

# Recovery of Intrinsic and Extrinsic Camera Parameters Using Perspective Views of Rectangles

T. N. Tan, G. D. Sullivan and K. D. Baker

Department of Computer Science

The University of Reading, Berkshire RG6 6AY, UK

Email: T.Tan@reading.ac.uk

## Abstract

This paper concerns the recovery of intrinsic and extrinsic camera parameters using perspective views of rectangles. Several algorithms are described. The intrinsic parameters (aspect ratio, effective focal length, and the principal point) are determined in closed-form using a minimum of four views. The recovery of the intrinsic parameters and the rotation angles is independent of the size of the rectangles. The absolute translation is recovered by knowing either the length of one side of the rectangle or the area of the rectangle or any other absolute information. The new method for recovering extrinsic parameters is shown to be significantly more robust than the widely used method based on vanishing points. Experiments with both synthetic and real images are presented.

## 1 Introduction

Many computer vision applications require knowledge about the optical characteristics (the intrinsic parameters) and the orientation and position (the extrinsic parameters) of the camera w.r.t. a reference frame (often referred to as the world coordinate system). The determination of such parameters (camera calibration) is a fundamental issue in computer vision, and has been studied extensively [10].

A typical camera calibration method uses a specially constructed and often complicated reference object or calibration object. The calibration object is usually marked with points (the control points) whose 3-D coordinates are known very accurately in a world coordinate system. Images of the calibration object are taken by the camera to be calibrated and processed to determine the image coordinates of the control points. The correspondences between the known 3-D world and 2-D image coordinates of the control points are then used to recover the intrinsic and extrinsic camera parameters [10-11]. While such approaches can deliver accurate camera parameters, they have serious drawbacks in practical applications. For example, in vision-based road traffic surveillance applications [19-21], it is impractical to use complicated calibration objects and alternative methods are more appropriate [12].

Recent work on camera calibration indicates that camera parameters may be determined without using any calibration objects [16, 13-15]. For example, Maybank and Faugeras [16] have shown that given point correspondences in several views of a scene taken by the same camera, the intrinsic camera parameters can be determined by solving a set of so-called Kruppa equations [16-17]. The solution of the general Kruppa equations is rather time-consuming and often leads to highly noise-sensitive results [13]. More

stable and efficient algorithms are reported in [13-15], but the algorithms require constrained (e.g., pure rotation [13]) and/or known camera motion [14-15]. Such requirements are difficult to meet in practical applications such as road traffic surveillance. Although camera self-calibration methods such as those mentioned above provide interesting and potentially very valuable alternatives to conventional methods, the state-of-the-art does not seem to allow convenient camera calibration for practical applications.

A number of camera calibration methods [2-9, 12] have recently been reported which either seek calibration cues (e.g., vanishing points [1] due to parallel lines) directly from the scene or only require very simple calibration objects (e.g., simple planar shapes, cubes, etc.). Such methods do not have the problems of current self-calibration methods, and may easily be implemented in practice. In this paper, we present a camera calibration method which uses images of rectangles. For completeness, the recovery of both intrinsic and extrinsic camera parameters is discussed. Because of the simplicity and ubiquity of rectangles, the current method is advantageous in practice. Rectangles have previously been used by Haralick [2] and Chang et. al. [5] but they assume known intrinsic parameters. For the recovery of extrinsic parameters, several techniques are described in this paper. The new techniques are shown to be much more robust than the widely used technique based on vanishing points [3-4, 6-9, 18].

