Research Article

Vision Measurement Scheme Using Single Camera Rotation

Shidu Dong

College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400050, China

Correspondence should be addressed to Shidu Dong; shidu.dong@yahoo.com

Received 10 April 2013; Accepted 8 May 2013

Academic Editors: A. Nikolaidis and J. P. Siebert

Copyright © 2013 Shidu Dong. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose vision measurement scheme for estimating the distance or size of the object in static scene, which requires single camera with 3-axis accelerometer sensor rotating around a fixed axis. First, we formulate the rotation matrix and translation vector from one coordinate system of the camera to another in terms of the rotation angle, which can be figured out from the readouts of the sensor. Second, with the camera calibration data and through coordinate system transformation, we propose a method for calculating the orientation and position of the rotation axis relative to camera coordinate system. Finally, given the rotation angle and the images of the object in static scene at two different positions, one before and the other after camera rotation, the 3D coordinate of the point on the object can be determined. Experimental results show the validity of our method.

1. Introduction

Nowadays, digital camera or mobile phone with camera is very popular. It is appealing and convenient, if they are utilized to estimate the distance or size of an object. For this purpose, stereo images with disparity should be taken [1, 2]. One obvious method for the stereo image acquisition is using two cameras with different view angles. With two images of an object from two cameras and the relative orientation and position of the two different viewpoints, through the correspondence between image points in the two views, the 3D coordinates of the points on the object can be determined [3, 4]. But, in general, mobile phone or professional camera has only one camera and cannot acquire two images from different views simultaneously. Fortunately, there have been many methods for stereo vision system with single camera. The methods may be broadly divided into three categories. First, to obtain virtual images from different viewpoints, additional optical devices are introduced, such as two planar mirrors [5], a biprism [6], convex mirrors [1, 7], or the double lobed hyperbolic mirrors [8]. But these optical devices are expensive and space consuming. Second, 3D information of an object is inferred directly from a still image under the knowledge of some geometrical scene constraints such as planarity of points and parallelism of lines and planes [9–11] or prior knowledge about the scene obtained from

the supervised learning [12]. Nevertheless, these methods require constrained scenes or extra computation for training the depth models. Third, 3D information is extracted from sequential images with respect to camera movement, which is often adopted in robot area. Due to the uncertainties in the sequential camera position, however, it is difficult to get the accurate 3D information in that method [1].

In this paper, we propose a novel vision measurement method for estimating the distance or size of the object in static scene, which requires single camera with 3-axis accelerometer rotating around a fixed axis. Through the 3-axis accelerometer sensor, the slope angle of the camera relative to gravity direction can be obtained [13]. The angle can uniquely determine the position of the camera if the camera is rotated around a fixed axis which is not parallel to gravity. Moreover, the relative position and orientation of the camera between two positions, one before and the other after camera rotation, can be determined by the rotation angle. Therefore, at the given two positions, if the camera is calibrated, the classical binocular view methods [4] can be adopted to extract the 3D coordinates of the points on objects. Unfortunately, it is very difficult for the user to rotate the camera to the same slope angle as the calibrated one.

To deal with this problem, we firstly formulate the rotation matrix and translation vector from one coordinate system of the camera to another in terms of the rotation

FIGURE 1: Position and rotation angle of the camera. G' denotes gravity division on the x - y plane of the sensor coordinate system. G_x and G_y denote the divisions of gravity on the x and y axes of the sensor coordinate system, respectively, and θ_1 and θ_2 denote the camera positions. β_{21} indicates the rotation angle.

angle. Then, with camera calibration data at various positions, we calculate the orientation and position of the rotation axis relative to the camera coordinate system. Accordingly, at given two positions, the rotation matrix and translation vector from one coordinate system of the camera to another can be calculated by rotation angle, and with the collected two images at the two positions, the 3D coordinate of the points on the object can be determined.

