Pose Estimation based on Four Coplanar Point Correspondences

Yang Yang
Research Institute of Robotics,
Shanghai Jiaotong University
Shanghai200240, P.R. China
yangyangme@sjtu.edu.cn

Qixin Cao
Research Institute of Robotics,
Shanghai Jiaotong University
Shanghai200240, P.R. China
qxcao@sjtu.edu.cn

Charles Lo
Research Institute of Robotics,
Shanghai Jiaotong University
Shanghai200240, P.R. China
charleslo77@gmail.com

Zhen Zhang
Research Institute of Robotics,
Shanghai Jiaotong University
Shanghai200240, P.R. China
zzh2200_0@126.com

Abstract

In this paper we present a pose estimation algorithm based on four coplanar point correspondences. Given four coplanar points and their corresponding image points under a perspective camera, plus the camera's intrinsic matrix, the camera's rotation and translation relative to the object plane is determined directly. In essence, the pose estimation problem is converted to the calculation of a planar homography between the object plane and the image plane. Experiments with both synthetic data and real images verify the correctness of this algorithm.

1. Introduction

Pose estimation is a hot spot in both photogrammetry and computer vision, and is widely used in robot navigation, hand-eye coordination, visual servoing, ect. Specifically, accurate and fast pose estimation is the foundation for augmented reality, in which case we need to compute the camera's pose relative to the marker with high precision in real time. Different kinds of features can be used for pose estimation, such as correspondence from points, lines, curves, and surfaces. Liu et al [1] used four corresponding straight lines to compute the camera's relative pose. The outstanding augmented reality toolkit, ARToolkit [2] also uses four straight lines. The line correspondence paradigm possesses the merit that lines are easy to detect. However, they also bear the disadvantages that they are inclined to fail in the case of occlusion. If part of the surrounding rectangle is occlude, ARToolkit cannot determine the pose of the marker. Pose estimation from curve or surface features is a relative new method [3], but it is too sophisticated and imposes more difficulties on the detection of these features. Therefore, we focus on the point correspondence based pose estimation problem.

Pose estimation from point correspondences is coined as the "perspective-n-point" problem (PnP) [4]. Based on whether or not the camera's intrinsic matrix is known at a priori, different number of points is used. According to Haralick [5], the "perspective-three-point" problem (P3P) was first solved by the German mathematician Grunert in 1841. However, it is no early than 1981 until this problem was brought into the computer vision community by Fischler and Bolles[4]. They derived a direct solution to this problem and revealed that through three points people cannot obtain a unique solution (generally there exists four ambiguities). Long Quan and Zhongdan Lan [6] proposed a family of linear methods that yield a unique solution to four point pose determination for generic reference points. However, their algorithm needs double SVD decomposition and is computational expensive. DeMenthon and Davis[7] combined the direct and iterative algorithms to solve the P4P problem with only 25 lines of code which is known as the POSIT algorithm. It is accurate and fast. But it requires that the four points must not be coplanar. Nister [8] used five points and solved the P5P problem by computing the coefficients of a tenth degree polynomial in closed form. In essence, he used the essential matrix between two views and obtained the relative rotation and translation through the decomposition of the essential matrix. Without the camera's intrinsic parameters given as a prior knowledge, there are lots of literatures using seven or eight point correspondence such as [9]. In essence, these literatures convert the pose estimation problem

978-0-7695-3735-1/09 $25.00 © 2009 IEEE
DOI 10.1109/FSKD.2009.310

into the computation of the fundamental matrix and obtain the relative rotation and translation by decomposing the fundamental matrix.

In this paper, we focus on the P4P problem. Specifically, all the four points lie on the same plane but none of the three are collinear. We focus on this coplanar P4P problem because it is of special importance for the marker based augmented reality, in which case the marker is usually a plane. The remainder of this paper is organized as follows: Section II gives a formal statement for the coplanar P4P problem. Section III illustrates our algorithm which determines a unique solution using four coplanar point correspondences. In section IV, experiments with both synthetic data and real images are presented. In the last section, we conclude our work and give some discussions.

2. Problem Statement

Given four coplanar points, among which none of the three are collinear, and given their corresponding image points under a perspective camera, plus the camera's intrinsic parameters (especially, the camera's focal length and principal point), determine the camera's pose (translation and rotation, six degrees of freedom) relative to the object plane.