## 2 Notations

The imaging geometry assumed in this paper is depicted in Fig.1. The camera is a pinhole

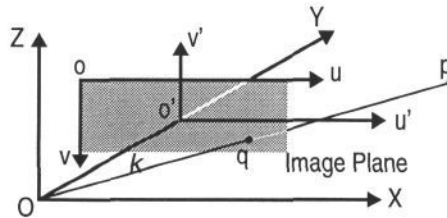


Figure 1: Illustration of imaging geometry.

camera with no lens distortion. The camera image plane  $u'-o'-v'$  is at a distance  $k$  in front of the focal point and is orthogonal to the optical axis. The camera image plane is spatially sampled along the horizontal and vertical directions to store a camera image frame as a raster image frame or simply an (image) frame in a computer. The raster image plane is shown in Fig.1 as  $u-o-v$ , with the abscissa axis pointing rightwards and the ordinate axis downwards. The origin of the raster image plane is at the top-left corner of the image frame. The camera image coordinates  $(u', v')$  and the observable raster image coordinates  $(u, v)$  are related to each other by the following equations:

$$u' = \alpha_u (u - u_0); \quad v' = \alpha_v (v_0 - v) \quad (1)$$

where  $\alpha_u$  and  $\alpha_v$  are the horizontal and vertical scaling factors, and  $(u_0, v_0)$  the raster image coordinates of the origin of the camera image plane. These four variables only depend on the characteristics of the imaging system.

The camera coordinate system (CCS) is initially aligned with a world coordinate

system (WCS). To allow the camera to view the 3-D world from an arbitrary viewpoint, the CCS is first rotated around its X-axis by an angle  $\phi$  (the tilt angle), then around its Y-axis by an angle  $\psi$  (the roll angle), and finally around its Z-axis by an angle  $\theta$  (the pan angle). The rotation is followed by a translation  $T_x$  along the X-axis,  $T_y$  along the Y-axis, and  $T_z$  along the Z-axis. The six variables  $\phi$ ,  $\psi$ ,  $\theta$ ,  $T_x$ ,  $T_y$  and  $T_z$  are the extrinsic camera parameters.

From now on, lowercase bold letters are used to denote (row) vectors. For example, we use  $\mathbf{p}_A$  to represent the camera coordinates of a world point  $A$ , and  $\mathbf{q}_a$  to represent the camera coordinates of the image  $a$  of point  $A$ . If  $a$  is located at  $(u_a, v_a)$  on the raster image plane,  $\mathbf{q}_a$  is given by

$$\mathbf{q}_a = (\alpha_u (u_a - u_0) \quad k \quad \alpha_v (v_0 - v_a)) \quad (2)$$

By definition, the focal point  $O$ , the image point  $a$ , and the world point  $A$  are collinear on the line of sight. Given the image position of  $a$ , the direction of the line of sight is uniquely determined and is given by the following unit direction vector

$$\hat{\mathbf{q}}_a = \mathbf{q}_a / |\mathbf{q}_a| \quad (3)$$

Let the distance from the focal point to point  $A$  be  $\lambda_A$ . Then  $\mathbf{p}_A$  is given by

$$\mathbf{p}_A = \lambda_A \hat{\mathbf{q}}_a \quad (4)$$

where  $\lambda_A$  is often called the range of point  $A$ .

For a given rectangle, we name its four corners in a clockwise manner as  $A$ ,  $B$ ,  $C$ , and  $D$ , and its centre as  $H$ . The image of the rectangle is similarly labelled (see Fig.2).

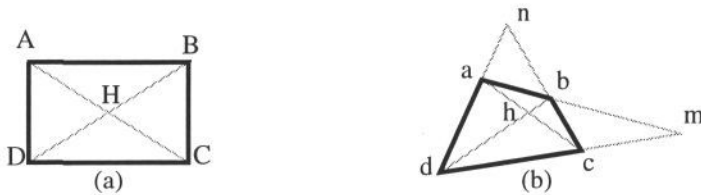


Figure 2: Labelling of a rectangle (a) and its image (b).

In the following two sections, we discuss the recovery of the intrinsic and the extrinsic camera parameters using perspective views of rectangles.

