

Camera Calibration Using Parallel Line Segments

Gaku Nakano

Central Research Labs, NEC Corporation

Kawasaki, Japan, 211-8666

Email: g-nakano@nec.com

Abstract—In this paper, we propose a camera calibration method for surveillance cameras that uses the image projection of parallel 3D line segments of the same length. We assume that vertical line segments are perpendicular to the ground plane and their bottom end points are on the ground plane. Under this assumption, the camera parameters can be directly determined from at least two line segments without estimating vanishing points. By extending the minimal solution, we devise a closed-form solution to the least squares case with more than two line segments. Lens distortion is jointly optimized in bundle adjustment. Evaluation of synthetic data showed that the optimal depression angle of a camera is around 50 degrees. In real data evaluation, we used the joints of pedestrians as vertical line segments. The experimental results on public datasets showed that the proposed method used with a human pose detector can accurately calibrate wide-angle cameras that have radial distortion.

I. INTRODUCTION

Surveillance camera calibration has been an important topic for industrial applications of computer vision. Both intrinsic and extrinsic parameters must be accurately calibrated to recognize what is happening in a 3D scene from recorded 2D images or videos, *e.g.* pedestrian analysis, traffic estimation. A general approach for camera calibration is to use a calibration object such as a checkerboard [1] or a box [2]. However, these approaches are often impractical for surveillance cameras because they require a calibration object large enough to cover the entire filmed area. In addition, traffic may need to be stopped or restricted during the calibration process if surveillance cameras are already installed.

To calibrate surveillance cameras without calibration objects, vanishing point-based methods have been proposed [3]–[8]. Using vanishing points for camera calibration is one of the classical approaches in computer vision [9]. For traffic surveillance cameras, Dubska *et al.* [3] used the direction of moving vehicles and the edges of the vehicles to detect vanishing points. Evans and Ferryman [4] estimated vanishing points from the foot and head positions of pedestrians observed by multiple cameras. For single camera calibration, Kusakuniran *et al.* [6] formulated a camera projection matrix from the vertical vanishing point and the horizontal vanishing line, which were estimated from a pedestrian’s trajectory. Huang *et al.* [5] estimated vanishing points by analyzing periodic foot movement, assuming that pedestrians walk in a straight line at a constant speed. Liu *et al.* [7] showed that vanishing points can be estimated from pedestrians on a noisy foreground mask based on prior knowledge about the distribution of the pedestrians’ relative heights.

The difference between the above methods is essentially their approach for estimating vanishing points. Reliable and accurate vanishing point estimation has been a classical issue in computer vision that is still being investigated today [10], [11]. This is because parallel 3D lines projected onto an image generally do not intersect at a single point due to image noise and limited image resolution, even when using the edges of buildings or road lanes. Therefore, estimating vanishing points from pedestrians is considerably more challenging than when man-made structures are available.

Another important issue not mentioned in the above works is lens distortion. Vanishing point-based methods implicitly assume that lens distortion is negligible. However, it is not unusual for surveillance cameras to have a wide-angle lens to include as much of the scene as possible. In such cases, vanishing point estimation may fail because straight lines become curved [12].

To overcome the above two issues, we propose a camera calibration method suitable for surveillance cameras. Our only assumption is that parallel 3D line segments of the same length are on the ground plane. We do not use any 3D information about the position of the line segments. The proposed method has the following two advantages compared with the vanishing point-based methods: 1) Without using vanishing points, the focal length and extrinsic parameters are directly derived by solving the projective equation of line segments. 2) The radial distortion can be estimated by performing bundle adjustment. We demonstrate the performance of the proposed method by conducting both synthetic and real data experiments. The real data experiments showed that the proposed method can successfully estimate camera parameters of wide-angle cameras by using human body joints as the line segments.

II. RELATED WORK

Calibration using 1-D Objects. Zhang [13] first proposed a camera calibration method using 1D objects consisting of three colinear points. The method assumes that one of the end points is fixed at a single point and the other end points are freely moving around the fixed point. Hammarstedt *et al.* [14] thoroughly analyzed the degeneracy, or the critical motions, of 1D objects. Wu *et al.* [15] proved that camera calibration with 1D objects is feasible when the objects follow a planar motion without fixing an end point. Due to the limitations of the object motion, 2D plane-based methods using a checkerboard [1] have been standard for single camera calibration rather than methods using 1D objects.

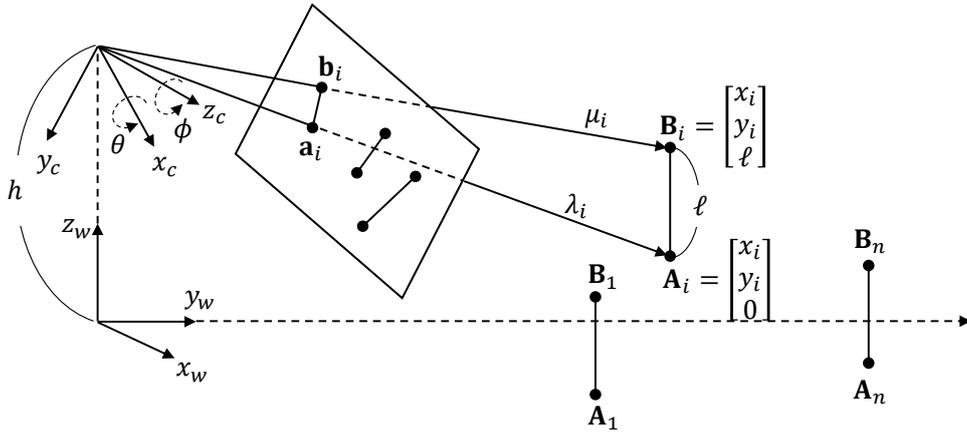


Fig. 1. Illustration of camera parameter estimation from parallel line segments.

