CALIBRATION OF MOTORIZED OBJECT RIG AND ITS APPLICATIONS

Pang-Hung Huang1, Yu-Pao Tsai2,3, Wan-Yen Lo1, Sheng-Wen Shih4, Chu-Song Chen2, and Yi-Ping Hung1,2,5

1Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
2Institute of Information Science, Academia Sinica, Taipei, Taiwan
3Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan
4Department of Computer Science and Information Engineering, National Chi Nan University, Nantou, Taiwan
5Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan

Email: hung@csie.ntu.edu.tw

ABSTRACT

Object Movies (OMs) have been successfully used in many applications. However, the techniques for acquiring OMs still need to be improved if high-quality and efficient OMs are desired. In this paper, we present a method for calibrating a motorized object rig to facilitate the acquisition of OMs. We first apply the CPC kinematic model to formulate the 3D configuration of the device, and then propose a method to estimate the parameters of the CPC model of the device. Furthermore, a visual tool is provided for users to adjust the controllable axes of the rig according to the estimated results. After this calibration, more accurate camera parameters can be obtained and then be used for different purposes. In this work, we use the parameters to reconstruct, from an OM, the 3D model of the 3D object, and then adjust the OM according to the center of the 3D model so that a high-quality OM can be obtained for rendering.

1. INTRODUCTION

Recently, image-based techniques for modeling and rendering high-quality and photo-realistic 3D objects have become a popular research topic. Having the advantage of being photo-realistic, object movie is especially suitable for delicate artifacts and thus has been widely applied to many areas, e.g., e-commerce, digital archive, digital museum [9], etc.. This technique was first proposed in Apple QuickTime VR [4]. An object movie is a set of images taken from different perspectives around a 3D object; when the images are played sequentially, the object seems to be rotating around itself. Each image in an OM is associated with a pair of distinctive pan and tilt angles of the viewing direction, and thus a particular image can be chosen and shown on screen according to mouse motion of the user. In this way, users can interactively rotate the virtual artifacts arbitrarily and enjoy freely manipulating the object.

To acquire object movies (OMs), we use the motorized object rig, AutoQTVR, developed by Texnai Inc. The motorized object rig is a computer-controlled 2-axis omniview shooting system, as shown in Fig. 1. It has two rotary axes: the pan-direction object rotator and the tilt-direction camera arm rotator. For convenience, we will refer to the rotation axes of the both rotators by the tilt and the pan axes, respectively.

Fig. 1. Motorized object rig – AutoQTVR.

In OM acquisition, the center of the object should be placed at the crossing point of the two rotation axes and the optical axis of the camera, as shown in Fig. 1. Otherwise, the acquired OM would have a bizarre rotation effect when it is browsed. As a result, to acquire high quality OMs that rotate smoothly, one should manage to make the three axes intersect at a common point first. However, since the optical axis of the camera is invisible, aligning these three axes is inherently a difficult problem. To our knowledge, there is no simple and efficient method for solving this three-axis alignment problem. In this paper, we propose a method to calibrate the motorized object rig to make the three axes as close as possible. With our calibration method, accurate camera parameters can be easily estimated and consequently the quality of the acquired OMs can be remarkably improved.

482
To calibrate the motorized object rig, we first develop a method to estimate the three axes of it. We then provide a visual tool for users to adjust the motorized object rig according to the estimated results. After the adjustment, the three axes will intersect at a common point \( C_0 \) as shown in Fig. 1. The details of the calibration process will be described in Section 2. Even though the three-axis alignment problem can be solved perfectly, in practice, it is still hard to align \( C_0 \) (the center of the object) exactly with \( C_S \). Thus, we further propose a method for adjusting all images of the acquired OMs by utilizing the estimated camera parameters. The details will be described in Section 3. The experimental results and conclusions will then be described in Section 4 and Section 5, respectively.

2. CALIBRATION OF MOTORIZED OBJECT RIG

To calibrate the motorized object rig, we first use the camera mounted on the AutoQTVR to capture some feature points, whose 3D positions are known beforehand. The 2D and 3D positions of the feature points are used to estimate the intrinsic and extrinsic camera parameters. With the estimated extrinsic camera parameters, we can reconstruct the kinematic model of the rig. Then, we apply a simple and practical model, completely and parameter continuous (CPC) model [1][13], to formulate the relation among the three axes. Finally, we provide a visual tool showing the axes for users to adjust the motorized object rig. If the intersections of the rays are not close enough, the user can adjust the motorized object rig according to the estimated result, and then the axes will be estimated again. The whole process will be repeated until the intersections of the rays are close enough. After calibration, reliable extrinsic parameters of the camera will be available with the kinematic model.