The paper is organized as follows: Section 2 provides a stereo vision system from single camera through rotation, and Section 3 indicates calculation method of the rotation matrix and translation vector from one coordinate system of the camera to another by the rotation angle. The calculation method for the position and orientation of rotation axis relative to the camera coordination system is proposed in Section 4, and the complete calibration and the 3D coordinate calculation of the point on an object are presented in Section 5. Experimental results are given in Section 6, and some conclusions and discussions are drawn in Section 7.

2. Stereo Vision through Camera Rotation

In what follows, we adopt the following hypotheses.

(H1) The camera is provided with a 3-axis accelerometer sensor, whose readouts, G_x , G_y , and G_z , are the divisions of gravity on the x , y , and z axes of the sensor coordinate system, respectively.

(H2) In calibration and measurement processes, the camera rotates around the fixed rotation axis which is parallel to the z axis of the sensor coordinate system and is not parallel to the direction of gravity.

Thus, in the course of the rotation, the readout G_z keeps steady. As a result, the division of gravity on the plane x - y of the sensor coordinate system G' also keeps steady

(see Figure 1). Therefore, the slope angle with respect to gravity division G' (position, for short) can be determined by

$$\theta = \arctan\left(\frac{G_x}{G_y}\right). \quad (1)$$

From position θ_1 to θ_2 , the rotation angle of device β_{21} is governed by

$$\beta_{21} = \theta_2 - \theta_1. \quad (2)$$

Two images, I_1 and I_2 , of an object are collected at the positions θ_1 and θ_2 , respectively. O_1 - xyz and O_2 - xyz denote the camera coordinate systems at θ_1 and θ_2 , respectively. (x_1, y_1, z_1) and (x_2, y_2, z_2) denote the coordinates of a point on the object relative to O_1 - xyz and O_2 - xyz , respectively, and (X_1, Y_1) and (X_2, Y_2) denote image coordinates of the point of the object on I_1 and I_2 (image coordinate for short), respectively.

The projection of object coordinates relative to O_1 - xyz and O_2 - xyz into image coordinates is summarized by the following forms:

$$\begin{aligned} s_1 \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix} &= \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}, \\ s_2 \begin{bmatrix} X_2 \\ Y_2 \\ 1 \end{bmatrix} &= \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}. \end{aligned} \quad (3)$$

Let

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = [R_{21} \quad T_{21}] \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}, \quad (4)$$

where R_{21} and T_{21} denote rotation matrix and translation vector between O_2 - xyz and O_1 - xyz .

Substituting (3) into (4), we get

$$\begin{aligned} \rho_2 \begin{bmatrix} X_2 \\ Y_2 \\ 1 \end{bmatrix} &= \begin{bmatrix} f_x r_1 + c_x r_7 & f_x r_2 + c_x r_8 & f_x r_3 + c_x r_9 & f_x t_x + c_x t_z \\ f_y r_4 + c_y r_7 & f_y r_5 + c_y r_8 & f_y r_6 + c_y r_9 & f_y t_y + c_y t_z \\ r_7 & r_8 & r_9 & t_z \end{bmatrix} \\ &\times \begin{bmatrix} (X_1 - c_x) \frac{z}{f_x} \\ (Y_1 - c_y) \frac{z}{f_y} \\ z_1 \\ 1 \end{bmatrix}. \end{aligned} \quad (5)$$

Thus, the 3D coordinate of the point on the object relative to O_1 -xyz (object coordinate) can be determined by

$$\begin{aligned} x_1 &= (X_1 - c_x) \frac{z_1}{f_x}, \\ y_1 &= (Y_1 - c_y) \frac{z_1}{f_y}, \\ z_1 &= \frac{f_x f_y (f_x t_x + c_x t_z - X_2 t_z)}{B}, \end{aligned} \quad (6)$$

where

$$\begin{aligned} B &= (X_2 - c_x) [r_7 f_y (X_1 - c_x) + r_8 f_x (Y_1 - c_y) + r_9 f_y f_x] \\ &\quad - f_x [f_y r_1 (X_1 - c_x) + f_x r_2 (Y_1 - c_y) + f_x r_3 f_y]. \end{aligned} \quad (7)$$

From (6), we can see that the object coordinate can be found provided that the intrinsic parameters f_x , f_y , c_x , and c_y , the rotation matrix R_{21} , and translation vector T_{21} are available.

