Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Atlanta scaled layouts from non-central panoramas

Bruno Berenguel-Baeta*, Jesus Bermudez-Cameo, Jose J. Guerrero

I3A, University of Zaragoza, Zaragoza 50018, Spain

ARTICLE INFO

Article history:
Received 23 June 2021
Revised 30 March 2022
Accepted 23 April 2022
Available online 25 April 2022

Keywords:
Omnidirectional vision
3D vision
Non-central cameras
Layout recovery
Scene understanding

ABSTRACT

In this work we present a novel approach for 3D layout recovery of indoor environments using a non-central acquisition system. From a single non-central panorama, full and scaled 3D lines can be independently recovered by geometry reasoning without additional nor scale assumptions. However, their sensitivity to noise and complex geometric modeling has led these panoramas and required algorithms being little investigated. Our new pipeline aims to extract the boundaries of the structural lines of an indoor environment with a neural network and exploit the properties of non-central projection systems in a new geometrical processing to recover scaled 3D layouts. The results of our experiments show that we improve state-of-the-art methods for layout recovery and line extraction in non-central projection systems. We completely solve the problem both in Manhattan and Atlanta environments, handling occlusions and retrieving the metric scale of the room without extra measurements. As far as the authors' knowledge goes, our approach is the first work using deep learning on non-central panoramas and recovering scaled layouts from single panoramas.

1. Introduction

Layout recovery and 3D understanding of indoor environments is a hot topic in computer vision research [1,2]. Recovering the information of an environment from a single view is an attractive tool for different applications such as virtual or augmented reality [3] and human pose estimation [4,5]. Previous works for layout recovery relied on pure geometrical processing [6]. Those methods usually required hard layout assumptions and iterative proccesses in order to obtain proper results. Besides, since many hypotheses and verifications should be made, these approaches derive in very slow implementations, not suitable for real time applications. The development of neural networks made the problem of layout recovery more accurate, efficient and faster. The high and low level features obtained by deep learning architectures have proven to be useful for structural recovery of indoor environments.

Through the development of algorithms for layout recovery, different kinds of acquisition systems have been used in order to obtain more information of the environment with as less images as possible. An example is the evolution from perspective images to equirectangular panoramas, which acquire much more information of the environment in a single image.With this extra information, better reconstructions from more complex environments could be made. In this paper, we propose to go a step further and evolve taking as acquisition system the non-central circular panoramas proposed in [7,8] (for simplicity, we will call these as non-central panoramas in the rest of the article). These panoramas provide 360 information of the environment and the image distortion of the non-central acquisition systems includes subtle differences allowing geometric 3D reasoning. In particular, the distortion of the curves fitting the projections of lines encodes the full 3D description of the line. This characteristic of non-central projection systems is a clear advantage for environment reconstruction, since it allows to recover the scale of the environment directly from the image, without measurement assumptions (e.g. camera position or room height). However, due to their sensitivity to noise and complex geometric reasoning, non-central panoramas have been little investigated. Fig. 1 shows two panoramas, one central and the other non-central, in the same environment and from the same position. From the non-central panorama we can recover a more accurate layout, including the real scale.

In this work we present the first proposal of layout recovery with single non-central panoramas and the first deep learning approach for this kind of images. We propose to adapt the neural network architecture of HorizonNet [9] to non-central circular panora-

* Corresponding author.
E-mail addresses: berenguel@unizar.es (B. Berenguel-Baeta), bermudez@unizar.es (J. Bermudez-Cameo), josechu.guerrero@unizar.es (J.J. Guerrero).

https://doi.org/10.1016/j.patcog.2022.108740

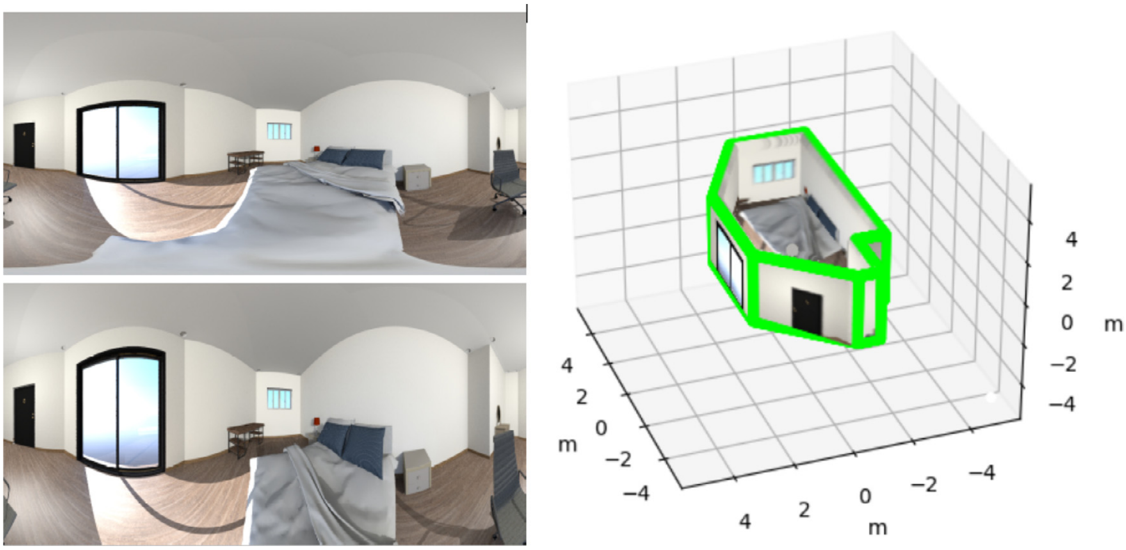


Fig. 1. Central (top-left) and non-central (bottom-left) panoramas from the same virtual environment taken in the same position. Both panoramas have similar appearance but there are subtle differences in favor of the second if we want to obtain 3D information. On the right, the scaled layout from a single non-central panorama in Atlanta world. The green wireframe shows the real 3D layout of the virtual environment.

mas for the extraction of the boundaries of the structural lines from indoor environments. As in [9], we assume that our panoramas are horizontally oriented and that the layout share the ceiling and floor heights. These restrictions will give strong priors in the geometrical proccessing. Taking advantage of the omnidirectional view of non-central panoramas and the unique properties of the non-central projection systems, we extract the 3D information of the structural lines provided by the network. The experiments performed show that our pipeline outperforms the state of the art in layout recovery by a margin. The main contributions of this paper are as follow: Two new geometrical solvers to obtain the layout of an environment in a Manhattan or Atlanta world assumption for non-central projection systems. First work that uses deep learning with non-central projection systems. First work of scaled layout recovery without extra measurements from Manhattan and Atlanta world assumptions, handling occlusions, from a single non-central panorama.

Preliminary results of our work have been presented in [10]. Here, the proposed approach is revised and completed, including an ablation study, a more complete experimentation and comparison with state of the art methods and more detailed explanations of the basis and implementation.

2. State of the art

Non-central projection systems have been extensely studied from their different acquisition systems [11]. Different works have set the fundamentals on catadioptric systems [12] based on conical [13] or spherical [14] mirrors. Other non-central images come from moving cameras, as the pushbroom camera [15] or the non-central panorama [7,8]. These non-central projection systems present geometrical properties that allow to recover 3D information from single images with geometric reasoning. In particular, several works exploit that a 3D line can be recovered with scale from a single non-central projection [16]. The fundamentals for this approach consist in computing the intersection of a line by four generic rays [17]. However, although it is theoretically possible to recover the scaled, full 3D reconstruction of a line from a single non-central projection, in practice the results are so sensitive to noise and therefore it is not possible to directly use these approaches with current non-central systems. For this reason, more recent works aim to improve the accuracy of 3D lines fitting by imposing structural constraints. As example, in [18] the line extraction is constrained to lines parallel to a known plane, which can be used for extracting horizontal lines from a non-oriented camera by using the gravity direction, for example from an IMU, as prior. Other constraints such as parallel lines or intersecting orthogonal lines are well studied and solved in [19], where they present a minimal solution to be included in a robust approach.

When the geometric constraints of structural lines are globally considered, lines and their intersections are enclosed in the concept of layout. The layout of indoor environments provides a strong prior for many computer vision tasks. Several works on virtual or augmented reality [3], object recognition [20–22], segmentation [23,24] and human pose estimation [4,5] rely on information of the environment, which is more easily obtainable once the layout is known. Many different methods have been developed in order to recover the layout of a room from different central cameras [25]. Particularly, in recent years, the use of omnidirectional central images is on the rise, since a single image can provide enough information to make an estimation for a whole room [26–28]. One of the first attempts for layout estimation is the work [29] which presents an implementation where many 3D layout hypotheses are generated and then ranked by a Support Vector Machine. Then the best ranked hypotheses are selected and compared with the input image to test its validity. More recent approaches take advantage of neural networks to estimate the layouts in a more efficient way. Corners for Layouts (CFL) [30] uses an encoder-decoder architecture with convolutions adapted to the spherical distortion of the equirectangular panorama. The output of the network are two heat-maps for the corners and edges that compose the structure of rooms. With a post-processing of this information, an up-to-scale reconstruction of Manhattan environments can be obtained. Other recent approaches combine the convolutional networks with recurrent neural networks, which allow to obtain dependencies along the image, extracting the boundaries of the structural lines. Relying on different geometry constraints, HorizonNet [9] and AtlantaNet [31] obtain a 1D representation of corners, as a probability of having a wall-wall intersection at a certain image column, and other 1D representation of the ceiling-wall and floor-wall intersections which form the structural lines of the room. This minimal representation allows to obtain a more precise approximation of the

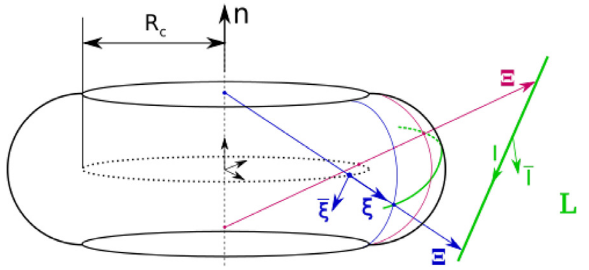


Fig. 2. Toroidal projection of non-central panoramas. For each point of the circular trajectory of optical centers with radius R_c, there is a ring in which the projection is central. A 3D line **L** is projected in the toroidal surface. A projecting ray Ξ intersects the line giving a point in the toroidal surface for a unique optical center.

layout of the room. As in other works, after a post-processing, an up-to-scale layout estimation can be obtained.

