Egocentric Indoor Localization from Coplanar Two-Line Room Layouts

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

The coplanar two-line room layout with two parallel junction lines is often seen in an egocentric indoor vision when facing a wall or walking in a corridor. However, camera pose estimation from this kind of room layouts cannot be handled by existing vanishing point-based algorithms or PnL (Perspective-n-Line) methods due to the lack of line correspondences. This includes a recently proposed PnL-IOC approach that introduces image outer corners (IOCs), i.e., the intersecting points between room layout boundaries and image borders, to create more auxiliary lines. In this paper, a new coplanar P3L (CP3L) method is proposed to handle the coplanar two-line room layouts by embedding a P3L (Perspective-three-Line) method into the NSGA-II, a multi-objective optimization method. The proposed CP3L algorithm jointly estimates the initial camera pose and the 3D correspondences of four IOCs related to the two junction lines, and optimizes the camera pose in the iterative Gauss-Newton algorithm. We also study and compare the robustness of CP3L solutions under different configurations of auxiliary lines from estimated IOCs. Experiment results on both simulated images and real ones from the Matterport3D-Layout database demonstrate the accuracy and robustness of the proposed method.

1. Introduction

In indoor egocentric vision, the spatial structure of a room can be represented by certain 3D layouts defined by a few junctions and boundary lines [13, 39, 43, 51, 53], and deep learning for room layout estimation has shown great promise recently [28, 38, 49]. On the other hand, camera pose estimation from room layouts has some advantages due to the fact that most indoor scenes conform to the Manhattan world assumption [11]. A layout-oriented PnL (Perspective-n-Line) method was recently proposed for camera pose estimation from 2D room layouts, which introduces image outer corners (IOCs) to provide sufficient PnL conditions [9]. Generally, there are 11 room layouts defined [28], among which there are two types of room layouts with two parallel and coplanar lines as shown in Fig. 1. These two layouts are popular in indoor egocentric vision when facing a wall or walking in a corridor. However, since there are only two given 2D/3D line correspondences, camera pose estimation becomes ill-posed and cannot be handled by existing methods, including the one in [9].

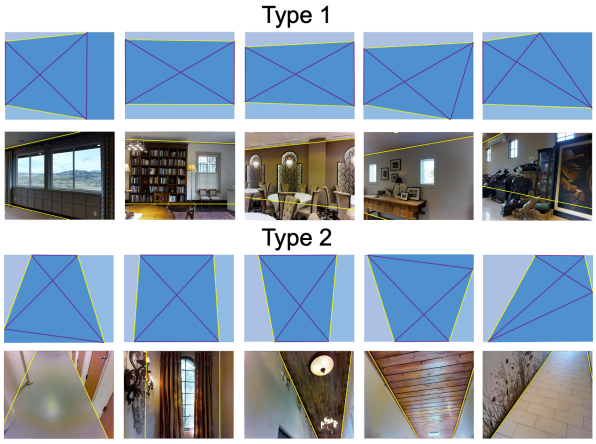

Figure 1. The two coplanar two-line room layouts under study where IOCs are connected by purple lines.

In this work, we develop a new coplanar P3L (CP3L) method to handle these two specific layouts by taking advantage of the idea of using IOCs in [9] to provide additional line correspondences. However, the initial rotation matrix cannot be obtained by only two given 2D/3D line correspondences. Therefore, we embed the P3L in the non-dominated sorting genetic algorithm II (NSGA-II) [15] which is a fast multi-objective genetic algorithm that has been widely used [4, 23, 33, 36, 42]. The proposed method optimizes 3D correspondences of IOCs by evaluating the fitness values based on their 3D correspondences, the rotation matrix, and the translation vector. The rotation and translation are then re-estimated from the given 2D/3D line correspondences created from the optimized IOCs. Therefore, the 3D correspondences of IOCs and the initial camera pose can be optimized iteratively to reach a final solution. To the best our knowledge, this work is the first attempt to attack this kind of coplanar two-line layouts.

2. Related work

A coplanar two-line layout provides two parallel lines in the scene according to the room structure and dimension, and the perspective projection of any set of parallel lines which are not parallel to the image plane will converge to a "vanishing point" [19]. Vanishing points can be determined by line pair intersections from parallel lines in the scene for most of the existing methods [1, 2, 8, 10, 21, 26, 32, 35]. Camera pose estimation from vanishing points is an effective approach [5, 6, 25, 46]. However, it needs at least two orthogonal vanishing points to determine the camera pose uniquely [20, 32, 55], and on the other hand, most coplanar two-line layouts from real-world images can only provide one vanishing point, as illustrated in Fig. 1.

The P3L (Perspective-3-Line) problem is the basis for dealing with the general PnL problem [48] because there are 6 DoFs for a 3D camera pose and each line correspondence offers two constraints. The P3L problem was addressed with an analytical method by solving an eighth-order polynomial in [18]. An algebraic P3L method was proposed in [7] that may not be stable in the presence of noise. By introducing two intermediate frames in [50], the P3L problem formulation can be simplified. However, the P3L solution usually cannot be uniquely determined [7]. In [9], a new PnL method was proposed based on room layouts, which introduced IOCs to change the P3L problem to a PnL ($n > 3$) problem, but it cannot handle two-line layouts because of only two given 2D/3D line correspondences.

Room layout estimation is a well studied topic in decades [34], which was mainly solved with geometry-based approaches in the early attempts [16, 17, 24, 40, 44]. With the advancement of deep learning, a wide range of highly challenging scenes can be represented by a set of well-defined layouts robustly and accurately [28, 30, 49, 52]. Moreover, the high quality datasets [12, 53, 54] published recently further promote the development of deep learning methods for room layout estimation which supports camera pose estimation for more location-aware vision tasks.

