Egocentric Indoor Localization from Room Layouts and Image Outer Corners

Xiaowei Chen and Guoliang Fan
School of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK, 74078 USA
{xiaowei.chen,guoliang.fan}@okstate.edu

Abstract

Egocentric indoor localization is an important issue for many in-home smart technologies. Room layouts have been used to characterize indoor scene images by a few typical space configurations defined by boundary lines and junctions, which are mostly detectable or inferable by deep learning methods. In this paper, we study camera pose estimation for egocentric indoor localization from room layouts that is cast as a PnL (Perspective-n-Line) problem. Specifically, image outer corners (IOCs), which are the intersecting points between image borders and room layout boundaries, are introduced to improve PnL optimization by involving additional auxiliary lines in an image. This leads to a new PnL-IOC algorithm where 3D correspondence estimation of IOCs are jointly solved with camera pose optimization in the iterative Gauss-Newton algorithm. Experiment results on both simulated and real images show the advantages of PnL-IOC on the accuracy and robustness of camera pose estimation over the existing PnL methods.

1. Introduction

With the recent advancement of wearable technologies, egocentric or first-person vision has become an active topic that has led to many useful tools [10, 28]. Room-level indoor localization is often a fundamental step for an egocentric vision-based in-home assistive tool that can deliver location-aware assistance or support indoor navigation [3, 32, 29, 23]. Most previous works use deep learning methods for indoor localization that involve a large number of labeled or annotated room images. On the other hand, because most indoor structures usually conform to the Manhattan world assumption [8], indoor scene images can be characterized by different room layouts defined by a few boundary lines and junctions (also called inner corners [26]). During the past decade, indoor layout estimation has emerged as an interesting and fast evolving topic with many deep learning-based methods proposed that show great promise and potential [21, 39, 33].

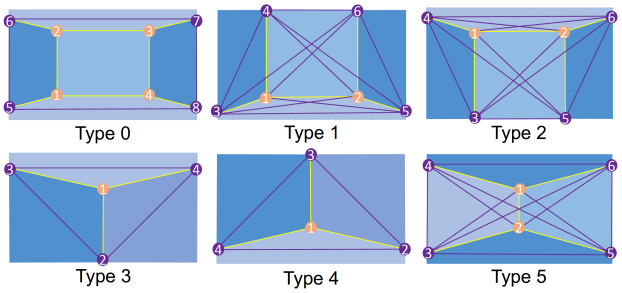

Figure 1. The six room layouts under study where the inner corners and IOCs are colored in yellow and purple, respectively.

In this work, we are interested in indoor localization from room layouts via camera pose estimation that involves n correspondences between 3D reference features and their 2D projections. When features are points, this is called the perspective-n-points (PnP) problem [22, 19, 24, 15]. When features are lines, it becomes the PnL (Perspective-n-Line) problem [38]. Given a room layout shown in Figure 1, there are a few boundary lines which intersect at inner corners, which are well defined in a layout map and whose 3D correspondences in the world frame may be available with some *a priori* condition (e.g., the room dimension). Therefore, indoor localization and camera pose estimation from room layouts can be converted to a PnL problem [38].

Of the 11 types of room layouts [21, 25], we focus on 6 of them (Figure 1) with at least three lines (the minimum case of PnL). We propose a new PnL method by introducing image outer corners (IOCs), the intersecting points between image borders and layout boundaries, which are used to create a preferable condition for the PnL solution by adding more line correspondences. Moreover, 3D correspondence estimation of IOCs is built in the PnL solution, leading to the proposed PnL-IOC method that has two advantages over existing ones: (1) It improves accuracy of camera pose estimation through IOCs whose 3D correspondences are initialized by solving a linear system and further optimized along with camera pose via the iterative Gaussian-Newton algorithm. (2) It achieves stable and robust results under different noise levels at both the inner and outer corners.

2. Related work

We briefly review related work in three areas: *room layout estimation*, *PnL/P3L*, and *recent PnL development*.

Since the introduction of spatial layout estimation by [18], room layout estimation has remained an active research topic. The early works [18, 34, 11, 36, 12] solved the room layout estimation problem using geometry-based methods, which took advantage of vanishing points estimation. With the development of deep learning, deep learning-based methods, which are robust and accurate when handling a wide range of highly challenging scenes, have been proposed [39, 21, 25]. Furthermore, some high quality datasets [6, 9, 41] published recently make deep learning methods more feasible and accurate. A detailed review for layout estimation can be found in [39].

In the PnL problem, at least three 2D/3D line correspondences are needed because there are 6 DoFs for a 3D camera pose and each line correspondence offers two constraints [17]. When $n = 3$, it is the P3L (Perspective-three-Line) problem that plays a fundamental role in dealing with the general PnL problem [38], because the latter is essentially constructed by the former. In [13], one early analytical method was proposed to solve the P3L problem that leads to a closed-form solution by solving an eighth-order polynomial. In [5], an algebraic P3L method was proposed that may not be stable in the presences of noise. In [4], a special case of P3L was addressed where three co-planar lines intersect at a point. In [31], a unique P3L problem was studied where three lines form a Z-shape in space. In [40], a geometric method was proposed by introducing two intermediate frames to simplify the P3L problem formulation. However, a well-known fact about the P3L problem is that the solution is not uniquely determined [5].

Most existing PnL studies focus on the cases where $n > 3$ where there are two kinds, *iterative* and *non-iterative*. The early iterative ones [14, 7, 20] are usually computationally costly and sensitive to initialization, and easily converge to a local minimum [37]. For recent non-iterative ones, several linear formulation based methods were proposed [1, 35, 30] that are sensitive to noise and cannot deal with small line sets ($n < 6$). Some non-iterative PnL methods [2, 27] were developed to deal with small sets that may not be stable due to the underlying linearization scheme. In [40], a non-iterative $O(n)$ solution, named Robust PnL (RPnL), was proposed for the cases of $n \geq 4$. Based on RPnL, the Accurate Subset-based PnL (ASPnL) method was proposed in [37] that is more accurate on small line sets. However, ASPnL cannot properly deal with the case when there are only three orthogonal lines intersecting at one junction point, and it was modified in [38] resulting in the SRPnL method, which can deal with the aforementioned case and deliver high accuracy on small line sets. However, SRPnL may struggle under more lines ($n \geq 8$) and strong noise.