In our proposal, we overcome the problem of structural lines extraction in non-central panoramas adapting the neural network of HorizonNet [9]. Then, we propose a new geometrical processing, which takes advantage of two new solvers that fit Manhattan and Atlanta layouts, to recover the 3D layout of indoor environments. Besides, exploiting the geometrical properties of non-central projection systems in our geometrical processing, we are able to recover the scale of the 3D layout without extra measurements, which no state-of-the-art method is able to do.

3. The non-central panorama

Central projection systems are those acquisition systems where all projecting rays that form the image intersect at a single point, called optical center. The pinhole camera model or the spherical panorama are examples of central projection systems. By contrast, non-central projection systems do not have a unique optical center, that means, the rays that form the images do not pass through a unique point. This nature leads to a harder modeling of the projection and management of the information. Nevertheless, this characteristic allows to obtain more geometric information from the image than from central projection systems. In particular, we can extract the 3D information of lines directly from a single image.

The non-central panorama is a projection model, presented in [32], with symmetry of revolution in which each projecting ray intersects both an axis $\mathbf{n}$ and a circle of radius R_c (see Fig. 2). In the last two decades some works have studied the geometrical properties of non-central panoramas [8,33] and other non-central projection systems [11,34]. Taking advantage of the unique geometrical properties of this acquisition systems, different applications have been already studied, as 3D line extraction [18,35].

The non-central panorama can be modeled as the projection of the environment into a toroidal surface (Fig. 2). The optical center is distributed in a circle of radius R_c, centered in an axis $\mathbf{n}$. For each optical center, we have a region where the projection is locally central, which correspond with one column in the panoramic image. We use Plücker coordinates [36] to define the backward projection function of the system as well as the 3D lines in the environment.

The notation to define the projection model and the math presented in this paper is as follows. For projecting rays and lines, defined in Plücker coordinates, we will use bold uppercase letters (e.g. Ξ, $\mathbf{L}$). For vectors that belong to $\mathbb{R}^3$, bold lower case letters (e.g. $\mathbf{n}$), while vectors of greater dimmensions are presented as upper case letters in Euler font (e.g. $\mathscr{W}$). Matrices are presented in the serif font as uppercase letters (e.g. A). Scalar values are presented as standard text (e.g. R_c, φ).

$$\varphi = atan2(y,x); \quad \phi = atan\left(\frac{z}{\sqrt{x^2+y^2}-wR_c}\right) \tag{1}$$

$$j = n_{columns}\frac{\varphi-\varphi_{ini}}{\varphi_{end}-\varphi_{ini}}; \quad i = m_{rows}\frac{\phi-\phi_{ini}}{\phi_{end}-\phi_{ini}} \tag{2}$$

The forward projection provides the pixel coordinates (i,j) for each 3D point $(x,y,z,w)^T$ defined in homogeneous coordinates. Each point is defined by two angles (ϕ,φ) from its corresponding optical center as defined in Eq. (1). The angles are transformed into pixel coordinates (2) taking into account the image resolution $(m_{rows}, n_{columns})$ and the horizontal (ϕ_{ini},ϕ_{end}) and vertical $(\varphi_{ini},\varphi_{end})$ fields of view.

$$\begin{aligned}\Xi &= (\boldsymbol{\xi};\bar{\boldsymbol{\xi}})\\ &= (\cos\phi\cos\varphi;\cos\phi\sin\varphi;\sin\phi;R_c\sin\phi\sin\varphi;\\ &\quad -R_c\sin\phi\cos\varphi;0)\end{aligned} \tag{3}$$

The backward projection model provides the projecting rays in Plücker coordinates from each pixel in the non-central panorama. The pixel coordinates are transformed back into spherical coordinates taking into account the image resolution and field of views. Then, the projecting ray (3) is computed considering the radius R_c of the non-central acquisition system.

3.1. Computing 3D lines from a non-central projection

Computing 3D lines from a single image is a particular property we want to exploit. Here we introduce how to compute a 3D line from a non-central projection system. Defining in Plücker coordinates a 3D line as $\mathbf{L} = (\mathbf{l}^T,\bar{\mathbf{l}}^T)^T \in \mathbb{P}^5$ (where $\mathbf{l} \in \mathbb{R}^3$ and $\bar{\mathbf{l}} \in \mathbb{R}^3$) and a projecting ray $\Xi = (\boldsymbol{\xi}^T,\bar{\boldsymbol{\xi}}^T)^T \in \mathbb{P}^5$, their intersection is defined by the side operator [36] as $side(\mathbf{L},\Xi) = \mathbf{l}^T\bar{\boldsymbol{\xi}} + \bar{\mathbf{l}}^T\boldsymbol{\xi} = 0$.

Given that a 3D line has four degrees of freedom, we need, at least, 4 independent equations to solve for $\mathbf{L}$. In general, four projecting rays from a 3D line generate four independent constraints from where we can compute the 3D line [17]. However, we can find some degenerate cases where four projecting rays do not generate independent constraints, e.g. the rays are coplanar with the revolution axis or with the plane containing the circle of optical centers.

4. Layout estimation proposal

Our proposal for layout estimation is a new pipeline composed by two main blocks (see Fig. 3). In a first block, we use a neural network to obtain the boundaries of the structural lines of an indoor environment from an image. On the second block, we geometrically process the information provided by the network, exploiting the properties of non-central projection systems and recovering the scaled layout.

With respect to the first block, [18] and [29] propose geometrical methods based on hypothesis generation-verification to extract lines and layouts respectively while [9] and [31] rely on the use of neural networks for layout recovery. In our proposal we combine both solutions in order to obtain the scaled layout from a single non-central panorama. The use of a neural network allows to obtain the structural lines of an environment faster than with classical approaches of hypothesis generation-verification. In the next section we define in more detail the network architecture proposed and its advantages and disadvantages. On the other hand, we aim to exploit the geometrical properties of non-central projection systems. In Section 5 we present in detail the different geometrical solutions proposed to solve the layout recovery problem and the geometrical pipeline proposed to obtain scaled layouts from single panoramas.

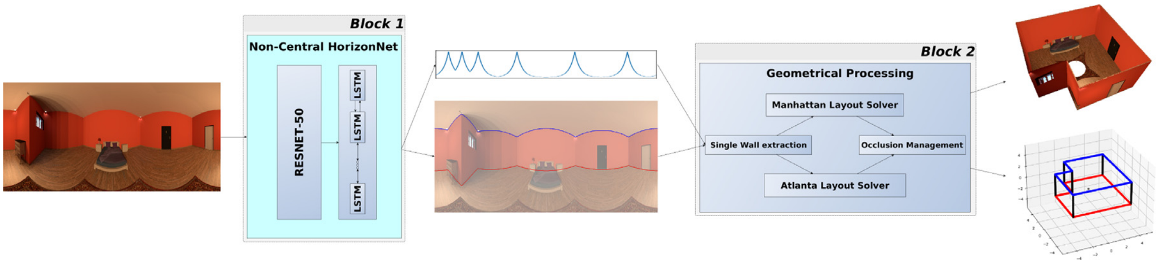


Fig. 3. Pipeline of our proposal. In a first stage, the neural network extracts the boundaries of the structural lines of the room as well as a probability of corner positions from the non-central panorama. On a second stage, our proposed geometrical processing exploits the properties of non-central projection systems to recover the 3D of the layout from the information provided by the network.

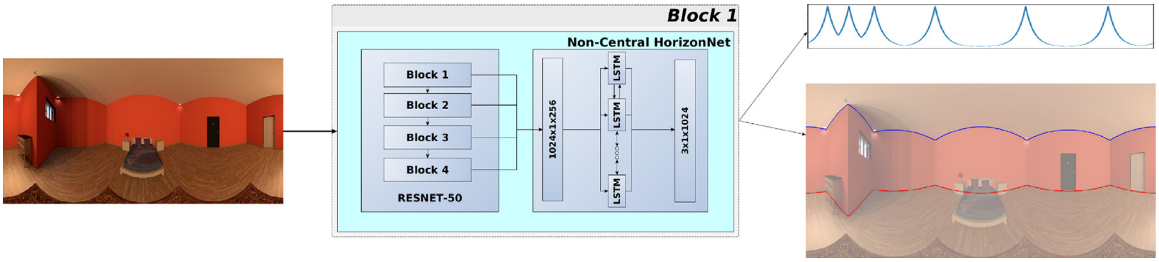


Fig. 4. The non-central panorama is processed by Non-central HorizonNet, which is an adaptation of the work [9]. First, it goes through a ResNet50, where high and low-level features are extracted. After a set of convolutions, the result is concatenated and fed to an array of bidirectional LSTMs. The network provides the boundaries of the structural lines of ceiling and floor, as well as the corners of the room as three separate 1D arrays.

4.1. Non-central horizonnet

We propose to adapt an existing neural network in order to obtain the structural lines of indoor environments from non-central circular panoramas (see Fig. 4). This architecture is divided in two parts: a convolutional part, formed by the first layers of ResNet-50 [37] and a set of convolutions; and a recurrent part, formed by a set of bi-directional LSTMs [38]. From this architecture, we obtain three 1D arrays with the boundaries information from the image. One of the arrays contains the probability of finding a wall-wall intersection in each column of the image. The other two 1D arrays provide the pixel of the intersection between the ceiling or the floor with the walls. From these three 1D arrays we obtain the boundaries of each of the structural lines that form the layout of the room.

The advantage of this architecture when dealing with non-central panoramas resides in how the network extracts the ceiling-wall and floor-wall intersections: column by column. Due to the bi-directional LSTMs, each column of the image is treated separately in order to recover the structural lines of the room. In our case, where non-central panoramas are used, this property is very interesting. As presented in Section 3, for each optical center, there are regions of the image that share the optical center. In particular, each column of the image is locally central, allowing the network to work in a central projection system for each separate column. Thus, this architecture proposed for central projection systems fits perfectly and is very suitable to extract the structural lines of a room from a non-central panorama.