3. Problem Statement

3.1. PnL problem statement

The PnL problem is about recovering rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ of a camera from n known 3D reference lines $L_i = (\mathbf{v}_i^w, \mathbf{P}_i^w)$ $(i = 1, ..., n)$ along with their corresponding 2D projections on the image plane denoted as $l_i = (s_i, e_i)$ [56], where $\mathbf{v}_i^w \in \mathbb{R}^3$ is the normalized vector for L_i, $\mathbf{P}_i^w \in \mathbb{R}^3$ is any point on L_i in the world coordinate frame, and s_i and e_i are the endpoints of l_i in the image plane. The problem is illustrated in Fig. 2 and to tackle with this problem, a new camera frame and a model frame are introduced into the re-projection model as two intermediate frames. The rotation from the world frame to the model frame is defined as $\mathbf{R}_w^m$, and similarly $\mathbf{R}_m^n$, $\mathbf{R}_n^c$, and $\mathbf{R}_w^c$ can be defined [9], where the new camera frame can be determined by rotating the original camera frame with $\mathbf{R}_w^m$, as $\mathbf{R}_n^c = (\mathbf{R}_w^m)^T$. The relationship among those rotation matrices can be defined as:

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

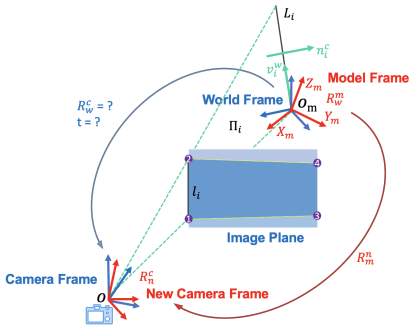

Figure 2. The PnL problem illustration.

A projection plane Π_i can be formed with a given 2D line l_i, the corresponding 3D line L_i, and the projection center O. The cross product of two points on l_i is calculated as the normal of Π_i, denoted by $\mathbf{n}_i^c$. With the geometrical constraints [22], $\mathbf{P}_i^c = \mathbf{R}_w^c \mathbf{P}_i^w + \mathbf{t}$, the coordinate of $\mathbf{P}_i^w$ in the camera coordinate frame, should be perpendicular to the normal $\mathbf{n}_i^c$, then

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

and an analytic solution of $\mathbf{t}$ can be obtained by Eq. (2) [48].

3.2. Coplanar P3L problem statement

If two 3D correspondences of IOCs are given which are displayed as the purple points in Fig. 3, the problem can be solved by a P3L method. Thus Camera pose estimation from coplanar two-line room layouts can be converted to a coplanar P3L (CP3L) problem. However, the unique camera pose can be only determined for PnL ($n > 3$) problem, and there are multiple P3L solutions [7, 31, 48]. To tackle with this issue, the camera height (cH) is introduced to obtain the unique camera pose. In egocentric vision, the cH is directly related to the user's height and is assumed to be available. The camera origin in the world frame $\mathbf{O}_c^w$ can be calculated based on $\mathbf{R}_w^c$ and $\mathbf{t}$ as

$$\mathbf{O}_c^w = -(\mathbf{R}_w^c)^T \mathbf{t}. \tag{3}$$

Because the world frame is based on the Manhattan room layout structure, one coordinate of O_w^c is the camera height that yields a constraint between $\mathbf{R}_w^c$ and $\mathbf{t}$ with regard to Eq. (3). This constraint can be stacked with Eq. (2) to obtain the unique camera pose.

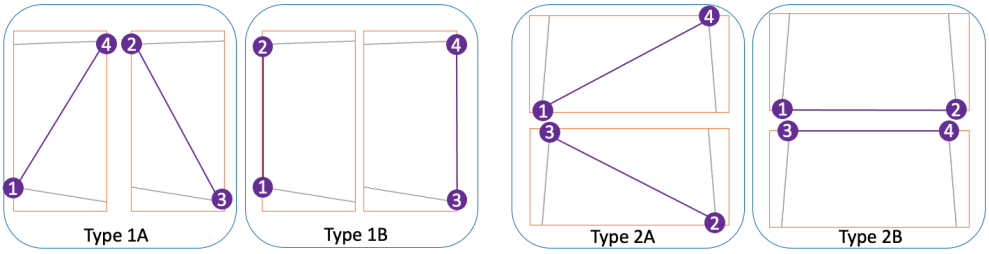

Figure 3. Different combinations for IOCs.

However, the range of 3D correspondences of IOCs is the only known information because the two given coplanar lines are the boundaries of a room that can be well defined by the room dimension and basic structure information. The four unknowns for 3D correspondences of IOCs are defined as $\mathbf{u}_f = [u_{f1}, u_{f2}, u_{f3}, u_{f4}]^T$, and those four unknowns are in the same direction axis. The range of the four unknowns can provide a good initial for $\mathbf{u}_f$. Thus a NSGA-II embedded P3L method is introduced. After initialing the first generation for $\mathbf{u}_f$, two IOCs are considered as one group, and then different combinations for four IOCs can generate 2 different situations for types 1 and 2 shown in Fig. 3. The reason why the four unknowns cannot be used together as one group is that optimizing four unknowns together will take a very long time to converge or even not usually converge because four unknowns mean so many possibilities and the result for camera pose estimation is discontinuous. Meanwhile, the reason why we only use two groups, instead of three or four, is that the information provided by type 1B or 2B is limited because the two IOCs are in the same line, which makes the optimization more difficult. Besides, these more groups will reduce the converged rate. Then, each group will be a CP3L problem.

4. Proposed Method

For the camera pose estimation of the two-line room layout types shown in Fig. 1, the yellow line correspondences can be easily defined. However, camera pose cannot be estimated only by two line correspondences using any PnL method, but if the purple lines can be defined, there will be more line correspondences. Therefore, 3D correspondence estimation of IOCs is the key for our proposed method. NSGA-II embedded with P3L is introduced to determine 3D correspondences of IOCs and the initial camera pose, and Gaussian-Newton is adopted after reducing the order for the cost function to optimize camera pose via IOC refinement. The proposed method can be described as the flow chart in Fig. 4 that is detailed in the following sections.