As shown in Figure 1, $Q_{i=1,2,3,4} \in \Re^4$ are the homogenous coordinates of the four points on the object plane Π, $Q_i = (x_i, y_i, z_i, w_i)^T$. $p_{i=1,2,3,4} \in \Re^3$ are the corresponding homogenous coordinates on the image plane. $p_i = (x_i', y_i', \lambda_i)^T$.

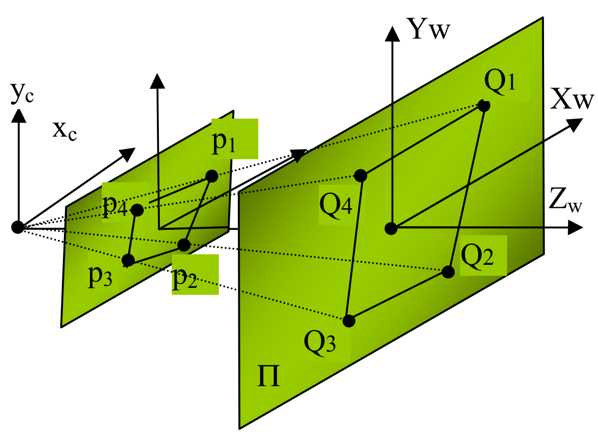

Figure 1. Graph showing the relationship between the object plane and the image plane

If the camera's intrinsic matrix is K, where $K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$ with f_x , f_y as the focal length, and c_x, c_y the principal point, we have

$$p = KR[I_{3\times3} \,|\, -C]Q$$

(1)

where R is a 3×3 rotation matrix representing the orientation of the camera coordinate frame, and C is a 3×1 vector representing the origin of the camera coordinate frame in the world coordinate frame. Since all the four points are coplanar, we can set the world coordinate frame in such a way that the plane Π coincides with the Oxy plane. That means all the four points Q_i can be represented as $Q_i = (x_i, y_i, 0, w_i)^T$ .Substitute Q into equation (1) we have

$$p = K[R_1 \quad R_2 \quad -RC]\tilde{Q}$$

(2)

where R_1, R_2 are the first and second column of R, $\tilde{Q}_i = (x_i, y_i, w_i)^T$. As long as the third coordinate of C is not equal to zero, i.e., the origin of the camera coordinate frame is not on the plane Π, $K[R_1 \quad R_2 \quad -RC]$ is reversible and hence we can use a planar homography H to relate p and $\tilde{Q}$:

$$p = K[R_1 \quad R_2 \quad -RC]\tilde{Q} = H\tilde{Q} \quad , \quad H \in \Re^{3\times3}$$

(3)

As stated above, the intrinsic matrix K is known at a priori, hence we can multiply the inverse of K at both sides of equation (3) and get the following form

$$p' = K^{-1}p = [R_1 \quad R_2 \quad -RC]\tilde{Q} = H'\tilde{Q}$$

(4)

where $p_i' = (u_i, v_i, \lambda_i)^T$. Usually the last element of the homogeneous vector p' is set to 1 through dividing itself by λ_i.

Hence the problem can be restated as: given the four points' coordinates in the world coordinate frame $Q_{i=1,2,3,4} = (x_i, y_i, 0, w_i)^T$, and their corresponding image plane coordinates $p'_{i=1,2,3,4} = (u_i, v_i, \lambda_i)^T$, plus the camera's intrinsic matrix $K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$, find the relative rotation matrix R and translation vector C.

3. Pose from Four Points

Through the above analysis, the pose estimation problem is converted to the calculation of a planar homography matrix. According to equation (4), p' equals to $H'\tilde{Q}$ up to a scale, there are eight unknown elements in H (nine elements minus one for the overall scale). Each point correspondence provides two constrains. Therefore, four non-collinear point correspondences can determine a unique H'.

As stated above, p' equals to $H'\tilde{Q}$ up to a scale, equation (4) can be expressed in the cross product form as

$$p' \times H'\tilde{Q} = 0$$

(5)

And

$$p' \times H'\tilde{Q} = \begin{pmatrix} vh^{3T}\tilde{Q} - \lambda_i h^{2T}\tilde{Q} \\ \lambda_i h^{1T}\tilde{Q} - uh^{3T}\tilde{Q} \\ uh^{2T}\tilde{Q} - vh^{1T}\tilde{Q} \end{pmatrix} = 0$$

(6)

where h^{iT} is the i-th row of H'.