HorizonNet imposes some restrictions required to adapt it for non-central panoramas. The main restriction is that the image must be oriented with the vertical direction. It means, that the wall-wall intersections form a straight vertical line in the image. Assuming this restriction, non-central panoramas must be acquired with the revolution axis of the system aligned with the gravity direction. This configuration introduces some disadvantages, since the depth and direction of lines parallel to the axis, the wall-wall intersection in this case, cannot be directly estimated (are one of the degenerated cases mentioned in Section 3.1). However, since we know the gravity direction and that the structural lines will be perpendicular to it, we can turn the disadvantage into advantage, exploiting this constrait in the geometrical processisng to estimate the 3D lines that form the layout.

The original network architecture is trained in PanoContext [29] and Stanford 2D-3D [39]. These datasets are formed by equirectangular panoramas obtained from indoor environments. In our proposal, we start with the network trained on these datasets, since the distortion of equirectangular panoramas and non-central panoramas are similar. Afterwards we add a fine tuning to learn the particular distortion of the non-central panorama. For that purpose, we train the network, starting with the weights presented in [9], with a dataset formed of non-central panoramas and 3D information of the environment. However, since non-central projection systems are little used, there is no public dataset available. To solve this problem, we have generated and used a dataset of non-central panoramas from synthetic environments to fill this gap in the resources (more details in Section 6.1). Once the network has been fine-tuned, it has learned the subtle different distortion of the non-central panoramas, providing more accurate information of the boundaries of the structural lines of the different environments.

5. Geometrical processing

The next step in our proposal is to take advantage of the geometrical properties of the non-central panorama in order to recover the 3D layout and the scale of the environment. To do that, we propose a geometrical pipeline, which includes different linear solvers, that takes as input the information provided by the network and outputs the 3D corners of the room. In this section, we present the geometrical problem, defined as a plane extraction problem, and we provide two different solutions to jointly obtain the whole layout of a room under different world assumptions. One of the solutions is for Altanta environments, which are more general and challenging. The second solution is for Manhattan environments, which can be seen as a special case of Atlanta, where we have more geometric restrictions. After these solutions,

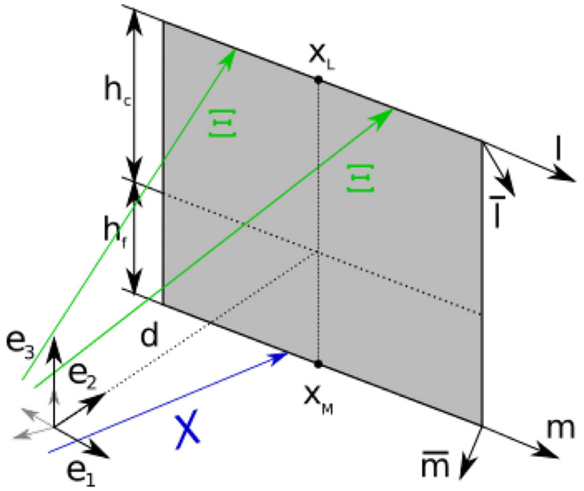


Fig. 5. Rays and wall parameter definition. The wall reference system is defined as $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$; $\boldsymbol{\Xi}$ and $\boldsymbol{X}$ are the projecting rays; $(\mathbf{l}, \bar{\mathbf{l}})$ and $(\mathbf{m}, \bar{\mathbf{m}})$ are the ceiling and floor lines that define the wall; $\mathbf{x_L}, \mathbf{x_M}$ are the closest points of the lines to the reference system; h_c, h_f and d are the distance from the reference system to the ceiling, floor and wall planes respectively.

we present our detailed geometrical pipeline, that includes several steps to make more robust our implementation as well as handling occluded walls in the environment.

5.1. Vertical wall extraction

Man-made environments are usually built by vertical walls that intersect the ceiling and the floor in two horizontal parallel straight lines. Thus, we define a wall as a set of two horizontal parallel lines $(\mathbf{L}, \mathbf{M})$ contained in a vertical plane (see Fig. 5). We define the ceiling line $\mathbf{L} = \left(\mathbf{l}^T, \bar{\mathbf{l}}^T\right)^T$ and the floor line $\mathbf{M} = \left(\mathbf{m}^T, \bar{\mathbf{m}}^T\right)^T$ in Plücker coordinates. We also define an orthonormal reference system placed in the origin and oriented with the vertical wall as $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. From the wall definition shown in Fig. 5, the lines direction coincide with the first component of the reference system $\mathbf{l} = \mathbf{m} = \mathbf{e}_1$. To define the momentum vector of the lines with respect to the reference system, we compute the cross product between the closest point of the line to the origin and the line direction as: $\bar{\mathbf{l}} = \mathbf{x_L} \times \mathbf{l}$ and $\bar{\mathbf{m}} = \mathbf{x_M} \times \mathbf{m}$, where $\mathbf{x_L}$ and $\mathbf{x_M}$ are the closest points of the lines to the origin. These points are defined as $\mathbf{x_L} = d\mathbf{e}_2 + h_c\mathbf{e}_3$ and $\mathbf{x_M} = d\mathbf{e}_2 + h_f\mathbf{e}_3$, where h_c, h_f and d refer to the distance from the reference system to the ceiling, floor and wall planes respectively.

$$side(\boldsymbol{\Xi}, \mathbf{L}) = \boldsymbol{\xi}^T\bar{\mathbf{l}} + \bar{\boldsymbol{\xi}}^T\mathbf{l} = \boldsymbol{\xi}^T(h_c\mathbf{e}_2 - d\mathbf{e}_3) + \bar{\boldsymbol{\xi}}^T\mathbf{e}_1 = 0 \tag{4}$$

$$side(\boldsymbol{X}, \mathbf{M}) = \boldsymbol{\chi}^T\bar{\mathbf{m}} + \bar{\boldsymbol{\chi}}^T\mathbf{m} = \boldsymbol{\chi}^T\left(h_f\mathbf{e}_2 - d\mathbf{e}_3\right) + \bar{\boldsymbol{\chi}}^T\mathbf{e}_1 = 0 \tag{5}$$

To compute the lines that define a wall from an image, we need the projecting rays of these lines. In our proposal, the neural network provides the pixel information of the boundaries of the projection of structural lines in the environment. From this pixel information, we can compute the projecting rays to the ceiling $\boldsymbol{\Xi} = \left(\boldsymbol{\xi}^T, \bar{\boldsymbol{\xi}}^T\right)^T$ and floor $\boldsymbol{X} = \left(\boldsymbol{\chi}^T, \bar{\boldsymbol{\chi}}^T\right)^T$ lines from the backprojection model seen in Section 3. Known the projecting rays, we aim to obtain the 3D lines that define each wall. The relation among the projecting rays and the wall lines is given by their intersection, defined with the side operator in Plücker coordinates (see Section 3.1). From our definition of the wall, Eqs. (4) and (5) define the intersection of the projecting rays and the walls of the layout. Trying to solve directly this ecuations may be difficult since, in general, is a non-linear problem. Instead, we propose a DLT-like [40] approach where we compute the solution as a linear problem.

In a first approach, we aim to extract each wall of the layout independently. Let the main direction of a wall be horizontal and described by the vector $\mathbf{u} = (u_x, u_y)^T$ such that $\mathbf{l} = \mathbf{m} = (u_x, u_y, 0)^T$. Then, the orthonormal reference system oriented with the wall can be re-defined as: $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\} = \{(u_x, u_y, 0)^T, (-u_y, u_x, 0)^T, (0, 0, 1)^T\}$. From this parameterization, Eqs. (4) and (5) can be writen as:

$$\begin{aligned} \bar{\xi}_1 u_x + \bar{\xi}_2 u_y + (\xi_2 u_x - \xi_1 u_y)h_c - d\xi_3 &= 0; \\ \bar{\chi}_1 u_x + \bar{\chi}_2 u_y + (\chi_2 u_x - \chi_1 u_y)h_f - d\chi_3 &= 0 \end{aligned} \tag{6}$$

These equations are non-linear since $\mathbf{u}$, h_c and h_f are coupled. At this point, we define the new variables $\mathbf{v} = h_c\mathbf{u}$ and $\mathbf{w} = h_f\mathbf{u}$. From this new variable definition, Eq. (6) become linear, obtaining the following equations:

$$\begin{aligned} \bar{\xi}_1 u_x + \bar{\xi}_2 u_y - \xi_1 v_y + \xi_2 v_x - d\xi_3 &= 0; \\ \bar{\chi}_1 u_x - \bar{\chi}_2 u_y - \chi_1 w_y + \chi_2 w_x - d\chi_3 &= 0 \end{aligned} \tag{7}$$

Now, we can build a linear system $\mathrm{A}\mathscr{W} = 0$, where the matrix A is full-filed with relations (7) and $\mathscr{W} = (\mathbf{u}^T, \mathbf{v}^T, \mathbf{w}^T, d)^T$ is the unknown wall homogeneous vector. Notice that $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ are independent variables which can be non-parallel. Since we have defined these vectors as proportional, to impose the parallelism we compute the null space of the system with a Singular Value Decomposition (SVD), obtaining a parametric solution which is a linear combination of singular vectors parameterized with λ_i. Notice that, two horizontal lines contained in a vertical plane have 4 dregrees of freedom. At this point we have two options to solve the problem. In one hand, a minimal solution would require 2 projecting rays for each line of the wall, describing the null space with three singular vectors and two parameters λ_1 and λ_2, such as $\mathscr{W} = \mathscr{W}_0 + \lambda_1\mathscr{W}_1 + \lambda_2\mathscr{W}_2$. By solving a system of two quadratic equations with action matrices or as a polynomial eigenvalue vector [41], we obtain a set of 4 different solutions which should be discriminated. On the other hand, since the network provides enought robust information of the structural lines, we propose to solve the over-determined case, taking at least 3 rays for each line of the wall.

$$\begin{aligned} \lambda(\mathbf{v_1} - h_c\mathbf{u_1}) &= h_c\mathbf{u_0} - \mathbf{v_0}; \\ \lambda\left(\mathbf{w_1} - h_f\mathbf{u_1}\right) &= h_f\mathbf{u_0} - \mathbf{w_0} \end{aligned} \tag{8}$$

In this over-determined case, the null space is described by a linear combination involving only one parameter λ, such as $\mathscr{W} = \mathscr{W}_0 + \lambda\mathscr{W}_1$. Imposing the parallelism restriction for $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ as shown in Eq. (8), we can derive two uncoupled quadratic equations for λ as:

$$\begin{aligned} &\left(u_{y0}u_{x1} - u_{x0}u_{y1}\right)\lambda^2 + \\ &\left(u_{x0}v_{y1} + v_{x0}u_{y1} - u_{y0}v_{x1} - v_{y0}u_{x1}\right)\lambda + \\ &\left(v_{y0}v_{x1} - v_{x0}v_{y1}\right) = 0 \end{aligned} \tag{9}$$

$$\begin{aligned} &\left(u_{y0}u_{x1} - u_{x0}u_{y1}\right)\lambda^2 + \\ &\left(u_{x0}v_{y1} + w_{x0}u_{y1} - u_{y0}w_{x1} - w_{y0}u_{x1}\right)\lambda + \\ &\left(w_{y0}v_{x1} - w_{x0}w_{y1}\right) = 0 \end{aligned} \tag{10}$$

Computing the solution for λ in each equation, we get four solutions. However, the solutions from Eqs. (9) and (10) are paired, which means that efectively we have only two different solutions for λ. The global orientation prior allows to easily discriminate which of the solutions is the correct one. Computing $\mathscr{W}$ for each

λ and extracting the ceiling h_c and floor h_f plane distances, we observe that only one of the solutions sets $h_c > h_f$. Taking the correct value of λ, we have defined the wall direction $\mathbf{u}$, the ceiling h_c, floor h_f and wall d planes distance to the acquisition system as well as the Plücker coordinates of the ceiling and floor lines that define the wall.