4.1. Camera pose estimation

Camera pose estimation is a vital step in the proposed method, which is mainly used to determine the rotation and translation after 3D correspondences of IOCs are generated

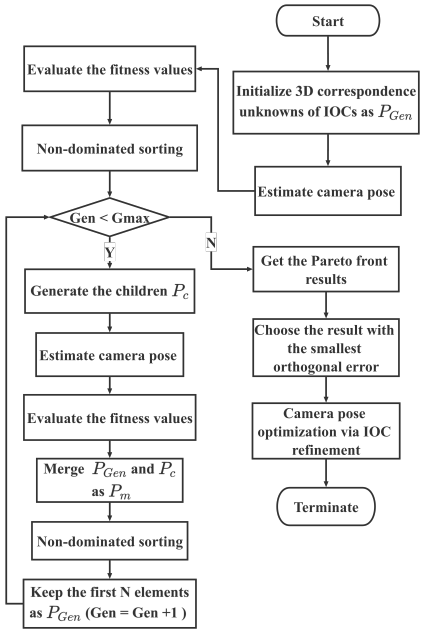

Figure 4. The proposed CP3L method.

in the initialization and iterative step as Fig. 4 shown. For a P3L problem displayed in Fig. 2, the rotation matrix $\mathbf{R}_w^m$ can be generated based on the line $L_0 = (\mathbf{v}_0^w, \mathbf{P}_0^w)$ with the longest projection length [9]. Therefore, $\mathbf{R}_m^n$ is the key to calculate rotation matrix. According to the Euler Angle definition [37], $\mathbf{R}_m^n$ can be expressed as

$$\mathbf{R}_m^n = Rot(Y, \beta)Rot(Z, \gamma)Rot(X, \alpha), \tag{4}$$

where $Rot(Y, \beta)$, $Rot(Z, \gamma)$, and $Rot(X, \alpha)$ denote the rotation and β, γ, and α denote the rotation angle around the Y-axis, Z-axis, and X-axis in the model frame, respectively [45]. $Rot(X, \alpha)$ can be easily obtained because α is the angle between $\mathbf{v}_0^m$ ($\mathbf{v}_0^m = \mathbf{R}_w^m \mathbf{v}_0^w$) and Z_m-axis. An eighth-order polynomial called the P3L polynomial is built by the 3-line subset $\{L_0 L_1 L_2\}$ formed from the remaining lines L_1 and L_2 together with line L_0 [48]. This polynomial can be used to determine $Rot(Z, \gamma)$ [45, 48], and at most 8 minima can be chosen as the candidate solutions. β can be retrieved via optimization method after γ is obtained, and at most 2 minima for $Rot(Y, \beta)$ will be determined [9]. Therefore, there will be up to 16 minima for $Rot(Z, \gamma)$ and

$Rot(Y, \beta)$ combinations, and $\mathbf{R}_m^n$ candidates can be determined via Eq. (4) for each minima combination. The rotation matrix $\mathbf{R}_w^c$ can be finally obtained with Eq. (1) .

Rotation matrix $\mathbf{R}_w^c$ can be polished through optimization as follows. Firstly, let $\mathbf{R}_w^c$ be expressed using Cayley-Gibbs-Rodriguez (CGR) parameter vector [41, 48], which is related to three variables and defined as $\mathbf{r}$. Then, a least-squares problem can be reconstructed and solved by a single Gauss-Newton step. The rotation and translation can be parameterized according to Eq. (2), and form the linear system as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{t} \end{bmatrix} = 0. \tag{5}$$

From Eq. (5), $\mathbf{t}$ can be represented by $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{r}$, and resubstituting $\mathbf{t}$ into Eq. (5) for m known 3D points on the n known lines in the world coordinate frame, the least-squares problem is obtained as follows

$$\varepsilon = \sum_{i=1}^{m} ||\mathbf{E}_i \mathbf{r}||^2, \tag{6}$$

where $\mathbf{E}_i = \mathbf{A}_i - (\mathbf{B}_i^T \mathbf{B}_i)^{-1} \mathbf{B}_i^T \mathbf{A}_i$, a 1×10 vector, can be calculated ahead. The traditional Gauss-Newton method can be adopted to solve the problem. The optimized initial $\mathbf{R}_w^c$ can be determined based on the refined $\mathbf{r}$ [45].

As mentioned before, we assume that the camera height (cH) in the world frame is available that is related to the user's height in egocentric vision. Letting

$$\mathbf{t} = [t_x \; t_y \; t_z]^T \quad and \quad \mathbf{R}_w^c = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix},$$

from Eq. (3) we have

$$\begin{aligned} &r_{11}t_x + r_{21}t_y + r_{31}t_z + cH = 0, \\ &r_{12}t_x + r_{22}t_y + r_{32}t_z + cH = 0, \\ &\text{or} \\ &r_{13}t_x + r_{23}t_y + r_{33}t_z + cH = 0. \end{aligned} \tag{7}$$

After the refined $\mathbf{r}$ is substituted into Eq. (5), a linear system can be obtained by being stacked with Eq. (7) and Eq. (5), which can solved by SVD, and translation vector $\mathbf{t}$ can be determined uniquely. The camera origin in the world frame $\mathbf{O}_c^w$ can be calculated by using Eq. (3), and $\mathbf{O}_c^w$ must be inside the room according to the coordinate system setting, which can be used to remove some incompatible solutions from all the candidate camera pose solutions. For each remaining R_w^c candidate, its orthogonal error

$$E_{or} = \sum_{i=1}^{n} [(\mathbf{n}_i^c)^T \mathbf{R}_w^c \mathbf{v}_i^w]^2 \tag{8}$$

can be evaluated, and the $\mathbf{R}_w^c$ with the smallest E_{or} and its corresponding $\mathbf{t}$ are selected as the final initial camera pose solution.

