bottom). By removing non-structural lines, the number of corners extracted is vastly reduced, yet the important ones remain detected. This reduction makes further stages of the method faster and more efficient, but also improves the reliability of the results since most corner candidates coming from clutter and irrelevant structures are not considered.

Panoramic images have the advantage of providing a full view of the room, allowing us to look around, up and down in the scene. Unlike conventional images where the ceiling and some walls use to be out of the FOV. Taking this into account, we carry out a classification of the detected corners following two criteria (See Fig.5):

1) Their position along the z axis: Corners detected below the horizon line $l_H(-z)$ in the image are considered as floor corner candidates and those detected above the $l_H(+z)$ as ceiling corner candidates.
2) Their position in the XY-plane: Since the camera is inside the room, we divide the scene into four quadrants around the center of the camera with the horizontal VPs as quadrant dividers, $\mathcal{Q} = \{q_1, q_2, q_3, q_4\}$. Hence, *e.g.* to determine when a corner belongs to the fourth quadrant: $c \in q_4 \iff c_x \in \mathbb{R}^+ \wedge c_y \in \mathbb{R}^-$.

Manhattan World rooms always have an even number of walls, and the number of corners in each quadrant will be an odd number so, this quadrant division allows to easily know some additional information to sample corners. For example, the simplest layout will include just one corner in each quadrant while more complex layouts will have three or even five corners in some of their quadrants.

B. *Layout hypotheses generation*

Many works simplify the layout generation problem by assuming that the room is a simple box of four walls, sometimes because of lack of information due to the use of conventional images with smaller FOV [7], [8], [18], or just to subtract complexity to the problem [24]. Here, we face more complex designs which will be faithful to the actual shapes of the rooms, introducing the possibility of estimating in-between hidden corners when required, *i.e.* when they are occluded by clutter or due to scene non convexity. We generate layout hypotheses by means of an iterative method that attempts to join consecutive corners with alternatively oriented walls following Manhattan assumption.

Our algorithm randomly generates at each iteration initial groups of corners, $\mathcal{G}_c$, which are ordered clockwise in the XY-plane. There is a relation between the number of corners randomly selected $N_{\mathcal{G}_c}$ and the maximum number of walls N_W^{max} that our algorithm is able to solve with them, $N_W^{max} = 2(N_{\mathcal{G}_c} - 1)$. In this way, we can adjust the complexity of the layouts just giving more or less freedom to the random function that selects the initial corners. This relation means that *e.g.* we can draw layouts with six walls from a minimum of four corners allowing the algorithm to introduce two new corners that maybe were not visible in the image. For this initial selection, we establish a minimum requirement for which there must be corners in at least three quadrants $\subseteq \mathcal{Q}$, thus, the corner in the remaining quadrant can be estimated assuming closed Manhattan layouts, and there must be at least one corner of each hemisphere, *e.g.* $\mathcal{G}_c = \{c_{q_2}^{ceiling}, c_{q_3}^{floor}, c_{q_4}^{ceiling}\}$. This last condition allows us to estimate the height of the room, *i.e.* the relative distance of the camera to the ceiling and floor planes. We proceed with the geometric reasoning in 2D as in the right side of the Fig.5, with a top view of the scene. While we do not have the 3D coordinates of the corners but just their direction vector (ray), we assume that all the candidate corners from each hemisphere intersect in a single ceiling and floor plane respectively. The vertical Manhattan direction is the normal direction of both planes (Ceiling-floor symmetry).

An example of hypotheses generation performing this operation is shown in Fig. 5. The corners above the l_H (c_1, c_2 and c_3) belong to the ceiling, so we intersect their rays (yellow) into a reference ceiling plane in a way that the walls connecting them can be obtained in 2D assuming Manhattan world. The corner below the l_H (c_4) belongs to the floor but, although we know that it is parallel to the ceiling plane, the distance between them is a priori unknown. We use the Manhattan world requirement to estimate the floor position along its ray (cyan), choosing the one that makes the projection of the point such that the walls connected by the corner are as perpendicular as possible.