3. Proposed Method

3.1. Problem statement

The PnL problem is illustrated in Figure 2 where the goal is to recover rotation $\mathbf{R}_w^c$ and translation $\mathbf{t}$ of a camera from n known 3D reference lines $L_i = (\mathbf{v}_i^w, P_i^w)$ $(i = 1, 2, ..., n)$ along with their corresponding 2D projections on the image plane denoted as l_i, where $\mathbf{v}_i^w \in \mathbb{R}^3$ is the normalized vector giving the direction of the line and $P_i^w \in \mathbb{R}^3$ is any point on the line in the world coordinate frame. Two intermediate frames are introduced into the reprojection model, the model frame and the new camera frame. The rotation of the model frame with respect to the world frame is $\mathbf{R}_w^m$, and the rotation of the new camera frame with respect to the model frame is $\mathbf{R}_m^n$. The rotation of the camera frame with respect to the new camera frame is $\mathbf{R}_n^c$, where the new camera frame can be obtained by rotating the original camera frame with $\mathbf{R}_w^m$, as $\mathbf{R}_n^c = (\mathbf{R}_w^m)^T$, and similarly $\mathbf{R}_w^c$ denotes the rotation of the camera frame with respect to the world frame. The relationship among those four 3×3 rotation matrices can be defined as follows:

$$\mathbf{R}_w^c = \mathbf{R}_n^c \mathbf{R}_m^n \mathbf{R}_w^m = (\mathbf{R}_w^m)^T \mathbf{R}_m^n \mathbf{R}_w^m. \tag{1}$$

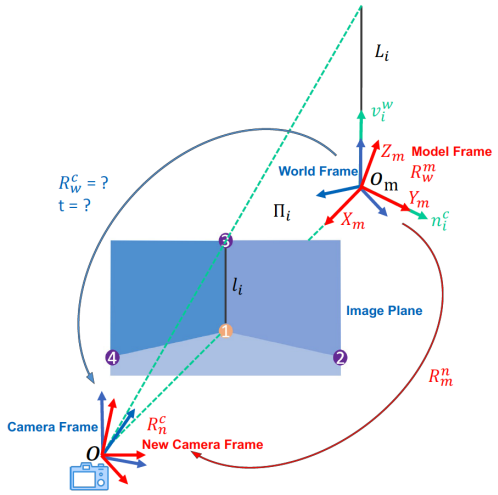

Figure 2. Illustration of the PnL-IOC problem.

Given a 2D line $l_i = (s_i, e_i)$, where s_i and e_i are the endpoints of l_i, its corresponding 3D line L_i and the projection center O can form a projection plane Π_i. The normal of Π_i can be easily achieved using the cross product of s_i and e_i, which can be defined as $\mathbf{n}_i^c$. Suppose P_i^w is any point on L_i, and by using the geometrical constraints [17] that P_i^c, the coordinate of P_i^w in the camera coordinate frame and $P_i^c = \mathbf{R}_w^c P_i^w + \mathbf{t}$, should be perpendicular to the normal $\mathbf{n}_i^c$ of the plane Π_i, there is a constraint

$$(\mathbf{n}_i^c)^T (\mathbf{R}_w^c P_i^w + \mathbf{t}) = 0 \qquad i = 1, 2, ..., n, \tag{2}$$

which leads to an analytic solution of $\mathbf{t}$ [38].

3.2. Determining initial rotation matrix

From all reference lines L_i, line $L_0 = (\mathbf{v}_0^w, P_0^w)$with the longest projection length can be selected, then it is used to calculate the corresponding normal $\mathbf{n}_0^c$. A new intermediate model frame $[O_m - X_m, Y_m, Z_m]$ can be formed from L_0 and $\mathbf{n}_0^c$ [37]. The origin of the model frame is matched with the world frame, and the Y-axis of the model frame is aligned with $\mathbf{n}_0^c$ to form the intermediate rotation matrix $\mathbf{R}_w^m = [X_m, Y_m, Z_m]$. After $\mathbf{R}_w^m$ is determined, the key to calculate $\mathbf{R}_w^c$ is to determine $\mathbf{R}_m^n$ according to Eq. (1), and $\mathbf{R}_m^n$ can be expressed by an Euler Angle as

$$\mathbf{R}_m^n = Rot(Y,\beta)Rot(Z,\gamma)Rot(X,\alpha), \tag{3}$$

in which $Rot(X,\alpha)$, $Rot(Y,\beta)$ and $Rot(Z,\gamma)$ denote rotation around the X-axis, Y-axis, and Z-axis in the model frame, respectively. From the Euler Angle definition, α is the angle between Z-axis and $\mathbf{v}_0^m = \mathbf{R}_w^m \mathbf{v}_0^w$ [37]. Therefore, if the two unknown variables β and γ are determined, the rotation matrix $\mathbf{R}_m^n$ can be obtained, then $\mathbf{R}_w^c$ can be calculated from $\mathbf{R}_m^n$ based on Eq. (1).

For determining $Rot(Z,\gamma)$, another line $L_1 = (\mathbf{v}_1^w, P_1)$ is selected, whose projection line length in 2D image plane is the second longest, then every remaining line L_k together with line L_0 and L_1 forms a 3-line subset $\{ L_0 L_1 L_k \mid k = 2, 3, ..., n\text{-}1 \}$, and all given lines can be divided into $n-2$ subsets. By using the P3L constraints [40], each subset can build an eighth-order polynomial called the P3L polynomial [38]. With the P3L polynomial, γ can be determined [37, 38], but there are at most 8 minima for the polynomial, which are chosen as the candidate solutions. After $Rot(Z,\gamma)$ is determined, from Eq. (3), only $Rot(Y,\beta)$ needs to be calculated. There are two methods to identify $Rot(Y,\beta)$. One method is solving $Rot(Y,\beta)$ alone, which is for room layouts type 1, type 2, type 3 and type 4, because the given 2D/3D line correspondences information in those layouts is limited, only 5 or 3 line correspondences. Therefore, the accurate $Rot(Y,\beta)$ and translation vector cannot be determined both at the same time. The second method is determining $Rot(Y,\beta)$ together with the translation vector for type 0 and type 5 room layouts. For type 0, there are 8 line correspondences so that rotation and translation restrict each other to yield a simultaneous result. For type 5, there are 5 line correspondences, but the experimental result shows that the second method is more suitable for type 5 and can achieve more accurate results.