### 3 Recovery of intrinsic parameters

Under perspective projection, the parallel sides of the rectangle  $AB$  and  $CD$ , and  $AD$  and  $BC$  intersect at vanishing points  $m$  and  $n$  respectively [1] (see Fig.2(b)). According to the properties of vanishing points, the lines connecting  $O$  (the focal point) and  $m$ , and  $O$  and  $n$  are parallel to  $AB$  ( $CD$ ) and  $AD$  ( $BC$ ) respectively. The unit direction vectors along  $Om$  and  $On$  are simply  $\hat{\mathbf{q}}_m$  and  $\hat{\mathbf{q}}_n$ . Since  $AB \perp AD$ , we then have

$$\hat{\mathbf{q}}_m \bullet \hat{\mathbf{q}}_n = 0 \quad (5)$$

From (2), (3) and (5), we obtain

$$\alpha_u^2 (u_m - u_0) (u_n - u_0) + k^2 + \alpha_v^2 (v_0 - v_m) (v_0 - v_n) = 0 \quad (6)$$

By letting  $\alpha = \frac{\alpha_v}{\alpha_u}$  and  $f = \frac{k}{\alpha_u}$ , we rewrite (6) as

$$\frac{1}{\alpha^2} (u_m - u_0) (u_n - u_0) + f^2 + (v_0 - v_m) (v_0 - v_n) = 0 \quad (7)$$

where  $\alpha$  (called the aspect ratio),  $f$  (called the effective focal length), and  $(u_0, v_0)$  (the principal point) are the four intrinsic camera parameters that are to be determined in this paper.

### 3.1 The algorithm

Equation (7) states that from a perspective view of a single rectangle, one can derive a constraint on the four intrinsic camera parameters. Hence for a total of  $N$  perspective views of a single rectangle taken by the same camera with the same parameters, there are  $N$  such constraints:

$$\frac{1}{\alpha^2} (u_{m,i} - u_0) (u_{n,i} - u_0) + f^2 + (v_0 - v_{m,i}) (v_0 - v_{n,i}) = 0; \quad i = 1, 2, \dots, N \quad (8)$$

Clearly,  $N$  must be no less than 4 to ensure that the intrinsic parameters can be determined from the constraints. The minimal number of views can be reduced from 4 to 3, 2 or 1 by placing multiple rectangles in the scene. For example, when the calibration object is a cube [8, 18], one only needs two views of the cube (since a single general view of a cube is equivalent to three views of a rectangle). The multiple rectangles do not have to be of the same size since the constraints do not depend on the size of the rectangles.

The constraint equations (8) are nonlinear in the four unknowns. However, a simple closed-form solution is possible. By subtracting the first equation (or any other equation) of (8) from the other  $N - 1$  remaining equations, we obtain

$$A_i u_0 + B_i (\alpha^2 v_0) + C_i \alpha^2 = D_i; \quad i = 2, 3, \dots, N \quad (9)$$

where

$$\begin{cases} A_i = (u_{m,i} - u_{m,1}) + (u_{n,i} - u_{n,1}) \\ B_i = (v_{m,i} - v_{m,1}) + (v_{n,i} - v_{n,1}) \\ C_i = v_{m,1} v_{n,1} - v_{m,i} v_{n,i} \\ D_i = u_{m,i} u_{n,i} - u_{m,1} u_{n,1} \end{cases} \quad (10)$$

The  $N - 1$  equations in (9) can easily be solved by using the standard linear least squares technique to get  $u_0$ ,  $\alpha^2 v_0$  and  $\alpha^2$ , hence three of the four unknowns. The remaining unknown, the effective focal length  $f$ , can then be determined by substituting  $u_0$ ,  $v_0$  and  $\alpha$  into (8).