5.2. Manhattan layout solver

Notice that in a Manhattan world assumption, there is a set of walls sharing the wall direction $\mathbf{u} = (u_x, u_y)^T$ and the complementary set of walls share the orthogonal direction $\mathbf{u}_\perp = (-u_y, u_x)^T$. Since we assume that the rooms have single ceiling and floor planes, all the walls share the ceiling h_c and floor h_f heights. Defining the projecting rays of the walls with direction $\mathbf{u}_\perp$ as $(\boldsymbol{Z}, \boldsymbol{\Psi})$, we can redefine the Eq. (7) for this set of walls as:

$$\begin{aligned} \overline{\zeta}_2 u_x - \overline{\zeta}_1 u_y - \zeta_2 v_y - \zeta_1 v_x - d_i \zeta_3 &= 0; \\ \overline{\psi}_2 u_x - \overline{\psi}_1 u_y - \psi_2 w_y - \psi_1 w_x - d_i \psi_3 &= 0 \end{aligned} \tag{11}$$

Then, we can extend the DLT-like approach to fit all the set of walls computing the null space of $\mathrm{A}\mathscr{L}_M = 0$, where $\mathscr{L}_M = (\mathbf{u}^T, \mathbf{v}^T, \mathbf{w}^T, d_1, \cdots, d_N)^T$, where N is the number of walls, and the matrix A is full-filed with relations (7) for a set of walls and (11) for the other. This approach allows to jointly obtain the main directions in the Manhattan world assumption, the height of the room and the walls locations.

5.3. Atlanta layout solver

In the case of Atlanta world assumption, each wall of the room could have a different horizontal direction, therefore we must find a new approach to obtain the whole layout. Notice that from the previously proposed solver, we can extract each wall independently. However, with this approach we do not impose that the walls share the ceiling and floor heights. Nevertheless, if the direction of each wall is known (e.g. extracting each wall independently), we can derive a new linear solution for the whole layout.

$$\begin{aligned} \overline{\xi_1}' + h_c \xi_2' - d\xi_3' &= 0; \\ \overline{\chi_1}' + h_f \chi_2' - d\chi_3' &= 0 \end{aligned} \tag{12}$$

Making a first independent wall extraction, we obtain the direction for each wall. Changing the refence system of the projecting rays from the acquisition system to each wall local reference system, Eqs. (4) and (5) become the linear expresions (12), where $\boldsymbol{\Xi}'$ and $\boldsymbol{X}'$ are the projecting rays in each wall reference system. Then we can solve the null space of the linear system $\mathrm{A}\mathscr{L}_A = 0$, where A is a matrix full-filed with Eq. (12) and $\mathscr{L}_A = (1, h_c, h_f, d_1, \cdots, d_N)^T$, where N is the number of walls in the environment. In this approach, once known the walls directions, we can simultaneously compute the room height and the walls location.

5.4. Detailed geometric pipeline

In order to improve the robustness of our method, we propose a new full geometric pipeline that includes the two new solvers presented before (see Fig. 6). This pipeline takes as input the information provided by the network and gives as output the 3D lines and corners that form the layout. This geometric pipeline is divided in two branches, one for Manhattan world assumption and other for Atlanta world. This is due to the different management of the occlusions in each world assumption.

The geometric pipeline starts with a RANSAC algorithm to filter possible spurious data. Here could rise a question: What is the advantage of using these structural deep learning based edges and corners over classic Canny edges if we still have to use a RANSAC approach? The main advantage is the huge reduction of the number of required hypotheses. Consider the number of hypotheses in a RANSAC approach n_{hyp} which is typically estimated by $n_{hyp} = \frac{\log(1-P)}{\log(1-(1-\epsilon)^k)}$, where P is the probability of not failing in the random search, ϵ is the rate of outliers and k the number of elements defining the hypothesis. Assuming that we want to extract a wall, with a probability of $P = 99.99\%$ of not failing, we need two lines, defined by 3 points each. We assume a rate of outliers of $\epsilon = 20\%$ in the input data. With our method, we have well defined which data belong to the ceiling line and which to the floor line and the data of each wall separately. So, we assume that our $\epsilon = 20\%$ and that we only need $k = 3$ samples to define the two lines, since data is defined by column and we can take one value for each line in each column. This computation leaves that our implementation needs $n_{hyp} = 12, 84$ hypotheses to define the best wall that fits the data. With state-of-the-art approaches as [18,35], where an edge detector is used, as Canny, the data is not as well defined and structured. Thus, the probability of a sample been an inlier is reduced by half, since it can be part of the ceiling or the floor. This also means that we need the double of samples, since they are not paired. This assumptions lead to a number of outliers of $\epsilon = 60\%$ and a number of samples $k = 6$, obtaining $n_{hyp} = 2244$ hypotheses per wall. This difference in the number of hypotheses is reduced by the neural network. Thus, with the solution for a single wall presented in Section 5.1 as hypothesis in the RANSAC, we get the 3D lines that better fit the information provided by the network. After this step, the pipeline branches depending on the world assumption.

Assuming a Manhattan world. We cluster the extracted walls into two classes corresponding to the main perpendicular directions. In this clustering, we label each wall with the index of one of the clusters corresponding to different perpendicular Manhattan directions. In a second step, we manage the occlusions in the Manhattan environment. Since a Manhattan world only have two main directions, these must be alternating in consecutive walls. If two consecutive walls have the same direction means that an occluded wall is between them. In this case, we add a perpendicular wall between the occluded and occluder walls to keep the alternation in the walls direction. Finally, once we know the number of walls and their Manhattan direction label (defined as $\mathbf{u}$ and $\mathbf{u}_\perp$ in Section 5.2) we apply the Manhattan layout solver. This solver will provide the walls direction that better fit the whole environment as well as the height of the room and the walls location. Once obtained the lines that define the walls, we can obtain the 3D corners of the room computing the intersection of these lines, which is easy since ceiling lines are co-planar, as well as floor lines.

Assuming an Atlanta world. We do not know the number of dominant directions in an Atlanta environment. Besides, we cannot find an occlusion looking at the walls direction, since two consecutive walls may have a similar orientation. Thus, Atlanta viroments must be tackled in a different way than the Manhattan ones.

Since Atlanta environments do not have a defined number of main directions, we cannot cluster the walls extracted in the RANSAC by their direction. By contrast, we consider this first wall direction estimation good enough and use it as initial value in our pipeline. Then, assuming that the walls direction is known, we can apply the solution for Atlanta environments presented in Section 5.3, where we jointly obtain the room height and the walls locations.

Once defined the lines that form the walls of the environment, we compute the 3D corners. Notice that if we compute the corners as lines intersection, we may make impossible layouts if there is

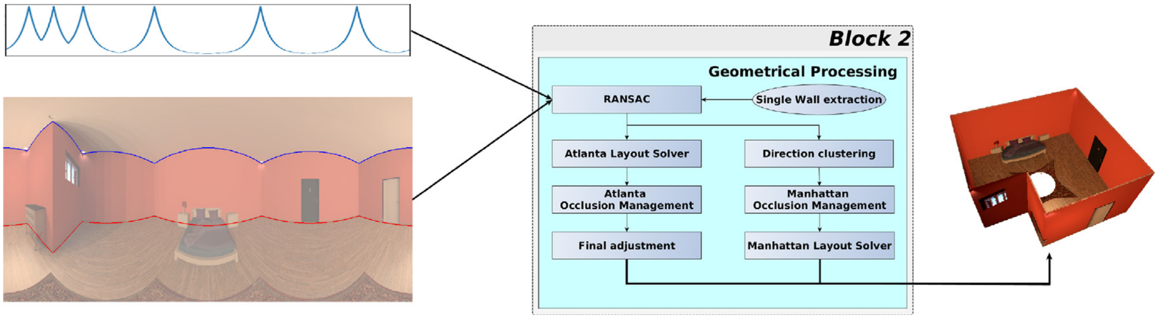


Fig. 6. Geometrical pipeline. The input is the pixel information provided by the network. First, a Ransac makes a first wall direction estimation. Then the pipeline branches. For Manhattan world assumption, we cluster the walls direction, then handle the possible occlusions and compute the direction labels of the walls. Finally we compute the Manhattan layout with the proposed solver. For Atlanta world assumption, once defined the walls directions, we implement the proposed layout solver. Then we search for occlusions. As last step, a final adjustment of the corners is made to obtain the final 3D layout.