With the camera calibration method proposed by Zhang [14], the camera intrinsic parameters and extrinsic parameters describing the camera motion around a static scene can be figured out. Let R_1 and T_1 denote the extrinsic parameters at θ_1 and R_2 and T_2 the extrinsic parameters at θ_2 . For simplicity, $P_w = (x, y, z)$, $P_1 = (x_1, y_1, z_1)$, and $P_2 = (x_2, y_2, z_2)$ stand for the coordinates of the point on the object relative to the world coordinate system, O_1 -xyz and O_2 -xyz, respectively. Thus,

$$P_1 = R_1 P_w + T_1, \quad P_2 = R_2 P_w + T_2. \quad (8)$$

From (8), we get

$$P_w = R_1^{-1} P_1 - R_1^{-1} T_1. \quad (9)$$

Substituting (9) into (8), we get

$$P_2 = R_2 R_1^{-1} P_1 - R_2 R_1^{-1} T_1 + T_2. \quad (10)$$

Equation (4) can be rewritten as

$$P_2 = R_{21} P_1 + T_{21}. \quad (11)$$

Thus,

$$R_{21} = R_2 R_1^{-1}, \quad T_{21} = T_2 - R_2 R_1^{-1} T_1. \quad (12)$$

Based on the previous discussions, the object coordinate can be determined by the following process.

- (1) The camera is calibrated, and the intrinsic parameters, f_x , f_y , c_x , and c_y , are obtained.
- (2) At the positions θ_1 and θ_2 , the camera is calibrated, and the corresponding extrinsic parameters, R_1 , T_1 , R_2 , and T_2 , are obtained. Then, R_{21} and T_{21} can be acquired from (12).

- (3) In the course of measurement, the camera is rotated carefully to the positions θ_1 and θ_2 . At these two positions, the two images of the object, I_1 and I_2 , are collected. Once the image coordinates of the object, (X_1, Y_1) and (X_2, Y_2) , are known, the object coordinate can be figured out by (6).

It should be noticed that it is rather difficult for the user to rotate the camera accurately to the positions θ_1 and θ_2 , where camera is calibrated.

3. Calculation Method of Rotation Matrix and Translation Vector

Intuitively, the orientation and position of O_2 -xyz relative to O_1 -xyz, R_{21} and T_{21} , depend on the camera rotation angle, β_{21} . Thus, R_{21} and T_{21} may be figured out by β_{21} .

Let O_R -xyz be the coordinate system associated with the rotation axis, which is unit vector on the z axis of the coordinate system. Let R_x and T_x denote the orientation and position of O_R -xyz relative to O_1 -xyz (orientation and position of the rotation axis). Its homogeneous matrix can be written as

$$M_X = \begin{bmatrix} R_X & T_X \\ 0 & 1 \end{bmatrix}. \quad (13)$$

The rotation matrix for the camera rotation around the z axis of O_R -xyz from θ_1 to θ_2 (device rotation matrix) can be modeled as

$$R_{\beta_{21}} = \begin{bmatrix} \cos \beta_{21} & -\sin \beta_{21} & 0 \\ \sin \beta_{21} & \cos \beta_{21} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (14)$$

Its homogeneous matrix can be written as

$$\text{Rot}(\beta_{21}) = \begin{bmatrix} R_{\beta_{21}} & 0 \\ 0 & 1 \end{bmatrix}. \quad (15)$$