### 3.2 Experimental results

The algorithm was applied to recover intrinsic camera parameters from real images. Seven views of a rectangle on a cardboard were taken. The images were of size 768x576

pixels and are shown in Fig.3. The Plessey corner detector [22] was applied to each image

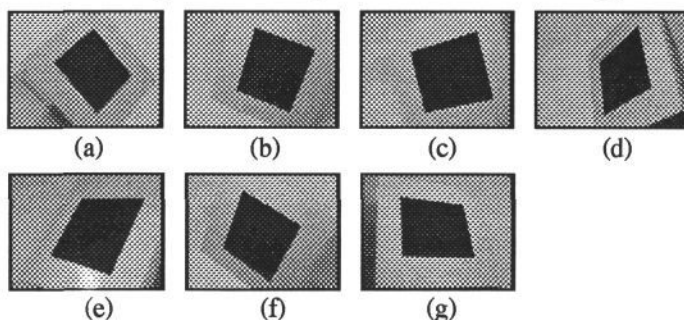


Figure 3: Seven perspective views of a rectangle. Images are of size 768x576 pixels.

to locate the four corners of the rectangle. The output of the Plessey corner detector was then used by the algorithm to determine the four intrinsic camera parameters. The recovered values are as follows

$$u_0 = 380.16; \quad v_0 = 290.98; \quad \alpha = 0.98; \quad f = 1093.40 \quad (11)$$

Since the ground-truth was unknown, we could not assess the accuracy of the recovered parameters quantitatively. Intuitively, however, the determined values appear reasonable. For example, the recovered principal point is close to the centre of the image, and the aspect ratio close to 1. The intrinsic parameters obtained by the algorithm have also been used successfully in structure, pose and motion recovery (see next section and also [23]).

## 4 Recovery of extrinsic parameters

Once the intrinsic parameters are determined, the six extrinsic parameters may be recovered from a single perspective view of a rectangle in a number of ways. Two methods are described in the following. The first method is based on vanishing points (the VP method) and the second on angle constraints (the AC method). The WCS is defined on the rectangle as illustrated in Fig.4. The transformation from the WCS to the CCS

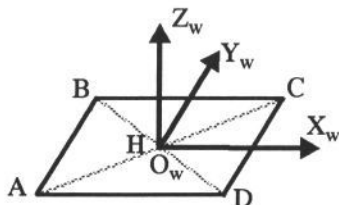


Figure 4: The world coordinate system defined on a rectangle.

comprises a rotation  $R_{wc}$  followed by a translation  $t_{wc}$ . The rotation may also be represented as the pan, tilt and roll angles.

### 4.1 The VP method: recovery using vanishing points

For calibration objects having parallel lines, a widely used method is based on vanishing points. Examples of methods of this kind may be found in [3-9, 12, 18]. For the sake of completeness, a similar method is described here.

Given the vanishing point  $m$  of AB and CD, and  $n$  of AD and BC, we can easily determine the unit direction vectors  $\hat{r}_y$  and  $\hat{r}_x$  (expressed in the camera coordinate system) along the  $Y_w$ - and the  $X_w$ -axis. They are simply  $\hat{r}_x = \hat{q}_n$  and  $\hat{r}_y = \hat{q}_m$ . The unit direction vector along the  $Z_w$ -axis is determined by the cross-product of  $\hat{r}_x$  and  $\hat{r}_y$ , i.e.,  $\hat{r}_z = \hat{r}_x \times \hat{r}_y$ . Hence the rotation matrix  $R_{wc}$  is given by

$$R_{wc} = \begin{bmatrix} \hat{r}_x^T & \hat{r}_y^T & \hat{r}_z^T \end{bmatrix} = \begin{bmatrix} \hat{q}_n^T & \hat{q}_m^T & (\hat{q}_n \times \hat{q}_m)^T \end{bmatrix} \quad (12)$$

From the known raster image coordinates of the four corners, we can easily compute those  $(u_h, v_h)$  of the centre H of the rectangle. These allow us to determine the unit direction vector  $\hat{q}_h$  along the line of sight of H. The translation vector  $t_{wc}$  is simply the camera coordinates of H and is therefore given by

$$t_{wc} = \lambda_H \hat{q}_h \quad (13)$$

where the range parameter  $\lambda_H$  is a unknown scaling factor.

