Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Pose Estimation for Generalized Imaging Device via Solving Non-Perspective *N* Point Problem

Chu-Song Chen and Wen-Yan Chang

Institute of Information Science, Academia Sinica, Taipei, Taiwan

Email: song@iis.sinica.edu.tw

Abstract

In this paper, we present a systematic method for pose estimation of such a generalized imaging device. We formulate it as a non-perspective n point (NPnP) problem. The case with exact solutions, n=3, is investigated comprehensively. Approximate solutions can also be found for n>3 with our approach in a least-squared-error manner. The proposed method can be used not only to perspective imaging devices, but also non-perspective ones.

1. Introduction

In the past, many methods were developed for solving the pose estimation problem for perspective imaging devices, where the imaging rays are assumed to intersect at a common point. For some applications such as tele-presence and image-based virtual reality, the perspective property has better to be taken into account because the generated images are supposed to be presented to humans. However, for some other applications such as automatic visual surveillance and mobile robot guidance, the imaging system need not comply with the perspective rule.

In fact, in recent years, many new types of imaging methodologies or devices violating the perspective rule were designed. That is, the imaging rays may not intersect at a common point. For example, Rademacher and Bischop introduced the concept of images with multiple centers of projection, which were applied to image-based rendering [11]. A linear pushbroom camera [6] contains multiple focal centers distributed in a line. It is also possible to acquire a non-perspective image in a single shot. For instance, wide-angle lens systems including sever projective distortions may have a locus of viewpoints [10]. An omni-directional vision sensor combining a camera and a conic mirror, which was employed for collision avoidance of robotics, is another example of non-perspective imaging devices [14].

However, it still lacks of systematic methods for pose estimation of an imaging device that is non-perspective. Therefore, how to design a general pose-estimation method for non-perspective imaging devices is important. In this paper, we propose a pose-estimation method for an arbitrary imaging device.

2. Problem Formulation

First, we formulate the model of the imaging devices considered in this paper. In essence, an imaging device captures the rays of lights in the 3D space. Since these rays are occluded by the physical occupation of the imaging device itself, the end points of these rays are inherently determined. Hence, an imaging device can be generally formulated via the three components, (*I, CCS, L*), defined below.

(1) $I(\cdot,\cdot)$: $D_I \rightarrow$ R×G×B is an image map ($D_I \subset \mathcal{R}^2$ is the domain of image I), and R, G, B are the sets consisting of the three primitive colors, respectively.

(2) *CCS*: an arbitrary Euclidean coordinate frame selected in the 3D space, which is referred to as the camera coordinate system (CCS).

(3) $L(\cdot,\cdot)$: $D_I \rightarrow \mathcal{R}^3 \times \mathcal{R}^3$ is a mapping from an image point, say (i, j), to the 3D ray represented as (c, v) with respect to CCS, which consists of all the 3D points that can be imaged at (i, j), where $c \in \mathcal{R}^3$ is the end point and $v \in \mathcal{R}^3$ is the normalized directional vector of this ray.

We call the model formulated above a *generalized imaging device* (GID) in this paper. Fig. 1 gives an illustration of the GID. The color grabbed in a particular point in the image, $I(i,j)$, $i,j \in D_I$ is thus a blending of the light intensities of the rays in $N(L;i,j)$, a neighborhood of $L(i,j)$. [1]

Given a GID **G**, let $\Gamma(\mathbf{G})=\{l$: a 3D line | $\exists(c,v)$ s.t. the ray specified by (c,v) is contained in $l\}$. If all of the lines contained in $\Gamma(\mathbf{G})$ intersect at a common point, then **G** is

[1] In [5], Grossberg and Nayar presented a more general imaging model, in which an image point corresponds to a bundle of rays, and it is useful for identifying the point spread function for each pixel. Since we focuses on the geometrical calibration of the imaging devices in this paper, the imaging model is formulated by associating an image point with a single ray, which simplifies considerably the problem for estimation of the parameters about rigid transformations.

