Single View Camera Calibration for Augmented Virtual Environments

Lu Wang* Suya You† Ulrich Neumann‡

Computer Graphics and Immersive Technologies Laboratory
University of Southern California

ABSTRACT

Augmented Virtual Environments (AVE) are very effective in the application of surveillance, in which multiple video streams are projected onto a 3D urban model for better visualization and comprehension of the dynamic scenes. One of the key issues in creating such systems is to estimate the parameters of each camera including the intrinsic parameters and its pose relative to the 3D model. Existing camera pose estimation approaches require known intrinsic parameters and at least three 2D to 3D feature (point or line) correspondences. This cannot always be satisfied in an AVE system. Moreover, due to noise, the estimated camera location may be far from the expectation of the users when the number of correspondences is small. Our approach combines the users' prior knowledge about the camera location and the constraints from the parallel relationship between lines with those from feature correspondences. With at least two feature correspondences, it can always output an estimation of the camera parameters that gives an accurate alignment between the projection of the image (or video) and the 3D model.

Keywords: Camera calibration, pose estimation, augmented virtual environment.

1 INTRODUCTION

In this paper camera calibration refers to estimating the intrinsic and extrinsic parameters of a camera. It is a key step in many augmented reality systems. In an Augmented Virtual Environment (AVE) [6], multiple live videos captured by surveillance cameras mounted on the top of buildings are projected onto a 3D urban model. Instead of watching these videos separately, the users can visualize them in a 3D context simultaneously, which helps the users better comprehend the dynamic scenes. To set up such a system, one of the main problems is to calibrate each camera so that the projection of the video transmitted from it can accurately align with the 3D model. Unlike camera tracking problems, the cameras in this application are static.

This calibration typically depends on the feature (point or line) correspondences between the 3D model and an image frame captured by each camera. The existing approaches can be classified into two groups. The first one, such as the DLT algorithm [4], computes the camera projection matrix by minimizing the reprojection error with least-square techniques and then decomposes it into the intrinsic calibration matrix and the extrinsic pose. To be stable, these approaches require a large number of 2D to 3D feature correspondences (at least 6). This can be hardly satisfied in an AVE system because the camera field of view is limited and the model that is generally constructed from aerial images or airborne LIDAR data lacks 3D details (See Fig.2 and Fig.3).

*e-mail: luwang@graphics.usc.edu
†e-mail: Suyay@graphics.usc.edu
‡e-mail:uneumann@graphics.usc.edu

IEEE Virtual Reality Conference 2007
March 10 - 14, Charlotte, North Carolina, USA
1-4244-0906-3/07/$20.00 ©2007 IEEE

In the second group of approaches, the intrinsic parameters of a camera are estimated beforehand with some calibration rigs, e.g. by the method in [8], and are assumed to be fixed. The camera pose is then estimated from at least three point or line correspondences between the image taken at the current viewpoint and the 3D scene [1, 3, 5]. The camera tracking algorithms in many augmented reality systems belong to this category [2, 7]. However, the two-step scheme increases the user interaction. More importantly, the intrinsic parameters of a camera in an AVE system may change after its installation, especially for PTZ cameras since the users can control their settings. In addition, when the number of available feature correspondences is small, the pose estimation is very sensitive to noise. Although the reprojection error can be quite small, the estimated camera location may be far from the place where the users think the camera is installed. This looks weird when the users try to display the cameras in the 3D model.

In general, the users know the approximate camera location. For instance, the camera is known to be mounted along an edge on the rooftop of a building. Our approach exploits this prior knowledge, which can not only make the calibration result more accurate but also reduce the number of required feature correspondences since the degree of freedom of the camera location is reduced. In the system, the users indicate the possible range of the X and Y coordinates of the camera location by drawing a circle or rectangle on the ground plane of the 3D model and directly input the range of its height. With at least two 2D to 3D feature correspondences, the approach can always give an estimation of the intrinsic and extrinsic parameters with the camera location inside the input range that minimizes the reprojection error.

Moreover, besides the classical feature correspondences, some additional constraints can also be used. For example, sometimes the counterpart of an image line cannot be found in the 3D model but it is parallel to a known 3D line. How to use this information is not clear in existing approaches whereas it can be easily incorporated into our framework. This helps to improve the calibration accuracy.