In Fig. 6 two more complex examples of layout hypotheses generation are shown. In the first example we present a valid layout hypothesis where an initial random group of candidate corners is selected $\mathcal{G}_c = \{c_1, c_2, c_3, c_5\}$. This means that the algorithm will be able to solve a layout hypothesis with $N_W^{max} = 6$. A joining corner process starts from c_1 finding then a floor spatial ray. In order to find the optimal corner position along this ray, the algorithm finds possibilities with its nearest corners and draws an intermediate solution, c_2. In

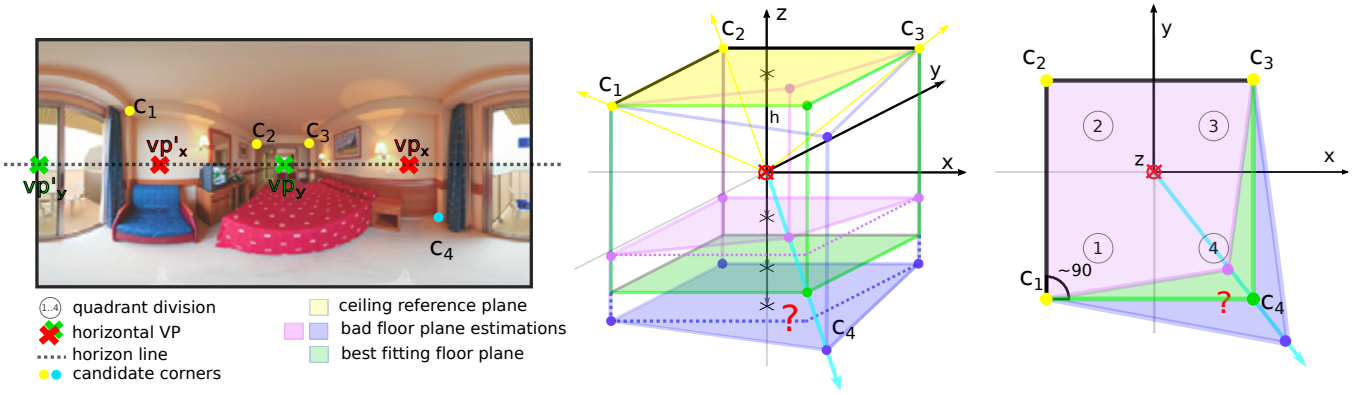


Fig. 5: **Room height**. We take advantage of the ceiling-floor symmetry to estimate the distance between both planes. We look for the solution that makes the projection of the floor corner such that the walls connected by the corners are as perpendicular as possible (Manhattan world assumption).

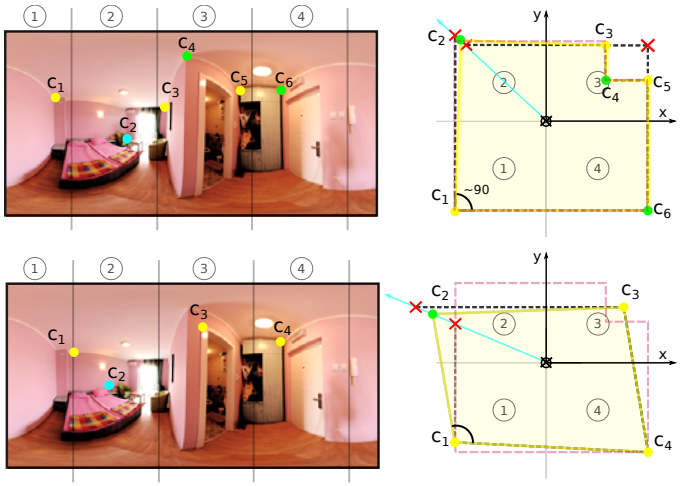


Fig. 6: **Layout hypothesis generation**: We show two examples of layout hypotheses generation. The first example corresponds to a valid hypothesis whereas the second one presents a non-valid discarded hypothesis.

the third quadrant, taking into account the direction $(x-y)$ from previous unions, our algorithm selects the best solution for c_4 by choosing the one which produces alternatively oriented consecutive walls. In the empty quadrant, Manhattan walls from nearest corners give c_6. For each union the Manhattan assumption is checked with a certain threshold ($90°\pm5°$). In the second example we show a non-valid layout hypothesis. Following the same idea, initial random corners are selected (c_1, c_2, c_3, c_4) and orderly joined conforming in this case a non-Manhattan layout, so it is rejected as hypothesis. When we get the corner floor position along its direction vector, we can obtain the distance between ceiling and floor planes that verifies the ray equation (This is illustrated in the **video attachment**).