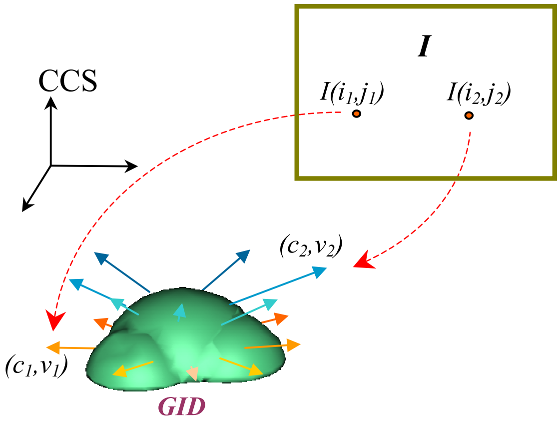


Figure 1. An illustrative example of the generalized imaging device (GID).

called *perspective*. Otherwise, **G** is *non-perspective*. For example, a common video camera is usually modeled as a perspective GID. The concept of GID is suitable for formulating the geometrical relation of optical apparatuses designed for capturing images in a 3D environment. Consider such a general definition of imaging devices, a basic problem is that: *Given a set of 3D points w.r.t a world coordinate system (WCS) and their projecting points in the image plane of a GID, how can the rigid transformation between the world and the camera coordinate systems be computed?* Such a fundamental problem is called the *perspective n point problem* (P*n*P) for perspective imaging devices [4][15][7][8][3][9]. In this paper, we refer to the problem as the *non-perspective n point problem* (NP*n*P) because the GIDs considered herein need not be perspective.

In the past, the P*n*P problem has been well investigated. Closed-form solutions have been formulated if three or four 3D/2D correspondences are adopted [4][7]. However, if more correspondences are used, closed-form solutions do not exist. Lowe [8] and Yuan [15] used the Newton-Raphson method for pose estimation under the assumption that approximate initial poses were provided. The Dementhon and Davis approach [3] first assumed that the camera model is orthographic. It obtains the rigid transformation by solving a linear system, and then uses a POSIT procedure to refine the result iteratively. Lu et al. [9] proposed an orthogonal iteration method for finding the camera poses.

3. Non-Perspective Three Point Problem

In an NP*n*P problem, n points in the 3D space, e.g., $P_1, P_2, \ldots, P_n$, w. r. t. a WCS are supposed to be imaged with a GID. Assume that their 2D image points are $(i_1, j_1), (i_2, j_2), \ldots, (i_n, j_n)$, respectively, where $(i_k, j_k)\in D_I$ for all $k = 1, \ldots, n$. We want to find the rigid transformation between CCS and WCS such that $Q_k=\boldsymbol{R}\cdot P_k+\boldsymbol{t}$, where $\boldsymbol{R}$ is a 3×3 rotation matrix, $\boldsymbol{t}$ is a 3×1 translation vector and Q_k is a coordinate in CCS that can be represented as $Q_k=s_k\cdot v_k+c_k$, in which $(c_k, v_k) = L(i_k, j_k)$ and s_k is a scale factor for all $k = 1, \ldots, n$.

First, we investigate the problem when n=3, which is the minimal number of 3D/2D correspondences allowing the solutions to be identified exactly. The induced problem is called the NP3P problem in this paper.

3.1. Solutions of the NP3P Problem

When n=3, the three points P_1, P_2, and P_3 form a triangle as shown in Fig. 2(a). Since the coordinates of P_1, P_2, and P_3 are known, the lengths of the three edges of the triangle, a, b, c, can be obtained, respectively. Consider the transformed points, Q_1, Q_2, and Q_3 lying on the corresponding rays, $\boldsymbol{l}_1$, $\boldsymbol{l}_2$, $\boldsymbol{l}_3$, respectively, where $\boldsymbol{l}_k = (c_k, v_k)$, $k = 1, 2, 3$. Denote l_k to be the full line containing the ray $\boldsymbol{l}_k$, $k = 1, 2, 3$, and let l_{ij} be the commonly orthogonal line between l_i and l_j, $i,j=1,2,3$ and $i\neq j$, respectively. Denote q_{ij} and q_{ji} to be the intersection points between l_i and l_{ij}, and l_j and l_{ij}, respectively. Let the distance between q_{ij} and q_{ji} be d_{ij}. Without loss of generality, consider the coordinate system whose origin is q_{12} and whose **x**-axis is defined to be along the same direction of l_{12}. In addition, the **y**-axis of this coordinate system is also defined to be along the direction of l_1, and the **z**-axis is defined to be the cross product of **x** and **y** axes, respectively. By using this coordinate system, Q_1 and Q_2 can be represented as follows:

$$Q_1=(0, t_1, 0) \text{ and } Q_2=(d_{12}, t_2\cos\theta_{12}, t_2\sin\theta_{12}),$$

where t_1 is the distance between q_{12} and Q_1, t_2 is the distance between q_{21} and Q_2, and θ_{12} is the angle between the directions of l_1 and l_2 (i.e., $\theta_{12} = acos(v_1,v_2)$). By using the property that the distance between Q_1 and Q_2 is a, the following constraint is satisfied:

$$d_{12}^2+(t_2\cos\theta_{12}-t_1)^2+(t_2\sin\theta_{12})^2=a^2$$

$$\Rightarrow \quad t_2^2-2t_1t_2\cos\theta_{12}+t_1^2=a^2-d_{12}^2 \quad (1)$$

Note that the parameters used for describing (1), including the distance between q_{12} (or q_{21}) and Q_1 (or Q_2) and the angle between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, are all independent of the coordinate systems being selected. Hence, consider the line pair (l_1, l_3), we also have the following constraint by using the property that the distance between Q_1 and Q_3 is b:

$$t_3^2-2(t_1-d_1)t_3\cos\theta_{13}+(t_1-d_1)^2=b^2-d_{13}, \quad (2)$$

where t_3 is the distance between q_{31} and Q_3 and t_1-d_1 is the distance between q_{13} and Q_1 (and thus the distance between q_{12} and q_{13} is $|d_1|$).

Similarly, consider the line pair (l_2, l_3), we also have

$$(t_3-d_3)^2-2(t_3-d_3)(t_2-d_2)\cos\theta_{23}+(t_2-d_2)^2=c^2-d_{23}^2, \quad (3)$$

where t_3-d_3 (or t_2-d_2) is the distance between q_{32} (or q_{23}) and Q_3 (or Q_2).

Equations (1), (2), and (3) give three constraints on the three unknowns, t_1, t_2 and t_3. Organizing the constraints in this way help us considerably to simplify the solutions of special cases, as will be shown in Section 3.2. Generally, since each of the (1), (2), and (3) is a quadratic polynomial equation associated with two unknowns, the solutions can be obtained by solving eighth-order polynomial equations with a single variable, as shown in the following. From (1) and (2), t_2 and t_3 can both be represented with t_1, respectively:

$$t_2=t_1\cos\theta_{12}\pm(a^2-d_{12}^2-t_1^2\sin^2\theta_{12})^{1/2} \quad (4)$$

$$t_3=(t_1-d_1)\cos\theta_{13}\pm[b^2-d_{13}^2-(t_1-d_1)^2\sin^2\theta_{13}]^{1/2} \quad (5)$$

Substituting (4) and (5) into (3), we can derive an equation of the following form:

$$A_2\pm A_1(B_2)^{1/2} = \pm B_1(C_2)^{1/2}\pm(B_2C_2)^{1/2}, \quad (6)$$

where A_2, B_2, C_2 are all second-order polynomials and A_1, B_1 are both first-order polynomials, in terms of t_1, respectively. Taking the square of both sides of (6), we obtain

$$A_2^2+A_1^2B_2\pm2A_2A_1(B_2)^{1/2} = B_1^2C_2+B_2C_2\pm2B_1C_2(B_2)^{1/2}$$

or equivalently,

$$A_2^2+A_1^2B_2-B_1^2C_2-B_2C_2 = 2(\pm B_1C_2\pm A_2A_1)(B_2)^{1/2}. \quad (7)$$

Taking the square of both sides of (7) yields eighth-order polynomial equations in terms of t_1.

Although there is no analytic way to solve a polynomial equation of eighth order, it is not difficult to find the solutions of them numerically. Then, the other coefficients, t_2 and t_3, can be obtained by substituting the solution of t_1 into (4) and (5). The priori knowledge of the 3D points located on the positive direction of the corresponding rays, i.e., $s_k>0$, $k = 1, 2, 3$, can be used to eliminate inappropriate solutions as well.

3.2. Special Cases of the NP3P Problem

To solve a general NP3P problem requires solving the eighth-order polynomial equations as described above. In this section, we investigate some special cases of the NP3P problem whose solutions can be obtained by solving polynomial equations whose orders are at most four, instead of eighth-order ones. Since there are analytical representations of the solutions of a fourth-order polynomial equation, the solutions of these special cases can be expressed in closed forms.

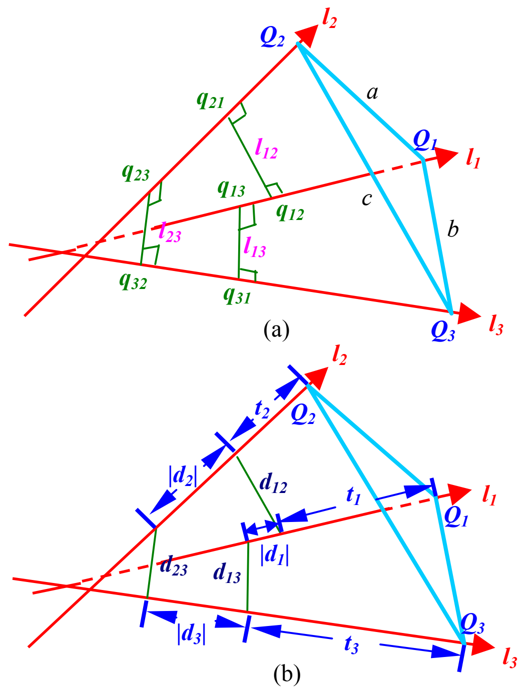


Figure 2. Illustration of the definitions associates with the NP3P problem.