Equation (6) can be rearranged as

$$\begin{pmatrix} 0_{1\times3} & -\lambda_i\tilde{Q}_i^T & v\tilde{Q}_i^T \\ \lambda_i\tilde{Q}_i^T & 0_{1\times3} & -u\tilde{Q}_i^T \\ -v\tilde{Q}_i^T & u\tilde{Q}_i^T & 0_{1\times3} \end{pmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = Ah = 0$$

(7)

where $A \in \Re^{3\times9}$, $h \in \Re^{9\times1}$. It is apparent that only two rows of A are linearly independent, hence we can reduce A to $\Re^{2\times9}$ and equation (7) becomes

$$Ah = \begin{pmatrix} 0_{1\times3} & -\lambda_i\tilde{Q}_i^T & v\tilde{Q}_i^T \\ \lambda_i\tilde{Q}_i^T & 0_{1\times3} & -u\tilde{Q}_i^T \end{pmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0$$

(8)

Each point correspondence provides such an equation. So with four point correspondences, we can stack each of such equation together and get a matrix $A \in \Re^{8\times9}$. Matrix A has rank 8 with h as its null space. There are plenty of methods to solve this kind of homogenous equations. For example, we can compute the SVD decomposition of A. h is the singular vector corresponding to the smallest singular value [10]. Once $h \in \Re^{9\times1}$ is obtained, we can rearrange it as $H' \in \Re^{3\times3}$. To obtain the rotation matrix from H', we need to normalize H' as the following

$$H' = \frac{H'}{\|H_1'\|}$$

(9)

In this way, the first and second columns of H' are converted to unit vectors and hence we get the first and second columns of the rotation matrix R (according to equation (4)). The third column of R is computed as the cross product of R_1 and R_2. The translation vector of the camera coordinate system's origin is computed as

$$C = -R^T H_3'$$

(10)

4. Experiments

4.1. With Synthetic Data

The experiment with synthetic data is carried out following seven steps:

1). Generate four random points on the object plane: $Q_i = (x_i, y_i, 0, 1)^T$. x_i and y_i follow a uniform distribution on a specified interval [a,b]. In our experiment we choose a=-3, b=3;

2). Generate three random angles rotX, rotY, rotZ which follow a uniform distribution on the interval $[-\pi, \pi]$. Hence the three rotation matrixes are computed as

$$Rx = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(rotX) & -\sin(rotX) \\ 0 & \sin(rotX) & \cos(rotX) \end{bmatrix},$$

$$Ry = \begin{bmatrix} \cos(rotY) & 0 & \sin(rotY) \\ 0 & 1 & 0 \\ -\sin(rotY) & 0 & \cos(rotY) \end{bmatrix},$$

$$Rz = \begin{bmatrix} \cos(rotZ) & -\sin(rotZ) & 0 \\ \sin(rotZ) & \cos(rotZ) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The overall rotation matrix is computed as $R = Rx \cdot Ry \cdot Rz$.

3) Generate a translation vector $C=(t_1,t_2,t_3)^T$. t_1, t_2 and t_3 follow a uniform distribution on an interval [-1,1];

4) Set the camera's intrinsic matrix equal to identity. Hence the overall camera matrix is

$$P_{cam}=K\left[R \quad -RC\right].$$

5) Following equation (1), the image points are computed as $p=KR[I_{3\times3}\,|-C]Q$.

6) Convert the image point p to its inhomogeneous form.

7) Computer the homography matrix H using the algorithm presented in section III and finally obtain the rotation matrix R_c and translation vector C_c. We use the subscript c to distinguish the computed result from the true value.

Figure 2 shows the result. The four object plane points are marked by a red *. The green lines represent the skeleton of the camera's true position. The red lines represent the skeleton of the camera's calculated position. They coincide together which indicates that the computed pose match the true pose correctly. (We adapt some functions of EGT [11] to plot this figure.)

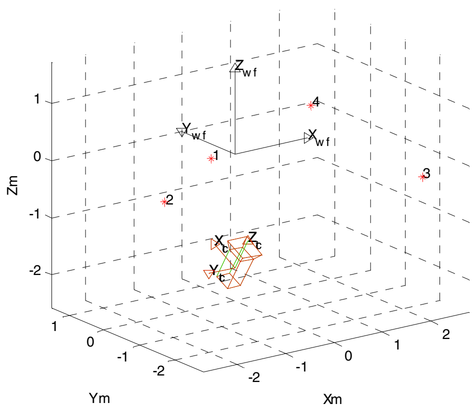

Figure 2. Experiment result with synthetic data.