3.2.1 Retrieving $Rot(Y,\beta)$ via optimization

From Eq. (3) , $\mathbf{R}_m^n$ can be expressed as:

$$\mathbf{R}_m^n = Rot(Y,\beta)\mathbf{R}' = \begin{bmatrix} u & 0 & v \\ 0 & 1 & 0 \\ -v & 0 & u \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}, \tag{4}$$

in which $\mathbf{R}' = Rot(Z,\gamma)Rot(X,\alpha)$, $u = \cos\beta$ and $v = \sin\beta$. As L_i lies on the plane Π_i, $\mathbf{v}_i^m = \mathbf{R}_w^m \mathbf{v}_i^w$ is perpendicular to the plane normal $\mathbf{n}_i^m = \mathbf{R}_w^m \mathbf{n}_i^c$. Therefore, $\mathbf{R}_m^n$ needs to satisfy the constraint that

$$(\mathbf{n}_i^m)^T \mathbf{R}_m^n \mathbf{v}_i^m = 0 \qquad i = 1, 2, ..., n. \tag{5}$$

In addition, there is a constraint that $u^2 + v^2 = 1$. By using these two constraints and denoting a new unknown $\mathbf{e} = [u, v, 1]^T$, a cost function can be represented as

$$E_{er} = \mathbf{e}^T \mathbf{G} \mathbf{e} + \lambda(1 - u^2 - v^2), \tag{6}$$

in which $\mathbf{G}$ obtained from Eq. (5) is a known 3×3 symmetric matrix, and λ is a Lagrange multiplier. The minima of Eq. (6) can be obtained by solving the polynomial system of its first-order optimality condition [37], then u and v can be determined. Once u and v are determined, $Rot(Y,\beta)$ can also be identified. There will be at most 2 minima for calculating $Rot(Y,\beta)$, and then up to 16 minima for determining $Rot(Z,\gamma)$ and $Rot(Y,\beta)$. For each minima, a candidate $\mathbf{R}_m^n$ can be determined via Eq. (4) and a candidate R_w^c can be obtained by using Eq. (1) .

3.2.2 Solving the rotation and the translation together

As $P_i^m = R_w^m P_i^w$ is also on the plane Π_i, we have a constraint as

$$(\mathbf{n}_i^m)^T (\mathbf{R}_m^n P_i^m + \mathbf{t}^m) = 0 \qquad i = 1, 2, ..., n, \tag{7}$$

where

$$\mathbf{t}^m = \mathbf{R}_w^m \mathbf{t} = [t_x^m \; t_y^m \; t_z^m]^T.$$

By substituting Eqs. (4) into (5) and (7) and stacking all these constraints, $2n$ homogeneous linear equations with parameter vector $[u, v, t_x^m, t_y^m, t_z^m, 1]$ can be obtained, and the rotation angle β and the translation vector $\mathbf{t}^m$ can be estimated [37]. Then, $\mathbf{R}_m^n$ and $\mathbf{R}_w^c$ can be determined by Eqs. (4) and (1), respectively. A few candidate solutions can be obtained, and the room layout constraints are used to find the suitable one.

3.3. Optimizing initial rotation matrix

A more accurate rotation matrix $\mathbf{R}_w^c$ can be obtained through optimizing the initial rotational matrix. Firstly let $\mathbf{s} = [s_1 \; s_2 \; s_3]^T$ be the Cayley-Gibbs-Rodriguez (CGR) parameter vector and $\mathbf{R}_w^c$ can be expressed using CGR parameterization [37] as

$$\mathbf{R}_w^c = \frac{1}{H} \begin{bmatrix} 1 + s_1^2 - s_2^2 - s_3^2 & 2s_1 s_2 - 2s_3 & 2s_1 s_3 + 2s_2 \\ 2s_1 s_2 + 2s_3 & 1 - s_1^2 + s_2^2 - s_3^2 & 2s_2 s_3 - 2s_1 \\ 2s_1 s_3 - 2s_2 & 2s_2 s_3 + 2s_1 & 1 - s_1^2 - s_2^2 + s_3^2 \end{bmatrix}, \tag{8}$$

where $H = 1+s_1^2+s_2^2+s_3^2$. Based on this definition, a least-squares problem with three variables can be reconstructed, and then solved by a single Gauss-Newton step. According to Eq. (2), the rotation and translation can be parameterized, and form the linear system as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{t} \end{bmatrix} = 0, \tag{9}$$

where $\mathbf{r} = [1, s_1, s_2, s_3, s_1^2, s_1 s_2, s_1 s_3, s_3^2, s_2 s_3, s_3^2]^T$.

From Eq. (9), $\mathbf{t} = -(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}\mathbf{r}$ and substituting $\mathbf{t}$ into Eq. (9), we have

$$\mathbf{E}\mathbf{r} = \mathbf{0}, \tag{10}$$

where $\mathbf{E} = \mathbf{A} - (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}$. Finally, we obtain the least-squares problem as follows

$$\varepsilon = \sum_{i=1}^{n} ||\mathbf{E}_i\mathbf{r}||^2, \tag{11}$$

where $\mathbf{E}_i$ is a 3×10 matrix that can be determined ahead, and the Gauss-Newton method can be used to solve the least-squares problem. Once the refined $\mathbf{r}$ is obtained, the optimized initial $\mathbf{R}_w^c$ and $\mathbf{t}$ can be determined. Then the camera origin in the world frame $\mathbf{O}_c^w$ can be evaluated based on $\mathbf{R}_w^c$ and $\mathbf{t}$ as

$$\mathbf{O}_c^w = -(\mathbf{R}_w^c)^T\mathbf{t}, \tag{12}$$

and because $\mathbf{O}_c^w$ must be inside the room, we can use this constraint to obtain the final $\mathbf{R}_w^c$ and $\mathbf{t}$ from several candidates mentioned above. This optimization step has been proven to drastically improve numerical precision [37].

3.4. 3D correspondence estimation of IOCs

In a given layout, IOCs are relatively easy to detect. Then we need to evaluate the 3D correspondences of IOCs in the world frame to get more 2D/3D line correspondences for camera pose estimation. Specifically, two methods are used for different layouts. For types 0, 1, 2, and 5 ($n > 3$), there are at least 5 known line correspondences, which is sufficient to determine the 3D correspondences of IOCs only by the following constraint

$$(\mathbf{n}^c)^T\mathbf{R}_w^c\mathbf{v}^w = 0, \tag{13}$$

in which $\mathbf{n}^c$ is the norm of the projection plane Π_i and supposed to be $\mathbf{n}_i^c$, and $\mathbf{v}^w$ is the direction vector of the i-th line in world frame and supposed to be $\mathbf{v}_i^w$. Here, we omit the subscript i, because they are general for every 2D/3D correspondence, and they can be presented as

$$\mathbf{n}^c = [n_x\ n_y\ n_z]^T$$
$$\mathbf{R}_w^c = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}.$$

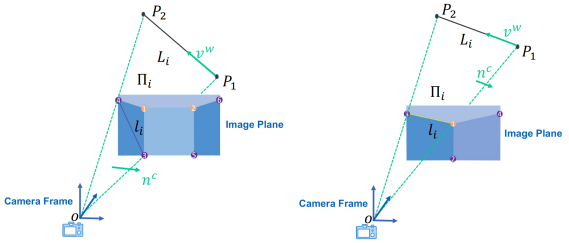

Figure 3. Estimating 3D correspondences of IOCs for type 2 (left) and type 3 (right).