2.1. Estimation of Camera Parameters

We adopt the method proposed by Zhang [12] to estimate the intrinsic camera parameters. The method performs camera calibration with at least two images of a known planar pattern captured at different orientations.

On the other hand, we adopt the method presented in [3] and [5] to estimate the extrinsic camera parameters, by first using the method proposed by Kato et al. [6] to obtain a set of initial extrinsic parameters, and then applying Iterative Closest Point (ICP) principle [2] to refine them.

2.2. CPC Model

A CPC model stands for the completely and parameter continuous kinematic model [13]. A complete model means the model provides enough parameters to express any variation of the actual robot structure, and parameter continuity implies no model singularity by adopting a singularity-free line representation [10].

This model was motivated by the special needs of robot calibration. It is assumed that the robot links are rigid. A CPC kinematic model for a revolution/prismatic joint can be represented as follows (we refer the reader to [13] for detail descriptions):

\[
^iT_{i+1} = \begin{cases} 
Q_i V_i, & \text{for revolute joint} \\
\text{Trans}(0, 0, q_i^l) & \text{for prismatic joint}
\end{cases}
\]

where \(^iT_{i+1}\) denotes the transformation matrix between any two consecutive joint frames, i.e., the \((i+1)\)-th reference frame to the \(i\)-th reference frame. \(Q_i\) is the motion matrix defined as follows:

\[
Q_i = \begin{cases} 
\text{Rot}((q_i),) & \text{for revolute joint} \\
\text{Trans}(0, 0, q_i^l) & \text{for prismatic joint}
\end{cases}
\]

\(q_i\) denotes joint value, which means the rotation angle for a revolute joint, or the amount of displacement for a prismatic joint, and \(V_i\) denotes the constant shape matrix. The shape matrix is a general transformation matrix given by

\[
V_i = [R | t] = R_i \text{Rot}((\beta_i)) \text{Trans}(l_{i,x}, l_{i,y}, l_{i,z})
\]

where

\[
R_i = \begin{bmatrix}
1 - \frac{b_{i,x}^2}{1 + b_{i,z}} & -b_{i,x}b_{i,y} & b_{i,x} & 0 \\
b_{i,x} & \frac{1}{1 + b_{i,z}} & b_{i,y} & 0 \\
-1 & \frac{b_{i,y}}{1 + b_{i,z}} & b_{i,z} & 0 \\
0 & -b_{i,x} & b_{i,z} & 1
\end{bmatrix}
\]

and

\[
\text{Trans}(l_{i,x}, l_{i,y}, l_{i,z}) = \begin{bmatrix}
1 & 0 & 0 & l_{i,x} \\
0 & 1 & 0 & l_{i,y} \\
0 & 0 & 1 & l_{i,z} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

The rotation matrix \(R_i\) is used to describe the relative orientation of the two consecutive joint axes, \(\text{Rot}((\beta))\) is used to align the \(x\)- and the \(y\)-axes. Notice that the CPC convention requires that any two consecutive joint axes have a nonnegative inner product, i.e., \(b_{i,z} \geq 0\). In general, this requirement can be achieved by changing the sign of one of the joint values of consecutive joints. This is because changing the sign of the joint value is equivalent to reversing the joint axis for both revolute and prismatic joints [11].
2.3. Kinematic Calibration Using the CPC Model

In this section, we will introduce how to apply the CPC model to estimate the transformation matrices among the coordinate systems defined on the motorized object rig. As shown in Fig. 2, we define three axes of three different reference frames on the rig. Let \( z_c, z_r, \) and \( z_p \) denote the \( z \)-axes of the coordinate system (CCS), the tilt-axis coordinate system (TCS), and the pan-axis coordinate system (PCS), respectively.

![Fig. 2. The schematic of motorized object rig.](image)

For convenience, let the camera be the “end-effector” of the motorized object rig. Thus, we can obtain the corresponding robot pose with the method described in Section 2.1. In general, the orientations of the \( x \)- and the \( y \)-axes of the coordinate systems need not to be specified in formulating the kinematics of the motorized object rig [11][13]. Therefore, the redundant parameter \( \beta_j \) can be set to zero, and the transformation matrix from object coordinate system (OCS) to camera coordinate system (CCS) can be simplified as follows:

\[
^{t}T_o = ^{c}T_c \times ^{T}T_p \times ^{p}T_o = Q_0 \times V_0 \times Q_1 \times V_1 \times Q_2 \times V_2, \tag{7}
\]

where \(^{t}T_o\) denotes the transformation matrix from coordinate system \( o \) to coordinate system \( b \).