4.2. With Real Images

To test the POSIT algorithm, literature [7] constructs a video based 3D mouse to control a cube. The 3D mouse is simply a combination of a web-camera and 4 small infrared sources. With the coordinates of these infrared sources in the object coordinate frame and their corresponding image coordinates, the relative pose of the camera is computed with POSIT algorithm and is used to control the cube such that the cube moves following the movement of the camera. Enlightened by them, we construct a similar video based 3D mouse shown in Figure 3(a). It is just a CD with four red marker equally distributed. The distance between the red marker and the CD's center is 49mm. We then use a web-camera and CamShift algorithm to track these four markers and obtain their centroids' image coordinates. The tracked markers are highlighted with red ellipses as shown in Figure 3(b). The object's pose relative the web-camera is computed using the algorithm proposed in this paper and is used to control a cube drawn with OpenGL. This experiment indicates that our algorithm can determine the relative pose correctly and rapidly.

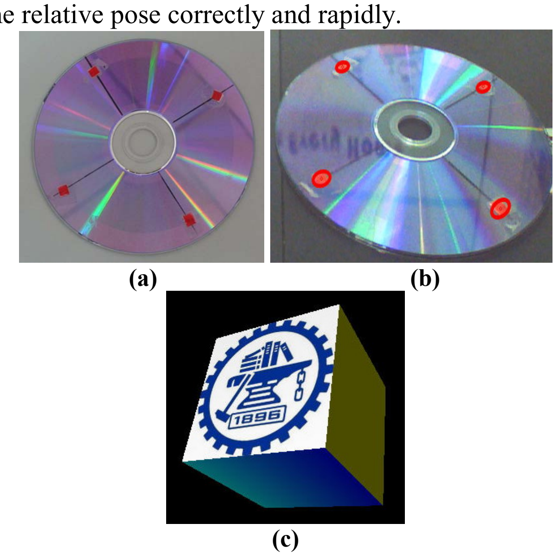

Figure 3. Experiment with real images. (a) shows the simple "3D mouse" with four markers. (b) shows the tracking of these four markers with CamShift algorithm. (c) shows the cube drawn with OpenGl control by the "3D mouse" using our pose estimation algorithm.

5. Conclusion

In this paper we demonstrate a method for solving the planar point correspondence based pose estimation problem. Because of the planarity, this problem is naturally converted into the computation of homography between planes. Through utilizing the mature homography computation algorithms, our method can be easily extended in the following ways. On one hand, more points can be used to improve the algorithm's robustness. For example, with more than four points, we can use RANSAC [4] to remove the disturbance of outlier points. On the other hand, because the duality of points and lines, line features can also be used to computer the homography and hence obtain the relative pose in a similar manner.

6. Acknowledgements

This work is supported by the National High Technology Research and Development Program of China under grant 2007AA041703-1, 2006AA040203 and 2007AA041602.

7. References

[1] Lijuan Qin, Yulan Hu, Yingzi Wei, Yue Zhou and Hong Wang, “Pose Determination of 3D Object Based on Four Straight Lines,” Fourth International Conference on Natural Computation, Jinan, vol. 6, 2008, pp. 188-192.

[2] http://www.washington.edu/artoolkit/

[3] Bodo Rosenhahn and Gerald Sommer, “Pose estimation of 3D free-form contours,” International Journal of Computer Vision, vol. 62, no. 3, 2005, pp. 267-289.

[4] Fischer and Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Graphics and Image Processing, vol. 24, no. 6, 1981, pp. 381–39.

[5] Robert M. Haralick, Chung-nan Lee, Karsten Ottenberg and Michael Nolle, “Review and analysis of solutions of the three point perspective pose estimation problem,” International Journal of Computer Vision, vol. 13, no. 3, 1994, pp. 331-356.

[6] Long Quan and Zhongdan Lan, “Linear N-Point Camera Pose Determination,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 8, 1999, pp. 774-780.

[7] Daniel F. DeMenthon and Larry S. Davis, “Model-based object pose in 25 lines of code,” International Journal of Computer Vision, vol. 15, no. 1, 1995, pp. 123-141.

[8] David Nister, “An Efficient Solution to the Five-Point Relative Pose Problem,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 6, 2004, pp. 756-770.

[9] Richard I. Hartley, “In defence of the eight-point algorithm,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, 1997, pp. 580-593.

[10] Richard I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge Univ. Press, 2000.

[11] G.L. Mariottini and D. Prattichizzo, "EGT: a Toolbox for Multiple View Geometry and Visual Servoing," IEEE Robotics and Automation Magazine, vol. 3, no. 12, 2005, pp. 26-39.