We use the type 2 layout as an example to discuss the solution process (Figure 3, left). $\mathbf{n}^c$ can be determined by two points that are either two IOCs or one IOC plus one inner corner, e.g., IOC 3 and 4. The 3D correspondences of these two points are denoted by P_1 and P_2, respectively. Let $P_1 = [x_1, y_1, z_1]$ and $P_2 = [x_2, y_2, z_2]$, then $\mathbf{v}^w = P_1 - P_2 = [x_1 - x_2, y_1 - y_2, z_1 - z_2]$. From Eq. (13), we have

$$C_x x_1 - C_x x_2 + C_y y_1 - C_y y_2 + C_z z_1 - C_z z_2 = 0, \tag{14}$$

where

$$C_x = n_x r_{11} + n_y r_{21} + n_z r_{31}$$
$$C_y = n_x r_{12} + n_y r_{22} + n_z r_{32}$$
$$C_z = n_x r_{13} + n_y r_{23} + n_z r_{33}.$$

Those unknown 3D correspondences of IOCs lie on the room layout boundaries, so there are only three different situations about which coordinate is unknown. For room layout type 0, type 1, type 2 and type 5, there are four IOCs and only one coordinate is missing for the 3D correspondence of each IOC. Therefore, we use a 5D vector to denote the four unknown coordinates $\mathbf{u} = [u_1, u_2, u_3, u_4, 1]^T$, then apply Eq. (14) to every IOC-contained line in the room layout, we have

$$\mathbf{C}\mathbf{u} = \mathbf{0}, \tag{15}$$

in which $\mathbf{C}$ is a $m \times 5$ matrix, generated by arranging the the coefficients of Eq. (14) for each IOC-contained line, and m is the number of IOC-contained lines in the room layout. The unknown axis coordinates can be estimated by solving the linear system in Eq. (15) with SVD method [16], and the estimated coordinates can be further refined using the method described in Sec. 3.5.

For types 3 and 4 ($n = 3$), there are only three known line correspondences, where three lines form a same junction. In this situation, we can estimate R_w^c with the method in Section 3.2 (i), but we cannot arrive at a unique translation vector [38, 5], and can instead only arrive at a scale value, which results the 3D correspondences of IOCs cannot

be determined uniquely. To get the unique value, we assume camera height (cH) is available. The camera height yields a constraint between $\mathbf{R}_w^c$ and translation $\mathbf{t}$ with regard to Eq. (12), because the Y coordinate of O_w^c is the camera height, but cH can be X or Z coordinate in the world coordinate system. Letting $\mathbf{t} = [t_x\ t_y\ t_z]^T$ and from Eq. (12) we have

$$r_{12}t_x + r_{22}t_y + r_{32}t_z + cH = 0. \tag{16}$$

Here, we use the room layout type 3 as the example to describe the solution process (Figure 3, right). Each point P^w(the subscript is omitted) on the line L_i, such as P_1 or P_2, defined as $P^w = [P_x\ P_y\ P_z]^T$, must satisfy the geometrical constraints in Eq. (2). Therefore, we have

$$A_xP_x + A_yP_y + A_zP_z + A_{t_x}t_x + A_{t_y}t_y + A_{t_z}t_z = 0, \tag{17}$$

where

$$\begin{aligned}
A_x &= n_xr_{11} + n_yr_{21} + n_zr_{31},\\
A_y &= n_xr_{12} + n_yr_{22} + n_zr_{32},\\
A_z &= n_xr_{13} + n_yr_{23} + n_zr_{33},\\
A_{t_x} &= n_x,\ A_{t_y} = n_y,\ A_{t_z} = n_z.
\end{aligned}$$

For type 3 and type 4, there are six unknown parameters including three unknown coordinates and the translation vector for every layout. Thus, we set all the unknowns as the parameter vector $[u_1, u_2, u_3, t_x, t_y, t_z, 1]^T$ and stack all constraints in Eq. (16) and Eq. (17) for related points in every line, to yield

$$\mathbf{A}[u_1, u_2, u_3, t_x, t_y, t_z, 1]^T = 0, \tag{18}$$

in which $\mathbf{A}$ can be obtained by A_x, A_y, A_z, A_{t_x}, A_{t_y}, A_{t_z}, known coordinates of 3D correspondences of IOCs and inner corners in every line. The initial unknown coordinates and translation vector can be estimated by solving the linear system in Eq. (18) with SVD.

3.5. Camera pose optimization via IOC refinement

To further improve camera pose estimation, we need to improve 3D correspondence estimation of IOCs. First, we jointly refine the rotation matrix and 3D correspondence of IOCs together, and use the refined 3D correspondences of IOCs to re-estimate the camera pose. The pose estimation problem is converted into a least-squares problem with three variables related to the rotation matrix R_w^c and the unknown 3D correspondences of IOCs. From Eq. (2), we have

$$(\mathbf{n}^c)^T\mathbf{R}_w^c\mathbf{P}^w = -(\mathbf{n}^c)^T\mathbf{t}, \tag{19}$$

in which $\mathbf{n}^c(\mathbf{n}_i^c)$ and $\mathbf{P}^w(\mathbf{P}_i^w)$ is general for every 2D/3D correspondence. $\mathbf{R}_w^c$ can be represented with the Cayley parameterization as Eq. (8). Now letting $P^w = [P_x, P_y, P_z]$, Eq. (19) can be transformed into the following matrix form

$$\mathbf{Mr} = \mathbf{Nt}, \tag{20}$$

where

$$\mathbf{N} = -(\mathbf{n}^c)^T,$$

$$\mathbf{M} = \begin{bmatrix}
n_xP_x + n_yP_y + n_zP_z\\
2n_zP_y - 2n_yP_z\\
2n_xP_z - 2n_zP_x\\
2n_yP_x - 2n_xP_y\\
n_xP_x - n_yP_y - n_zP_z\\
2n_yP_x + 2n_xP_y\\
2n_zP_x + 2n_xP_z\\
n_yP_y - n_xP_x - n_zP_z\\
2n_zP_y + 2n_yP_z\\
n_zP_z - n_xP_x - n_yP_y
\end{bmatrix},$$

and $r = [1, s_1, s_2, s_3, s_1^2, s_1s_2, s_1s_3, s_3^2, s_2s_3, s_3^2]^T$. Here we need to add unknown coordinates from 3D correspondences of IOCs to the parameter vector. The unknown coordinate is on the X-axis, Y-axis, or Z-axis, then the unknown parameter needs to be extracted from matrix $\mathbf{M}$ and added to the parameter vector. According to three different situations, the added part is

$$\begin{aligned}
&\mathbf{r}_x = [s_4, s_4s_2, s_4s_3, s_4s_1^2, s_4s_1s_2, s_4s_1s_3, s_4s_2^2, s_4s_3^2]^T,\\
&\mathbf{r}_y = [s_5, s_5s_1, s_5s_3, s_5s_1^2, s_5s_1s_2, s_5s_2^2, s_5s_2s_3, s_5s_3^2]^T,\\
&\text{or}\\
&\mathbf{r}_z = [s_6, s_6s_1, s_6s_2, s_6s_1^2, s_6s_1s_3, s_6s_2^2, s_6s_2s_3, s_6s_3^2]^T.
\end{aligned}$$