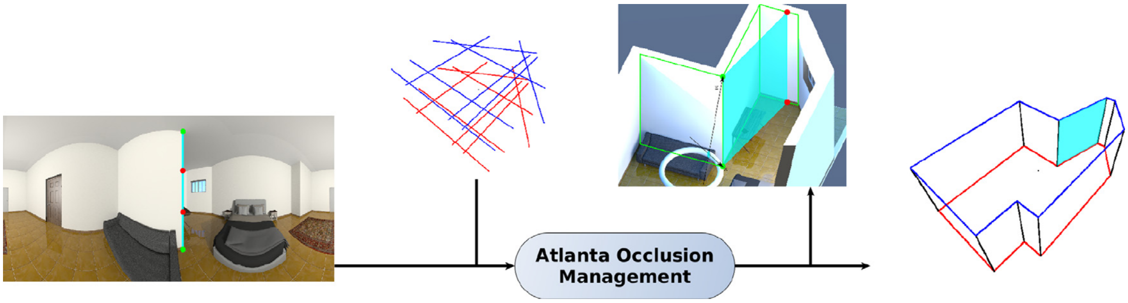


Fig. 7. Atlanta occlusion management. We generate a new wall (in blue) which is co-planar with the projecting rays Ξ and $\boldsymbol{X}$ correspondng to the visible corner (dots in green). This new wall defines two new corners (dots in red) in a partialy occluded wall. The green wireframe is the ground truth layout.

an occlusion in the environment. To avoid this problem, we compute each corner as the point of the computed 3D line that minimises the Euclidean metric distance in $\mathbb{E}^3$ [35] between the 3D line and the projecting ray corresponding to the corner provided by the network. Each projecting ray crosses with two 3D lines corresponding to two consecutive walls where two different cases may appear. When the computed 3D corners coincide in a single point, no occluded wall is detected and this point is a corner of the environment. By contrast, if the computed corners are different 3D points (i.e. the distance between them is higher than a threshold), we have found an occluded wall between the two 3D lines and we insert a wall in the layout model (see Fig. 7). Since in an Atlanta world assumption there is no restriction about the walls direction, we assume that the occluded wall and the projecting ray of the corner lie in the same plane.

Finally, we make a final adjustment where we fine tune the 3D lines direction and position. We minimise the least-square reprojecting error of the computed 3D lines with the pixel coordinates of the boundaries [35] provided by the network. This final adjustment refines the position of the 3D corners. The movement of the corners out of the plane defined by the projecting rays of the corners is penalized. This extra soft-restriction allows managing the infinite posible solutions caused by occlusions during the optimization step.

6. Experiments

The pipeline of our proposal is divided into two main blocks: a neural network that extracts the boundaries of the structural lines of an indoor environment from a single panorama and a geometrical processing that takes as input the output of the network and recovers the 3D scaled layout. In order to evaluate our proposal, we have performed a set of experiments. We independently evaluate the performance of both main blocks: the proposed neural network and geometric pipeline. Additionally, we make a comparison with state of the art methods for line estimation in non-central panoramas and for layout recovery from single panoramas. Before describing the experiments, we present the image dataset used to train the neural network and perform the experiments.

6.1. Non-central panorama dataset

Currently, we can find a great amount of image datasets, from perspective images [42] to omnidirectional panoramas [39,43]. However, non-central projection systems have never been used with deep learning before, thus there is not a dataset of this kind of images. This is a big problem in our case, since we need a large amount of images to train a deep learning architecture. So, we present a new dataset obtained with our synthetic generator of realistic non-central panoramas. It includes semantic, depth and 3D information of the environment.

We generate random layouts, from 4 to 14 walls, in a Manhattan world assumption. Then, with a probability set by the user, corners of these layouts are clipped and substituted by oblique walls to generate Atlanta world layouts. Once obtained the structure of the room, we compute the free space to place objects in it. We have two kinds of objects: those that are placed next to a wall in a fixed orientation (beds, desks, wardrobes, TVs) and those that are placed in the middle of the room at any orientation (chairs, sofas, carpets). These objects are taken from different pools where one is chosen randomly and placed in the room if has a free space in it. Additionally, the ceiling, floor and walls materials and colour are taken from different pools and chosen randomly for each new environment. After the virtual environment is generated, we set the illumination conditions. We have different pre-defined ambient illuminations and we also randomly place spot lights to give the environment a more realistic view.

Once defined the virtual enviroment, we use POV-Ray[1] to render RGB and semantic images and MegaPOV[2] for depth maps. These images are generated by a ray tracing method, which can follow ad-hoc programmable camera projection models. In our case, we use the projection model presented in Section 3 for non-central circular panoramas. The center of the acquisition system is placed in a random position for each room generated, obtaining a greater variability in the walls distortion along the dataset. This allows to generate a set of images, from different rooms and in different positions, not only in the center of the room as many existing datasets.

For this work, we have generated a dataset of non-central panoramas to fine-tune the deep learning architecture presented in Section 4.1. The dataset[3] is formed by more than **2600** images, taken from different positions inside the environments, from around **650** different rooms, from 4 to 14 walls, combining Manhattan and Atlanta environments. We propose a division of the dataset in three blocks: training set, formed by 1677 panoramas; validation set, formed by 399 panoramas; and test set, formed by 499 panoramas. Each set includes Atlanta and Manhattan environments of different number of walls, and we make sure that there are no equal layouts in different sets.

6.2. Ablation study: non-central horizonnet

In this work, we have adapted an existing neural network to work with non-central panoramas, fine-tuning it with the dataset proposed in the previous section. We take the weights for the network that minimise the validation error during the training to perform the different experiments.

We evaluate and compare different state-of-the-art networks for layout extraction in order to verify our selection. The different networks have been evaluated before and after a fine tuning with our proposed dataset of non-central panoramas. We have considered HorizonNet [9], Corners for Layouts [30] and LayoutNetv2 [1].

To evaluate the performance of boundary and corner extractors, we use similar metrics as in [1,30]. We compare the probability maps of the output of the network with their respective labels (each network provides different output resolutions and different probability maps). We try to make the comparison of methods as fair as posible, evaluating the outputs in a similar way and with their respective parameters for the label generation. In order to include HorizonNet in this comparison, we generate the probability map of edges and corners from the output of the network (which are 1D arrays with pixel information) in a similar way than [1,30]. The metrics defined for the evaluation are: Precision (P), Recall (R), Accuracy (Acc) and Intersection over Union (IoU). In Table 1 we present the results of this comparison.

Table 1
Evaluation and comparison of different state-of-the-art methods for layout recovery. We compare HorizonNet [9] (HN), Corners for layouts [30] (CFL) and LayoutNet v2 [1] (LNv2) with the weights provided in their works (Base Line) and after a fine tuning with non-central panoramas (Fine Tuning). The metrics in bold represent the best result (higher is better). All networks are tested in the test partition of the dataset proposed in Section 6.1.

		Base Line			Fine Tuning		
		HN [9]	CFL [30]	LNv2 [1]	HN	CFL	LNv2
Corners	P	**0.379**	0.298	0.352	**0.806**	0.644	0.457
	Acc	**0.995**	0.994	0.985	**0.998**	0.997	0.984
	R	0.204	**0.236**	0.160	0.792	0.543	**0.817**
	IoU	**0.154**	0.150	0.133	**0.643**	0.418	0.400
Edges	P	0.068	**0.085**	0.064	**0.476**	0.231	0.115
	Acc	**0.983**	**0.983**	0.945	**0.995**	0.991	0.951
	R	0.153	**0.202**	0.084	**0.544**	0.373	0.158
	IoU	0.047	**0.060**	0.036	**0.382**	0.166	0.068

6.3. Ablation study: geometric solvers

In this section, we evaluate the proposed solvers presented in Section 5. To do so, we use the ground truth information of the test partition of the dataset to evaluate the sensitivity to noise of the geometric solvers for line extraction. We compare our results with the state of the art method for line extraction in non-central panoramas [18]. Since we are focusing on the geometric approach, we omite the environments with occlusions.

As input information, we use the projecting rays of the boundaries of the structural lines of the indoor environment, taken with sub-pixel accuracy from the ground truth information of the dataset. To evaluate the sensitivity to noise of the solvers, we add increasing Gaussian noise to the ground truth projecting rays at sub-pixel level. For the evaluation of our Manhattan layout solver, we use the walls direction to label the walls in the two main Manhattan directions. For the Atlanta layout solver, we need these wall directions to compute the projecting rays in the wall reference system. The method proposed in [18] compute the lines which are parallel to a known plane, which in our case is the horizontal plane.

To make the evaluation and comparison, we use the same metrics defined in [18]. Computed a 3D line $\mathbf{L} = (\mathbf{l}, \bar{\mathbf{l}})^T$, we compute the direction error as: $\epsilon_{dir} = \arccos(\mathbf{l}^T \cdot \mathbf{l}_{gt})$, measured in degrees. We also compute the depth error of the line as: $\epsilon_{depth} = |||\bar{\mathbf{l}}|| - ||\bar{\mathbf{l}}_{gt}|||$, measured in meters. Additionally, we use a common metric in layout recovery works, the corner error (CE). We compute the corners of the layout as line intersections and compute the L2 distance to the ground truth corners to define the metric.

The evaluation and comparison of our methods with the proposed in [18] is shown in Fig. 8. In blue is shown our Manhattan solver, in green is shown our Atlanta solver while in red is shown the method presented in [18].

6.4. Full pipeline validation

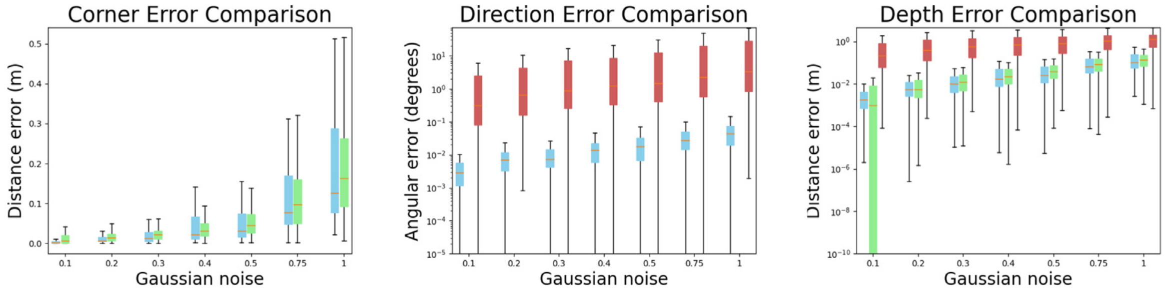
In this section, we analyse the performance of the geometric pipeline proposed in Section 5.4 against the use of only the geometric solvers from Sections 5.2 and 5.3. The question is: if we have two geometric solvers that obtains the whole layout, why we need a more complex geometric pipeline? The answer comes in two parts. First, the geometrical solvers are not able to handle possible occlusion in the environment. Second, we have observed that the input information to the geometric block of our full pipeline (Fig. 3) can be noisy and with spurious data. As seen in previous Section 6.3, the geometrical solvers work well with refined data, however they may lead to impossible layouts if the input is too noisy or if the fitting is corrupted by spurious information.