C. *Layout hypotheses evaluation*

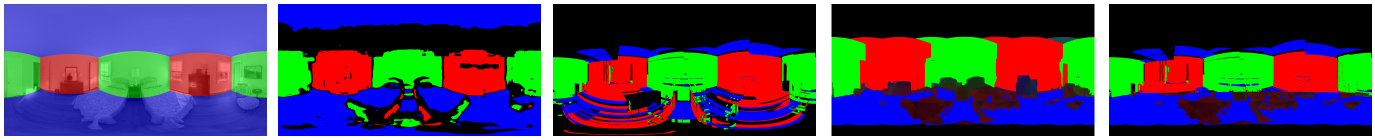
In the hypotheses generation stage we obtain a certain number of layout hypotheses (N_h). In the evaluation process we determine which one is the best, and therefore, the final result. For each hypothesis H_i, we generate a labeled image $\mathcal{I}^{H_i}$, in which each pixel encodes the orientation of the surface (e.g. wall in x, wall in y or floor/ceiling in z). In Fig. 7(a) there is an example of a labeled map $\mathcal{I}^{H_i}$ where each label has different color. Then, we evaluate the hypotheses fitting to a *reference map* $\mathcal{I}^R$ that roughly encodes the orientation of the pixels, and can be obtained from several methods. We compute the ratio of pixels that are equally oriented in $\mathcal{I}^R$ and $\mathcal{I}^{H_i}$ over the total size of the image, that we call Equally Oriented Pixel ratio (EOP):

$$EOP\left(\mathcal{I}^{H_i}, \mathcal{I}^R\right) = \frac{1}{M \cdot N} \sum_{x,y,z}^{P} \sum_{i,j}^{M,N} \mathcal{I}^{H_i} \& \mathcal{I}^{\mathcal{R}},$$

being M and N are the height and width of the images $\mathcal{I}$ and P the number of channels (corresponding to the labels, *i.e.* orientations *x,y,z*).

In this work, we test four methods to compute the reference map $\mathcal{I}^{\mathcal{R}}$, three of them from the literature and one proposed in this paper. Orientation Map [13], $\mathcal{I}^{OM}$ (Fig. 7(c)), and Geometric Context [7], $\mathcal{I}^{GC}$ (Fig. 7(d)) are two methods widely used over years. Recently, researches [24], [9] have started to combine the strengths of both of them in one single map that we call Merge Map, $\mathcal{I}^{MM}$ (Fig. 7(e)).

We propose a fourth method, *Normal Map* ($\mathcal{I}^{NM}$), applying another recent deep learning method to our task. We choose the work from Eigen and Fergus [5], which proposes a multiscale convolutional network that returns depth prediction, surface normal estimation and semantic labeling of indoor images. Here, we take advantage of the surface normal estimation to create the reference map. As expected, this network has been trained with conventional images, so it does not work properly with panoramic images. To address this problem, we adapt the CNN to our image geometry by splitting the panorama in perspective images as in SectionII-B. In this case, in order to stitch them back to the panorama, we need to rotate the normals to set them in a common reference frame. Overlapping areas are tackled



(a) Layout hypothesis ($\mathcal{I}^{H_i}$) (b) Normals Map ($\mathcal{I}^{NM}$) (c) Orientation Map ($\mathcal{I}^{OM}$) (d) Geometric Context ($\mathcal{I}^{GC}$) (e) Merge Map ($\mathcal{I}^{MM}$)

Fig. 7: (a) Example of labeled image generated from layout hypotheses. (b)-(e) Visual representation of how each of the *reference maps* $\mathcal{I}^{\mathcal{R}}$, looks like.

