

An Interactive Scene Annotation Tool for Video Surveillance

Wenze HU^{1,2}, Jianting WEN^{2,3}, Haifeng GONG^{2,4}, Yongtian WANG¹

¹Beijing Institute of Technology, Beijing, China

²Lotus Hill Research Institute, Ezhou, Hubei, China

³State Key Lab of Remote Sensing Science, Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China

⁴Dept. of Statistics, University of California, Los Angeles, USA

{wzhu.lhi, jianting.wen, hfgong.lhi}@gmail.com, wyt@bit.edu.cn

Abstract

An interactive scene annotation tool for video surveillance is presented in this paper. The annotation process is divided into three stages. 1) Camera rough calibration; 2) Calibration refinement; 3) Major surfaces annotation. Inputs are then rendered in a 3D environment, which again help users check calibration accuracy and annotation correctness. Experiments show that this tool is easy to use and attains acceptable annotation accuracy. The interactive procedure helps users without knowledge in computer vision to complete camera calibration as well as surface annotation.

1. Introduction

In order to develop high performance video surveillance systems, researchers fused 3D scene context into various modules of surveillance systems [2] [3] and received considerable performance improvement. This paper mainly address the issue of how to conveniently extract 3D information from surveillance scenes.

In this paper, 3D scene context mainly refers to camera calibration information and coarse surface configuration. Correspondingly, the labeling process is divided into 2 parts, camera calibration and surface layout annotation. For the first part, there are mainly two classes of methods which address the problem of calibrating surveillance cameras. One category is by extracting point pairs from moving objects in the scene, especially from pedestrians. Generally, these points are labeled or estimated head feet point pairs, from which camera's projection matrix can be calculated [6], or estimated through Bayesian paradigm [4]. Another category utilizes special landmarks such as lane markings on the road [5] to estimate horizon lines and vanishing points. Most of these methods try to automatically estimate camera parameters. However, when applied to

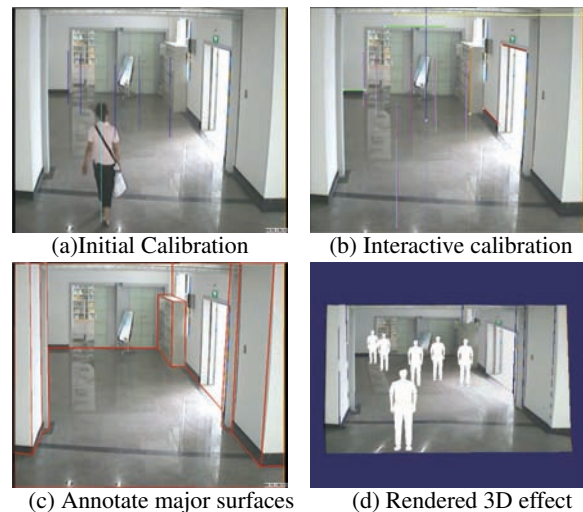


Figure 1. Camera calibration and major surface layout recovery from video.

real applications, performance of these automatic methods highly depend on other modules, such as tracking and recognition, and is difficult to improve once fail.

To better utilize above algorithms while ensures convenience and accuracy, we propose an interactive calibration method to recover camera calibration information and major surfaces layout. Through manually labeled vertical objects of approximately the same height, camera internal and external parameters are roughly estimated using algorithm in Section 2. After that, pairs of parallel lines and a vertical line are given, and the vanishing point estimation is refined by adjusting these lines. Then, the estimated horizon line and re-projected vertical poles are displayed on the image, serving as calibration accuracy indicators. These indicators further help users add and adjust other inputs, such as vertical poles and principle point. After calibration, major surfaces are labeled and reconstructed according to the algorithm in Section 2.

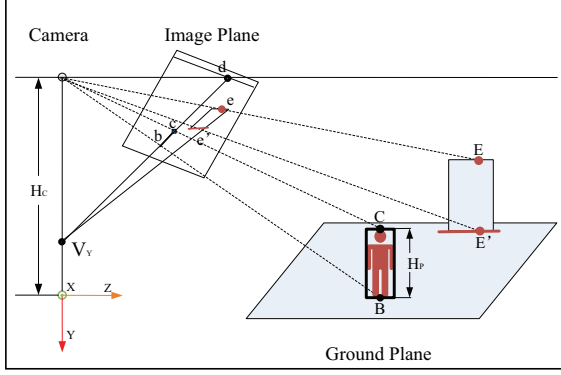


Figure 2. Feet point estimation using cross ratio theorem.