We can add the unknowns vectors $\mathbf{r}_x, \mathbf{r}_y$, or $\mathbf{r}_z$ to the parameter vector according to different situations. The new parameter vector will be $\hat{\mathbf{r}} = [\mathbf{r}^T\ \mathbf{r}_x^T]^T$, $\hat{\mathbf{r}} = [\mathbf{r}^T\ \mathbf{r}_y^T]^T$ or $\hat{\mathbf{r}} = [\mathbf{r}^T\ \mathbf{r}_z^T]^T$, and $\mathbf{M}$ will become $\hat{\mathbf{M}}$ as

$$\begin{bmatrix}
n_yP_y + n_zP_z\\
2n_zP_y - 2n_yP_z\\
2n_xP_z\\
-2n_xP_y\\
-n_yP_y - n_zP_z\\
2n_xP_y\\
2n_xP_z\\
n_yP_y - n_zP_z\\
2n_zP_y + 2n_yP_z\\
n_zP_z - n_yP_y\\
n_x\\
-2n_z\\
2n_y\\
n_x\\
2n_y\\
2n_z\\
-n_x\\
-n_x
\end{bmatrix},
\begin{bmatrix}
n_xP_x + n_zP_z\\
-2n_yP_z\\
2n_xP_z - 2n_zP_x\\
2n_yP_x\\
n_xP_x - n_zP_z\\
2n_yP_x\\
2n_zP_x + 2n_xP_z\\
-n_xP_x - n_zP_z\\
2n_yP_z\\
n_zP_z - n_xP_x\\
n_y\\
2n_z\\
-2n_x\\
-n_y\\
2n_x\\
n_y\\
2n_z\\
-n_y
\end{bmatrix}
\text{or}
\begin{bmatrix}
n_xP_x + n_yP_y\\
2n_zP_y\\
-2n_zP_x\\
2n_yP_x - 2n_xP_y\\
n_xP_x - n_yP_y\\
2n_yP_x + 2n_xP_y\\
2n_zP_x\\
n_yP_y - n_xP_x\\
2n_zP_y\\
-n_xP_x - n_yP_y\\
n_z\\
-2n_y\\
2n_x\\
-n_z\\
2n_x\\
-n_z\\
2n_y\\
n_z
\end{bmatrix}.$$

However, for the 3D correspondences of inner corners, $\mathbf{M}$ will keep same with $\hat{\mathbf{M}}$, because the coordinates of the 3D correspondences of inner corners are given and the variables in (19) are just r and t.The equations can be listed according to different situations. Then (20) can be written as

$$\hat{\mathbf{M}}\hat{\mathbf{r}} = \mathbf{Nt}. \tag{21}$$

Eq. (21) is satisfied for every reference point, hence

$$\begin{bmatrix} \hat{\mathbf{M}}_1^T \\ \hat{\mathbf{M}}_2^T \\ \vdots \\ \hat{\mathbf{M}}_n^T \end{bmatrix} \hat{\mathbf{r}} = \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \\ \vdots \\ \mathbf{N}_n \end{bmatrix} \mathbf{t} \Longleftrightarrow \widetilde{\mathbf{M}}\hat{\mathbf{r}} = \widetilde{\mathbf{N}}\mathbf{t} \Longleftrightarrow \mathbf{t} = \mathbf{C}\hat{\mathbf{s}}, \tag{22}$$

where $\mathbf{C} = (\widetilde{\mathbf{N}}^{\mathbf{T}}\widetilde{\mathbf{N}})^{-1}\widetilde{\mathbf{N}}^{\mathbf{T}}\widetilde{\mathbf{M}}$, $\hat{\mathbf{r}}$ will change according to the unknown 3D correspondences of IOCs, and we obtain the least-squares problem as follows

$$\hat{\varepsilon} = \sum_{i=1}^{n} ||(\widetilde{\mathbf{M}} - \widetilde{\mathbf{N}}\mathbf{C})\hat{\mathbf{r}}||^2 = \sum_{i=1}^{n} ||\mathbf{E}\hat{\mathbf{r}}||^2. \tag{23}$$

However, this cost function is the 3rd order, and we need to do order reduction in order to use Gauss-Newton. We solve this problem using a relinearization technique [22]. Let $s_7 = s_1^2, s_8 = s_1 s_2, s_9 = s_1 s_3, s_{10} = s_2^2, s_{11} = s_2 s_3, s_{12} = s_3^2$. Although we introduce five more parameters, we have five more equations, which allow us to reduce the order successfully. Then we can use Gauss-Newton similar to the one discussed in Section 3.3 to refine $\mathbf{R}_c^w$ and 3D correspondences of IOCs. Afterwards, there will be more 2D/3D line correspondences which can be used to determine the rotation matrix and translation vector using the methods in Section 3.2 and 3.3. The proposed method, referred to as PnL-IOC, is presented in Algorithm 1.

Algorithm 1: The proposed PnL-IOC method.

Input : 2D/3D line correspondence of the specific layout
Output: Rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$

$Rot(X, \alpha) \leftarrow$ determined by the longest line
$R_z List \leftarrow Rot(Z, \gamma)$ determined by P3L polynomial
for $i \leftarrow 1$ **to** *length of* $R_z List$ **do**
  if *type 1, type 2, type 3, type 4* **then**
    $R_y List \leftarrow Rot(Y, \beta)$ determined by 3.2.1
    for $j \leftarrow 1$ **to** *length of* $R_y List$ **do**
      $\mathbf{R}_w^c \leftarrow$the orthogonal error minimal result
    end for
  else
    $R_y List \leftarrow Rot(Y, \beta)$ determined by 3.2.2
    for $j \leftarrow 1$ **to** *length of* $R_y List$ **do**
      $\mathbf{R}_w^c \leftarrow$the orthogonal error minimal result
    end for
  end if
end for
if *type 0, type 1, type 2, type 5* **then**
  3D correspondences of IOCs determined by Eq. (15)
else
  3D correspondences of IOCs determined by Eq. (18)
end if
Refine 3D correspondences of IOCs and rotation matrix $\mathbf{R}_w^c$
Reestimate $\mathbf{R}$ and $\mathbf{t}$ with additional refined 2D/3D line correspondences by repeating step 1 to step 12
return $\mathbf{R}, \mathbf{t}$

4. Experiment results

The PnL-IOC method is evaluated on both synthetic data and real images, and compared with leading PnL methods listed below. All methods are implemented in MATLAB on a MacPro with a 2.3 GHz CPU and 8GB of RAM.