4.2. Fitness function determination

The performance of genetic algorithms depends to a large degree on the fitness functions therefore the fitness functions need to be carefully selected to match the specifics of credit scoring [27]. Type 1A in Fig. 3 is set as an example to discuss the process to determine fitness functions for our proposed method. The IOC 1 and IOC 4 estimation combination is set as group 1 and the IOC 2 and IOC 3 estimation combination is set as group 2. After the 3D information of IOCs is generated by NSGA-II for group 1 or 2, three 2D/3D line correspondences can be determined, which is cast as a P3L problem and camera pose can be estimated with the steps described in 4.1. After the initial rotation matrix $\mathbf{R}_g$ and translation vector $\mathbf{t}_g$ are obtained, the orthogonal error O_{er} can be evaluated according to Eq. (2) and Eq. (8) as

$$O_{er} = \sum_{i=1}^{3} ||(\mathbf{n}_i^c)^T (\mathbf{R}_g \mathbf{P}_i^w + \mathbf{t}_g)|| + ||(\mathbf{n}_i^c)^T \mathbf{R}_g \mathbf{v}_i^w||, \tag{9}$$

where $\mathbf{P}_i^w$ and $\mathbf{v}_i^w$ are the known point and normalized vector for line L_i. Eq. (9) for group 1 and 2 can be considered as two fitness functions, and the orthogonal errors for group 1 and 2, O_{er1} and O_{er2}, can be considered as two fitness values. Meanwhile, the estimated rotation matrices and translation vectors for group 1 and 2 are supposed to be the same. Therefore, the functions about rotation matrix difference $O_{\mathbf{R}}$ and translation vector difference $O_{\mathbf{t}}$ between two groups can be used as the other two fitness functions, which are defined as

$$\begin{aligned} O_{\mathbf{R}}(deg) &= \max_{k \in 1,2,3} \angle(\mathbf{R}_{g1}(:,k), \mathbf{R}_{g2}(:,k)) \times \frac{180}{\pi}, \\ O_{\mathbf{t}}(\%) &= \frac{||\mathbf{t}_{g1} - \mathbf{t}_{g2}||}{||\mathbf{t}_{g2}||} \times 100, \end{aligned} \tag{10}$$

where $\mathbf{R}_{g1}$ and $\mathbf{t}_{g1}$ are the estimated rotation matrix and translation vector for group 1, and $\mathbf{R}_{g2}$ and $\mathbf{t}_{g2}$ are for group 2. $\mathbf{R}_{g1}(:,k)$ and $\mathbf{R}_{g2}(:,k)$ are the k-th column of $\mathbf{R}_{g1}$ and $\mathbf{R}_{g2}$, respectively. $\angle$ represents the angle difference between $\mathbf{R}_{g1}(:,k)$ and $\mathbf{R}_{g2}(:,k)$. Then those four fitness functions are optimized in every iterative by changing the value for 3D correspondences of IOCs with NSGA-II.

4.3. Non-dominated sorting

For each genetic iteration, we calculate those aforementioned four fitness values for each individual that is a possible solution for the unknown vector $\mathbf{u}_f$ of 3D correspondences of IOCs [15]. The fitness values of each individual can be used to find the non-dominated set by comparing four fitness values. If all four fitness values of one individual are better than others, then this individual is a non-dominated individual, and repeat this process to find all

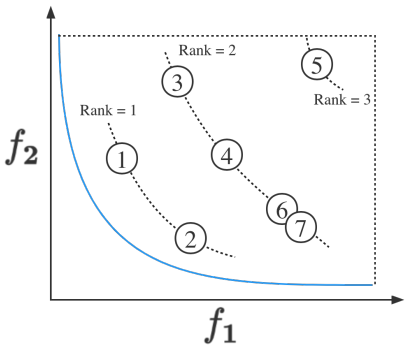

Figure 5. Non-dominated sorting.

the non-dominated individuals. After non-dominated determination [15], there will be several non-dominated fronts shown in Fig. 5 and the individuals in different fronts have a different rank value. The smaller rank value means that the individual is dominated by fewer other individuals. Afterwards, the crowding distance can be calculated for the individuals in the same non-dominated front, and the crowding distance for the edged individuals of the non-dominated fronts is defined as infinity to improve the diversity of the algorithm. The crowding distance is the sum of the distances between two adjacent individuals for four fitness functions, and for the individual k the distance can be defined as d_k^c

$$d_k^c = \sum_{i=1}^{4} \frac{||f_i(k+1) - f_i(k-1)||}{f_i^{max} - f_i^{min}} \quad k = 2, 3, ..., n-1, \tag{11}$$

where f_i^{max} and f_i^{min} are the maximum and minimum for the specific fitness values f_i in each genetic iteration, $f_i(k+1)$ and $f_i(k-1)$ are the two adjacent individuals of $f_i(k)$, and n is the total individual number in one non-dominated front. Then, a new sorted set can be obtained as the parents after sorting by the non-dominated rank and the crowding distance, and the better one has the smaller rank and the bigger crowding distance [15].

4.4. Child individual generation

To obtain the better children in the iterative, the traditional steps of genetic algorithm, selection, crossover, and mutation [47], are used to generate the child individuals. For the selection step, the tournament selection is used, which mainly selects two individuals randomly and then saves the one with the bigger rank value. After this step, the worse solutions will be saved and used for crossover and mutation, which can make the worse one have the possibility to become a better solution.

4.4.1 Child generation using crossover

For the crossover step, a parameter called crossover possibility (p_c) needs to be set, and the crossover will only be done when the random number is smaller than p_c. The simulated binary crossover (SBX) operator [14] is mainly used to generate the child individual. Suppose there are two parent individuals p_1 and p_2, then the children c, can be generated as

$$c = \alpha(p_1 + p_2) + \alpha\beta(p_1 - p_2), \tag{12}$$

where

$$\beta = \begin{cases} (2r)^{1/(1+\eta_c)} & 0 < r <= 0.5, \\ (2(1-r))^{-(1+1/\eta_c))} & 0.5 < r < 1, \end{cases}$$

$$\alpha = \frac{p_2(rank)}{p_2(rank) + p_1(rank)},$$

where r is generated randomly from 0 to 1, and η_c is crossover distribution index for crossover operators [15], whose value is inversely proportional to the amount of perturbation in the design variables. η_c will be chosen according to the specific situation and α can be determined by the rank of p_1 and p_2.