Once the orientation and position of the rectangle are known, the camera coordinates of the four corners can easily be determined. The equation (expressed in the camera coordinate system) of the plane supporting the rectangle is defined by

$$(\hat{q}_n \times \hat{q}_m) \cdot (p - t_{wc}) = 0 \quad (14)$$

where  $p$  is the vector representing the camera coordinates of a point on the supporting plane. Then the camera coordinates of a corner point are simply the intersection of the supporting plane and the line of sight of that point. For example, the coordinates of corner A are given by the following

$$p_A = \frac{\hat{q}_n \times \hat{q}_m \cdot t_{wc}}{\hat{q}_n \times \hat{q}_m \cdot \hat{q}_a} \hat{q}_a \quad (15)$$

and those of B, C and D are computed similarly.

The scaling factor  $\lambda_H$  in (13) and (15) may be determined by knowing one of the following: a) the length of one side of the rectangle; b) the area of the rectangle; or c) the absolute range of one of the corners.

## 4.2 The AC method: recovery using angle constraints

The VP method often depends critically on the accuracy of the vanishing points [7]. Unfortunately, the accuracy of the vanishing points cannot always be guaranteed, particularly when the perspective distortion is not severe. For example, when the depth change across an object is small compared with the nominal depth of the object, perspective projection is very close to scaled orthographic projection. In this case, parallel lines in 3D project to almost parallel lines on the image plane, and the detection of vanishing points is highly unstable. Therefore, from the practical point of view, a method not involving the use of vanishing points is perhaps preferable. Such a method is presented in the following.

Let the position vectors (i.e., the camera coordinates) of the four corners be  $p_A$ ,  $p_B$ ,  $p_C$  and  $p_D$ . From (4), they are given by

$$p_A = \lambda_A \hat{q}_a; \quad p_B = \lambda_B \hat{q}_b; \quad p_C = \lambda_C \hat{q}_c; \quad p_D = \lambda_D \hat{q}_d \quad (16)$$

where  $\lambda_A$ ,  $\lambda_B$ ,  $\lambda_C$  and  $\lambda_D$  are the range parameters that are to be determined in the following.

From the simple fact that the four angles of a rectangle are all right angles, we have

$$\begin{aligned} (p_A - p_B) \cdot (p_A - p_D) &= 0; & (p_B - p_A) \cdot (p_B - p_C) &= 0 \\ (p_C - p_B) \cdot (p_C - p_D) &= 0; & (p_D - p_A) \cdot (p_D - p_C) &= 0 \end{aligned} \quad (17)$$

By substituting (16) into (17), we obtain

$$\lambda_A^2 - Q_{ab}\lambda_A\lambda_B - Q_{ad}\lambda_A\lambda_D + Q_{bd}\lambda_B\lambda_D = 0 \quad (18)$$

$$\lambda_B^2 - Q_{ab}\lambda_A\lambda_B - Q_{bc}\lambda_B\lambda_C + Q_{ac}\lambda_A\lambda_C = 0 \quad (19)$$

$$\lambda_C^2 - Q_{bc}\lambda_B\lambda_C - Q_{cd}\lambda_C\lambda_D + Q_{bd}\lambda_B\lambda_D = 0 \quad (20)$$

$$\lambda_D^2 - Q_{ad}\lambda_A\lambda_D - Q_{cd}\lambda_C\lambda_D + Q_{ac}\lambda_A\lambda_C = 0 \quad (21)$$

where  $Q_{ij} = \hat{q}_i \cdot \hat{q}_j$ . The absolute values of the four unknown range parameters cannot be determined from the above four equations since the equations are homogeneous in the unknowns. We then arbitrarily set  $\lambda_A = 1$  to determine the relative values of the range parameters. With  $\lambda_A = 1$ , Equations (18)-(21) become

$$1 - Q_{ab}\lambda_B - Q_{ad}\lambda_D + Q_{bd}\lambda_B\lambda_D = 0 \quad (22)$$

$$\lambda_B^2 - Q_{ab}\lambda_B - Q_{bc}\lambda_B\lambda_C + Q_{ac}\lambda_C = 0 \quad (23)$$

$$\lambda_C^2 - Q_{bc}\lambda_B\lambda_C - Q_{cd}\lambda_C\lambda_D + Q_{bd}\lambda_B\lambda_D = 0 \quad (24)$$

$$\lambda_D^2 - Q_{ad}\lambda_D - Q_{cd}\lambda_C\lambda_D + Q_{ac}\lambda_C = 0 \quad (25)$$