We test how much the performance of our method improves with and without the proposed geometric pipeline. For that purpose, we compute the corners of the test-set layouts of our dataset in two different cases. In the first case, we use as input of our geometric block the labels of the network ('Network labels' in Table 2), that means, the best information that the network would be able to provide. On a second case, we use as input of the geometric block the predictions provided by the network ('Network predictions' in Table 2), which are noisier than the labels. The metrics used to make the comparison are: Corner Error (CE), defined as the L2 distance between the computed corners and the ground truth;

[1] The Persistence of Vision Raytracer http://www.povray.org/, website visited in 2020

[2] MegaPOV http://megapov.inetart.net/, website visited in 2020

[3] The dataset will be available in github.com/jesusbermudezcameo/NonCentralIndoorDataset



(a) (b) (c)

Fig. 8. Experimental results to evaluate and compare our proposed Manhattan and Atlanta layout solvers against a state-of-the-art method for line extraction [18]. In blue are the Manhattan solver results. In green are the Atlanta solver results. In red are the state of the art method results. (a) shows the Corner error, in meters, of our proposed solvers. (b) shows the Direction error of our solvers compared to [18], in degrees and logarithmic scale. Notice that the direction error of the Atlanta solver is zero since we provide the lines direction as prior. (c) shows the Depth error of our proposed solvers compared to [18], in meters and logarithmic scale.

Table 2
Comparison of layout recovery with only the solvers or the whole proposed geometric pipeline.

		Network labels		Network predictions	
	World assumption	CE (m)	IoU (%)	CE (m)	IoU (%)
Solvers	Manhattan	0.0297	98.0036	0.5569	81.7364
	Atlanta	0.8575	57.3289	1.2860	42.6701
Pipeline	Manhattan	0.0218	98.4753	0.2109	86.8104
	Atlanta	0.1391	92.5012	0.4811	76.0218

and the Intersection over Union (IoU) of the volumes of the computed layout and the ground truth. In this experiment, 'Solvers' refer to the use of only the geometric solvers presented in Section 5. We use the wall extractor (see Section 5.1) to compute the wall labels in the Manhattan case and the walls' direction in the Atlanta case, and then the layout solver to obtain the corners of the room. 'Pipeline' refers to the use of the whole geometric pipeline proposed in Section 5.4. In order to make a more fair comparison, environments with occlusions have not been taken into account to compute the metrics (if so, the performance difference would be much greater). Table 2 shows the results of this experiment.

6.5. State of the art comparison

We have performed three different tests to compare our work with the state-of-the-art methods for layout recovery from a single panorama. In a first experiment, we have fine-tuned HorizonNet [9] with equirectangular panoramas of the same virtual environments of our dataset. On a second experiment, we compare the results of different state-of-the-art works with ours in Manhattan and Atlanta environments. Finally, on the third experiment we compare our proposal with state-of-the-art methods on the Stanford 2D-3DS [39] dataset, labelled as cuboid rooms. In the Fig. 10 we present qualitative results of our proposal in the two datasets evaluated.

For our first experiment, we have generated a dual test-set, with equirectangular and non-central panoramas taken in the same position in the same Manhattan environments. We use this set-up to compare two different methods and two different acquisition systems in the same test-set. Then, we recover the layout with HorizonNet, from central panoramas, and with our method, from non-central panoramas, for Manhattan environments. Notice that HorizonNet does not obtain the scale of the layout. Instead, it assumes a camera height and computes the 3D corners with this extra measurement. Our method does extract the scale of the environment, so this measurement is computed and not given. To take this into account, we compare the metrics in two different cases. In a first case (No-scale), we normalize the layouts, predicted and ground truth, with the camera height. In the second case (Metric), we use the real camera height to compute the scale in the prediction from HorizonNet. In the case of our proposal, the scale is computed directly from the image, without any extra measurement. Table 3 shows the results of this comparison. The metrics used are the Corner Error (CE) and the Intersection over Union (IoU) defined before.

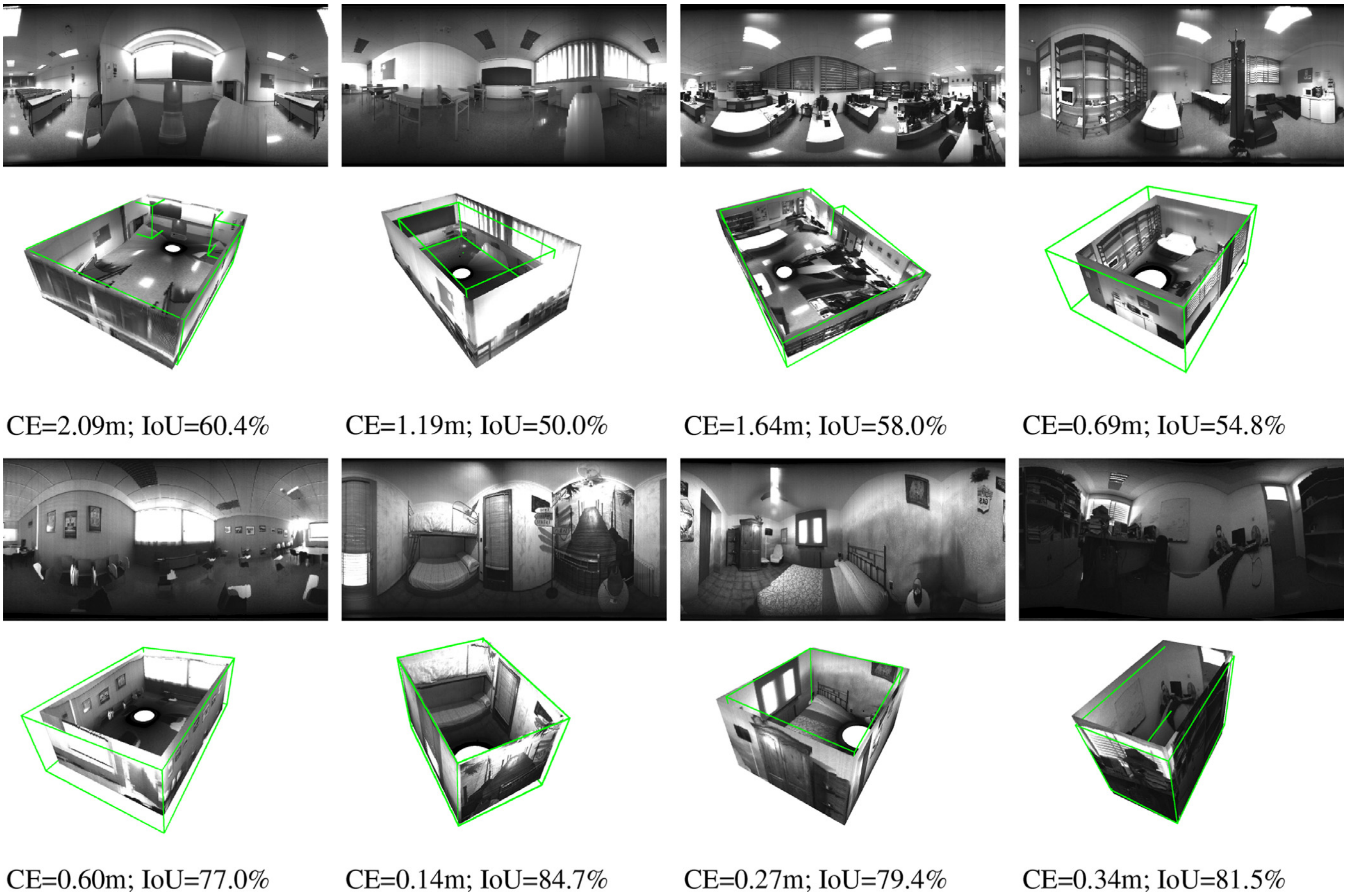
Table 3
Comparison of methods: fine tuned HorizonNet [9] (HorizonNet FT) and our method. Evaluation made in selected Manhattan environments from the test-set of the presented dataset. HorizonNet FT uses the camera height for computing the scale while our proposal does not use extra measurements to compute the 3D layout.

	Scale assumption	CE	IoU (%)
HorizonNet	No-scale	0.3380	82.6742
FT	Metric	0.3504	82.6742
Ours	No-scale	0.2024	87.6208
	Metric	0.2271	87.6208

Table 4
Comparison of different state-of-the-art methods for 3D layout recovery.

	Manhattan World assumption			
	3D IoU (u2s)	3D IoU	CEN	CE
CFL [30]	78.87	-	0.75	-
HorizonNet [9]	82.66	-	**0.69**	-
AtlantaNet [31]	83.94	-	0.71	-
Ours	**93.87**	**86.16**	0.78	**0.223**
			Atlanta World assumption	
HorizonNet [9]	73.53	-	-	-
AtlantaNet [31]	80.01	-	-	-
Ours	**90.46**	**76.02**	**1.5**	**0.481**
	higher is better		*smaller is better*	

On a second experiment, we compare our method with other state-of-the-art methods. This second comparison is not as fair as the previous one since each method has been trained and tested on different datasets, so the results can depend on the dataset used and not only on the method. The metrics used for the comparison are: 3D IoU, which refer to the 3D intersection over union of the predicted layout and the ground truth; 3D IoU(u2s), which refer to the up-to-scale intersection over union of the layout; CEN, which refer to the Corner Error Normalized computed as the L2 distance of the corners divided by the diagonal of the layout's bounding box; CE, which refers to the Corner Error computed as the L2 dis-



CE=2.09m; IoU=60.4% CE=1.19m; IoU=50.0% CE=1.64m; IoU=58.0% CE=0.69m; IoU=54.8%

CE=0.60m; IoU=77.0% CE=0.14m; IoU=84.7% CE=0.27m; IoU=79.4% CE=0.34m; IoU=81.5%

Fig. 9. Qualitative and quantitative demonstration of the proposed scaled layout recovery with real images. As qualitative evaluation, in green is a wire-frame of the room layout.

Table 5
Comparison of different state-of-the-art methods for 3D layout recovery evaluated in the Stanford 2D3DS [39] dataset.