4.4.2 Child generation using mutation

After we get the new generation by crossover, the polynomial mutation can be adopted with a random probability to improve the possibility to get the better solution [14], and the new children c_n, can be obtained by the following equations

$$c_n = ||c_o + (c_{max} - c_{min})\delta||, \tag{13}$$

where

$$\delta = \begin{cases} (2r)^{1/(1+\eta_m)} & 0 < r < 0.5 \\ (2(1-r))^{-(1+1/\eta_m))} & 0.5 <= r < 1 \end{cases}$$

where r is generated randomly from 0 to 1, and η_m is mutation distribution index for mutation operators [15], whose value is inversely proportional to the amount of perturbation in the design variables. η_m will be chosen according to the specific situation. c_o is the individual before mutation, and c_{max} and c_{min} are the maximum and minimum in the range possible values for $\mathbf{u}_f$, respectively.

Those new generated child individuals are merged with the parents, and the merged ones are used to repeat the non-dominated sorting steps mentioned in 4.3. After this, the first N (population size) elements are selected to form the new parents. All the above steps will be repeated as the process displayed in Fig. 4 until we get the quite good result for the 3D correspondences of IOCs and initial camera pose. The results will be represented as a Pareto front, and a threshold is set for all the fitness functions to remove some unreasonable solutions, and then choose the solution with the smallest orthogonal error from the remaining solutions as the result for the further refinement.

4.5. Camera pose optimization via IOC refinement

After 3D correspondences of IOCs and initial camera pose are obtained, they can be jointly refined together, and the optimization problem is converted into a least-squares problem with the unknown 3D correspondences of IOCs and three variables related to the rotation matrix $\mathbf{R}$. From Eq. (2), we have

$$(\mathbf{n}_i^c)^T \mathbf{R} \mathbf{P}_i^w = -(\mathbf{n}_i^c)^T \mathbf{t}, \tag{14}$$

where $\mathbf{R}$ can be represented with the Cayley parameterization vector $\mathbf{s} = [s_1 \ s_2 \ s_3]^T$ [45]. Then, Eq. (14) can be represented as the following matrix form

$$\mathbf{M}\mathbf{r} = \mathbf{N}\mathbf{t}, \tag{15}$$

where $\mathbf{r}$ is constructed by the variable vector about $\mathbf{R}$ and the unknowns for 3D correspondences of IOCs. The unknown coordinate might be on the X-axis, Y-axis, or Z-axis because of the different definition for the world coordinate frame. The common part of the variable vector $\mathbf{r}$ is defined as $\mathbf{cr} = [1, s_1, s_2, s_3, s_1^2, s_1 s_2, s_1 s_3, s_3^2, s_2 s_3, s_3^2]^T$, which is Cayley parameterization form for the rotation matrix. According to three different situations, the added variable vector can be defined as $\mathbf{r}_x$, $\mathbf{r}_y$, or $\mathbf{r}_z$, which can be added to the common part vector and the variable vector will be $\mathbf{r} = [\mathbf{cr}^T \ \mathbf{r}_x^T]^T, [\mathbf{cr}^T \ \mathbf{r}_y^T]^T$, or $[\mathbf{cr}^T \ \mathbf{r}_z^T]^T$, and $\mathbf{M}$ can be represented according to different $\mathbf{r}$ [9]. Eq. (14) is satisfied for n reference point, hence

$$\widetilde{\mathbf{M}}\mathbf{r} = \widetilde{\mathbf{N}}\mathbf{t} \Longleftrightarrow \mathbf{t} = \mathbf{C}\mathbf{r}, \tag{16}$$

where

$$\widetilde{\mathbf{M}} = [\mathbf{M}_1^T, \mathbf{M}_2^T, ..., \mathbf{M}_n^T]^T,$$
$$\mathbf{N} = -(\mathbf{n}^c)^T, \quad \widetilde{\mathbf{N}} = [\mathbf{N}_1^T, \mathbf{N}_2^T, ..., \mathbf{N}_n^T]^T,$$
$$\mathbf{C} = (\widetilde{\mathbf{N}}^\mathbf{T}\widetilde{\mathbf{N}})^{-1}\widetilde{\mathbf{N}}^\mathbf{T}\widetilde{\mathbf{M}},$$

and the least-squares problem can be obtained as follows

$$\hat{\varepsilon} = \sum_{i=1}^{n} ||(\widetilde{\mathbf{M}} - \widetilde{\mathbf{N}}\mathbf{C})\mathbf{r}||^2 = \sum_{i=1}^{n} ||\mathbf{E}\mathbf{r}||^2. \tag{17}$$

However, the traditional Gauss-Newton can be only adopted when the cost function is the 2nd order [3], and this cost function is the 3rd order, so the function order needs to be reduced. This issue is solved by using a re-linearization technique [29]. Let $s_4 = s_1^2, s_5 = s_1 s_2, s_6 = s_1 s_3, s_7 = s_2^2, s_8 = s_2 s_3, s_9 = s_3^2$. Although five more variables are introduced here, five more equations are also added, which allow us to reduce the order successfully. Then, the traditional Gauss-Newton method can be used, which is similar to the camera pose optimization part discussed in 4.1, to refine $\mathbf{R}$ and 3D correspondences of IOCs. After this refinement, the translation vector $\mathbf{t}$ can be determined according to Eq. (16).

The whole proposed method, referred to as CP3L, is presented in Algorithm 1.

Algorithm 1: CP3L method.