Vanishing point estimation in radially distorted images.

Wildenauer and Micusik [16] derived a closed-form solution to estimate radial distortion from a vanishing point. For urban environments, the Manhattan world assumption is frequently used for simultaneously estimating vanishing points and radial distortion [10], [17].

Structure from motion using line segments. Line segments have been widely used to determine structure from motion [18], [19] and visual SLAM [20], [21] in poorly textured scenes where feature points are not available. To improve robustness in various environments, point-line based methods, which integrate both point and line information, have been proposed [22], [23].

III. PRELIMINARIES

A. Problem Formulation

In this paper, we refer to Figure 1 to explain the calibration problem and basic equations. Without loss of generality, we can set the origin of the world coordinate system on the ground plane ($z_w = 0$) and place a camera at height h on the z_w -axis. Then, the camera direction, or the z_c -axis, can be set parallel to the y_w -axis of the world coordinate system. In this configuration, the extrinsic parameter of the camera is defined by the position $[0, 0, h]^T$ and two rotational angles θ and ϕ around the x_w - and z_w -axes, respectively [7].

The surveillance camera is assumed to be a modern digital camera with zero skew, equal aspect ratio, and the principal point at the center of the image. Thus, focal length f is the only intrinsic parameter that needs to be estimated.

All line segments have the same length ℓ and are perpendicular to the ground plane, *i.e.* vertical line segments¹. Each line segment consists of two distinct end points, the bottom point $\mathbf{A}_i = [x_i, y_i, 0]^T$ and the top point $\mathbf{B}_i = [x_i, y_i, \ell]^T$. The only prior knowledge about the line segments is that they are the

¹For simplicity, we only discuss vertical line segments in this paper; however, our methodology is not limited to vertical line segments but can be applied to parallel line segments in any direction.

same length ℓ^2 . Their locations on the ground plane (x_i, y_i) are not given.

A pair of end points \mathbf{A}_i and \mathbf{B}_i of an i -th line segment are observed as image points \mathbf{a}_i and \mathbf{b}_i , respectively. The projective equation can be written by

$$\begin{aligned} \lambda_i \mathbf{a}_i &= \mathbf{K}(\mathbf{R}\mathbf{A}_i + \mathbf{t}), \\ \mu_i \mathbf{b}_i &= \mathbf{K}(\mathbf{R}\mathbf{B}_i + \mathbf{t}), \end{aligned} \quad (1)$$

where

$$\mathbf{K} = \text{diag}(f, f, 1), \quad (2)$$

$$\mathbf{R} = \mathbf{R}_z(\phi) \mathbf{R}_x(\theta)$$

$$= \begin{bmatrix} \cos \phi & -\cos \theta \sin \phi & \sin \theta \sin \phi \\ \sin \phi & \cos \theta \cos \phi & -\sin \theta \cos \phi \\ 0 & \sin \theta & \cos \theta \end{bmatrix}, \quad (3)$$

$$\mathbf{t} = -\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} = -h\mathbf{r}_3, \quad (4)$$

λ_i, μ_i : projective depth.

Note that the image points \mathbf{a}_i and \mathbf{b}_i are represented as their homogeneous coordinates, 3×1 vectors.

B. Number of Line Segments Needed to Solve Problem

The goal of the calibration problem shown in Eq. (1) is to find the intrinsic parameter (focal length f), extrinsic parameters (camera height h and rotational angles θ and ϕ), projective depth (λ_i and μ_i), and the position (x_i and y_i) and length ℓ of each line segment.

Given n line segments, the total number of unknown variables is $4n + 5$, which is $4n$ for $\lambda_i, \mu_i, x_i, y_i$ and 5 for θ, ϕ, f, h, ℓ . However, the degrees of freedom is actually $4n + 4$ due to the scale ambiguity between $\lambda_i, \mu_i, x_i, y_i, h, \ell$. Since the projection of end points, \mathbf{a}_i and \mathbf{b}_i , are represented by 3×1 vectors of the homogeneous coordinates, a line segment on an image plane gives us 6 constraints. Thus, the number of line segments needed to solve the calibration problem is $n \geq 2$.

²The proposed method does not necessarily require the absolute value of ℓ . See Section IV-C

IV. PROPOSED METHOD

A. Minimal Case: 2 Line Segments

This section describes the solution to the calibration problem with $n = 2$ line segments. First, we find the projective depth. Then, we recover the focal length followed by the length of the line segments, the rotation matrix, the 3D coordinates of the end points, and the translation vector.

When observing two line segments, we can write four constraint equations based on Eq. (1) as follows:

$$\lambda_1 \mathbf{a}_1 = K(x_1 \mathbf{r}_1 + y_1 \mathbf{r}_2 - h \mathbf{r}_3), \quad (5)$$

$$\lambda_2 \mathbf{a}_2 = K(x_2 \mathbf{r}_1 + y_2 \mathbf{r}_2 - h \mathbf{r}_3), \quad (6)$$

$$\mu_1 \mathbf{b}_1 = K(x_1 \mathbf{r}_1 + y_1 \mathbf{r}_2 + \ell \mathbf{r}_3 - h \mathbf{r}_3), \quad (7)$$

$$\mu_2 \mathbf{b}_2 = K(x_2 \mathbf{r}_1 + y_2 \mathbf{r}_2 + \ell \mathbf{r}_3 - h \mathbf{r}_3). \quad (8)$$

where \mathbf{r}_j denotes the j -th column of \mathbf{R} . Subtracting Eq. (7) from Eq. (5) and Eq. (8) from Eq. (6), we obtain

$$\mu_1 \mathbf{b}_1 - \lambda_1 \mathbf{a}_1 = \ell \mathbf{K} \mathbf{r}_3, \quad (9)$$

$$\mu_2 \mathbf{b}_2 - \lambda_2 \mathbf{a}_2 = \ell \mathbf{K} \mathbf{r}_3. \quad (10)$$