	3D IoU (u2s)	3D IoU	CEN	CE
CFL [30]	65.23	-	1.44	-
HorizonNet [9]	83.51	-	**0.62**	-
LayoutNet v2 [1]	82.66	-	0.66	-
AtlantaNet [31]	83.94	-	0.71	-
Ours	**88.19**	**58.19**	2.02	**0.878**
	higher is better		*smaller is better*	

tance of the corners in meters. Table 4 shows the results of this experiment.

The third proposed experiment aims to compare the performance of different state-of-the-art methods for layout recovery in the same dataset. In this case, we choose Stanford 2D3DS [39], where the layouts are labelled as cuboids. For this experiment, we have generated a set of non-central panoramas from the colour and depth information of the dataset. With the new dataset we make a fine tuning of our network and evaluate our proposal. We have observed that some images present gitches and blank spaces that lead our algorithm to failures where no layout can be recovered (~2.5% of the test-set). We present the results of this experiment in Table 5 considering the cases where the layout can be recovered.

6.6. Qualitative experiments

In this section we present different examples of real non-central panoramas and the scaled layout recovery with our pipeline. The acquisition and anotation of these real examples has been made manually by the authors.

Fig. 9 shows the real non-central panoramas and their corresponding reconstruction using our pipeline. These panoramas have different calibrations and have been taken in different environments and also in different illumination conditions. Besides, due to the acquisition system, these images are grey-scale, which provide less information to the network. To solve this problem, we have trained the network on grey-scale panoramas, creating a second set of weights for this gray-scale version of the network. As a qualitative demonstration, we show a green wireframe that represents the real layout of each environment, measured with a laser meter and reconstructed and aligned with the results. We have computed the same metrics as in other experiments, obtaining an average corner error of CE $= 1.01m$ and an average 3D intersection over union of IoU $= 65.57\%$. We also present the corner error and intersection over union of each image in the Fig. 9.

7. Discussion

In this section we analyse and discuss the results obtained in the previous experiments. Sections 6.2, 6.3 and 6.4 evaluate different parts of our pipeline. In the first one, we have compared three state of the art networks for layout recovery. This experiment supports our initial intuition and we can claim that HorizonNet [9] is our best option among the evaluated networks for boundary extraction. Even though before the fine tuning it is not the best option, the architecture of HorizonNet is really suitable for structural line extraction in non-central panoramas and provides the best performance after fine tuning. In 6.3 we compare

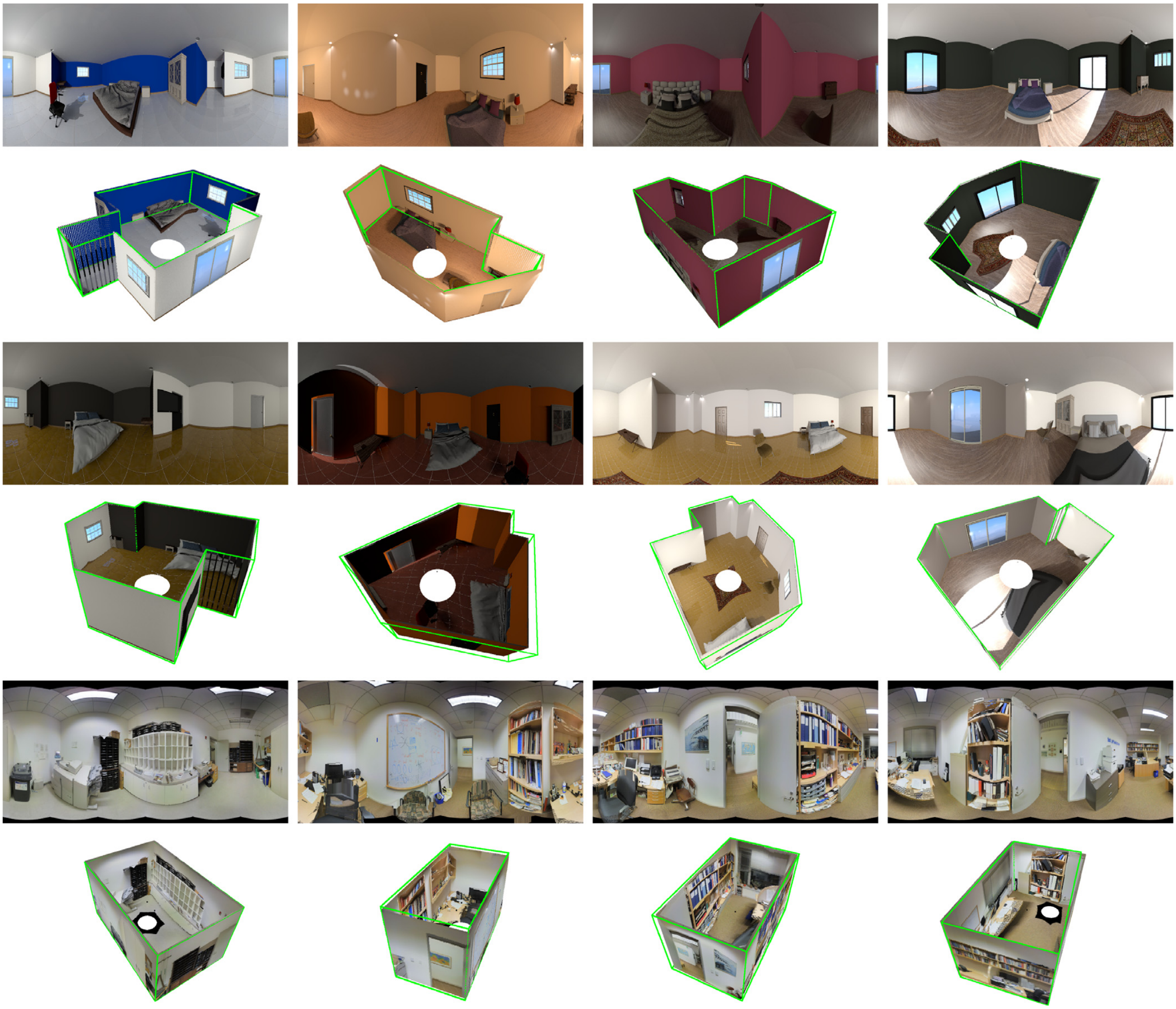


Fig. 10. Qualitative demonstration of the proposed scaled layout recovery with synthetic images (first and second rows) and real images adapted from the Stanford dataset (last row). As qualitative evaluation, in green is a wire-frame of the room layout.

our new layout solvers against a state-of-the-art line extractor for non-central panoramas. We observe that our approach outperforms previous methods for line extraction in both Manhattan and Atlanta world assumptions by a large margin. Finally, the evaluation of our geometric block validates the use of a more complex geometric pipeline. From the results presented in Table 2, we observe that the difference in performance between the solvers and the pipeline using the network labels in Manhattan environments is quite small. However, for Atlanta environments and with the use of the network predictions, the performance of the full geometric pipeline is significantly better.

In Section 6.5 we compare our method with different state of the art implementations for layout recovery. In a first experiment, we make a comparison with HorizonNet. This is the most fair comparison made since we use panoramas from the same virtual environments and same locations on both methods in order to recover their layout. The results show that our proposal outperforms HorizonNet in the two cases of study: with extra measurements for HorizonNet and in up-to-scale reconstructions. On the second experiment, we make a comparison with other state of the art methods. In this experiment, the datasets used for the experiments are different, so the results do not completly show the performance of each method. Nevertheless, making the comparison of the results of each method, our proposal presents a better performance in most of the metrics. Besides, we are able to recover the scale of the environment without extra measurements while other methods need a metric measure to scale the layout (e.g the camera or room height). Furthermore, we want to highlight the results in Atlanta environments. State-of-the-art methods do not refer in their works the management of occlusions in Atlanta environments, although they do manage occlusion in Manhattan world. Our proposal does handle occlusions in Atlanta environments, as well as in Manhattan environments. On the third experiment, we compare different state-of-the-art methods for cuboid layout extraction. Even though our metrics drop significantly against the results shown from our dataset, we still outperform state-of-the-art methods in the up to scale intersection over union. This drop in performance can be explained by the specialization into cuboid layouts of the dataset labelling. Looking for a more general solution may reduce the performance on more specific tasks. Our proposal aims

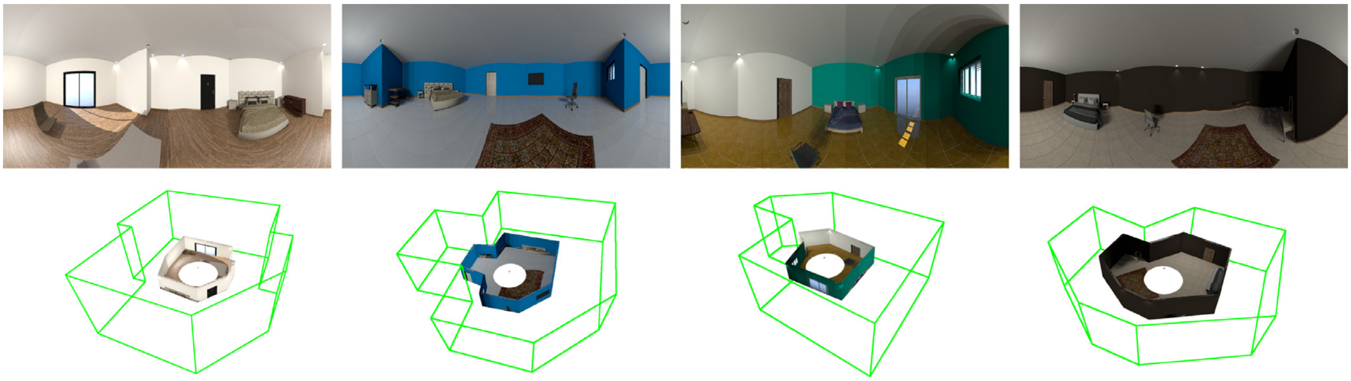


Fig. 11. We present cases where our method fails in the scale recovery.

to obtain the layout from a greater variability of rooms (our synthetic dataset has rooms from 4 to 14 walls) which means that our algorithm is not as particularized to cuboids as other state of the art methods.

As a qualitative demonstration, in Section 6.6 we present some examples of layout recovery from real images, taken by the authors. We observe that the performance varies from environment to environment. We are able to recover the layout of most of the environments and, in several cases, we achieve a good 3D scaled reconstruction of the environment. The performance in the 3D reconstruction is limited due to the noisy output of the network when dealing with these real images. This is the effect of the lack of real images for training the network and the artifacts that these real images present in the acquisition process.