From (22) and (25), we get  $\lambda_B$  and  $\lambda_C$  expressed in terms of  $\lambda_D$ :

$$\lambda_B = \frac{Q_{ad}\lambda_D - 1}{Q_{bd}\lambda_B\lambda_D - Q_{ab}}; \quad \lambda_C = \frac{Q_{ad}\lambda_D - \lambda_D^2}{Q_{ac} - Q_{cd}\lambda_C\lambda_D} \quad (26)$$

By substituting (26) into (23) and (24), we get respectively

$$C_4\lambda_D^4 + C_3\lambda_D^3 + C_2\lambda_D^2 + C_1\lambda_D + C_0 = 0 \quad (27)$$

and

$$B_4\lambda_D^4 + B_3\lambda_D^3 + B_2\lambda_D^2 + B_1\lambda_D + B_0 = 0 \quad (28)$$

where the coefficients of the polynomial equations are computable from known  $Q_{ij}$ . From (27) and (28), we can get a third-order polynomial equation on  $\lambda_D$ :

$$D_3\lambda_D^3 + D_2\lambda_D^2 + D_1\lambda_D + D_0 = 0 \quad (29)$$

where  $D_i$  are known coefficients. The third-order polynomial equation can easily be solved to obtain three possible solutions for  $\lambda_D$ . However, the valid solution should satisfy (27) or (28) and should be positive. Such constraints in general eliminate two solutions, leaving only one valid solution for  $\lambda_D$ . Once  $\lambda_D$  is known,  $\lambda_B$  and  $\lambda_C$  are computed from (26). The unknown global scale in the range parameters may be fixed by knowing the length of one side of the rectangle or the area of the rectangle. The known range parameters can be substituted into (16) to obtain the camera coordinates of the four corners. The camera coordinates are then used to determine the rotation matrix  $\mathbf{R}_{wc}$  and the translation vector  $\mathbf{t}_{wc}$ :

$$\begin{cases} \hat{\mathbf{r}}_x = \frac{1}{2} \left( \frac{\mathbf{p}_D - \mathbf{p}_A}{|\mathbf{p}_D - \mathbf{p}_A|} + \frac{\mathbf{p}_C - \mathbf{p}_B}{|\mathbf{p}_C - \mathbf{p}_B|} \right) \\ \hat{\mathbf{r}}_y = \frac{1}{2} \left( \frac{\mathbf{p}_B - \mathbf{p}_A}{|\mathbf{p}_B - \mathbf{p}_A|} + \frac{\mathbf{p}_C - \mathbf{p}_D}{|\mathbf{p}_C - \mathbf{p}_D|} \right) \\ \hat{\mathbf{r}}_z = \hat{\mathbf{r}}_x \times \hat{\mathbf{r}}_y \\ \mathbf{R}_{wc} = \begin{bmatrix} \hat{\mathbf{r}}_x^T & \hat{\mathbf{r}}_y^T & \hat{\mathbf{r}}_z^T \end{bmatrix} \\ \mathbf{t}_{wc} = \frac{1}{4} (\mathbf{p}_A + \mathbf{p}_B + \mathbf{p}_C + \mathbf{p}_D) \end{cases} \quad (30)$$

It can easily be shown that  $\mathbf{R}_{wc}$  defined above is independent of the size of the rectangle. This is consistent with previous results [2].