Case 1 [Linear Pushbroom]: All the rays of a linear pushbroom camera model [6] are emitted from a line and orthogonal to this line. The linear pushbroom camera can be used to model X-ray imageries and local behaviors of satellite images [6]. We will show that its pose estimation problem has analytical forms of solutions. Consider that in this case, $d_1=d_2=d_3=0$, and thus (1), (2), (3) become (8), (9), (10), respectively.

$$t_2^2-2t_1t_2\cos\theta_{12}+t_1^2=a', \quad (8)$$

$$t_3^2-2t_1t_3\cos\theta_{13}+t_1^2=b', \quad (9)$$

$$t_3^2-2t_3t_2\cos\theta_{23}+t_2^2=c', \quad (10)$$

where $a'=a^2-d_{12}^2$, $b'=b^2-d_{13}^2$, and $c'=c^2-d_{23}^2$, respectively. Let $x_1 = 1/t_1$, $x_2 = t_2/t_1$, $x_3 = t_3/t_1$, a fourth order polynomial equation in terms of x_3 can be derived, and thus the pose estimation problem of a linear pushbroom camera can be solved analytically.

Case 2 [Partially Parallel]: We call an NP3P problem *partially parallel* if any two of the three lines, l_1, l_2, l_3, are parallel. Without loss of generality, assume that l_1 and l_2 are parallel. In this case, $\cos\theta_{12}=1$, and thus (1) becomes

$$(t_2-t_1)^2=a' \quad (11)$$

That is, $t_2=t_1\pm(a')^{1/2}$. In particular, we can also choose appropriately the coordinate that makes $d_1=0$. Hence, (2) becomes

$$t_3=t_1\cos\theta_{13}\pm[-t_1^2\sin^2\theta_{13}+b']^{1/2} \quad (12)$$

Substituting both (11) and (12) into (3) yields fourth-order polynomial equations in terms of t_1. Hence, analytical solutions can be derived for the partially parallel case as well.

4. Non-Perspective *N* Point (NP*n*P) Problem

The analysis in Section 3 shows that exact solutions can be identified for the NPnP problem when n=3. However, when n>3, exact solutions may not exist due to image noises. It is therefore necessary to find approximate solutions. In this paper, we develop a systematic method that finds an initial estimate of the approximate solutions first, as introduced in Section 4.1. Then, an iterative optimization procedure is proposed for refining the solutions, as introduced in Section 4.2.

4.1. Initialization for the NP*n*P Problem

The idea of our approach to initialization of the NPnP problem is to exploit the solutions of the three-point case. In general, there are $n!/(3!(n-3)!)$ triples of 3D/2D correspondences that can be served as initial estimates. Among them, to find a better one is desired -- better in the sense that the triple gives a more accurate estimate than the other triples. Since in general, the closer the image points are to each other, the less accurate the estimate is. It is thus better to use the triple of image points where the triangle formed by it has large enough area. The initialization procedure is shown as follows.

Algorithm 1: {Consider n 3D points, $P_1, P_2, ..., P_n$, w.r.t. a WCS, and the corresponding 2D image points $p_1, p_2, ..., p_n$, w.r.t. the CCS, where $p_m\in D_I$ for all $m = 1, ...,n$. Assume that the full lines containing the rays associated with these image points to be $l_1, l_2, ..., l_n$, respectively.}

Step 1. Repeat *Steps 1.1-1.4* K times, where K is a positive integer.

*1.1***.** Select three points among $\{p_1, p_2, ..., p_n\}$ in the image randomly. Assume that they are p_i, p_j, and p_k, respectively.

1.2. If l_i, l_j, and l_k are either too coplanar or the area of the triangle formed by p_i, p_j, and p_k is too small, then go back to *Step 1*.

1.3. Compute the rigid transformations between WCS and CCS, which are associated with the three point-line pairs, (P_i, l_i), (P_j, l_j), and (P_k, l_k).

1.4. For each rigid transformation computed in Step *1.3*, say, $(\boldsymbol{R}, \boldsymbol{t})$, transform all the other 3D points with this rigid transformation by $P'_m=\boldsymbol{R}P_m+\boldsymbol{t}$, where $m=1, ...,n$. Compute the sum of squared distances (SSDs) between P'_m and l_m, $m=1, ...,n$. Record both the SSD value e and its corresponding rigid transformation $(\boldsymbol{R}, \boldsymbol{t})$.