- *RPnL* A non-iterative method, which works well for both non-redundant ($n \leq 6$) and redundant line correspondences. However, it is not that accurate in some cases, because it is a suboptimal method [40].
- *LPnL-Bar-ENull* A linear method, which parameterizes reference lines using barycentric coordinates, and uses the null space solver to tackle the PnL problem. This method is suitable for the cases where $n > 6$ [38].
- *ASPnL* A subset-based PnL method, which is improved based on the RPnL method and represents the state-of-the-art method. However, this method is not suitable for room layout type 3 and type 4, because the rotation and translation cannot be determined at the same time for those two types [38].
- *SRPnL* An improved subset-based PnL method, which is improved based on the ASPnL method and has good performance. However, when the line correspondences number is increasing, the result is worse than that by ASPnL and RPnL [37].

4.1. Experiments with synthetic data

4.1.1 Synthetic data

Using a virtual perspective camera with image size of 640×640 pixels and focal length of 180 pixels, the 3D reference lines are generated based on different room layout types. For a specific room layout type, we fix the 3D correspondences coordinates of inner corners in the world frame and the initial rotation angle and translation vector, then we randomly change the rotation angle in three different angles in the range of [-5, 5] and translation vector in three different directions in the range of [-3, 3], making sure that the generated lines can form a specific room layout. Then we project these 3D lines onto the 2D image plane using the ground-truth rotation $\mathbf{R}_{true}$ and translation $\mathbf{t}_{true}$. Some randomly generated room layouts are shown in Figure 4.

The error metric is defined the same as in [37, 38, 15]. $\mathbf{R}$ and $\mathbf{t}$ denote the estimate results for rotation matrix and translation vector, respectitively. Then rotation error ($Err_{\mathbf{R}}$) and translation error ($Err_{\mathbf{t}}$) will be calculated as

$$Err_{\mathbf{R}}(deg) = \max_{k \in 1,2,3} \angle(\mathbf{R}_{true}(:,k), \mathbf{R}(:,k)) \times \frac{180}{\pi},$$
$$Err_{\mathbf{t}}(\%) = \frac{||\mathbf{t} - \mathbf{t}_{true}||}{||\mathbf{t}_{true}||} \times 100, \tag{24}$$

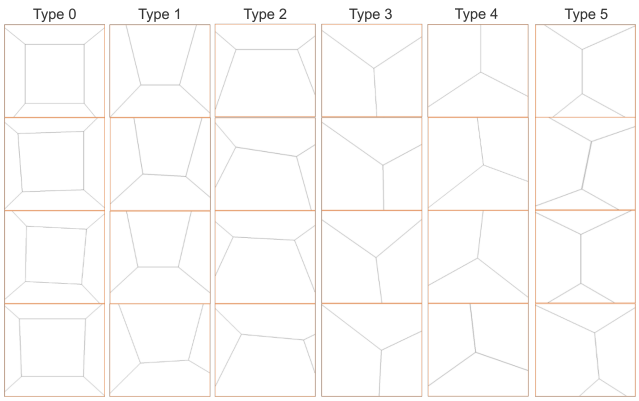

Figure 4. Some randomly generated room layout images.

where $\mathbf{R}_{true}(:,k)$ and $\mathbf{R}(:,k)$ are the k-th column of $\mathbf{R}_{true}$ and $\mathbf{R}$, respectively. $\angle$ represents the angle difference between $\mathbf{R}_{true}(:,k)$ and $\mathbf{R}(:,k)$. Depending on the experiment, a different level of white Gaussian noise was added to the 2D image plane.

4.1.2 Different layout results with varying noise

The experiment mainly tests the effects of noise on the accuracy of all methods for every room layout. We varied the noise deviation level δ from 1 to 10 pixels. At each noise level, we conducted 1000 independent tests once and ran three times, and calculated the mean and median errors of rotation and translation. Figure 5 shows that our proposed method yields a steady result as the noise is increased for all room layouts, and the mean errors of rotation and translation are increased almost linearly with the noise levels. For type 3, we find that the proposed method result is almost the same as the SRPnL method, because for type 3 the translation vector is a scale value if determined by the given information. After introducing the camera height, we can determine the unique result. However, there always exists a solution whose orthogonal error is the smallest when the noise is added. The solution obtained is the best for the given information. For type 3, the advantage of our method is not obvious, but for type 4 other PnL methods do not achieve the right rotation. Overall, only our PnL-IOC method is robust enough for every room layout.

4.1.3 Computational efficiency

Table 1 shows the computational time with fixed $\delta = 2$, where we conducted 1000 tests and show the average running time in seconds (s). From the results, our proposed method is comparable to the others, and even better than SRPnL for some room types. Considering the high accuracy and robustness, our method is still competitive.

Methods	Time (seconds)					
	type 0	type 1	type 2	type 3	type 4	type 5
ASPnL	0.0066	0.0065	0.0060	0.0072	0.0053	0.0056
LPnL-Bar-ENull	0.0038	0.0040	0.0016	N/A	N/A	0.0032
RPnL	0.0027	0.0022	0.0023	N/A	N/A	0.0020
SRPnL	0.0109	0.0091	0.0086	0.0437	0.0393	0.0090
PnL-IOC	0.0073	0.0122	0.0121	0.0277	0.0164	0.0177

Table 1. A comparison of the computational efficiency.

4.2. Experiments with Real Images

We also applied the aforementioned PnL algorithms on a set of room layout images with a known 3D line model. We collected some room layout images in the entry area of our office. For each image, we detected the inner corners and IOCs manually, and set 3D correspondences based on the room dimension information, then established the line correspondences between the image lines and the 3D line model. We tested the compared algorithms for every room layout type. In order to demonstrate the accuracy of the result, we projected the 3D line model into the image and estimate all the corresponding points reprojection error using the estimated camera pose. Figure 6 shows the type 4 and type 5 room layout results, and Table 2 shows the reprojection error for different room types in the real world. From Table 2, the proposed method again outperform others quantitatively for all six layouts being tested.

Methods	Reprojection error (pixels)					
	type 0	type 1	type 2	type 3	type 4	type 5
ASPnL	17.014	4.7613	21.723	1154.51	1154.51	50.052
LPnL-Bar-ENull	20.659	5.5603	814.60	N/A	N/A	65.780
RPnL	18.308	5.6313	203.27	N/A	N/A	187.72
SRPnL	600.65	4.7613	21.723	7.91e-12	2.53e+03	287.11
PnL-IOC	**16.765**	**4.3776**	**6.2870**	**4.04e-12**	**8.30e-13**	**28.551**

Table 2. Comparison of the Reprojection error for real images.