Since the right-hand side of both equations are equal each other, we have a linear equation in the projective depth $\lambda_1, \lambda_2, \mu_1, \mu_2$:

$$-\lambda_1 \mathbf{a}_1 + \lambda_2 \mathbf{a}_2 + \mu_1 \mathbf{b}_1 - \mu_2 \mathbf{b}_2 = \mathbf{0} \quad (10)$$

or in matrix form

$$\mathbf{M} \mathbf{v} = \begin{bmatrix} -\mathbf{a}_1 & \mathbf{a}_2 & \mathbf{b}_1 & -\mathbf{b}_2 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \mu_1 \\ \mu_2 \end{bmatrix} = \mathbf{0}. \quad (11)$$

Here, \mathbf{M} is a 3×4 matrix of rank three. Therefore, \mathbf{v} can be determined as the nullspace vector of \mathbf{M} up to scale and sign. The scale ambiguity of the nullspace vector corresponds to the scale ambiguity between $\lambda_i, \mu_i, x_i, y_i, h, \ell$. We cannot determine the scale without the absolute value of the scene. To resolve the sign ambiguity, we consider the orientation of the projective depth, all of which are positive in this paper, as shown in Figure 1. Hence, we can obtain a unique solution of $\lambda_1, \lambda_2, \mu_1, \mu_2$ by

$$\mathbf{v} = \text{abs}(\text{null}(\mathbf{M})). \quad (12)$$

Substituting $\lambda_1, \lambda_2, \mu_1, \mu_2$ into Eqs. (5)–(8), we obtain the following two equations:

$$\mathbf{K}^{-1}(\mu_1 \mathbf{b}_1 - \lambda_1 \mathbf{a}_1) = \ell \mathbf{r}_3, \quad (13)$$

$$\mathbf{K}^{-1}(\lambda_2 \mathbf{a}_2 - \lambda_1 \mathbf{a}_1) = (x_2 - x_1) \mathbf{r}_1 + (y_2 - y_1) \mathbf{r}_2.$$

The orthogonality condition of rotation matrix, $\mathbf{R}^T \mathbf{R} = \mathbf{I}$, leads to a constraint equation in f as follows:

$$\mathbf{c}^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{d} = \ell(x_2 - x_1) \mathbf{r}_3^T \mathbf{r}_1 + \ell(y_2 - y_1) \mathbf{r}_3^T \mathbf{r}_2 = 0 \quad (14)$$

where $\mathbf{c} = \mu_1 \mathbf{b}_1 - \lambda_1 \mathbf{a}_1$ and $\mathbf{d} = \lambda_2 \mathbf{a}_2 - \lambda_1 \mathbf{a}_1$. Letting c_j and d_j be the j -th element of \mathbf{c} and \mathbf{d} , respectively, focal length f can be given by the closed-form

$$f = \sqrt{\frac{c_1 d_1 + c_2 d_2}{c_3 d_3}}. \quad (15)$$

Because of the norm constraint $\|\mathbf{r}_3\| = 1$, we can obtain the line segment length ℓ and the third column of the rotation matrix \mathbf{r}_3 by

$$\ell = \|\mathbf{K}^{-1} \mathbf{c}\|, \quad \mathbf{r}_3 = \frac{1}{\ell} \mathbf{K}^{-1} \mathbf{c}. \quad (16)$$

From Eq. (3), the two rotational angles can be calculated by

$$\theta = \cos^{-1} r_{33}, \quad \phi = \tan^{-1} \left(-\frac{r_{13}}{r_{23}} \right), \quad (17)$$

where $\mathbf{r}_3 = [r_{13}, r_{23}, r_{33}]^T$. By back-substituting θ and ϕ into Eq. (3), we can recover the rotation matrix.

The rest of unknown variables are the positions of the line segments (x_i, y_i) , camera height h , and translation vector \mathbf{t} . As described in Eq. (4), the translation vector \mathbf{t} can be calculated if the camera height h is given. In other words, we need to solve for x_i, y_i , and h .

Eqs. (5) and (6) can be rewritten in matrix form:

$$\begin{bmatrix} \lambda_1 \mathbf{a}_1 & \lambda_2 \mathbf{a}_2 \end{bmatrix} = \mathbf{K} \mathbf{R} \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ -h & -h \end{bmatrix}. \quad (18)$$

Hence, we can obtain the position (x_i, y_i) and the camera height h by

$$\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ h & h \end{bmatrix} = \mathbf{Q} \mathbf{R}^T \mathbf{K}^{-1} \begin{bmatrix} \lambda_1 \mathbf{a}_1 & \lambda_2 \mathbf{a}_2 \end{bmatrix}, \quad (19)$$

where $\mathbf{Q} = \text{diag}(1, 1, -1)$.

B. Least Squares Case: $n > 2$ Line Segments

This section derives a solution to the least squares case by extending the minimal solution described in the previous section.