In the Fig. 11 we show cases where the scale recovery of the environment fails. Analysing those cases, we found different possible sources of error. One of these sources is the information provided by the network. When the images present environments which are too cluttered, occluding practically any boundary of the floor-wall intersection, the output of the network is not very accurate estimating the floor boundary. This lead to a worse 3D reconstruction. Besides, the artifacts in real images fool the network making it belive that there are more wall-wall boundaries than there really are. Other source of error is the occlusion management in Atlanta environments. We found out that an incorrect definition of the 3D corners of the room when dealing with an occlusion lead the final adjustment to under-estimate the scale of the room. This seem logical since the closer the 3D points are to the acquisition reference system, the lower the reprojection error will be. A final source of error is the effective baseline of the non-central acquisition system. This problem is also more evident in the real images. With a smaller effective baseline, related with the radius of the non-central panorama, the accuracy to compute the scale of the room is reduced.

8. Conclusions

In this paper we have proposed a new pipeline that completely solves the layout recovery problem from a single image (i.e. reconstructing Manhattan and Atlanta environments with scale). We have presented the first application of non-central panoramas that is comparable with state of the art methods for layout recovery, even improving in some of the metrics. We introduce the first indoor dataset of non-central panoramas automatically generated. This dataset provides a good resource for many researchers to further investigate the geometrical properties of the non-central projection system.

The experiments presented in this paper show that our proposal can achieve great results. However, there is still room to grow when real images enter into action. Since real images are hard to obtain, the current approach cannot handle these non-central panoramas as well as with the synthetically generated. Nevertheless, the results are promising and may encourage the development of commertial devices able to obtain non-central panoramas.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by project RTI2018-096903-B-100 (AEI/ FEDER, UE).

References

[1] C. Zou, J.W. Su, C.H. Peng, A. Colburn, Q. Shan, P. Wonka, H.K. Chu, D. Hoiem, Manhattan room layout reconstruction from a single 360 image: a comparative study of state-of-the-art methods, Int. J. Comput. Vis. 129 (2021) 1410–1431.
[2] M. Naseer, S. Khan, F. Porikli, Indoor scene understanding in 2.5/3d for autonomous agents: a survey, IEEE Access (2018) 1859–1887.
[3] K. Karsch, V. Hedau, D. Forsyth, D. Hoiem, Rendering synthetic objects into legacy photographs, ACM Trans. Gr. 30 (6) (2011) 1–12.
[4] D.F. Fouhey, V. Delaitre, A. Gupta, A.A. Efros, I. Laptev, J. Sivic, People watching: human actions as a cue for single view geometry, Int. J. Comput. Vis. 110 (3) (2014) 259–274.
[5] E.S. dos Reis, L.A. Seewald, et al., Monocular multi-person pose estimation: a survey, Pattern Recognit. 118 (2021).
[6] H. Wei, L. Wang, Understanding of indoor scenes based on projection of spatial rectangles, Pattern Recognit. 81 (2018).
[7] Y. Li, H.-Y. Shum, C.-K. Tang, R. Szeliski, Stereo reconstruction from multiperspective panoramas, Trans. Pattern Anal. Mach. Intell. 26 (1) (2004) 45–62.
[8] M. Menem, T. Pajdla, Constraints on perspective images and circular panoramas, in: British Machine Vision Conference, 2004, pp. 1–10.
[9] C. Sun, C.-W. Hsiao, M. Sun, H.-T. Chen, Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2019, pp. 1047–1056.
[10] B. Berenguel-Baeta, J. Bermudez-Cameo, J.J. Guerrero, Scaled 360 layouts: Revisiting non-central panoramas, in: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2021, pp. 3702–3705.
[11] A. Agrawal, Y. Taguchi, S. Ramalingam, Analytical forward projection for axial non-central dioptric and catadioptric cameras, in: European Conference on Computer Vision, Springer, 2010, pp. 129–143.
[12] G. Lopez-Nicolas, C. Sagues, Unitary torus model for conical mirror based catadioptric system, Comput. Vis. Image Underst. 126 (2014) 67–79.
[13] J. Bermudez-Cameo, G. Lopez-Nicolas, J.J. Guerrero, Fitting line projections in non-central catadioptric cameras with revolution symmetry, Comput. Vis. Image Underst. 167 (2018) 134–152.
[14] A. Agrawal, Y. Taguchi, S. Ramalingam, Beyond alhazen's problem: Analytical projection model for non-central catadioptric cameras with quadric mirrors, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 2993–3000.
[15] R. Gupta, R.I. Hartley, Linear pushbroom cameras, Trans. Pattern Anal. Mach. Intell. 19 (9) (1997) 963–975.

[16] S. Gasparini, V. Caglioti, Line localization from single catadioptric images, Int J Comput Vis 94 (3) (2011) 361–374.
[17] S. Teller, M. Hohmeyer, Determining the lines through four lines, J. Graph. Tools 4 (3) (1999) 11–22.
[18] J. Bermudez-Cameo, C. Demonceaux, G. Lopez-Nicolas, J. Guerrero, Line reconstruction using prior knowledge in single non-central view, in: British Machine Vision Conference, 2016.
[19] J. Bermudez-Cameo, J.P. Barreto, G. Lopez-Nicolas, J.J. Guerrero, Minimal solution for computing pairs of lines in non-central cameras, in: Asian Conference on Computer Vision, Springer, 2014, pp. 585–597.
[20] S.Y. Bao, M. Sun, S. Savarese, Toward coherent object detection and scene layout understanding, Image Vis. Comput. 29 (9) (2011) 569–579.
[21] S. Song, J. Xiao, Deep sliding shapes for amodal 3D object detection in rgb-d images, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 808–816.
[22] Y. Nie, S. Guo, J. Chang, X. Han, J. Huang, S.-M. Hu, J.J. Zhang, Shallow2deep: indoor scene modeling by single image understanding, Pattern Recognit. 103 (2020).
[23] Z. Wang, R. Song, P. Duan, X. Li, Efnet: enhancement-fusion network for semantic segmentation, Pattern Recognit. 118 (2021).
[24] Q. Zhou, X. Wu, S. Zhang, B. Kang, Z. Ge, L.J. Latecki, Contextual ensemble network for semantic segmentation, Pattern Recognit. 122 (2022).
[25] C. Jung, C. Kim, Real-time estimation of 3d scene geometry from a single image, Pattern Recognit. 45 (2012).
[26] K. Fukano, Y. Mochizuki, S. Iizuka, E. Simo-Serra, A. Sugimoto, H. Ishikawa, Room reconstruction from a single spherical image by higher-order energy minimization, in: International Conference on Pattern Recognition, IEEE, 2016, pp. 1768–1773.
[27] S. Rao, V. Kumar, D. Kifer, C.L. Giles, A. Mali, Omnilayout: Room layout reconstruction from indoor spherical panoramas, in: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2021, pp. 3706–3715.
[28] G. Pintore, E. Almansa, M. Agus, E. Gobbetti, Deep3dlayout: 3D reconstruction of an indoor layout from a spherical panoramic image, ACM Trans. Graph. TOG 40 (6) (2021) 1–12.
[29] Y. Zhang, S. Song, P. Tan, J. Xiao, Panocontext: a whole-room 3D context model for panoramic scene understanding, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 668–686.
[30] C. Fernandez-Labrador, J.M. Facil, A. Perez-Yus, C. Demonceaux, J. Civera, J.J. Guerrero, Corners for layout: end-to-end layout recovery from 360 images, Robot. Autom. Lett. 5 (2) (2020) 1255–1262.
[31] G. Pintore, M. Agus, E. Gobbetti, Atlantanet: inferring the 3D indoor layout from a single 360 image beyond the manhattan world assumption, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 432–448.
[32] H.-Y. Shum, R. Szeliski, Stereo reconstruction from multiperspective panoramas, in: Proceedings of the International Conference on Computer Vision, volume 1, IEEE, 1999, pp. 14–21.
[33] H. Bakstein, T. Pajdla, An overview of non-central cameras, Computer vision winter workshop, volume 2, 2001.
[34] L. Perdigoto, H. Araujo, Calibration of mirror position and extrinsic parameters in axial non-central catadioptric systems, Comput. Vis. Image Underst. 117 (8) (2013) 909–921.
[35] J. Bermudez-Cameo, O. Saurer, G. Lopez-Nicolas, J.J. Guerrero, M. Pollefeys, Exploiting line metric reconstruction from non-central circular panoramas, Pattern Recognit. Lett. 94 (2017) 30–37.
[36] H. Pottmann, J. Wallner, Computational Line Geometry, Springer Science & Business Media, 2001.
[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 770–778.
[38] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, Trans. Signal Process. 45 (11) (1997) 2673–2681.
[39] I. Armeni, S. Sax, A.R. Zamir, S. Savarese, Joint 2d-3d-semantic data for indoor scene understanding, arXiv preprint arXiv:1702.01105 (2017).
[40] Y.I. Abdel-Aziz, H. Karara, M. Hauck, Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry, Photogramm. Eng. Remote Sens. 81 (2) (2015) 103–107.
[41] Z. Kukelova, M. Bujnak, T. Pajdla, Polynomial eigenvalue solutions to minimal problems in computer vision, Trans. Pattern Anal. Mach. Intell. 34 (7) (2011) 1381–1393.
[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[43] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, Y. Zhang, Matterport3d: learning from rgb-d data in indoor environments, in: International Conference on 3D Vision, IEEE, 2017, pp. 667–676.

Bruno Berenguel-Baeta graduated in Industrial Engineering from the University of Zaragoza in 2017. Currectly he is a Ph.D. student in the same institution in the Department of Computer Science and Systems Engineering. His research interests are in the area of computer vision and scene understanding, focusing non-conventional cameras.

Jesus Bermudez-Cameo obtained his Ph.D. in 2016 from the University of Zaragoza, and currently he is an Assistant Professor at the Department of Computer Science and Systems Engineering. His research interests are in the area of computer vision and robotics,focusing in omnidirectional cameras and its applications.

Jose J. Guerrero obtained the Ph.D. in 1996 from the University of Zaragoza, and currently he is Full Professor at the Department of Computer Science and Systems Engineering. His research interests are in the area of computer vision, particularly in 3D visual perception, robotics, omnidirectional vision and vision-based navigation.