Since the motorized object rig is composed of two revolution joints, the motion matrix \( Q_0 \) is a constant matrix which can be set to identity, whereas \( Q_1 \) and \( Q_2 \) are the rotation matrices about the \( z_r \)- and the \( z_p \)-axes, respectively. The equations of \( Q_0, Q_1, \) and \( Q_2 \) are given by

\[
Q_0 = I_{4 \times 4}
\]

\[
Q_1 = \text{Rot}_z(\theta_t), \text{ where } \theta_t = \text{sign}_t \times q_t^i \tag{8}
\]

\[
Q_2 = \text{Rot}_z(\phi_p), \text{ where } \phi_p = \text{sign}_p \times q_p^i
\]

where \( \text{sign}_t \) and \( \text{sign}_p \) are either +1 or -1, and \( q_t^i \) and \( q_p^i \) are the rotation angle about the tilt and the pan axes, respectively. Substituting (8) into (7), we have

\[
^{c}T_o = V_0 \times \text{Rot}_z(\theta_t) \times V_1 \times \text{Rot}_z(\phi_p) \times V_2
\]

\[
= R_0 \times \text{Trans}(l_0) \times \text{Rot}_z(\theta_t) \times R_1 \times \text{Trans}(l_1) \times \text{Rot}_z(\phi_p) \times R_2 \times \text{Trans}(l_2)
\]

\[
= \begin{bmatrix}
R_0 \times \text{Rot}_z(\theta_t) \times R_1 \times \text{Rot}_z(\phi_p) \times R_2 \\
0
\end{bmatrix}
\]

\[
\times \begin{bmatrix}
l_0 \times l_1 + R_0 \times \text{Rot}_z(\theta_t) \times l_2 + R_1 \times \text{Rot}_z(\phi_p) \times l_2
\end{bmatrix}
\]

\[
\times \begin{bmatrix}
l_0 \times l_1 + R_0 \times \text{Rot}_z(\theta_t) \times l_2 + R_1 \times \text{Rot}_z(\phi_p) \times l_2
\end{bmatrix}
\]

\[
\times \begin{bmatrix}
l_0 \times l_1 + R_0 \times \text{Rot}_z(\theta_t) \times l_2 + R_1 \times \text{Rot}_z(\phi_p) \times l_2
\end{bmatrix}
\]

where \( r^c_a \) and \( t^c_a \) are the rotation matrix and the translation vector of the transformation matrix \(^{c}T_o\). From (9), we have

\[
^{c}r_o(\theta_t, \phi_p) = R_0 \times \text{Rot}_z(\theta_t) \times R_1 \times \text{Rot}_z(\phi_p) \times R_2 \tag{10}
\]

and

\[
^{c}t_o(\theta_t, \phi_p) = R_0 \times l_0 + R_1 \times \text{Rot}_z(\theta_t) \times l_1 + R_2 \times \text{Rot}_z(\phi_p) \times l_2
\]

(11)

In the following subsections, we will show how to solve the parameters, \( R_0, l_0, R_1, l_1, R_2, l_2 \) in (10) and (11).

2.3.1. Rotation Parts

In order to simplify the calibration process, we calibrate one axis at a time. Therefore, when calibrating the tilt-axis, the pan-axis is held still, i.e., \( \varphi_p \) can be regarded as a constant, and thus \( R_1 \times \text{Rot}_z(\phi_p) \times R_2 \) becomes a constant term denoted by \( X \). By substituting \( X \) into (10), we have

\[
^{c}r_o(\theta_t, \phi_p) = R_0 \times \text{Rot}_z(\theta_t) \times X \tag{12}
\]

Equation (12) can be rewritten in the following form

\[
X = \text{Rot}_z(-\theta_t) R_0^{-1} \times ^{c}r_o(\theta_t, \phi_p) \tag{13}
\]