Given $n > 2$ line segments, we can obtain a linear equation in the i -th projective depth similar to Eq. (10):

$$-\lambda_1 \mathbf{a}_1 + \lambda_i \mathbf{a}_i + \mu_1 \mathbf{b}_1 - \mu_i \mathbf{b}_i = \mathbf{0}. \quad (20)$$

By stacking n of these equations, we can build a minimization problem for finding the projective depth

$$\min_{\mathbf{v}} \|\mathbf{M} \mathbf{v}\|^2, \quad \text{s.t. } \mathbf{v} > 0, \quad (21)$$

where

$$\mathbf{M} = \begin{bmatrix} -\mathbf{a}_1 & \mathbf{a}_2 & & \mathbf{b}_1 & -\mathbf{b}_2 & & \\ \vdots & & \ddots & \vdots & & \ddots & \\ -\mathbf{a}_1 & & & \mathbf{a}_n & \mathbf{b}_1 & & -\mathbf{b}_n \end{bmatrix}, \quad (22)$$

$$\mathbf{v} = [\lambda_1 \quad \cdots \quad \lambda_n \quad \mu_1 \quad \cdots \quad \mu_n]^T.$$

Since \mathbf{M} is of size $3(n-1) \times 2n$, we can solve it with $n > 2$ line segments in a least-square sense. We can solve Eq. (21) by using a non-negative least squares method, which is implemented as `nnls` in SciPy and `lsqnonneg` in MATLAB. If we replace the positive constraint $\mathbf{v} > 0$ with a unit norm constraint $\|\mathbf{v}\|^2 = 1$ to take a simpler approach, the DLT method [9] can be applied to determine \mathbf{v} as the eigenvector of $\mathbf{M}^T \mathbf{M}$ associated with the smallest eigenvalue.

To find the focal length, we can extend Eq. (14) to obtain an optimization problem as follows:

$$\min_f \sum_{i=2}^n (\mathbf{c}_i^\top \mathbf{K}^{-\top} \mathbf{K}^{-1} \mathbf{d}_i)^2, \quad \text{s.t. } f > 0, \quad (23)$$

where $\mathbf{c}_i = \mu_i \mathbf{b}_i - \lambda_i \mathbf{a}_i$ and $\mathbf{d}_i = \lambda_i \mathbf{a}_i - \lambda_1 \mathbf{a}_1$. Similarly to the discussion on Eq. (21), we can solve f in the closed-form without considering the constraint $f > 0$,

$$f = \sqrt{-\frac{\sum (c_{i,1} d_{i,1} + c_{i,2} d_{i,2})(c_{i,3} d_{i,3})}{\sum (c_{i,3} d_{i,3})^2}}, \quad (24)$$

where $c_{i,j}$ and $d_{i,j}$ denote the j -th elements of the vectors \mathbf{c}_i and \mathbf{d}_i , respectively.

Based on Eq. (16), the third column of the rotation matrix \mathbf{r}_3 can be obtained by solving

$$\min_{\mathbf{r}_3} \sum_{i=1}^n \left\| [\mathbf{K}^{-1} \mathbf{c}_i]_{\times} \mathbf{r}_3 \right\|^2, \quad \text{s.t. } \|\mathbf{r}_3\| = 1, \quad (25)$$

where $[\]_{\times}$ is a matrix representation of the vector cross product. Hence, \mathbf{r}_3 can be computed as the eigenvector of $\sum -[\mathbf{K}^{-1} \mathbf{c}_i]_{\times}^2$ corresponding to the smallest eigenvalue.

By inserting \mathbf{r}_3 into Eq. (16), we can formulate ℓ as

$$\min_{\ell} \sum_{i=1}^n \left\| \ell \mathbf{r}_3 - \mathbf{K}^{-1} \mathbf{c}_i \right\|^2, \quad \text{s.t. } \ell > 0. \quad (26)$$

If we ignore the positive constraint, the solution can be written in the closed-form

$$\ell = \mathbf{r}_3^\top \left(\frac{1}{n} \sum_{i=1}^n \mathbf{K}^{-1} \mathbf{c}_i \right). \quad (27)$$

Finally, the position (x_i, y_i) and the camera height h can be estimated by solving

$$\min_{x_i, y_i, h} \sum_{i=1}^n \left\| \begin{bmatrix} x_i & x_i \\ y_i & y_i \\ h & h - \ell \end{bmatrix} - [\mathbf{p}_i \quad \mathbf{q}_i] \right\|^2 \quad (28)$$

s.t. $h > 0$,

where $[\mathbf{p}_i \quad \mathbf{q}_i] = \mathbf{Q} \mathbf{R}^\top \mathbf{K}^{-1} [\lambda_i \mathbf{a}_i \quad \mu_i \mathbf{b}_i]$. We disregard the positive constraint $h > 0$ and obtain the solution in the closed-form

$$\begin{aligned} x_i &= \frac{1}{2}(p_{i,1} + q_{i,1}), \\ y_i &= \frac{1}{2}(p_{i,2} + q_{i,2}), \\ h &= \frac{1}{2n} \sum_{i=1}^n (p_{i,3} + q_{i,3} + \ell), \end{aligned} \quad (29)$$

where $p_{i,j}$ and $q_{i,j}$ are the j -th elements of the vectors \mathbf{p}_i and \mathbf{q}_i , respectively.

C. Recovering Absolute Length

As mentioned in Section III-B, the proposed method does not handle absolute value due to scale ambiguity. In other words, the proposed method estimates the ratio between $\lambda_i, \mu_i, x_i, y_i, \ell, h$. The absolute value cannot be recovered unless one of the real sizes is known, e.g. $h_{\text{real}} \leftarrow \ell_{\text{real}} / \ell h$. Even if one of the absolute values is given in advance, the proposed method can still be applied because the degree of freedom is still $4n + 4$. In practical situations, we can utilize vertical bars with known length, such as utility poles and road marks, as well as people with known height.

D. Bundle Adjustment with Lens Distortion

The proposed method in Sections IV-A and IV-B finds the solution by minimizing algebraic error, that is not physically meaningful. To refine the parameters, we perform bundle adjustment to minimize the reprojection error.