The remainder of this paper is organized as follows: Section 2 describes the details of the method. Section 3 demonstrates some experimental results including simulations and real tests. We conclude in Section 4.

2 METHOD

Initially, the camera is assumed to be a natural camera (i.e.unit aspect ratio, no skew and the principal point is at the image center). There are 7 parameters to be estimated including the focal length, camera location and rotation. The rest intrinsic parameters are refined in a final optimization. To obtain an initial estimation, an exhaustive search is applied. It is impractical to search directly in the 7-D parameter space. Our strategy is to search in the 4-D subspace spanned by the focal length and the camera location. At each sample point, the possible camera rotation is analytically calculated. More details are given below.

2.1 Camera rotation estimation

Given the camera location and focal length, the camera rotation can be computed analytically from two 2D to 3D point or line correspondences. As shown in Fig.1(a), O is the camera center. X,Y,Z

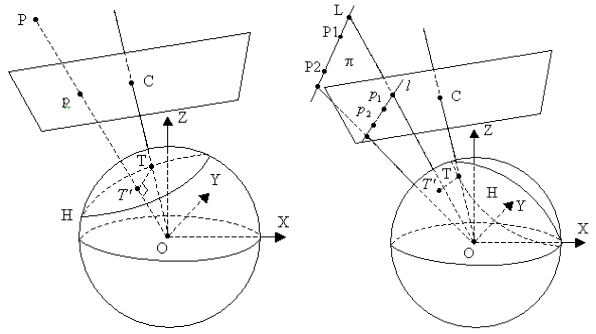
are directions parallel to the three coordinate axes of the world frame respectively. Image point p corresponds to 3D point P. OC is the principal axis of the camera and T is its intersection with the unit sphere centered at O. T' is the projection of T onto OP. The line equation of OP is determined by the camera location and the coordinates of P in the world frame. From the image coordinates of p and the focal length, the angle $\angle COP$ and the distance $|T'O|$ can be computed. Therefore, T is on a circle that is the intersection of H and the unit sphere, where H is a plane perpendicular to OP with its distance from O as $|T'O|$. Denote the coordinates of T as (t_x, t_y, t_z). The following equations exist:

$$\mathbf{H}^{\mathrm{T}}(t_x, t_y, t_z, 1) = 0, \tag{1}$$

$$\mathbf{H} = (\frac{\mathbf{P}-\mathbf{O}}{|\mathbf{P}-\mathbf{O}|}, -|T'O|), \tag{2}$$

$$|T'O| = \frac{f}{\sqrt{|\mathbf{p}-\mathbf{c}|^2 + f^2}}, \tag{3}$$

where $\mathbf{P}$ and $\mathbf{O}$ are the coordinates of P and O in the world frame. $\mathbf{p}$ and $\mathbf{c}$ are the image coordinates of p and the image center respectively. f is the focal length.



(a) (b)

Figure 1: (a)Point correspondence. (b)Line correspondence.

A similar constraint on point T can be obtained from a line correspondence. As shown in Fig.1(b), l is the image projection of a 3D line L. L is represented by two 3D points P_1 and P_2, and l by two image points p_1 and p_2. The equation of the plane π passing through L and O is determined by the camera location and the coordinates of P_1 and P_2 in the world frame. T' is the projection of T onto π. From the focal length and the line equation of l in the image, the distance $|TT'|$ can be computed. Therefore, T is on a circle that is the intersection of H and the unit sphere, where H is the plane parallel to π with its distance from O as $|TT'|$. We have the following equations:

$$\mathbf{H}^{\mathrm{T}}(t_x, t_y, t_z, 1) = 0, \tag{4}$$

$$\mathbf{H} = (\frac{(\mathbf{P_2}-\mathbf{O}) \times (\mathbf{P_1}-\mathbf{O})}{|(\mathbf{P_2}-\mathbf{O}) \times (\mathbf{P_1}-\mathbf{O})|}, -|TT'|), \tag{5}$$