Input : Two 2D/3D line correspondences
Output: Rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$

Initialize the population P_{Gen} for $\mathbf{u}_f$ ($Gen = 0$)
Estimate initial $\mathbf{R}_0$ and $\mathbf{t}_0$ with camera height aware P3L method
Evaluate fitness values for each individual using Eq. (9) and (10)
$P_{Gen} \leftarrow$ Non-dominated sorting
$P_{Gen} \leftarrow$ Crowd distance calculation using Eq. (11)
$P_{Gen} \leftarrow$ Sort by non-dominated rank and crowd distance
while $Gen < Gmax$ **do**
  $P_c \leftarrow$ Selection by binary tournament
  $P_c \leftarrow$ Crossover and Mutation using Eq. (12) and (13)
  $\mathbf{R}_{Gen}, \mathbf{t}_{Gen} \leftarrow$ P3L method based on P_c
  $P_c \leftarrow$ Fitness assessment using Eq. (9) and (10)
  $P_m \leftarrow$ Merge P_{Gen} and P_c
  $P_m \leftarrow$ Find non-dominated set
  $P_m \leftarrow$ Calculate the crowding distance using Eq. (11)
  $P_m \leftarrow$ Sort by non-dominated rank and crowd distance
  $Gen \leftarrow$ Gen +1
  $P_{Gen} \leftarrow$ Keep the first N (population size) elements
end while
Choose the result with the smallest orthogonal error from the Pareto front results
Refine 3D correspondences of IOCs and $\mathbf{R}$ using Eq. (17)
Calculate translation vector $\mathbf{t}$ using Eq. (16)
return $\mathbf{R}, \mathbf{t}$

5. Experiment results

The proposed CP3L algorithm is tested and validated thoroughly on both synthetic data and real-world images. All the results are measured by the error metric defined the same as in [9,48], and rotation error ($Err_\mathbf{R}$) and translation error ($Err_\mathbf{t}$) will be calculated as

$$Err_\mathbf{R}(deg) = \max_{k \in 1,2,3} \angle(\mathbf{R}_{true}(:,k), \mathbf{R}(:,k)) \times \frac{180}{\pi},$$
$$Err_\mathbf{t}(\%) = \frac{||\mathbf{t} - \mathbf{t}_{true}||}{||\mathbf{t}_{true}||} \times 100, \tag{18}$$

where $\mathbf{R}_{true}$ and $\mathbf{t}_{true}$ denote the ground-truth for rotation matrix and translation vector, and $\mathbf{R}$ and $\mathbf{t}$ denote the estimate results for rotation matrix and translation vector, respectively. The mean and median of rotation error and translation error will be calculated. For the real images, in addition to the rotation and translation error, the estimated layout lines are drawn according to the estimated 2D point coordinates of IOCs, and the reprojection errors R_{er} are listed under the real image result shown in Fig. 8. For types 1 and 2, NSGA-II is introduced and there are some parameters needed to be determined, including population size, archive size, iteration number, crossover probability, and mutation probability [15]. After experiments, those parameters are confirmed as 100, 50, 100, 0.8, and 0.1, respectively. All methods are implemented in MATLAB on a MacPro with a 2.3 GHz CPU and 8GB of RAM.

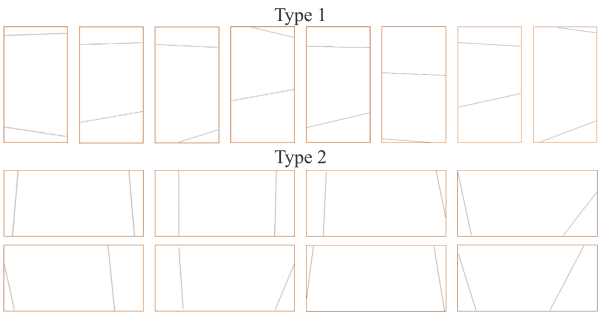

Figure 6. Some randomly generated room layout images.

5.1. Experiments with synthetic data

5.1.1 Synthetic data

A virtual perspective camera with image size of 320×640 pixels and focal length of 100 pixels for type 1, and 640×320 pixels and focal length of 200 pixels for type 2 is used to generate the 3D reference lines. The proper initial camera origin in the camera frame, initial rotation angle, and translation vector are fixed to generate a specific room layout. Then the rotation angle is randomly changed in three different angle directions in the range of [-4, 4] and translation vector is changed in three different directions in the range of [-3, 3] to assure that the generated lines conform to a specific room layout. Then these 3D lines are projected onto the 2D image plane using $\mathbf{R}_{true}$ and $\mathbf{t}_{true}$. Some randomly generated room layouts are shown in Fig. 6.

5.1.2 Different layout results with varying noise

This experiment tests the effects of noise levels on the two different IOC combinations for the two room layouts. The noise deviation level δ is varied from 1 to 10 pixels. At each noise level, 30 independent tests are conducted, and the mean and median errors of rotation and translation are calculated, as shown in Fig. 7. As the noise increases, the rotation errors are increased almost linearly, but the translation errors are less stable. The main reason is that the translation vector is determined from the 3D correspondences of IOCs estimated by NSGA-II, whose errors could propagate to the estimation of the translation vector. Furthermore, for the different IOC combinations in Fig. 3, the results for using the auxiliary lines connecting two IOCs from different layout boundaries are more stable than the other combination. In other words, the results of types 1A and 2A has the better result than those of types 1B and 2B, respectively. It is worth noting that the proposed method cannot handle the case when the two layout boundaries are parallel in the image plane, which is the same constraint in camera pose estimation from vanishing points [5]. Therefore, we avoid this situation when generating the simulated data.

5.1.3 Computational efficiency

The computational time for type 1A and type 2A is quite similar, and the average time is 91 seconds. The computational time for type 1B and type 2B is quite similar, and the average time is 157 seconds. This result means that Group A has the advantage on the computational time, and there is no other mathematical methods to estimate camera pose for this situation. Therefore, our method is also competitive for those room layout types.