Although we have not discussed lens distortion, it should be estimated because many surveillance cameras have a wide-angle lens to capture a broad area in practical applications. To deal with lens distortion, we initialize the distortion parameter as zero and optimize it by bundle adjustment.

The minimization of reprojection error is formulated by

$$\min_{\substack{f, k_1, \mathbf{R}, \mathbf{t} \\ x_i, y_i, h}} \sum_{i=1}^n \text{dist}(\mathbf{a}_i, \mathbf{A}_i)^2 + \text{dist}(\mathbf{b}_i, \mathbf{B}_i)^2 \quad (30)$$

where k_1 is the first term of radial distortion and $\text{dist}(\cdot, \cdot)$ is a function for calculating the L2-distance between an observed image point and the corresponding 3D point projected onto the image plane according to the camera parameters. Note that we exclude the length of the line segments ℓ in Eq. (30) to fix the scale.

In Section IV-B, we showed that the least squares case can be solved by either of the non-negative least squares methods or the closed-form solutions. Though both methods can be used to estimate the initial guess for bundle adjustment, the non-negative least squares methods are preferable from a theoretical point of view. The computational cost of non-negative least squares is greater than that of the closed-form solutions, but there are no significant numerical differences in the final output from bundle adjustment as long as we have conducted preliminary experiments. For this reason, we use the closed-form solutions in the experiments.

E. Degeneracy

In this section, we discuss the degeneracy of the proposed method. When degeneracy occurs, we cannot estimate or improve the camera parameters even with additional line segments. According to Eq. (21), the degeneracy condition can be analytically determined by finding image points or 3D points satisfying $\text{rank}(\mathbf{M}) < 2n - 1$. Due to limitations of space, we do not derive rigorous mathematical proof; rather, we only show examples that can happen in practical situations.

In a simple situation, any two of the four end points are colinear. This can be observed if two line segments are located

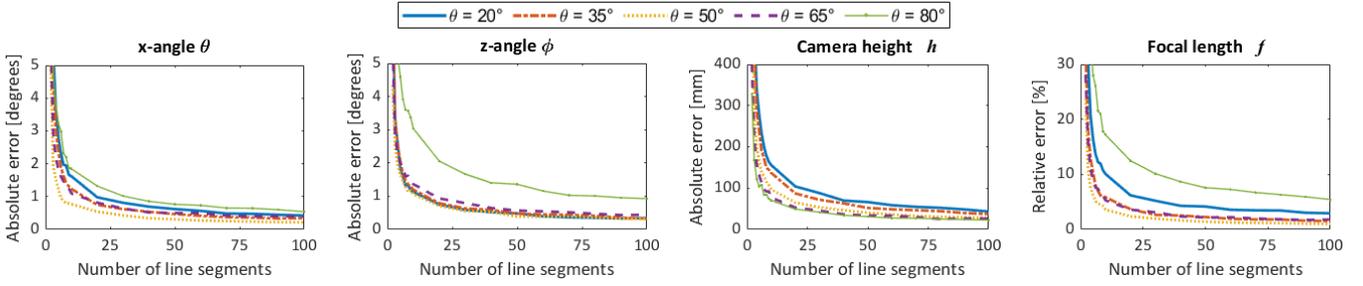


Fig. 2. Median error with respect to the number of line segments. $2 \leq n \leq 100$ and $\sigma = 2.0$. 1000 independent trails for each n .

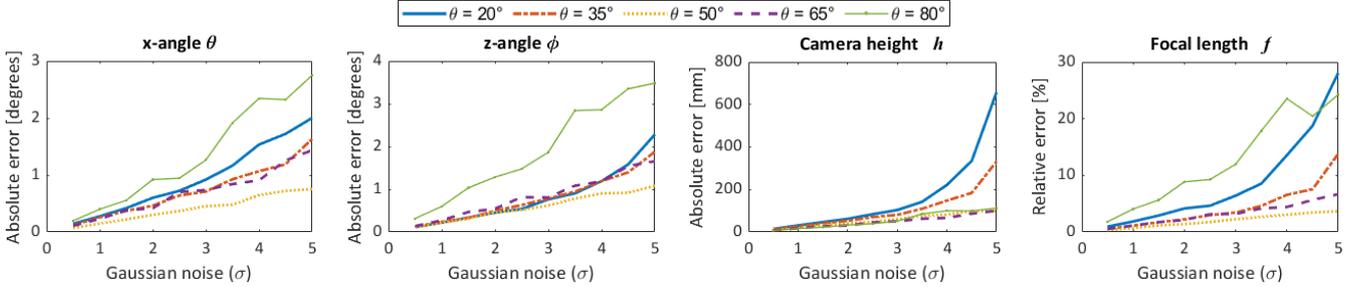


Fig. 3. Median error with respect to noise level on image points. $n = 50$ and $0.5 \leq \sigma \leq 10$. 1000 independent trails for each σ .

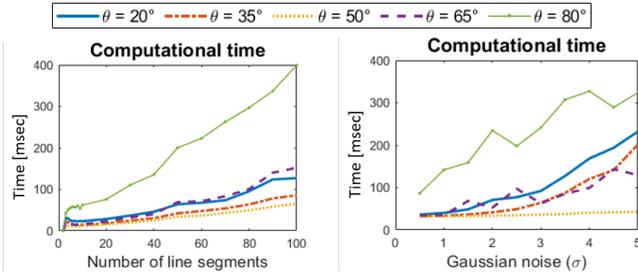


Fig. 4. Mean computational time.