The coordinate system transformation can be represented as a graph (Figure 2). A directed edge represents a relationship between two coordinate systems and is associated with a homogeneous transformation. From Figure 2, we can get

$$M_{21} M_X = M_X \text{Rot}(\beta_{21}), \quad (16)$$

where

$$M_{21} = \begin{bmatrix} R_{21} & T_{21} \\ 0 & 1 \end{bmatrix}. \quad (17)$$

Substituting (13), (15), and (17) into (16), we get

$$\begin{bmatrix} R_{21} & T_{21} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_X & T_X \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_X & T_X \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{\beta_{21}} & 0 \\ 0 & 1 \end{bmatrix}, \quad (18)$$

$$\begin{bmatrix} R_{21} R_X & R_{21} T_X + T_{21} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_X R_{\beta_{21}} & T_X \\ 0 & 1 \end{bmatrix}, \quad (19)$$

$$R_{21} = R_X R_{\beta_{21}} R_X^{-1}, \quad (20)$$

$$T_{21} = T_X - R_{21} T_X. \quad (21)$$

FIGURE 2: Coordinate system transformation represented by a graph.

From (20) and (21), we can see that the R_{21} and T_{21} can be calculated by the rotation angle β_{21} provided that R_x and T_x are available.

4. Calculation Method of Position and Orientation of Rotation Axis

4.1. Orientation of Rotation Axis. Suppose that the camera be calibrated at different positions. At position θ_i , we get camera calibration data. Let $\beta_{i1} = \theta_i - \theta_1$, and let $R_{\beta_{i1}}$ denote the device rotation matrix with respect to rotation angle β_{i1} . R_{i1} and T_{i1} denote rotation matrix and translation vector between camera coordinate systems at the positions θ_i and θ_1 . Equation (20) can be written as

$$R_{i1}R_X = R_X R_{\beta_{i1}}. \quad (22)$$

When the rotation axis is fixed, R_x would be constant. Thus, given the values of R_{i1} , T_{i1} , and β_{i1} , we can solve (22) for R_x .

Using a normalized quaternion to define the rotation between two coordinate systems provides a simple and elegant way to formulate successive rotations [15–17]. Given rotation matrix

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}, \quad (23)$$

it can be transformed to the quaternion with the following equation [18]:

$$\begin{aligned} q_0 &= \sqrt{1 + r_1 + r_5 + r_9}, \\ q_x &= \frac{r_8 - r_6}{4 \times q_0}, \quad q_y = \frac{r_7 - r_3}{4 \times q_0}, \quad q_z = \frac{r_4 - r_2}{4 \times q_0}. \end{aligned} \quad (24)$$

Similarly, the quaternion can be transformed to the rotation matrix with the following equation [18]:

$$R = \begin{bmatrix} q_0^2 + q_x^2 + q_y^2 + q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_x q_y + q_0 q_z) & q_0^2 + q_y^2 - q_x^2 - q_z^2 & 2(q_y q_z - q_0 q_x) \\ 2(q_x q_z - q_0 q_y) & 2(q_y q_z + q_0 q_x) & q_0^2 + q_z^2 - q_x^2 - q_y^2 \end{bmatrix}. \quad (25)$$

Let q_{Ai} , q_X , and q_{Bi} denote the quaternion of R_{i1} , R_x , and $R_{\beta_{i1}}$, respectively. With quaternion, the sequence of rotation can be formulated as an equation without involving rotation matrices [15]. As a result, the problem of solving $R_{i1}R_X = R_X R_{\beta_{i1}}$ can be transformed into an equivalent problem involving the corresponding quaternion as follows [15]:

$$q_{Ai} \otimes q_X = q_X \otimes q_{Bi}. \quad (26)$$

Since the quaternion multiplication can be written in matrix form and with notations introduced in [19], we have the following [16]:

$$q_{Ai} \otimes q_X = Q(q_{Ai})q_X, \quad q_X \otimes q_{Bi} = W(q_{Bi})q_X, \quad (27)$$

where, letting $q = [q_0, q_x, q_y, q_z]$,

$$Q(q) = \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & -q_z & q_y \\ q_y & q_z & q_0 & -q_x \\ q_z & -q_y & q_x & q_0 \end{bmatrix}, \quad (28)$$

$$W(q) = \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & q_z & -q_y \\ q_y & -q_z & q_0 & q_x \\ q_z & q_y & -q_x & q_0 \end{bmatrix}.$$