### 4.3 Experimental results

Experiments were carried out to compare the performance of the VP and the AC method using known parameters of a calibrated camera in an outdoor traffic scene [20-21]. The projection of a rectangle was computed and the ideal image coordinates of the four corners were then perturbed by values randomly chosen from a given interval  $[-\varepsilon, +\varepsilon]$ . The relative errors of the extrinsic parameters recovered by the two methods as functions of  $\varepsilon$  are shown in Fig.5. It can be seen that the AC method performs significantly better than the VP method, particularly for the pan angle and the translation.

The performance of the AC method was further assessed by applying it to the seven images shown in Fig.3. Since only the size (length and width) of the rectangle was known, the AC method was used to recover the dimension of the rectangle and the output was compared with the ground-truth. The results are summarised in Table 1. The intrinsic

Image	Fig.3(a)	Fig.3(b)	Fig.3(c)	Fig.3(d)	Fig.3(e)	Fig.3(f)	Fig.3(g)
Length	140.08	140.44	140.47	140.92	139.64	141.22	140.36
Width	123.93	123.61	123.58	123.19	124.32	122.92	123.68

Table 1 : Recovered dimension of the rectangle shown in Fig.3. The ground-truth is Length=140mm and Width=124mm.



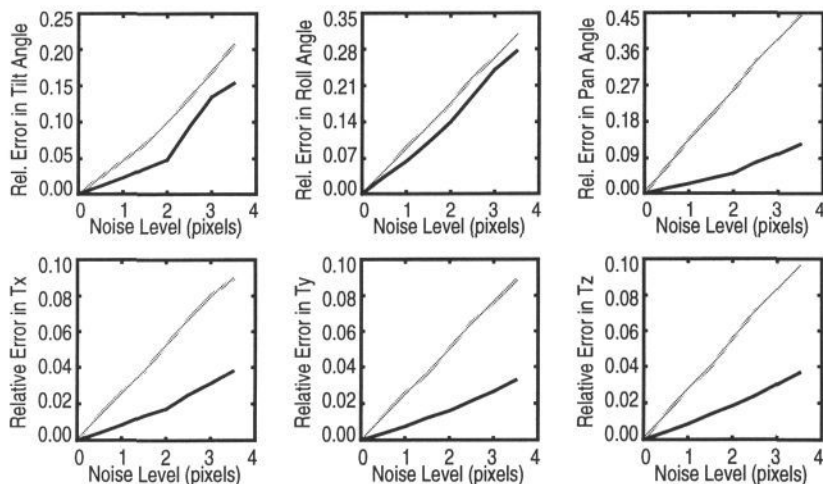


Figure 5: Relative errors of the six extrinsic camera parameters recovered by the VP algorithm (grey) and the AC method (dark).

camera parameters used in the recovery were those obtained by the algorithm described in Section 3.1. It can be seen that the dimension recovered from all images is very close to the ground-truth, indicating good performance of the AC method and the algorithm of Section 3.1.

## 5 Conclusions

We have discussed the recovery of intrinsic camera parameters based on 4 views of any rectangles and that of extrinsic camera parameters using a single perspective view of a rectangle. Several algorithms have been described. The simplicity and ubiquity of rectangles greatly facilitate efficient and convenient implementation of the algorithms in practical applications. It has been shown that the intrinsic parameters can be determined in closed-form using a minimum of four rectangle views. The recovery of the intrinsic parameters and the rotation angles is independent of the size of the rectangles. The new method for recovering the extrinsic parameters has been found much more robust than the conventional method based on vanishing points. Experiments with both synthetic and real images have been presented to demonstrate the performance of the algorithms.

## References

- [1] R. M. Haralick, Using Perspective Transformations in Scene Analysis, *CGIP*, vol.13, 1980, pp.191-220.
- [2] R. M. Haralick, Determining Camera Parameters from The Perspective Projection of A Rectangle, *Pattern Recognition*, vol.22, 1989, pp.225-230.
- [3] W. Chen and B. C. Jiang, 3-D Camera Calibration Using Vanishing Point Concept, *Pattern Recognition*, vol.24, 1991, vol.57-67.
- [4] B. Caprile and V. Torre, Using Vanishing Points for Camera Calibration, *Int. J. Comput. Vision*, vol.4, 1990, pp.127-140.