2 The Annotation Algorithm

For most of the surveillance applications, a pinhole camera model is enough to obtain acceptable accuracy. As is illustrated in Fig.2, we set the origin of WCS at CCS origin's projection point on the ground plane, Y axis points into the ground plane, and Z axis along the ground projection direction of camera's heading direction. In this way, the translation vector T of projection matrix $P \sim A \cdot [R|T]$ has only one degree of freedom, i.e. camera's height H_C and camera's pan will be zero. Without loss of generality, we also assume internal parameter skew is zero and aspect ratio equals one.

2.1 Camera Calibration Algorithm

Given 3 vanishing points V_X, V_Y and V_Z , camera internal parameters can be numerically calculated [1]. Since we also want to estimate external parameters, images of vertical poles as well as their real length should also be provided. Using calibration method in [6], camera's focal length and rotation matrix R can be derived.

As illustrated in Fig.2, the height of camera H_C is estimated by

$$\frac{H_P}{H_C} = 1 - \frac{|cd| \cdot |bV_Y|}{|bd| \cdot |cV_Y|} \quad (1)$$

where b and c are the lower and upper point of the vertical pole (e.g., image of a standing pedestrian's feet and head point), d is the intersection of line bc and horizontal line, and H_P the real height of pole's corresponding object (e.g., the height of a pedestrian). This relation is derived from cross ratio theorem.

2.2 Surface Layout Reconstruction

This subsection introduces how to reconstruct the vertices of surfaces from corresponding image points and the projection matrix P . These vertices can be classified into three cases. Without specific explanation, we use upper case character to represent 3D point, and use Fig.2 as an illustration.

Case 1. 3D point on the ground plane. This corresponds to recover point B from b . With the placement of WCS, point B will be the form $(B_X, 0, B_Z, 1)^T$. By eliminating the second column from P , homography H between image plane and ground can be derived, and then $B' = (B_X, B_Z, 1)^T \sim H^{-1} \cdot b$.

Case 2. 3D point with known Y-axis coordinate. It is the case of estimating C from c . Assume the line BC is perpendicular to ground plane, then point C 's position can be determined from B or equivalently its image b . Firstly, $|bc|$ satisfies the following equation,

$$\frac{|bc|}{|cd|} / \frac{|V_Y b|}{|V_Y d|} = \frac{|H_P|}{|H_P - H_C|} \quad (2)$$

where H_P is the known Y value C_Y and H_C the camera height. This is also derived from the cross ratio theorem. Note that point b is on line $V_Y d$. These two constraints are enough to get the position of b . After that, C_X and C_Z can be calculated as is in the previous case.

Case 3. 3D point which is on an annotated wall. This is the case of estimating E from e' and the image of wall's intersection line with ground plane. Compared with Case 2, this case's only difference is that E_Y is not known. To get E_Y , we either need to know E 's projection point E' on the ground plane or its image e' . If we further know the image of intersection line between ground plane and this wall, such as the brown line in Fig.2, then e' is simply the intersection between this line and $V_Y e$. In practice, we can let users label this intersection line first, and then annotate other lines forming that wall's boundary. After got e' , we can use Eq.2 to get the height of point E .

Determined by the application, most of the major surfaces in surveillance video are manmade, and are usually vertical or parallel to the ground plane. So, in practice, the problem of reconstructing vertices of these major surfaces can be categorized into one of the three cases above.

3 Interactive Annotation Procedure

This procedure is divided into 3 stages, which are specified as follows.

3.1 Initial Calibration Using Vertical Poles

At the initial stage, users are requested to label at least three vertical poles from a video sequence. These vertical poles can be either walking human or any other vertical objects of approximately the same height, such as cars, lampstandard, etc. As illustrated in Fig.3, vanishing point V_Y and its corresponding horizon line can be estimated by extending and joining these vertical poles, and further optimized by RANSAC [6].

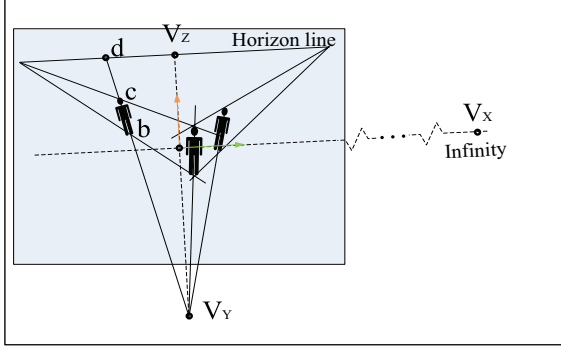


Figure 3. Estimate vanishing points from vertical poles.