By maneuver the tilt axis to two different joint values, \( \theta_t \) and \( \theta_j \), from (12) and (13), we have

\[
^{c}r_o(\theta_t, \phi_p) \times ^{c}r_o(\theta_j, \phi_p)^{-1} \times R_0 = R_0 \times \text{Rot}_z(\theta_j - \theta_t) \tag{14}
\]

Multiplying \([0 0 1]^T\) on both sides of (14), we have

\[
^{c}r_o(\theta_t, \phi_p) \times ^{c}r_o(\theta_j, \phi_p)^{-1} \times \bar{b}_0 = \bar{b}_0
\]

\[
\Rightarrow \left[ ^{c}r_o(\theta_t, \phi_p) \times ^{c}r_o(\theta_j, \phi_p)^{-1} \right] \times \bar{b}_0 = \varepsilon \approx 0
\]

\[
\Rightarrow \bar{A}b_0 = \varepsilon
\]

where \( \varepsilon \) denotes the error vector induced by the observation noise, and \( \bar{b}_0 \) can be estimated by minimizing \( \|\varepsilon\|^2 \). It is well known that \( \bar{b}_0 \) is the unit eigenvector of
Again, by solving an eigenvalue problem, we obtain \( \tilde{b}_1 \) which leads to the rotation matrix \( R_1 \). The sign parameter \( \text{sign}_p \) for \( \phi_p \) and also be determined by minimizing an objective function similar to (16).

The final orientation parameter \( R_2 \) can be computed with the following objective function derived from (10).

\[
\min_{R_2} \left\| e_{r_0}(\theta_i, \phi_p) - R_0 \cdot \text{Rot}_z(\theta_i) \cdot R_1 \cdot \text{Rot}_z(\phi_j) \cdot R_2 \right\|_F^2
\]

subject to \( R_1^2 R_2 = I \) and \( \det R_2 = 1 \). This constrained optimization problem can be solved with a method similar to the one proposed in [2].

### 2.3.2. Translation Parts

By substituting the estimated rotation matrices into (11), we have the following equations for the translation parameters:

\[
i_t = M_{3x3} \tilde{e}_t,
\]

where \( M_{3x3} = \begin{bmatrix} l_{0,1} & l_{0,2} & l_{0,3} \\ l_{1,0} & l_{1,2} & l_{1,3} \\ l_{2,0} & l_{2,1} & l_{2,2} \end{bmatrix} \). By moving the pan and the tilt joints to different positions, we have an over-determined system of the translation parameters which can be solved using the least square method.

### 2.3.3. Axes Adjustment

After solving the kinematic parameters of the motorized object rig, we can compute its forward kinematic model as follows:

\[
\begin{align*}
T_i &= V_i \times Q_i \times V_i \times Q_i \\
&= R_{0} \times \text{Trans}(l_z) \times \text{Rot}(\theta) \times R_{0} \times \text{Rot}(\phi) \times R_{0} \times \text{Trans}(l_z) \\
&= T_0
\end{align*}
\]

(21)

Given the tilt angle, \( \theta_n \) and the pan angle, \( \phi_p \), we can use (21) to determine the pose of the camera. Also, the forward kinematic model can be used to find the representations of \( z_n \), \( z_p \) and \( z_t \) axes, i.e., the orientation and position of these three axes. First, the transformation matrix from the reference frame of the tilt axis to the CCS can be determined as \( e_{T_t} = V_0 \). Thus, the unit direction vectors of the tilt axis \( z_n \), denoted by \( \text{ori}_{tilt} \), can be derived as follows

\[
\text{ori}_{tilt} = e_{T_t} \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T = V_0 \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T,
\]

(22)

The position of the tilt axis, denoted by \( \text{pos}_{tilt} \), is given by

\[
\text{pos}_{tilt} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T.
\]
\[ \text{pos}_{\text{lux}} = \epsilon T_x \times \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} t \]
\[ = V_0 \times \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} t'. \]  
(23)

Similarly, the orientation and position of the pan axis \( z_p \), denoted by \( \text{ori}_{\text{pan}} \) and \( \text{pos}_{\text{pan}} \), can be found to be
\[ \text{ori}_{\text{pan}} = \epsilon T_x \times T_p \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} t', \]
\[ = V_0 \times \text{Rot}_z(\theta_p) \times V_1 \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} t'. \]  
\[ \text{pos}_{\text{pan}} = \epsilon T_x \times T_p \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} t', \]
\[ = V_0 \times \text{Rot}_z(\theta_p) \times V_1 \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} t'. \]  
(24)
(25)

respectively.