	EOP	*Computing Time*
Normal Map ($\mathcal{I}^{NM}$)	**0.925±0.061**	243.36±1.42
Orientation Map ($\mathcal{I}^{OM}$)	0.906±0.133	**23.54±4.16**
Geometric Context ($\mathcal{I}^{GC}$)	0.883±0.114	174.07±13.28
Merge Map ($\mathcal{I}^{MM}$)	0.923±0.147	197.61±17.44

TABLE I: Ratio of equally-oriented pixels when comparing the best final hypotheses, $\mathcal{I}^{H_b}$, with the ground truth $\mathcal{I}^{GT}$, evaluating in each case with a *reference map*. Also the computing time in seconds of generating each map is shown.

in this case by doing the per-pixel average to achieve a continuity of the overall image. Then we apply an angular threshold to determine whether or not the normals from each pixel belong to a main direction (VPs) and label them accordingly. Resulting normal map is shown in Fig. 7(b). It can be noticed that the ceiling is the worst estimated part by the CNN since black pixels means uncertain areas (*i.e.* not belonging to any main direction). This happens because the CNN was trained with images where ceiling does not usually appear, making it difficult for the net to predict them.

IV. Experiments

We have evaluated our proposal using full-view panoramas of indoor scenarios from two public datasets. In particular, most of our quantitative results have been obtained from a subset of 85 panoramas of bedrooms and living rooms of the SUN360 dataset [20]. Additionally, we also show some results using the Stanford (2D-3D-S) dataset [1]. For each panorama we have manually created the ground truth as a labeled image $\mathcal{I}^{GT}$, similar to those in Fig. 7, where each pixel encodes the direction of the surface it belongs to. A previous ground truth was provided by [24], but was unusable for us since images were labeled following the box-shaped rooms simplification. The accuracy of our results is evaluated by computing $EOP\left(\mathcal{I}^{H_b}, \mathcal{I}^{GT}\right)$, measuring the ratio of equally-oriented pixels between the best hypothesis and the ground truth. Each EOP value shown is a median of 10 times performing the experiment. The number of hypotheses drawn (N_h) is specified in each experiment. For the experiments we allow the algorithm to initially select from three to five corners, *i.e.* to solve layouts with four to eight walls. Some examples of final layout estimations and 3D models are shown in Fig. 12. This submission includes a video which illustrates the procedure and shows some additional results.

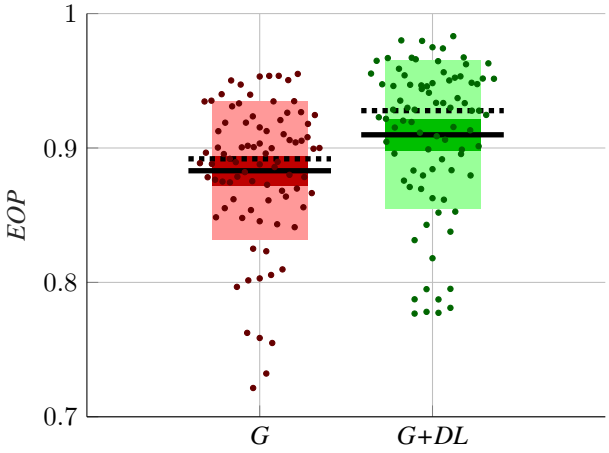


Fig. 8: **Advantages of combining.** Here we highlight the advantages of using structural lines from Geometry and Deep Learning combination [16] over lines obtained only with Geometry. The mean is represented in solid black and the median in dotted black. Also the standard deviation is shown in light color and jittered raw data are plotted for each group.

a) ***Edge map advantages****:* A comparative study showing the effects of selecting structural lines (Section II-B) can be found in Fig. 8. For this experiment we choose $N_h = 100$ and the $\mathcal{I}^{NM}$ as reference map. Every single image evaluated is represented as a point: green if geometry and deep learning (G+DL) are used to obtain structural lines, and red if just geometry (G) is used. The graph demonstrates the improvement when combining both techniques, highlighted by the mean and especially median values: 0.889 vs. 0.925. Thus, the experiment proves that the inclusion of DL techniques in the pipeline of the process clearly benefits the approach. In particular, the detection of structural lines allows to remove clutter effectively, which translates into better accuracy.