5. Conclusion

In this work, we presented a new PnL approach to egocentric indoor localization by camera pose estimation from room layouts. Specifically, we have introduced IOCs to facilitate the PnL solution by adding more 2D/3D additional line correspondences. To the best of our knowledge, this is the first attempt to use IOCs in a room layout to solve the PnL problem. The key idea of our method is to initialize the 3D correspondences of IOCs by solving a linear system and to further optimize these 3D correspondences along with camera pose via the iterative Gaussian-Newton algorithm. The experimental results demonstrate that our method is more accurate and robust compared with the existing PnL methods at a comparable computational load.

Acknowledgment

This work is supported in part by the US National Institutes of Health (NIH) Grant R15AG061833 and the Oklahoma Center for the Advancement of Science and Technology (OCAST) Grant HR18-069.

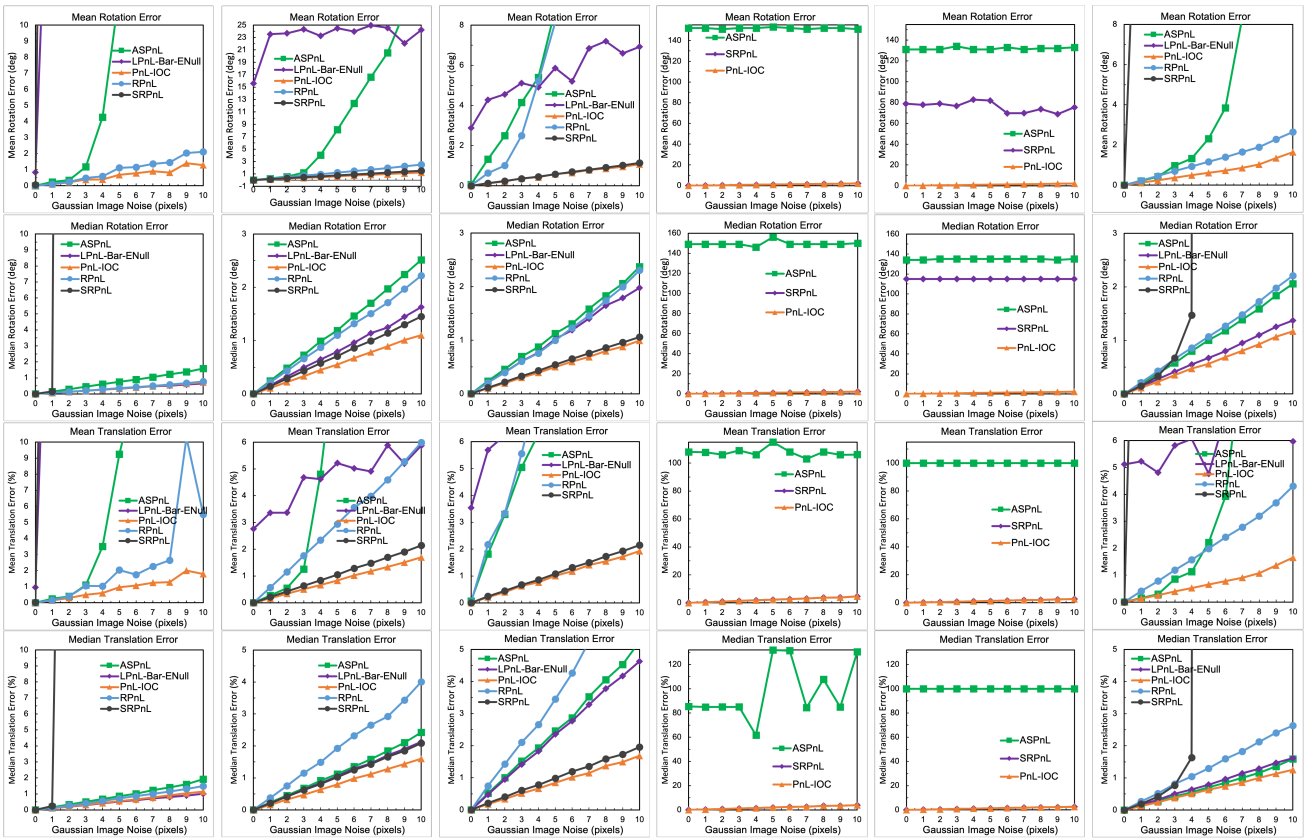

Figure 5. Experimental results on the simulated data under different noise levels ($\delta = 1, ..., 10$). From top to bottom: the mean/median rotation errors and the mean/median translation errors. From left to right, the results for type 0, type 1, type 2, type 3, type 4, type 5.

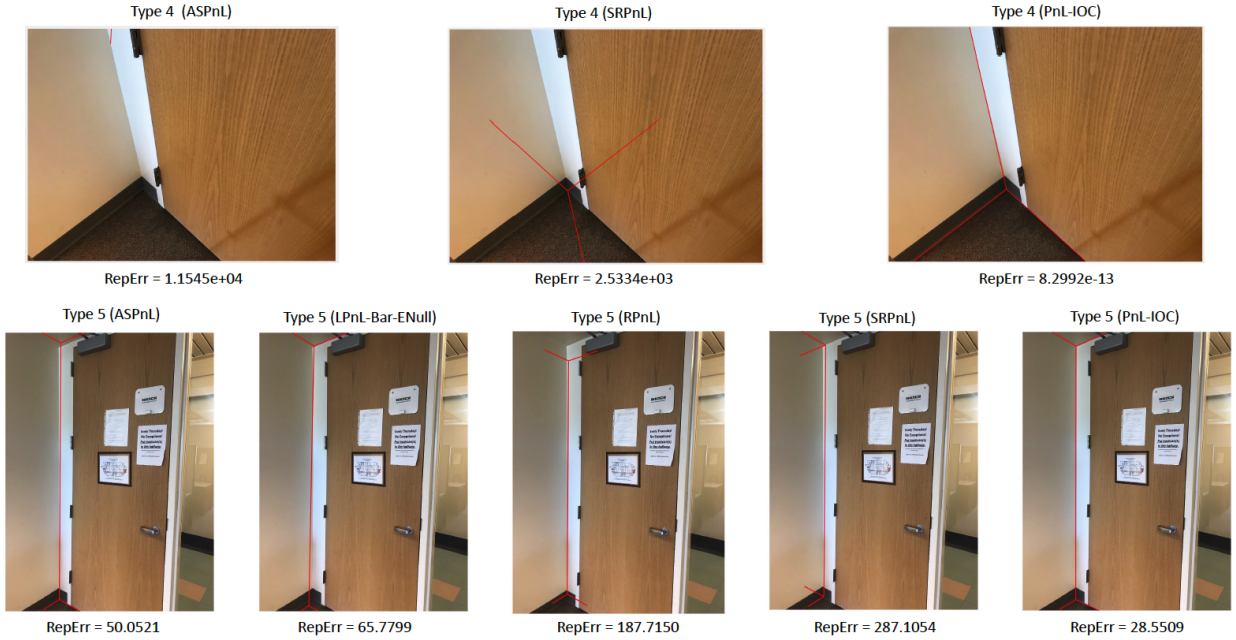

Figure 6. Camera pose estimation from real world images using our method and other PnL methods. The first row results are for type 4 room layout and the second row for type 5 layout, where RepErr is the reprojection error.