At this initial stage, image of camera’s principle point can be assumed at image’s center. According to the placement of WCS, vanishing point of $(0, 0, 1, 0)^T$ must also lie on horizon line’s perpendicular line which passes through principle point, and V_X lies on the line which is parallel to the horizon line and passes through principle point. Together with the constraint that V_Z and V_X must lie on the horizon line, these two vanishing points can be easily calculated. In practice, V_Z is set on the dashed line’s infinity, which is assumed to be the intersection point of these two parallel lines. Given these three vanishing points, vertical poles and their real heights, camera parameters can be estimated through methods in Subsection 2.1.

3.2 Interactively Adding and Adjusting Constraints

Because of the assumption on principle point’s position and inaccuracy of labeled head-foot point pairs, estimated vanishing points in the last subsection are not very accurate. In this stage, we mainly focus on optimize them by interactively optimizing the estimation of horizon line and V_Y .

One convenient way of optimizing the horizon line is to let users add more parallel line pairs, such as lane marks on the road, grids formed by floor tiles, etc. Specifically, we let users click and drag two pairs of colored lines to place them on image of parallel line pairs which are also parallel to ground plane, corresponding to the red and green lines shown in Fig.1(b). To conveniently tell users the labeling accuracy, every time they adjust these lines, camera is re-calibrated and feet points are re-projected. Re-projected feet points are displayed on the image, connecting with the original head points, which are shown as the pink lines in Fig.1(b). Re-projected feet point is got by projecting the estimate point B in Fig.2, using point c , length H_P and Eq.2. Compared with the labeled vertical pole, users can immediately percept the calibra-

tion accuracy, and adjust the position of these parallel line pairs accordingly. Besides displaying the re-projection effect, a numerical re-projection error is also shown on the screen, which is calculated as $Error = \frac{1}{N} \sum_{i=1}^N \sqrt{[(u_{f_i} - u'_{f_i})^2 + (v_{f_i} - v'_{f_i})^2]}$, where N is the number of vertical poles, and $(u_{f_i}, v_{f_i}), (u'_{f_i}, v'_{f_i})$ are coordinates of i -th original and re-projected feet point. At the same time, the calculated horizon line corresponding to the yellow line in Fig.1(b) is also displayed, which expressively shows camera’s tilt, roll and camera’s height.

In the same manner, users can add another vertical line to improve the estimation of V_Y , which is the brown line in Fig.1(b). This line can be placed on a vertical pole with arbitrary length, such as image of two wall’s intersection line. Usually, this line’s end points shall be more distinctive and easy to label than that of head-foot pairs, therefore reduces the inaccurate head-foot point pairs’ effect on estimating V_Y . Besides, principle point is also shown on the center of image, serving as an adjustment option. If users feel that any labeled vertical pole is not very accurate, they can also replace it with new ones at this stage. Note that every time any input is changed, camera will be re-calibrated and indicator lines described above will be updated and displayed.

3.3 Major Surface Annotation

After camera calibration, users can begin to label major surfaces of the scene. In this paper, major surfaces refer to ground plane, walls and other surfaces constraining objects activity such as plane abstracted from stairs, etc.

The problem of recovering major surfaces’ vertices has three cases represented in Section 2.2. To find out a vertex’s category, we classify major surfaces into surfaces parallel to ground plane (called parallel planes), and vertical ones. So, vertices in parallel plane belong to case 1 of Section 2.2. For vertices on vertical surfaces, we let users first annotate two vertices on the ground plane, so the rest vertices belong to case 3 of Section 2.2. An example of surface area chart is shown in Fig.1(c). After annotation, major surfaces are re-constructed and rendered using 3D render engine, as is shown in Fig.4. Users can drag, rotate and zoom the model in arbitrary angle and scale to check the calibration accuracy and labeling correctness.