Moreover, these two matrices are orthogonal [16], that is,

$$[Q(q)]^T Q(q) = [W(q)]^T W(q) = I. \quad (29)$$

Thus,

$$Q(q_{Ai})q_X = W(q_{Bi})q_X,$$

$$\begin{aligned} &\|Q(q_{Ai})q_X - W(q_{Bi})q_X\|^2 \\ &= [Q(q_{Ai})q_X - W(q_{Bi})q_X]^T [Q(q_{Ai})q_X - W(q_{Bi})q_X] \\ &= q_X^T \{2I - [Q(q_{Ai})]^T W(q_{Bi}) - [W(q_{Bi})]^T Q(q_{Ai})\} q_X \\ &= q_X^T [2I - C_i] q_X, \end{aligned} \quad (30)$$

where

$$C_i = [Q(q_{Ai})]^T W(q_{Bi}) + [W(q_{Bi})]^T Q(q_{Ai}). \quad (31)$$

Thus, the total error function allowing us to compute q_X becomes

$$\begin{aligned} f(q_x) &= \sum_{i=1}^n q_X^T [2I - C_i] q_X \\ &= q_X^T \left[2nI - \sum_{i=1}^n C_i \right] q_X \\ &= q_X^T [2nI - S] q_X, \end{aligned} \quad (32)$$

where n is the number of positions of the camera:

$$S = \sum_{i=1}^n C_i, \quad (33)$$

q_x is the unit quaternion. Therefore, q_x can be obtained by solving the following problem:

$$\begin{aligned} \min \quad & f(q_x) = \min \{q_x^T [2nI - S] q_x\} \\ \text{s.t.} \quad & q_x^T q_x = 1. \end{aligned} \quad (34)$$

4.2. *Position of Rotation Axis.* At position θ_i , (21) can be written as

$$(I - R_{i1}) T_X = T_{i1}. \quad (35)$$

When the rotation axis is fixed, T_x would be constant. Thus, given the values of R_{i1} and T_{i1} , we can solve (35) for T_X .

Let $E_i = [-T_{i1}, I - R_{i1}]$, $T_Y = [1, T_x^T]^T$, and thus

$$\begin{aligned} E_i T_Y &= 0, \\ \|E_i T_Y\|_2^2 &= T_Y^T E_i^T E_i T_Y. \end{aligned} \quad (36)$$

Let n denote the number of positions of the camera. The total error function is

$$g(T_Y) = \sum_{i=1}^n T_Y^T E_i^T E_i T_Y = T_Y^T \left[\sum_{i=1}^n E_i^T E_i \right] T_Y. \quad (37)$$

Since the camera rotation axis is approximately vertical to T_X , the value of t_z approaches zero. Thus, $T_y = [t_{y0}, t_{y1}, t_{y2}, t_{y3}]$ can be obtained by solving the following problem:

$$\begin{aligned} \min \quad & g(T_Y) = \min T_Y^T \left[\sum_{i=1}^n E_i^T E_i \right] T_Y \\ \text{s.t.} \quad & t_{y0} = 1 \\ & t_{y3} < 1. \end{aligned} \quad (38)$$

5. Calculation of Position and Orientation of Rotation Axis and 3D Coordinate

5.1. *Calculation of Position and Orientation of Rotation Axis.* Based on the previous discussions, the complete process of calculation of position and orientation of rotation axis is outlined below:

- (1) The chessboard (see Figure 3) is printed and plastered on plane. By rotating and moving the camera properly, a set of chessboard images is collected. Then, the values of the intrinsic parameters of the camera, f_x , f_y , c_x , and c_y , are obtained by calling the method proposed in [14].

FIGURE 3: Chessboard.