By using equations (22)-(25), the positions and orientations of the three axes of \( z_v, z_t, \) and \( z_p \) can be evaluated and then can be illustrated as shown in Fig. 4 (a). The positions of these three axes can be adjusted to minimize the distance among them. According to our experiences, when the maximum distance among these three axes is smaller than a threshold value of 15 mm, the effect of the miss-alignment of these three axes is negligible.

3. APPLICATIONS

After the calibration mentioned in Section 2 has been done, the axes of the motorized object rig can be adjusted to proper positions and the reliable camera parameters can be obtained. Such reliable camera parameters can then be applied to many areas.

3.1. Reconstruction of Visual Hull

Visual hull [8] is an approximate geometric representation of an object. Kutulakos and Seitz [7] have proposed an algorithm, space carving, to compute the 3D shape of an unknown, convex-shaped object from multiple photographs taken at known viewpoints.

Let us consider the pinhole camera model, which projects a 3-D point \( P \) in the scene to a 2-D point \( p \) in the image as follows:
\[ p \cong K[R] \begin{bmatrix} P \end{bmatrix} t. \]  
(26)

where \( K \) is the intrinsic parameters, \( R \) is rotation matrix, and \( t \) is translation vector. Now, consider a voxel \( V \) of the volume \( V \), we can regard \( V \) as a 3D point in space. From (26), we can calculate the projection point \( p \) in the image, and if it belongs to background, the voxel \( V \) will be carved. Since the camera parameters could be well estimated with the method proposed in Section 2, a more subtle 3D model could be obtained in this way.

3.2. Centralization of Object Movies

When acquiring OMs, the user needs to match the rotation center of the object (\( C_0 \)) with that of shooting (\( C_S \)). However, it is difficult to match \( C_0 \) with \( C_S \) exactly, so we propose a centralization method to adjust the acquired OMs.

After reconstructing the object, we can obtain a coarse 3D model. Then we calculate the center of the object by averaging all the positions of voxels of the visual hull. Assume \( p_i \) is the center of image plane \( I_i \), and \( Q \) is the center of the visual hull. Since the intrinsic and extrinsic parameters of each image are known, we can obtain \( q_i \), the projection of \( Q \) on \( I_i \), by (26). Then calculate the translation vector from \( q_i \) to \( p_i \) and shift the image with the vector so that \( q_i \) is shifted to the center of the image, as shown in Fig. 3. After centralization, the quality of the resulted OM can be improved.

![Figure 3. The schematic of centralization process.](image)

3.3. Others

Besides the applications described in the previous sections, there are still many applications of using the estimated camera parameters. For example, we can reconstruct the lighting conditions of the object and perform relighting on the object, or use the reconstructed 3D model and the camera parameters to improve the image segmentation results.

4. EXPERIMENTAL RESULTS

To calibrate the parameters of the motorized object rig, we found that only 12 images are enough to obtain a set of highly accurate parameters. That is, we only need to take 12 pictures at each adjustment-calibration process, and the processing time needed, including capturing and processing, is about 7 minutes.
Fig. 4 shows the result before aligning the three axes of the rig where the estimates of the three axes are shown in Fig. 4(a), and the acquired OM of a toy shark is shown in Fig. 4(b). The estimation and adjustment process is repeated five times to align the three axes of the rig and the result is shown in Fig. 5. From the frontal view of Fig. 5(d), we show that the tilt axis can be effectively adjusted to be perpendicular to the pan axis and optical axis of camera with our method. Moreover, from the top view of Fig. 5(d), the intersections of the three axes are close enough. Some images of the OM of the toy shark are shown in Fig. 5(a). After the visual hull of the shark is constructed, shown in Fig. 5(c), the centralization process can be performed, and the resulted OM is shown in Fig. 5(b).

**5. CONCLUSION**

In this paper, we presented a method for calibrating the motorized object rig, and introduced a visual tool for users to adjust the axes of the motorized object rig. After adjustment, the distances among all the three axes of the motorized object rig can be minimized, and more reliable camera parameters could be obtained after the calibration process. Furthermore, by utilizing the obtained camera parameters, we proposed a software method for automatically adjusting the acquired OM to improve its quality. This work should be useful for promoting future adoption of OMs.

**ACKNOWLEDGEMENTS**

This work is supported in part by the grants of NSC- 93-2422-H-002-022 and NSC 93-2422-H-001-0004.

**REFERENCES**