5.2. Experiments with Real Images

We also applied the proposed CP3L method on a set of room layout images with a known 3D line model from Matterport3d-Layout Dataset [53]. Matterport3d-Layout Dataset is a large scale dataset with 3D layout ground truth, which has good layout diversity. It also provides depth information that can be used to recover 3D points ground truth with rotation and translation ground truth together. However, the coordinate system for different images is set differently, so we need to figure out the coordinate system setting, then use different equations to estimate camera pose. 17 images for type 1 and 7 images for type 2 are collected. Tab. 1 shows the rotation error and translation error for different combinations listed in Fig. 3, and the proposed method can give a quite accurate result. Moreover, the results with refinement and without refinement are compared, which shows that the polishing step can improve the result. Similar with the result from the synthetic data, the result from combination A is better than it from B for different types. Moreover, Fig. 8 demonstrates that the proposed method can recover the camera pose quite well. From the real image experiment, there is another situation. When there is a short section of a long corridor or a long wall, the range of the 3D correspondences of IOCs cannot be initialized properly, and there will be multiple solutions for this situation.

Type	Mean of the rotation error (deg)	
	With refinement	Without refinement
type 1A (17)	**0.6062**	0.6093
type 1B (17)	1.6181	1.7894
type 2A (7)	**0.2510**	0.2612
type 2B (7)	2.1670	2.2984
Type	**Mean of the translation error (%)**	
	With refinement	Without refinement
type 1A (17)	**3.8215**	3.8554
type 1B (17)	7.2583	7.5320
type 2A (7)	**4.6976**	4.7305
type 2B (7)	6.6223	6.8615

Table 1. The mean rotation and translation errors for types 1 and 2 layouts with two different IOC combinations (Fig. 3). The number in () is the number of images in each case.

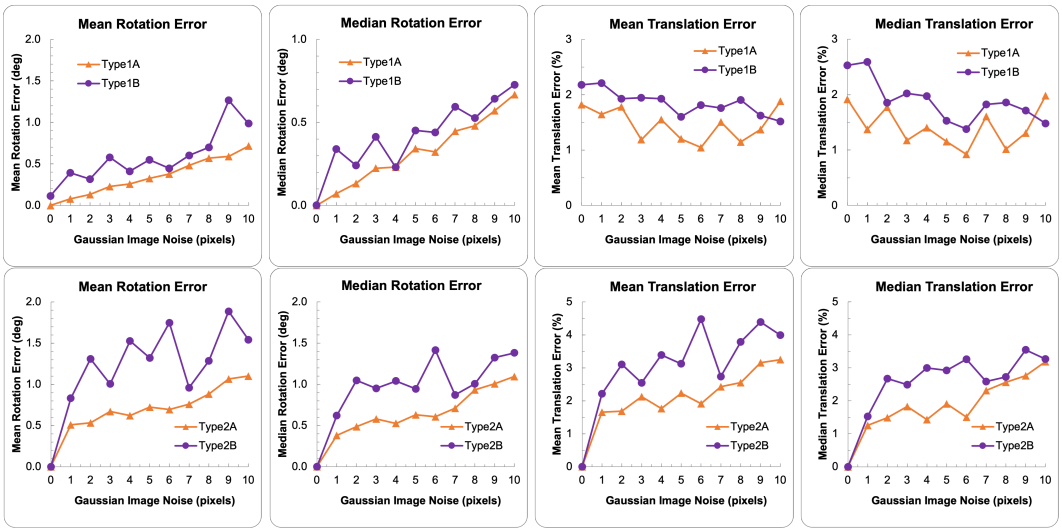

Figure 7. Experimental results on the simulated data under different noise levels ($\delta = 1, ..., 10$). From left to right: the mean/median rotation errors and the mean/median translation errors. From top to bottom, the results for type 1, type 2.

Figure 8. Camera pose estimation from real-world images using our method. R_{er} is the reprojection error.

6. Conclusion

In this work, we study camera pose estimation from coplanar two-line room layouts that are often-seen in egocentric vision applications. The proposed CP3L algorithm is inspired and motivated by the recent PnL-IOC method [9] that still cannot handle this kind of coplanar two-line layouts due to the limited given information. The proposed CP3L incorporates the multi-objective NSGA-II optimization in the P3L method to estimate 3D correspondences of IOCs from a two-line layout that yields two additional line correspondences for a valid P3L solution. *To the best of our knowledge, this is the first attempt to estimate the camera pose for this kind of challenging layouts.* The capability of camera pose estimation from common room layouts enables and facilitates many location-aware egocentric vision applications, such as indoor localization, way-finding, and navigation. Nevertheless, generalization to non-Manhattan room layouts is necessary to make the proposed research applicable to different indoor structures and environments.

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.

References

[1] M. Antunes and J. P. Barreto. A global approach for the detection of vanishing points and mutually orthogonal vanishing directions. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1336–1343, 2013. 2

[2] P. Beardsley and D. Murray. Camera calibration using vanishing points. In David Hogg and Roger Boyle, editors, *BMVC92*, pages 416–425, London, 1992. Springer London. 2

[3] J. V. Burke and M. C. Ferris. A gauss–newton method for convex composite optimization. *Mathematical Programming*, 71(2):179–194, 1995. 6

[4] X. Cai, P. Wang, L. Du, Z. Cui, W. Zhang, and J. Chen. Multi-objective three-dimensional dv-hop localization algorithm with NSGA-II. *IEEE Sensors Journal*, 19(21):10003–10015, 2019. 1

[5] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International journal of computer vision*, 4(2):127–139, 1990. 2, 7

[6] H. Chang and F. Tsai. Vanishing point extraction and refinement for robust camera calibration. *Sensors*, 18(1), 2018. 2

[7] H. Chen. Pose determination from line-to-plane correspondences: existence condition and closed-form solutions. In *Proc. ICCV*, 1990. 2

[8] W. Chen and B. C. Jiang. 3-d camera calibration using vanishing point concept. *Pattern Recognition*, 24(1):57–67, 1991. 2

[9] X. Chen and G. Fan. Egocentric indoor localization from room layouts and image outer corners. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3449–3458, 2021. 1, 2, 3, 6, 8