$$|TT'| = \frac{d}{\sqrt{|f^2 + d^2|}}, \tag{6}$$

where $\mathbf{P_1}$, $\mathbf{P_2}$ and $\mathbf{O}$ are the coordinates of P_1, P_2 and O in the world frame. d is the distance from the image center to l in the image plane. Note that the two end points of L should be ordered so that the vector $(\mathbf{P_2}-\mathbf{O}) \times (\mathbf{P_1}-\mathbf{O})$ in Eq.5 points toward the image center C. To guarantee this, the direction of $\overrightarrow{P_2P_1}$ should correspond to $\overrightarrow{p_2p_1}$ and the vector $(\mathbf{p_2}-\mathbf{c}) \times (\mathbf{p_1}-\mathbf{c})$ is in the direction from O to C, where $\mathbf{p_1}$, $\mathbf{p_2}$ and $\mathbf{c}$ are the image coordinates of p_1, p_2 and the image center respectively.

Therefore, one point or line correspondence gives a linear constraint on T, the intersection of the principal axis with the unit sphere. With two feature correspondences, the coordinates of T can be fully determined by:

$$\begin{cases} t_x^2 + t_y^2 + t_z^2 = 1; \\ \mathbf{H_1}^{\mathrm{T}}(t_x, t_y, t_z, 1) = 0; \\ \mathbf{H_2}^{\mathrm{T}}(t_x, t_y, t_z, 1) = 0. \end{cases} \tag{7}$$

If $\mathbf{H_1} \neq \mathbf{H_2}$, the above equation has at most two solutions. After the coordinates of T are obtained, the direction of the principal axis in the world frame is known. The left degree of freedom of the camera rotation is the one around the principal axis. Its angle can be calculated from one of the feature correspondences by rotating the image frame around the principle axis so that the image projection of the 3D feature coincides with its 2D counterpart. By verifying the consistency of such angles computed from both feature correspondences, only one of the two solutions of Eq.7 is kept.

2.2 Searching strategy and the complete algorithm

Our approach searches in the 4-D space spanned by the camera location and focal length. The users indicate the horizontal (X and Y directions) searching range of the camera location by drawing a circle or a rectangle on the ground plane of the 3D model. Typically, its area is smaller than $50 \times 50m^2$. The range of the camera height from the ground plane is also input by the users and generally it is within 0 to $50m$. The searching step in each dimension of the camera location is $1m$ in our experiments. The typical focal length is between 200 and 10000 pixels. Its searching step is adaptive and computed by $f/50$, where f is the current focal length. Therefore, the total number of samples in the searching space is generally within 2×10^7.

At each sample point, the possible camera rotation is computed from two feature correspondences with the method presented in section 2.1. To obtain a more accurate estimation, two features with the maximum distance in the image are chosen for this purpose. After this, the camera projection matrix is known. The geometric error of the feature correspondences is then estimated. For a point correspondence, the geometric error is defined as the angle between the vector from the camera center to the 3D point and the vector from the camera center to the image point. The geometric error of a line correspondence is defined to be the angle between the plane passing the camera center and the 3D line and the plane passing the camera center and the image line.

Besides feature correspondences, some extra constraints can also be used. For instance, sometimes the 3D counterpart of an image line cannot be located in the 3D model but it is parallel to a known 3D direction. If there are multiple such image lines that are parallel to the same direction, their intersection is the corresponding vanishing point. This vanishing point and the point at infinity on that direction can be used as a usual point correspondence. If only one such image line is found, the available constraint is that the vanishing point of the 3D direction should be on the image line. The geometric error of this constraint is defined to be the angle between the plane passing through the camera center and the image line and the vector from the camera center to the point at infinity on the 3D direction. The reason that we define the geometric error with angles is to make the error measurement for vanishing points comparable with that for ordinary features.

The average geometric error of all available constraints can be expressed as:

$$E = \frac{1}{N}\left(\sum_i \theta(p_i, P_i) + \sum_j \theta(l_j, L_j) + \sum_k \theta(l_k, d_k)\right), \tag{8}$$

where N is the total number of constraints. The three items in the outer brace are the geometric error of all the point correspondences, all the line correspondences and all the parallelism constraints respectively.