- (2) The camera is fixed on a camera tripod, and the rotation axis of the camera, which is parallel to the z axis of the sensor coordinate system, lies in an approximately horizontal plane.
- (3) By rotating the camera around the fixed axis to the different positions, another set of chessboard images are collected.
- (4) The extrinsic parameters of the camera at the positions θ_i , R_i , and T_i are obtained by calling the function `cvFindExtrinsicCameraParams2` in OpenCV [20].
- (5) The rotation matrix and translation vector between the camera coordinate systems at the positions θ_i and θ_1 , R_{i1} , and T_{i1} are figured out by (12). The rotation angle β_{i1} and its corresponding rotation matrix $R_{\beta_{i1}}$ are also calculated by (2) and (14).
- (6) The R_{i1} and $R_{\beta_{i1}}$ are converted into quaternions q_{Ai} and q_{Bi} , respectively, by using the method proposed by Bar-Itzhack [18].
- (7) q_x is found by solving problem (34). As a result, the R_x can be obtained.
- (8) T_X is obtained by solving problem (38).

5.2. *3D Coordinate Calculation.* Based on the previous discussions, we present the complete process of 3D coordinate calculation as follows:

- (1) The camera is fixed on a camera tripod whose rotation axis lies in an approximately horizontal plane.
- (2) At certain position θ_1 , the image of the object I_1 is collected.
- (3) By rotating the camera on the fixed axis to another position θ_2 , we get another image of the object I_2 .
- (4) The rotation angle β_{21} and its corresponding rotation matrix $R_{\beta_{21}}$ are figured out by (2) and (14).
- (5) With $R_{\beta_{21}}$, R_{21} and T_{21} are calculated by (20) and (21).
- (6) The image coordinate of the point of the object, (X_1, Y_1) , on the image I_1 is appointed manually.

- (7) The corresponding image coordinate in the image I_2 , (X_2, Y_2) , can be determined by stereo correspondence method, for example, the function FindStereoCorrespondenceBM in Opencv [20] or by manual.
- (8) The 3D coordinate of the point on the object relative to O_1 -xyz, (x_1, y_1, z_1) , can be figured out by (6).

6. Experiments

Since digital camera with 3-axis accelerometer sensor is not available for us, iPhone 4, which has the sensor, is adopted. To simulate digital camera, the phone is placed in a box which is fixed on a camera tripod. And it is ensured that the z axis of the sensor (experiment results show that the axis is parallel to the optical axis of the camera) is parallel to the rotation axis of the camera tripod, so that the value of G_z keeps steady in the course of the rotation. In calibration course, the distance between the camera and the chessboard is about 1000 mm.

Figure 4 illustrates the curve of the quaternion of the rotation matrix between the camera coordinate systems at the i th position and 1st position, with respect to the rotation angle. The quaternion $q' = [q'_0, q'_x, q'_y, q'_z]$ was calculated by the proposed method with rotation angle, while $q = [q_0, q_x, q_y, q_z]$ was converted directly from rotation matrix R_{i1} , which was from calibration data. From the graph, one can see that the proposed method can calculate the rotation matrix by rotation angle.

Figure 5 plots the curve of translation vector between the camera coordinate systems at the i th position and 1st position, with respect to rotation angle. The vector $T' = [T'_x, T'_y, T'_z]$ was calculated by the proposed method with rotation angle, while $T = [T_x, T_y, T_z]$ was directly from calibration data T_{i1} . From the graphs, one can see that the proposed method can calculate effectively translation vector by rotation angle.

In order to estimate the accuracy of 3D coordinate calculated by the proposed method, the chessboard which has a bigger block than the one for calibration is printed. The width of the block is 46.6 mm, and the distance between the chessboard and the camera is about 1200 mm. The distance between two neighbor corners rather than the distance between the camera and the chessboard is calculated, because the measurement of the former by manual is easier. For simplicity, the corners of the blocks in images are automatically detected by calling the Opencv function “cvFindChessboardCorners” [19]. Figure 6 depicts the measurement error ratio of the distance between two corners with respect to rotation angle.