[10] R. Cipolla, T. Drummond, and D. P. Robertson. Camera calibration from vanishing points in image of architectural scenes. In *BMVC*, volume 99, pages 382–391. Citeseer, 1999. 2

[11] J. M. Coughlan and A. L. Yuille. Manhattan World: Orientation and Outlier Detection by Bayesian Inference. *Neural Computation*, 15(5):1063–1088, 2003. 1

[12] 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. 2

[13] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[14] K. Deb, R. B. Agrawal, et al. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995. 5

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. 1, 4, 5, 6

[16] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *Proc. CVPR*, 2012. 2

[17] 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. 2

[18] 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. 2

[19] J. D. Foley and A. Van Dam. *Fundamentals of interactive computer graphics*. Addison-Wesley Longman Publishing Co., Inc., 1982. 2

[20] L. Grammatikopoulos, G. Karras, and E. Petsa. Camera calibration combining images with two vanishing points. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 35(5):99–104, 2004. 2

[21] E. Guillou, D. Meneveaux, E. Maisel, and K. Bouatouch. Using vanishing points for camera calibration and coarse 3d reconstruction from a single image. *The Visual Computer*, 16(7):396–410, 2000. 2

[22] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2

[23] Z. He, G. Yen, and J. Zhang. Fuzzy-based pareto optimality for many-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 18(2):269–285, 2014. 1

[24] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1849–1856, 2009. 2

[25] Z. Kim. Geometry of vanishing points and its application to external calibration and realtime pose estimation. 2006. 2

[26] J. Kogecka and W. Zhang. Efficient computation of vanishing points. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 1, pages 223–228 vol.1, 2002. 2

[27] V. Kozeny. Genetic algorithms for credit scoring: Alternative fitness function performance comparison. *Expert Systems with applications*, 42(6):2998–3004, 2015. 4

[28] C.-Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. RoomNet: End-to-End Room Layout Estimation. In *Proc. ICCV*, 2017. 1, 2

[29] 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. 6

[30] 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. 2

[31] C. Liu, F. Zhu, J. Ou, and Y. Yu. Z-shaped perspective-three-line problem's unique solution conditions. In *2010 Third International Conference on Intelligent Networks and Intelligent Systems*, pages 132–135, 2010. 2

[32] F. M. Mirzaei and S. I. Roumeliotis. Optimal estimation of vanishing points in a manhattan world. In *2011 International Conference on Computer Vision*, pages 2454–2461, 2011. 2

[33] K. Mitra. Multiobjective optimization of an industrial grinding operation under uncertainty. *Chemical Engineering Science*, 64(23):5043–5056, 2009. 1

[34] N. Mohan and M. Kumar. Room layout estimation in indoor environment: a review. *Multimedia Tools and Applications*, pages 1–31, 2021. 2

[35] Y. Y. Moon, Z. W. Geem, and G.-T. Han. Vanishing point detection for self-driving car using harmony search algorithm. *Swarm and evolutionary computation*, 41:111–119, 2018. 2

[36] M. Orouskhani, D. Shi, and X. Cheng. A fuzzy adaptive dynamic NSGA-II with fuzzy-based borda ranking method and its application to multimedia data analysis. *IEEE Transactions on Fuzzy Systems*, 29(1):118–128, 2021. 1

[37] R. Pio. Euler angle transformations. *IEEE Transactions on Automatic Control*, 11(4):707–715, 1966. 3

[38] Y. Ren, S. Li, C. Chen, and C.-C. J. Kuo. A Coarse-to-Fine Indoor Layout Estimation (CFILE) Method. In *Proc. ACCV*, 2017. 1

[39] A. Rituerto, A. C. Murillo, and J. J. Guerrero. 3d layout propagation to improve object recognition in egocentric videos. In *European Conference on Computer Vision*, pages 839–852. Springer, 2014. 1

[40] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *Proc. CVPR*, 2012. 2

[41] M. D. Shuster. A survey of attitude representations. *Navigation*, 8(9):439–517, 1993. 4

[42] Y. Sun, F. Lin, and H. Xu. Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II. *Wireless Personal Communications*, 102, 09 2018. 1

[43] B. S. Tjan, P. J. Beckmann, R. Roy, N. Giudice, and G. E. Legge. Digital sign system for indoor wayfinding for the visually impaired. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pages 30–30. IEEE, 2005. 1

[44] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *Proc. ECCV*, 2010. 2

[45] 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. 3, 4, 6

[46] Y. Wang. An efficient algorithm for uav indoor pose estimation using vanishing geometry. In *MVA*, pages 361–364. Citeseer, 2011. 2

[47] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85, 1994. 5

[48] 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. 2, 3, 4, 6

[49] 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. 1, 2

[50] L. Zhang, C. Xu, K.-M. Lee, and R. Koch. Robust and Efficient Pose Estimation from Line Correspondences. In *Proc. ACCV*, 2013. 2

[51] W. Zhang, W. Zhang, K. Liu, and J. Gu. Learning to predict high-quality edge maps for room layout estimation. *IEEE Transactions on Multimedia*, 19(5):935–943, 2016. 1

[52] W. D. Zhang, W. Zhang, K. Liu, and J. Gu. Learning to predict high-quality edge maps for room layout estimation. *IEEE Transactions on Multimedia*, 19(5):935–943, 2017. 2

[53] W. D. Zhang, W. Zhang, and Y. Zhang. Geolayout: Geometry driven room layout estimation based on depth maps of planes. *CoRR*, abs/2008.06286, 2020. 1, 2, 7

[54] 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. 2

[55] L. Zhao, C. Wu, and J. Ning. A camera calibration method based on two orthogonal vanishing points. *Concurrency and Computation: Practice and Experience*, 26(5):1185–1199, 2014. 2

[56] L. Zhou, Y. Yang, M. Abello, and M. Kaess. A robust and efficient algorithm for the pnl problem using algebraic distance to approximate the reprojection distance. In *AAAI*, 2019. 2