Our approach finds the sample point in the searching space with the minimum geometric error. The camera parameters computed at this sample point are used to initialize an iterative non-linear optimization to further reduce the geometric error. It is also in this stage the other intrinsic parameters including aspect ratio, skew and principal point are refined. Levenberg-Marquardt algorithm is applied as the optimizer in our experiments [4].

In practice, the following strategy can be used to improve the searching speed. The searching for the camera location is started from the center point of the range input by the users. Generally, it is the most likely camera location. The sample points with the camera location closer to this center point are checked earlier. Once the average geometric error at a sample point is found to be smaller than a threshold (3 degree), the non-linear optimizer will be called. At this stage, the maximum number of its iterations is limited to be 10 in our experiments. If the resultant average geometric error is already small enough (< 0.3 degree), this sample point will be returned. Otherwise, the searching will continue until it returns at some sample point or finally finds the one with the minimum geometric error. In the end, the non-linear optimizer will be called again with a large upper bound of the iteration number. Moreover, a soft constraint is added in this optimization to guarantee that the camera location is not too far from the center point of its searching range.

The complete calibration algorithm is summarized as follows:

1. The users provide the horizontal range of the camera location by drawing a circle or a rectangle on the ground plane of the 3D model and input the range of the camera height.

2. 2D to 3D feature correspondences and the parallel relationship between image lines and 3D directions are manually selected. Two feature correspondences with the maximum distance in the image are selected for the camera rotation estimation.

3. Sample the searching space of the camera location and the focal length. The sample points are sorted in the increasing order of their distance from the center point of the searching range of the camera location. Initialize $min(E) = \infty$. Start searching.

4. If all of the sample points have been visited, go to step 6. Otherwise, take the next sample point and compute the possible camera rotation. Denote the camera parameters obtained at this sample point as Γ'. Estimate its average geometric error E. If $E < min(E)$, set $min(E) = E$ and $\Gamma = \Gamma'$.

5. If $E < 3$, Levenberg-Marquardt algorithm is called with Γ' as the initial value. The maximum number of iterations is set to 10. After the optimization, if $E < 0.3$, go to step 6; otherwise, go to step 4.

6. Run Levenberg-Marquardt optimizer with Γ as the initial value. The parameters to be refined include all the intrinsic and extrinsic parameters.

3 Experimental results

We did extensive experiments including simulations and tests on real images. In simulations, zero-mean Gaussian noise is added to the coordinates of the 2D and 3D points. By changing its variance, the stability of our approach can be examined. Table 1 shows the result of one simulation in which the calibration is based on two point and two line correspondences. The size of the simulated image is 640×480. The horizontal searching range of the camera location is a circle with $(-82.64, -40.97)$ as the center and $25m$ as the radius. The camera height is searched from 0 to $40m$. σ_1 is the standard deviation of the Gaussian noise added to the coordinates of the 3D points. It changes from $0.25m$ to $2m$, representing the typical error range of a 3D model. σ_2 is the standard deviation of the Gaussian noise added to the image coordinates. It varies between 1 and 4 pixels. f is the focal length. r_x, r_y and r_z are the Euler angles (in radians) of the camera rotation. (t_x, t_y, t_z) is the camera location. Table 1 shows the comparison between the ground truth of these parameters and the output of our approach. M is the average reprojection error of the feature correspondences that is measured as the average distance (in pixels) between the image projection of the 3D features and their 2D counterparts. Since the number of feature correspondences is small, the estimated camera parameters have obvious deviation from their true values even at the lowest noise level. However, the reprojection error is small enough, which means the image is aligned with the 3D model very well. This satisfies the requirement in an AVE system.

Table 2 shows the result of another simulation in which two point and one line correspondences are used and the camera location is limited on a 3D line, simulating the case where the camera is installed on one edge of a building's rooftop.

In real experiments, we used the 3D model of a campus created from aerial images. As shown in Fig.2(a), the green circle is drawn by the user as the horizontal searching range of the camera location. Fig.2(b) and Fig.2(c) illustrate the feature correspondences between the 3D model and the image. P corresponds to 2D point p. 3D lines L_2 and L_3 correspond to 2D lines l_2 and l_3 respectively. The corresponding 3D lines of l_1 and l_4 cannot be located in the 3D model but we know that they are parallel to L_1 and L_2 respectively. With the calibration result, the image can be projected onto the 3D model. Fig.2(d) and Fig.2(e) are two synthetic views. Although the available feature correspondences are quite few (only 3) in this example, the image can still be aligned with the 3D model very well. It demonstrates the advantage of our approach to effectively exploit the constraints from the parallel relationship between lines.