b) ***Reference maps comparison****:* We compare the performance using the four alternative reference maps at hypotheses evaluation step (Section III-C). Here, we use $N_h = 100$ as well. Table I shows the median EOP value and the computing time of creating each map. In terms of accuracy, $\mathcal{I}^{NM}$ and $\mathcal{I}^{MM}$ perform similarly in median, although the smaller standard deviation of the $\mathcal{I}^{NM}$ indicates more consistent results. Both are considerably better than $\mathcal{I}^{OM}$ and $\mathcal{I}^{GC}$. However, the $\mathcal{I}^{OM}$ is about ten times faster to compute than the $\mathcal{I}^{NM}$ and, therefore, its usage would be

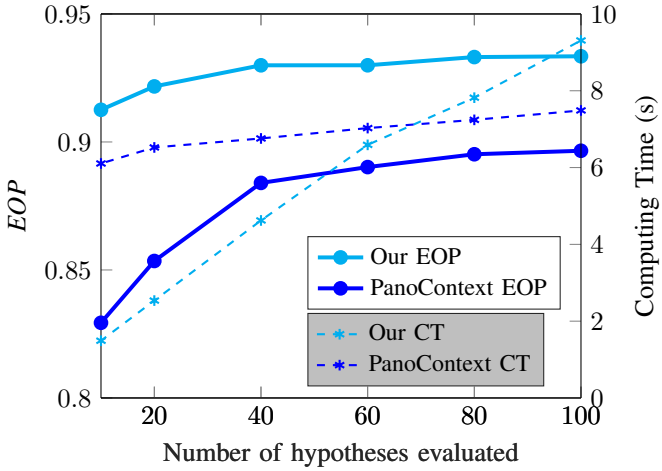


Fig. 9: **Comparison with PanoContext** [24] (with only four-wall rooms). We show the ratio of equally-oriented pixels and computing time against the number of hypotheses. Our method outperforms PanoContext and is able to provide much better results and much faster with fewer hypotheses.

recommendable if the priority lies in getting fast results in spite of losing some accuracy. The smaller standard deviation on the computing time of the $\mathcal{I}^{NM}$ shows that it does not vary through images, unlike the others whose time depends on scene-specific features such as the number of lines.

*c) **Comparison with the state of the art**:* We perform a comparison with PanoContext [24] since it is, to our knowledge, the only directly related method with available code. We establish the comparison with the first stage of their algorithm that reaches the same point as our work does, since after layout extraction they introduce object detection in the method. To evaluate accuracy, in order to carry out a direct and fair comparison we only compare numerically the four-wall rooms cases, removing more complex shaped ones from the experiment. In Fig. 9 we show the EOP ratio and the computing time necessary to generate the hypotheses for each method, varying the number of hypotheses N_h. Our method clearly outperforms [24], being the difference larger when only a few hypotheses are considered. Although the difference decreases as the amount of hypotheses rises, when both methods reach a stable EOP value our proposal continues giving better results. Moreover, with just 10 hypotheses (91,26%) our method beats [24] with 100 hypotheses (89.66%). This shows the good performance of our structural lines selection which increases the likelihood of getting good hypotheses with only a few attempts. Computing times show again bigger difference when fewer hypothesis are evaluated. Only rooms up to 4 walls are considered here to be fair with [24], but our method is also able to deal with more complex rooms (see Fig. 10).

*d) **Different datasets**:* Besides the 85 images from the SUN360 dataset, we additionally tested our method with 25 panoramas from the Stanford (2D-3D-S) dataset. In Table II we show the EOP we reach in both datasets. Several reasons

Fig. 10: **Comparison with PanoContext** [24] in complex geometries. Our method (cyan) is able to find 6 walls whereas [24] (dark blue) always finds just 4 walls.

Dataset	Category	EOP (N_h =100)
LSUN360	bedroom	0.921
	livingroom	0.933
Stanford (2D-3D-S)	area1	0.873
	area3	0.885

TABLE II: Ratio of equally-oriented pixels evaluated in different scenarios from two public datasets.

can be associated with the fact that our proposal works better with SUN360 dataset. On the one hand, panoramas from the Stanford dataset do not cover full view vertically, leaving a black mask that can lead to confusions in the limits when extracting structural lines. On the other hand, indoor scenes represented in the second dataset show more challenging scenarios like cluttered laboratories or corridors instead of bedrooms and living-rooms (see Fig. 11). Still, our method achieves more than 87% of Equally Oriented Pixels in this dataset.