V. EXPERIMENTS

This section reports the experimental results of the proposed method. First, we conducted a synthetic data experiment to evaluate the numerical accuracy of the proposed method with respect to variations in the number of the line segments, image noise sensitivity, and computational time. Then, we performed three real data experiments: 1) quantitative comparison with the existing method using a checkerboard pattern [1], 2) quantitative evaluation on publicly available datasets by combining with a human pose detector, and 3) planar image rectification as another application of the proposed method. We implemented the proposed method on MATLAB and ran all experiments on a PC with Core i9-7020X.

A. Synthetic Data Evaluation

We quantitatively evaluated the performance of the proposed method on synthetic data with the ground truth values. For the simulation, we set the camera to $f = 400$ [pixels] with the image resolution 640×480 [pixels], which is a wide-angle camera having 77.3° horizontal field of view (HFOV). In the generated 3D scenes, the camera was located on $h = 2.5$ [m] with rotational angles $\phi = 0^\circ$ and $\theta \in \{20^\circ, 35^\circ, 50^\circ, 65^\circ, 80^\circ\}$. We configured various θ to check the stability against depression angles. Vertical line segments of the length $\ell = 0.5$ [m] were randomly generated on the ground plane in the field of view range. Smaller θ enables the camera to shoot wider range of area where line segments are generated far from the origin.

First, we measured the estimation accuracy with respect to the number of line segments. We generated $2 \leq n \leq 100$ line segments and added Gaussian image noise with zero mean and $\sigma = 2.0$ [pixels] standard deviation to the projected image points. Figure 2 shows the median error of 1000 independent

close to each other in 3D space. Furthermore, if the camera direction (x_c -axis) is close to $\theta = 90^\circ$, two end points are projected onto 2D positions very close to each other on the image plane. In another case, if the camera direction is approximately $\theta = 0^\circ$ and almost perpendicular to vertical line segments, their end points on the image plane become colinear. In these situations, we can avoid degeneracy if other line segments in a different direction are available, *e.g.* horizontal line segments.

Interestingly, the proposed method is not degenerate if line segments are on a quadratic curve, such as a circle in 3D space, which is one of the critical conditions for calibration methods using 1D objects [13], [14]. Calibration methods using 1D objects assume that one of the objects' end points shares a single 3D point. On the other hand, for the proposed method, line segments can be scattered as long as their bottom point is on the same plane. Therefore, the proposed method is considerably more flexible for practical applications.



Fig. 5. Part of the images used for a comparison with a checkerboard calibration method. The image resolution is 960×720 .

TABLE I
ESTIMATION RESULT OF INTRINSIC PARAMETERS USING THE CHECKERBOARD IMAGES SHOWN IN FIGURE 5. NUMBERS WITH AN ASTERISK WERE NOT OPTIMIZED IN BUNDLE ADJUSTMENT.

Method	f	k_1	(c_x, c_y)
OpenCV	981.7	-0.0501	(487.0, 369.2)
	974.2	-0.0485	(480.0*, 360.0*)
Proposed closed-form w/ BA	966.2	0.0*	(480.0*, 360.0*)
	980.7	-0.0470	(480.0*, 360.0*)

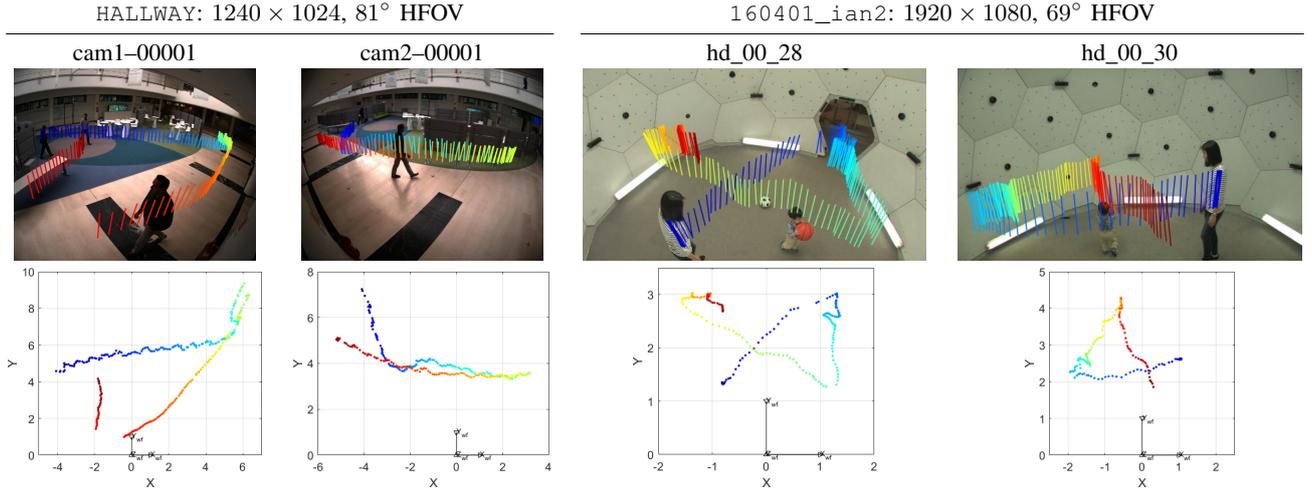


Fig. 6. Visualization of real data experiment. *Top*: Vertical line segments, *i.e.* connection between neck and mid-hip detected by OpenPose [24]. The change in line color represents the time course. *Bottom*: 3D trajectory of pedestrian reconstructed by proposed method. Each color dot corresponds to the line of the same color in the top images.

trials for each n . The proposed method demonstrated the highest accuracy for $35^\circ \leq \theta \leq 65^\circ$ except for the camera height. However, the results also indicate that the accuracy of $\theta \in \{20^\circ, 80^\circ\}$ can be improved if we use $n > 100$ line segments.