Another real test is shown in Fig.3. Four point correspondences (red points in Fig.3(b) and Fig.3(c)) and four line correspondences (yellow lines in Fig.3(b) and Fig.3(c)) are available. However, they are all gathered in a small area of the image and the error from the 3D model is nonnegligible. Without the constraint on the camera location, the estimated camera location of the traditional pose estimation algorithms may be far from the one expected by the users. This is illustrated in Fig.3(a). The green rectangle indicates the region within which the user thinks the camera is installed while the red dot represents the camera location obtained with the method in [1]. The white dot shows the camera location estimated by our approach, which is inside the green rectangle. Fig.3(d) and Fig.3(e) are two views of the image projection.

In practice, our algorithm is fast. On a general PC, it returns in two seconds in most cases. In the worst case we have encountered, it outputs the solution in 10 seconds.

4 Conclusion

The challenges in the single view camera calibration in AVE systems come from the limited number of 2D to 3D feature correspondences and the inaccuracy of the 3D model. In addition, the intrinsic camera parameters are not always available. Since the purpose is for visualization, the requirement of this calibration is to accurately align the image (or video) with the 3D model while the precision of each individual camera parameter is not very important. However, the estimated camera location should not be far from the place where the users think the camera is installed.

To satisfy the above requirements, our approach integrates linear searching, the technique to find analytical solution and non-linear

	Ground Truth	$\sigma_1 = 0.25$ $\sigma_2 = 1$	$\sigma_1 = 0.50$ $\sigma_2 = 2$	$\sigma_1 = 1.00$ $\sigma_2 = 3$	$\sigma_1 = 2.00$ $\sigma_2 = 4$
f	563.0	594.3	620.4	560.0	479.2
r_x	-0.081	-0.098	-0.101	-0.124	-0.161
r_y	-0.339	-0.328	-0.327	-0.347	-0.362
r_z	-0.058	-0.044	-0.052	-0.027	-0.036
t_x	-73.86	-73.34	-72.11	-75.08	-74.37
t_y	-30.67	-31.97	-31.97	-30.03	-30.51
t_z	29.24	32.85	35.96	26.13	18.18
M	-	0.4	2.1	3.0	6.2

Table 1: Simulation 1. σ_1 (in meters) and σ_2 (in pixels) are the standard deviation of the Gaussian noise added to the coordinates of the 3D and 2D points respectively. f is the focal length (in pixels). r_x, r_y, r_z are the Euler angles (in radians) of the camera rotation. (t_x, t_y, t_z) is the camera location (in pixels). M is the average reprojection error (in pixels).

	Ground Truth	$\sigma_1 = 0.25$ $\sigma_2 = 1$	$\sigma_1 = 0.50$ $\sigma_2 = 2$	$\sigma_1 = 1.00$ $\sigma_2 = 3$	$\sigma_1 = 2.00$ $\sigma_2 = 4$
f	502.0	512.3	550.4	602.3	412.2
r_x	0.892	1.172	0.610	1.576	0.578
r_y	0.176	0.571	0.233	-0.450	0.314
r_z	0.628	0.348	0.454	0.732	0.405
t_x	-10.32	-11.14	-12.31	-15.08	-18.50
t_y	-10.51	-9.25	-10.37	-11.03	-9.80
t_z	8.97	10.01	11.12	14.13	18.40
M	-	0.7	1.3	4.1	10.2

Table 2: Simulation 2. σ_1 and σ_2 are the standard deviation of the Gaussian noise added to the coordinates of the 3D and 2D points respectively. f is the focal length. r_x, r_y, r_z are the Euler angles of the camera rotation. (t_x, t_y, t_z) is the camera location. M is the average reprojection error.