- [5] H. D. Chang et. al., A Closed Form 3-D Self-Localisation for A Mobile Robot Using A Single Camera and A Rectangular Guide-Mark, *Proc. of ICARCV92*, Singapore, 1992, pp.CV-12.3.1-5.
- [6] L. L. Wang and W. H. Tsai, Camera Calibration by Vanishing Lines for 3-D Computer Vision, *IEEE Trans. PAMI*, vol.13, 1991, pp.370-376.
- [7] K. Kanatani, Statistical Analysis of Focal-Length Calibration Using Vanishing Points, *IEEE Trans. RA*, vol.8, 1992, pp.767-775.
- [8] L. L. Wang and W. H. Tsai, Computing Camera Parameters Using Vanishing Line Information from A Rectangular Parallelepiped, *Mach. Vision and Applications*, vol.3, 1990, pp.129-141.
- [9] B. Odone et. al., On the Understanding of Motion of Rigid Objects, *Proc. of SPIE*, vol.726, 1987, pp.197-205.
- [10] R. Y. Tsai, Synopsis of Recent Progress on Camera Calibration for 3D Machine Vision, in *Robotics Review* (O. Khatib et. al., Eds.), MIT Press, 1989.
- [11] T. N. Tan, G. D. Sullivan and K. D. Baker, On Computing the Perspective Transformation Matrix and Camera Parameters, *Proc. of BMVC93*, pp.125-134.
- [12] A. D. Worrall, G. D. Sullivan and K. D. Baker, A Simple Intuitive Camera Calibration Tool for Natural Images, *Proc. of BMVC94*, pp.781-790.
- [13] R. I. Hartley, Self-Calibration from Multiple Views with A Rotating Camera, *Proc. of ECCV94*, vol.1, 1994, pp.471-478.
- [14] L. Dron, Dynamic Camera Self-Calibration from Controlled Motion Sequences, *Proc. of CVPR93*, 1993, pp.501-506.
- [15] F. Du and M. Brady, Self-Calibration of the Intrinsic Parameters of Cameras for Active Vision Systems, *Proc. of CVPR93*, 1993, 477-482.
- [16] S. J. Maybank and O. D. Faugeras, A Theory of Self-Calibration of A Moving Camera, *Int. J. Comput. Vision*, vol.8, 1992, pp.123-151.
- [17] S. D. Hppisley-Cox and J. Porrill, Auto-Calibration - Kruppa's equations and the intrinsic parameters of a camera, *Proc. of BMVC94*, pp.771-779.
- [18] G. Q. Wei, Z. Y. He and S. D. Ma, Camera Calibration by Vanishing Point and Cross Ratio, *Proc. of ICASSP89*, vol.3, 1989, pp.1630-1633.
- [19] G. D. Sullivan, Visual Interpretation of Known Objects in Constrained Scenes, *Phil. Trans. Royal Soc. London, Series B: Biol. Sci.*, vol.337, 1992, pp.361-370.
- [20] T. N. Tan, G. D. Sullivan and K. D. Baker, Recognising Objects on the Ground Plane, *Image and Vision Computing*, vol.12, 1994, pp.164-172.
- [21] T. N. Tan, G. D. Sullivan and K. D. Baker, Fast Vehicle Localisation and Recognition Without Line Extraction and Matching, *Proc. of BMVC94*, 1994, pp.85-94.
- [22] J. A. Noble, Finding Corners, *Proc. of 3rd Alvey Vision Conf.*, Cambridge, England, September 1987, pp.267-274.
- [23] T. N. Tan, Structure, Pose and Motion of Bilateral Symmetric Objects, *Proc. of BMVC95*.