References

[1] Y. I. Abdel-Aziz, H. M. Karara, and M. Hauck. Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry*. 81(2):103–107, 2015.

[2] A. Ansar and K. Daniilidis. Linear Pose Estimation from Points or Lines. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25:578– 589, 06 2003.

[3] V. Bettadapura, I. Essa, and C. Pantofaru. Egocentric Field-of-View Localization Using First-Person Point-of-View Devices. In *Proc. ICCV*, 2015.

[4] V. Caglioti. The planar three-line junction perspective problem with application to the recognition of polygonal patterns. *Pattern Recognition*, 26(11):1603–1618, 1993.

[5] H. Chen. Pose determination from line-to-plane correspondences: existence condition and closed-form solutions. In *Proc. ICCV*, 1990.

[6] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *Proc. CVPR*, 2013.

[7] S. Christy and R. Horaud. Iterative Pose Computation from Line Correspondences. *Computer Vision and Image Understanding*, 73(1):137–144, 1999.

[8] J. M. Coughlan and A. L. Yuille. Manhattan World: Orientation and Outlier Detection by Bayesian Inference. *Neural Computation*, 15(5):1063–1088, 2003.

[9] A. Dai, A. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CoRR*, abs/1702.04405, 2017.

[10] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *Proc. ECCV*, 2018.

[11] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *Proc. CVPR*, 2012.

[12] L. Del Pero, J. Bowdish, B. Kermgard, E. Hartley, and K. Barnard. Understanding bayesian rooms using composite 3d object models. In *Proc. CVPR*, pages 153–160, 2013.

[13] M. Dhome, M. Richetin, J.-T. Lapreste, and G. Rives. Determination of the attitude of 3D objects from a single perspective view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, 1989.

[14] F. Dornaika and C. Garcia. Pose Estimation using Point and Line Correspondences. *Real-Time Imaging*, 5(3):215–230, 1999.

[15] L. Ferraz, X. Binefa, and F. Moreno-Noguer. Very Fast Solution to the PnP Problem with Algebraic Outlier Rejection. In *Proc. CVPR*, 2014.

[16] W. Gander. *Least Squares Fit of Point Clouds*, pages 339–349. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[17] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[18] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *Proc. ICCV*, 2009.

[19] J. A. Hesch and S. I. Roumeliotis. A Direct Least-Squares (DLS) method for PnP. In *Proc. ICCV*, 2011.

[20] R. Kumar and A.R. Hanson. Robust Methods for Estimating Pose and a Sensitivity Analysis. *CVGIP: Image Understanding*, 60(3):313–342, 1994.

[21] C.-Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. RoomNet: End-to-End Room Layout Estimation. In *Proc. ICCV*, 2017.

[22] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81, 02 2009.

[23] L. Li, Q. Xu, V. Chandrasekhar, J.-H. Lim, C. Tan, and M. A. Mukawa. A Wearable Virtual Usher for Vision-Based Cognitive Indoor Navigation. *IEEE Transactions on Cybernetics*, 47(4):841–854, 2017.

[24] S. Li, C. Xu, and M. Xie. A Robust O(n) Solution to the Perspective-n-Point Problem. *IEEE transactions on pattern analysis and machine intelligence*, 34, 01 2012.

[25] H. J. Lin, S.-W. Huang, S.-H. Lai, and C.-K. Chiang. Indoor Scene Layout Estimation from a Single Image. In *Proc. ICPR*, 2018.

[26] H.-J. Lin and S. Lai. Deeproom: 3d room layout and pose estimation from a single image. In *ACPR*, 2019.

[27] F. M. Mirzaei and S. I. Roumeliotis. Globally optimal pose estimation from line correspondences. In *Proc. ICRA*, 2011.

[28] T.-H.-C. Nguyen, J.-C. Nebel, and F. Florez-Revuelta. Recognition of Activities of Daily Living with Egocentric Vision: A Review. *Sensors*, 16(1), 2016.

[29] S. Orlando, A. Furnari, S. Battiato, and G. Farinella. Image Based Localization with Simulated Egocentric Navigations. In *Proc. ICCV*, 2019.

[30] B. Přibyl, P. Zemčík, and M. Čadík. Camera Pose Estimation from Lines using Plücker Coordinates. In *Proc. BMVC*, 2015.

[31] L. Qin and F. Zhu. A New Method for Pose Estimation from Line Correspondences. *Acta Automatica Sinica*, 34(2):130–134, 2008.

[32] F. Ragusa, A. Furnari, S. Battiato, G. Signorello, and G. M. Farinella. Egocentric Visitors Localization in Cultural Sites. *J. Comput. Cult. Herit.*, 12(2), Apr. 2019.

[33] Y. Ren, S. Li, C. Chen, and C.-C. J. Kuo. A Coarse-to-Fine Indoor Layout Estimation (CFILE) Method. In *Proc. ACCV*, 2017.

[34] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *Proc. CVPR*, 2012.

[35] M. Silva, R. Ferreira, and J. Gaspar. Camera Calibration using a Color-Depth Camera: Points and Lines Based DLT including Radial Distortion. In *Proc. IROS*, 2012.

[36] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *Proc. ECCV*, 2010.

[37] P. Wang, G. Xu, Y. Cheng, and Q. Yu. Camera pose estimation from lines: a fast, robust and general method. *Machine Vision and Applications*, vol. 30, 2019.

[38] C. Xu, L. Zhang, L. Cheng, and R. Koch. Pose Estimation from Line Correspondences: A Complete Analysis and a Series of Solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1209–1222, 2017.

[39] C. Yan, B. Shao, H. Zhao, R. Ning, Y. Zhang, and F. Xu. 3D Room Layout Estimation From a Single RGB Image. *IEEE Transactions on Multimedia*, 22(11):3014–3024, 2020.

[40] L. Zhang, C. Xu, K.-M. Lee, and R. Koch. Robust and Efficient Pose Estimation from Line Correspondences. In *Proc. ACCV*, 2013.

[41] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *CoRR*, abs/1612.07429, 2016.