Next, we evaluated the robustness by varying image noise, where $n = 50$ and $0.5 \leq \sigma \leq 5$. The results of over 1000 independent trials are shown in Figure 3. For a small noise level $\sigma \leq 3$, there were no significant differences between various θ . Similarly to the previous experiment, the proposed method with $\theta = 50^\circ$ shows the highest robustness.

Finally, we report computational time of the proposed method. Figure 4 shows the mean runtime of the above two tests. Whereas $\theta = 50^\circ$ was the fastest and most stable against image noise, the runtime of $\theta = 80^\circ$ was significantly longer than that of the other angles. The cause for the long runtime is likely an inaccurate initial guess for bundle adjustment. However, all results were less than 500 msec which is sufficiently fast enough for practical applications.

Given to the above results, θ of about 50° is suitable for the proposed method. This coincides with the well-known fact reported in the literature [1] that camera calibration methods

using a checkerboard perform reliably when tilting the board about 45° towards a camera.

B. Real Data Evaluation

1) *Comparison with checkerboard method*: We compared intrinsic parameter estimation by the proposed method with the existing method using a checkerboard [1] implemented in OpenCV. A camera was positioned at around $h \approx 40$ [cm] high and rotated around $\theta \approx 45^\circ$ and $\phi \approx 0^\circ$. Since the above configuration was manually set and the OpenCV method calculates extrinsic parameters for each image independently, we compared only intrinsic parameters in this experiment.

We placed a 9×6 checkerboard pattern on a desk so that the grid pattern was perpendicular to the desk plane. Figure 5 shows four of 14 captured images which have a resolution of 960×720 . The OpenCV method used all detected corners with known 3D and 2D coordinates, *i.e.* 54 corners per image. Meanwhile, the proposed method only used top and bottom corners with image coordinates, *i.e.* nine line segments per image. As default values, we fixed the principal point at the image center: $(c_x, c_y) = (480, 360)$.

TABLE II
 QUANTITATIVE RESULTS OF THE REAL DATA SHOWN IN FIGURE 6. NUMBERS WITH AN ASTERISK WERE NOT OPTIMIZED IN BUNDLE ADJUSTMENT.

Sequence	Method	f	(k_1, k_2, k_3)	(c_x, c_y)	θ [deg]	ϕ [deg]	h [m]
HALLWAY cam1-00001	Ground truth	747.7	$(-0.3559, 0.1542, -0.0328)$	$(646.2, 518.5)$	27.73	6.894	2.780
	Proposed w/ BA	715.8	$(-0.2169, 0.0^*, 0.0^*)$	$(640.0^*, 512.0^*)$	28.20	1.665	2.731
HALLWAY cam2-00001	Ground truth	731.0	$(-0.3615, 0.1560, -0.0331)$	$(650.4, 518.2)$	30.35	1.399	2.788
	Proposed w/ BA	713.8	$(-0.2129, 0.0^*, 0.0^*)$	$(640.0^*, 512.0^*)$	27.65	1.665	2.659
160401_ian2 hd_00_28	Ground truth	1396.5	$(-0.2867, 0.1860, -0.0509)$	$(955.0, 562.6)$	40.02	2.261	3.128
	Proposed w/ BA	1386.4	$(-0.2038, 0.0^*, 0.0^*)$	$(960.0^*, 540.0^*)$	41.77	-0.722	3.150
160401_ian2 hd_00_30	Ground truth	1407.9	$(-0.2838, 0.1887, -0.0537)$	$(948.2, 562.5)$	12.78	-3.616	1.974
	Proposed w/ BA	1330.4	$(-0.2462, 0.0^*, 0.0^*)$	$(960.0^*, 540.0^*)$	11.93	-3.158	1.944

Table I shows the estimated values of focal length f and the first radial distortion term k_1 . Although only image points were available, the proposed method successfully estimates the parameters including radial distortion comparable with the OpenCV method.

2) *Evaluation on open dataset:* We evaluated the proposed method on two publicly available datasets: HALLWAY sequence from Task Decomposition dataset [25]³ and 160401_ian2 sequence from Panoptic Studio dataset [26]⁴. The two datasets provide various video sequences of people performing simple actions and social interactions captured by calibrated multi-view cameras. We chose a total of four videos, two from each dataset, that show people walking around.

First, we selected a pedestrian in each video and used OpenPose [24] to detect 2D keypoints of the pedestrians (neck and mid-hip) through all video frames, as shown in the top row of Figure 6. The detected points were treated as the two end points of a vertical line segment. Then, we estimated the camera parameters as well as the 3D position of the line segments in the absolute scale based on the pedestrian’s height, which was measured by triangulation using the provided calibration data. The bottom row of Figure 6 visualizes the reconstructed trajectories of the pedestrian for each video. Note that the proposed method utilizes the absolute length of the line segments only for solving the scale ambiguity. For trajectory visualization, we did not use any 3D points obtained by the above triangulation procedure.

Table II summarizes the quantitative result. On the 160401_ian2 sequence, we successfully obtained both intrinsic and extrinsic parameters close to the ground truth even though the proposed method deals with partial intrinsic parameters. The estimated intrinsic parameters, especially the radial distortion term, for the HALLWAY sequence seemed less accurate. However, the undistorted images in Figure 7 indicate that the difference in their visual quality is minimal.

These results indicate that we can calibrate surveillance cameras in the absolute scale by observing a pedestrian with known height. In addition, the proposed method is more

feasible in practice than the existing methods [3]–[8] which do not consider lens distortion.

3) *Planar image rectification:* This section describes an application of the proposed method other than camera calibration. As described in Section IV, the proposed method estimates (x_i, y_i) coordinates and the length ℓ . Since two bottom points and their corresponding top points are obtained, we can calculate the homography matrix for metric rectification of planar images.