4 Experiments

This tool has already been integrated into our intelligent surveillance system, and has successfully improved other module’s performance[3]. Besides, most of our testers and clients report that this tool is easy to

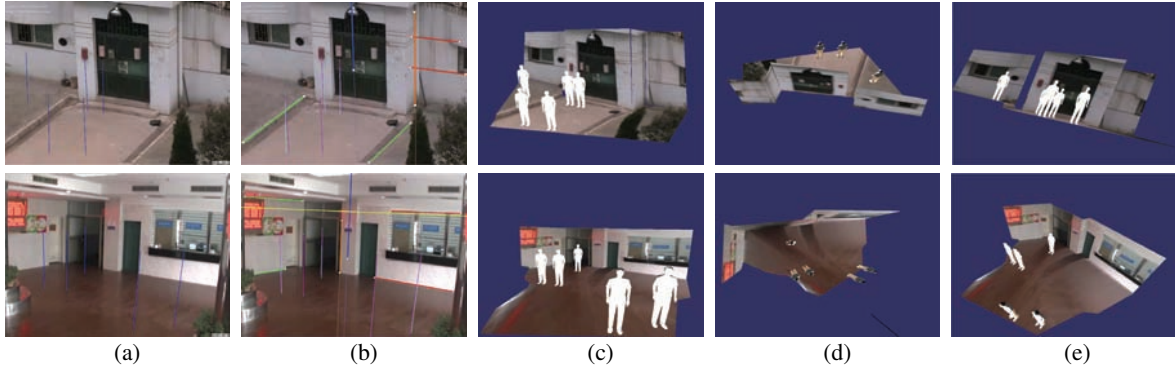


Figure 4. Experiment results of other scenes

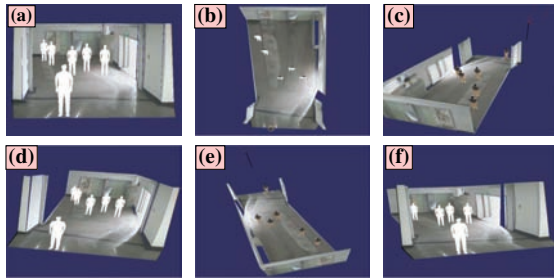


Figure 5. Different views of a reconstructed scene. (a) Original View; (b) Top View; (c)-(f) Other novel views;

use and the final rendered result is useful and impressive.

To demonstrate this tool's effectiveness, we show images of rendered major surfaces, together with 3D pedestrian models placed on vertical poles. If camera's position is in view port, it is also rendered (Red box in Fig.4(c)(e)). Through the top view (Fig.4(b)), we can see that the walls are roughly perpendicular to each other, which indirectly confirms the accuracy of the calibration method.

To show this tool's convenience, we let a user without knowledge of computer vision to annotate the scene in Fig.1(a). In the annotation process, we recorded the average re-projection errors in every 15 seconds interval during the interactive refining process, and show the result in Fig.4. Through this figure, we can see that the user finished the interactive calibration in about 80 seconds, and the re-projection error falls quickly with time, which demonstrates this tool is very easy to use.

Limited by space requirement, we only show two other reconstructed scenes in Fig.4, three views per each. From these views, we can again verify the calibration correctness.

5 Conclusion

An interactive scene annotation tool is introduced in this paper. Through this tool, camera calibration accu-

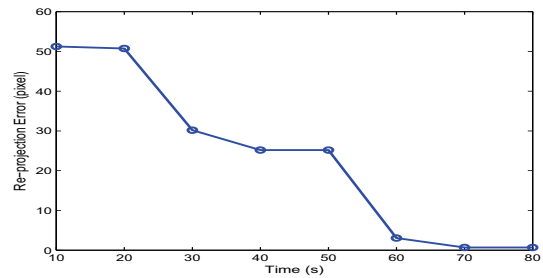


Figure 6. Convergence of re-projection error with user's adjustment

racy can be visualized to assist users adjust inputs, and major surfaces are reconstructed to help users check the labeling correctness and accuracy. Experiment results show that this tool provides users an easy to use environment for scene annotation in video surveillance application.

Acknowledgement

This work is supported by Natural Science Foundation of China (60673198), Hi-Tech Research and Development Program of China (2006AA02Z4E5), and Innovation Team Development Program of the Chinese Ministry of Education (IRT0606) .

References

- [1] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [2] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.
- [3] W. Hu, H. Gong, S.-C. Zhu, and Y. Wang. An integrated background model for video surveillance based on primal sketch and 3d scene geometry. In *CVPR*, 2008.
- [4] N. Krahnstoeber and P. Mendonca. Bayesian autocalibration for surveillance. In *ICCV*, 2005.
- [5] Y. Li, F. Zhu, Y. Ai, and F.-Y. Wang. On automatic and dynamic camera calibration based on traffic visual surveillance. *IVS*, 2007.
- [6] F. Lv, T. Zhao, and R. Nevatia. Self-calibration of a camera from video of a walking human. In *ICPR*, 2002.