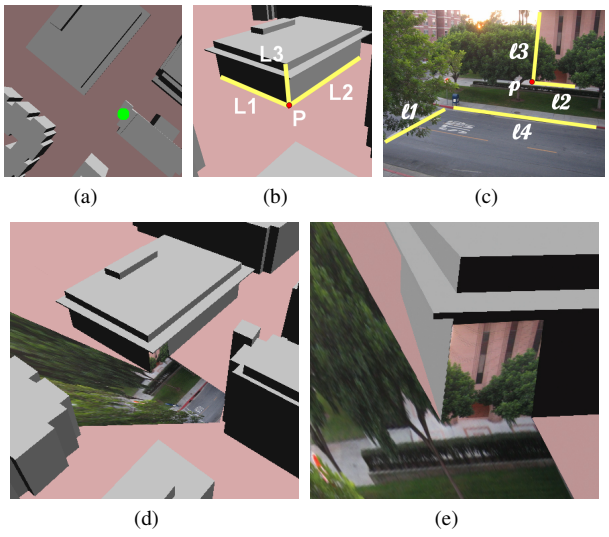


Figure 2: (a):The green circle is the horizontal searching range of the camera location. (b)-(c):Three feature correspondences and two pairs of parallel lines. (d)-(e):Two synthetic views of the image projection.

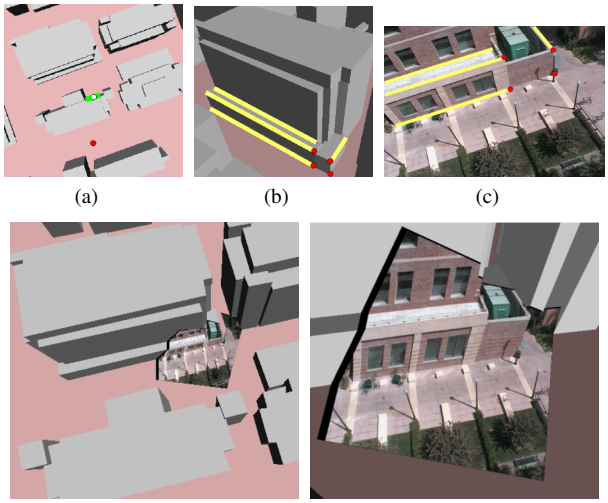


Figure 3: (a):The green rectangle is the horizontal searching range of the camera location. (b)-(c):Four point correspondences and four line correspondences. (d)-(e):Two synthetic views of the image projection.

optimization. It utilizes the prior knowledge of the users about the approximate camera location. Some additional constraints such as the parallel relationship between lines can also be effectively incorporated into the calibration process. Therefore the result is more accurate and the number of feature correspondences required to uniquely determine the camera parameters can be reduced.

The approach is robust with regard to noise. With at least two feature correspondences, it can always give a solution that satisfies all the given constraints (as long as $\mathbf{H_1} \neq \mathbf{H_2}$ in Eq.7). Although the solution may not be unique when the number of constraints is too small, it will not affect the purpose of visualization.

REFERENCES

[1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003.

[2] G.Simon, V.Lepetit, and M.-O.Berger. Computer vision methods for registration: mixing 3d knowledge and 2d correspondences for accurate image composition. *Proceedings of the first international workshop on augmented reality*, pages 111–127, 1998.

[3] R. M. Haralic, C. N. Lee, K. Ottenberg, and M. Nlle. Analysis and solutions of the three points perspective pose estimation problem. *Computer Vision and Pattern Recognition Conference*, pages 592–598, 1991.

[4] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge, second edition, 2003.

[5] Y. Liu, T. S. Huang, and O. D. Faugeras. Determination of camera location from 2-d to 3-d line and point correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, 1990.

[6] U. Neumann, S. You, J. Hu, B. Jiang, and J. W. Lee. Augmented virtual environments (ave): dynamic fusion of imagery and 3d models. *IEEE Virtual Reality*, pages 61–67, 2003.

[7] P. Santos, A. Stork, A. Buaes, and J. Jorge. Ptrack: introducing a novel iterative geometric pose estimation for a marker-based single camera tracking system. *IEEE Virtual Reality*, pages 143–150, 2006.

[8] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.