7. Conclusions

This paper proposed a stereo vision system with single camera, which requires digital camera with 3-axis accelerometer sensor rotating around the fixed axis which is parallel to the axis z of the sensor. Under these conditions, the slope angle relative to gravity, which can be figured out from the readouts of the sensor, can determine the camera position, and the rotation angle between two positions can determine the rotation matrix and translation vector from

FIGURE 4: Curve of the quaternion of rotation matrix with respect to rotation angle; $q = [q_0, q_x, q_y, q_z]$ is converted from rotation matrix, and $q' = [q'_0, q'_x, q'_y, q'_z]$ is calculated by the proposed method.

FIGURE 5: Curve of translation vector with respect to rotation angle; $T = [T_x, T_y, T_z]$ was directly from calibration data, and $T' = [T'_x, T'_y, T'_z]$ was calculated by (23) with rotation angle.

one coordinate system of the camera to another. Accordingly, given the rotation angle and the images of the object at two different positions, one before and the other after camera rotation, the 3D coordinates of the points on the object can be determined. Theoretical analysis and experimental results show the validity of our method.

It should be noticed that few digital cameras are provided with 3-axis accelerometer sensor. However, to obtain stereo vision, we believe that the inexpensive sensor embedded in

FIGURE 6: Measurement error ratio (%) of the distance between two corners with respect to rotation angle.

digital camera is worthy. Moreover, due to higher image quality and larger focus range, higher accuracy and larger range of measurement may be obtained. Furthermore, the smart phone which has the sensor is popular. If a mini fixed rotation axis is built in a corner of the phone and it does not move in the course of rotation, with the proposed method, the phone may estimate the size of the object being focused on and distance between the phone and the object.

Acknowledgment

This work is supported by Science and Technology Research Project of Chongqing's Education Committee (KJ110806).

References

- [1] Y. Sooyeong and N. Ahuja, "An omnidirectional stereo vision system using a single camera," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 861–865, August 2006.
- [2] D. Marr and T. Poggio, "A computational theory of human stereo vision," *Proceedings of the Royal Society of London*, vol. 204, no. 1156, pp. 301–328, 1979.
- [3] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.
- [4] H. C. Longuet-higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.
- [5] J. Gluckman and S. K. Nayar, "Catadioptric stereo using planar mirrors," *International Journal of Computer Vision*, vol. 44, no. 1, pp. 65–79, 2001.
- [6] D. Lee and I. Kweon, "A novel stereo camera system by a biprism," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 528–541, 2000.
- [7] T. Svoboda and T. Pajdla, "Epipolar geometry for central catadioptric cameras," *International Journal of Computer Vision*, vol. 49, no. 1, pp. 23–37, 2002.
- [8] D. Southwell, A. Basu, M. Fiala, and J. Reyda, "Panoramic stereo," *International Conference on Pattern Recognition*, vol. 1, pp. 378–382, 1996.
- [9] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, 2000.
- [10] A. Criminisi, "Single-view metrology: algorithms and applications," in *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, vol. 2449 of *Lecture Notes in Computer Science*, pp. 224–239, 2002.
- [11] G. Wang, Z. Hu, F. Wu, and H. Tsui, "Single view metrology from scene constraints," *Image and Vision Computing*, vol. 23, no. 9, pp. 831–840, 2005.
- [12] A. Saxena, S. H. Chung, and A. Y. Ng, "3-D depth reconstruction from a single still image," *International Journal of Computer Vision*, vol. 76, no. 1, pp. 53–69, 2008.
- [13] J. Rekimoto, "Tilting operations for small screen interfaces," in *Proceedings of the 9th ACM Annual Symposium on User Interface Software and Technology*, pp. 167–168, November 1996.
- [14] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [15] J. C. K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *International Journal of Robotics Research*, vol. 10, no. 3, pp. 240–254, 1991.
- [16] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.
- [17] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 4647–4653, October 2006.
- [18] I. Y. Bar-Itzhack, "New method for extracting the quaternion from a rotation matrix," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 6, pp. 1085–1087, 2000.
- [19] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-D location parameters using dual number quaternions," *Image Understanding*, vol. 54, no. 3, pp. 358–367, 1991.
- [20] B. Gary and K. Adrian, *Learning OpenCV*, O'Reilly Media, 2008.