Figure 8 shows the qualitative results of planar image rectification on Stanford Mobile Visual Search dataset [27]⁵. We manually selected two parallel line segments on the images. It is widely known that metric rectification can be recovered using vanishing points [9]. However, the proposed method can directly perform metric rectification without estimating vanishing points.

VI. CONCLUSION

We proposed a camera calibration method that uses parallel line segments of the same length and is suitable for surveillance cameras. We showed that at least two line segments provide sufficient constraints for determining focal length and extrinsic parameters. We also extended the least squares case for more than two line segments. The radial lens distortion was additionally optimized in bundle adjustment. Furthermore, we discussed degeneracy conditions that cause unstable calibration. We conducted both synthetic and real data experiments to evaluate the proposed method. The synthetic data experiments showed that the proposed method is stable at a depression angle of about 50 degrees, which is a similar condition for the existing calibration method using a planar pattern. In the real data experiments, we used a pose detector to detect the backbone of pedestrians as vertical line segments. We demonstrated that the proposed method correctly calibrated wide-angle cameras which was previously not feasible with the existing methods that use vanishing points.

³https://tev-static.fb.com/DATABASES/TASK_DECOMPOSITION.html

⁴<http://dome.db.perception.cs.cmu.edu/>

⁵<https://purl.stanford.edu/rb470rw0983>



Fig. 7. Undistorted images of HALLWAY sequence using the intrinsic parameter in Table II. *Top*: Ground truth. *Bottom*: Proposed method.

REFERENCES

- [1] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.
- [2] R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.
- [3] M. Dubska, A. Herout, R. Juranek, and J. Sochor, "Fully automatic roadside camera calibration for traffic surveillance," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 3, pp. 1162–1171, June 2015.
- [4] M. Evans and J. Ferryman, "Surveillance camera calibration from observations of a pedestrian," 10 2010, pp. 64 – 71.
- [5] S. Huang, X. Ying, J. Rong, Z. Shang, and H. Zha, "Camera calibration from periodic motion of a pedestrian," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3025–3033.
- [6] W. Kusanniran, H. Li, and J. Zhang, "A direct method to self-calibrate a surveillance camera by observing a walking pedestrian," in *2009 Digital Image Computing: Techniques and Applications*. IEEE, 2009, pp. 250–255.
- [7] J. Liu, R. T. Collins, and Y. Liu, "Surveillance camera autocalibration based on pedestrian height distributions," in *British Machine Vision Conference (BMVC)*, vol. 2, 2011.
- [8] J. Sochor, R. Juranek, and A. Herout, "Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement," *Computer Vision and Image Understanding*, vol. 161, pp. 87 – 98, 2017.
- [9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [10] M. Antunes, J. P. Barreto, D. Aouada, and B. Ottersten, "Unsupervised vanishing point detection and camera calibration from a single manhattan image with radial distortion," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [11] H. Li, J. Zhao, J.-C. Bazin, W. Chen, Z. Liu, and Y.-H. Liu, "Quasi-globally optimal and efficient vanishing point estimation in manhattan world," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- [12] F. Devernay and O. Faugeras, "Straight lines have to be straight," *Machine vision and applications*, vol. 13, no. 1, pp. 14–24, 2001.
- [13] Z. Zhang, "Camera calibration with one-dimensional objects," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 7, pp. 892–899, 2004.
- [14] P. Hammarstedt, P. Sturm, and A. Heyden, "Closed-form solutions and degenerate cases for camera calibration with one-dimensional objects," in *10th International Conference on Computer Vision (ICCV'05)*, vol. 1. IEEE Computer Society, 2005, pp. 317–324.



Fig. 8. Qualitative results of planar image rectification on Stanford Mobile Visual Search dataset.

- [15] F. Wu, Z. Hu, and H. Zhu, "Camera calibration with moving one-dimensional objects," *Pattern Recognition*, vol. 38, no. 5, pp. 755–765, 2005.
- [16] H. Wildenauer and B. Micusik, "Closed form solution for radial distortion estimation from a single vanishing point," in *BMVC*, vol. 1, 2013, p. 2.
- [17] H. Chang and F. Tsai, "Vanishing point extraction and refinement for robust camera calibration," *Sensors*, vol. 18, no. 1, p. 63, 2018.
- [18] A. Bartoli and P. F. Sturm, "Multiple-view structure and motion from line correspondences," in *ICCV*, vol. 3, 2003, p. 207.
- [19] G. Schindler, P. Krishnamurthy, and F. Dellaert, "Line-based structure from motion for urban environments," in *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*. IEEE, 2006, pp. 846–853.
- [20] K. Hirose and H. Saito, "Fast line description for line-based slam," in *BMVC*, 2012, pp. 1–11.
- [21] D. Ruifang, V. Frémont, S. Lacroix, I. Fantoni, and L. Changan, "Line-based monocular graph slam," in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2017, pp. 494–500.
- [22] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PI-slam: Real-time monocular visual slam with points and lines," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4503–4508.
- [23] S. J. Lee and S. S. Hwang, "Elaborate monocular point and line slam with robust initialization," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- [24] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.
- [25] T. Hu, S. Messelodi, and O. Lanz, "Wide-area multi-camera multi-object tracking with dynamic task decomposition," in *Proceedings of the International Conference on Distributed Smart Cameras*. ACM, 2014, p. 7.
- [26] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social motion capture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3334–3342.
- [27] V. R. Chandrasekhar, D. M. Chen, S. S. Tsai, N.-M. Cheung, H. Chen, G. Takacs, Y. Reznik, R. Vedantham, R. Grzeszczuk, J. Bach *et al.*, "The stanford mobile visual search data set," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 117–122.