V. Conclusion

We propose a novel entire pipeline which converts 360° panoramas into flexible, closed, 3D reconstructions of the rooms represented in the images. Our experimental results show that the proposed algorithm has a good performance in scene interpretation of full-view images and outperforms the state of the art not only in terms of accuracy but also in speed. As future work we consider to train a CNN able to work with both conventional and omnidirectional images.

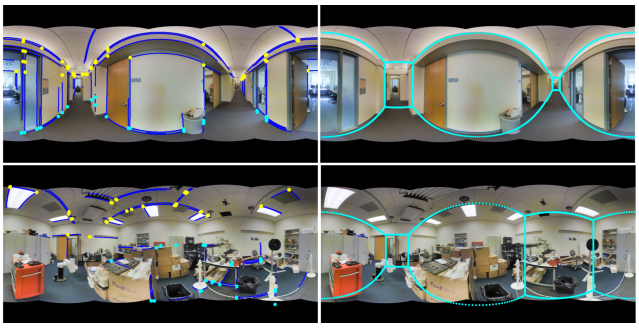


Fig. 11: **Top**: challenging corridor well estimated by our approach in Stanford (2D-3D-S) dataset. **Bottom**: a clear case of failure.

Fig. 12: Final layout estimations handling **different geometries** (cyan) compared with their ground truth (red).

References

[1] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv*, Feb. 2017.

[2] J.-C. Bazin, Y. Seo, and M. Pollefeys. Globally optimal consensus set maximization through rotation search. In *Asian Conference on Computer Vision*, pages 539–551, 2012.

[3] S. Dasgupta, K. Fang, K. Chen, and S. Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–624, 2016.

[4] E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3D reconstruction from a single indoor image. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2418–2428, 2006.

[5] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE Int. Conf. on Computer Vision*, pages 2650–2658, 2015.

[6] Á. González. Measurement of areas on a sphere using fibonacci and latitude-longitude lattices. *Mathematical Geosciences*, 42(1):49, 2010.

[7] 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.

[8] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. *European Conference on Computer Vision*, pages 224–237, 2010.

[9] A. B. Jahromi and G. Sohn. Geometric context and orientation map combination for indoor corridor modeling using a single image. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41, 2016.

[10] H. Jia and S. Li. Estimating structure of indoor scene from a single full-view image. In *IEEE International Conference on Robotics and Automation*, pages 4851–4858, 2015.

[11] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Fourth Int. Conf. on 3D Vision*, pages 239–248. IEEE, 2016.

[12] C. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. RoomNet: End-to-end room layout estimation. In *IEEE International Conference on Computer Vision*, 2017.

[13] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2136–2143, 2009.

[14] G. Lopez-Nicolas, J. Omedes, and J.J. Guerrero. Spatial layout recovery from a single omnidirectional image and its matching-free sequential propagation. *Robotics and Autonomous Systems*, 62(9):1271–1281, 2014.

[15] R. Lukierski, S. Leutenegger, and A. J. Davison. Room layout estimation from rapid omnidirectional exploration. In *IEEE International Conference on Robotics and Automation*, pages 6315–6322, 2017.

[16] A. Mallya and S. Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *IEEE International Conference on Computer Vision*, pages 936–944, 2015.

[17] 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*, pages 396–412, 2016.

[18] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3D layout and object reasoning from single images. In *IEEE Int. Conf. on Computer Vision*, pages 353–360, 2013.

[19] A. G. Schwing and R. Urtasun. Efficient exact inference for 3d indoor scene understanding. In *European Conference on Computer Vision*, pages 299–313, 2012.

[20] J. Xiao, K. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2695–2702, 2012.

[21] J. Xu, B. Stenger, T. Kerola, and T. Tung. Pano2CAD: Room layout from a single panorama image. In *IEEE Winter Conference on Applications of Computer Vision*, pages 354–362, 2017.

[22] H. Yang and H. Zhang. Efficient 3D room shape recovery from a single panorama. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5422–5430, 2016.

[23] W. Zhang, W. Zhang, K. Liu, and J. Gu. Learning to predict high-quality edge maps for room layout estimation. *Transactions on Multimedia*, 19(5):935–943, 2017.

[24] 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*, pages 668–686, 2014.