Setp2. Let e be the smallest among all the SSD values recorded in *Step 1.3*, and let $(\boldsymbol{R}, \boldsymbol{t})$ be the recorded rigid transformation corresponding to e. Output $(\boldsymbol{R}, \boldsymbol{t})$.

The above algorithm computes a rigid transformation with the smallest error among K random selections of the triples of 3D/2D correspondences. It can serve as a good initial estimate for the NPnP problem. The iteration times, K, is selected by balancing between accuracy and time. The larger the K, the more the triples are used, and thus the higher the chance of having a more accurate solution. However, too large a K leads to a long period of execution time. In our implementation, K is usually selected to be 100.

Another issue worthy to be addressed is the computation of the rigid transformations associated with the three pairs, (P_i, l_i), (P_j, l_j), and (P_k, l_k), in Step *1.3*. A generally effective way is to use the method for solving the NP3P problem as introduced in Section 3.1, which obtains rigid transformations that transforms P_i, P_j, and P_k to lying in l_i, l_j, and l_k, respectively. However, the computational efficiency may be diminished because of the following two reasons. (1) It requires finding all the real-number solutions of several eighth-order polynomial equations that can only be solved numerically. (2) There may be many solutions satisfying an NP3P problem and all of them need to be further processed in Step *1.4*.

To increase the efficiency of this algorithm, we suggest using a perspective camera to approximate the GID being considered, and the computations involved in Step *1.3* are thus reduced to finding solutions of a P3P problem instead of an NP3P one. Since the solutions of a P3P problem can be represented analytically, it is easy to identify the real-number solutions among them. In addition, the number of real-number solutions of a P3P problem is at most 4, which is much smaller than that of the NP3P one. Although the solution obtained by solving a P3P problem is an approximated one compared with its NP3P counterpart, it usually suffices to be an initial estimate for an NPnP problem by considering that the NP3P solution itself serves as an approximation to the NPnP problem.

In the remainder of this section, we focus on how to approximate a GID with a perspective camera. This problem is equivalent to finding a virtual center and a virtual image plane. First, the virtual center is obtained as a 3D point, P_c, that minimizes the following criterion:

$$\sum_m dist^2(P_c, l_m), \tag{13}$$

where $dist(P_c, l_m)$ is the distance between P_c and l_m.

Now, consider the selections of the virtual image plane. When a virtual plane is selected, the intersection points, $g_1, ..., g_n$, of this plane and all the rays of the GID can be computed respectively. The line l'_m passing through P_c and g_m then serves as an approximated line of l_m for all $m = 1, ...,n$. Hence, given a 3D point P lying in l'_m, its distance to l_m is, however, dependent on how far P is away from g_m. The farther is P away from g_m, the larger is $dist(P, l_m)$. Accordingly, the accuracy of such an approximation is distance-dependent. When P is away from the virtual plane (e.g., P is a distant 3D point), then the approximation is likely to be very poor. Since no prior knowledge about the locations of the 3D points to be imaged with a GID is given, we propose an *infinite-plane approximation* strategy for approximating the 3D rays of GID by using the rays of a perspective imaging device. In this strategy, the virtual plane is selected as the infinite plane, and line l'_m is thus parallel to line l_m that it approximates for all $m = 1, ...,n$. Hence, the distance between l_m and l'_m is a constant, $dist(P_c, l_m)$. The advantage of such an approximation strategy is that the accuracy of the approximation is independent of the locations of the 3D points, which allows the 3D points to be treated evenly in the pose-estimation process.

After approximating the GID to be processed with a perspective imaging model described above, the method for solving the P3P problem as introduced in Section 3.2 can then be used to find the required rigid transformations in Step *1.3* of Algorithm 1.

4.2. Convergent Iterations for NP*n*P

Given an initial estimate of the rigid transformation between WCS and CCS, we further refine it by minimizing an objective function iteratively. Consider that the projection of a point P onto a ray $\boldsymbol{l} = (c,v)$ can be represented as:

$$Proj(P;\boldsymbol{l}) = vv^T(P-c)+c. \tag{14}$$

The orthogonal vector from P to $Proj(P;\boldsymbol{l})$ is thus

$$Proj(P;\boldsymbol{l})-P = (v_iv_i^T-I)(P-c_i), \tag{15}$$

where I is the 3 by 3 identity matrix. The length of the vector defined in (15) is therefore the distance between P and $\boldsymbol{l}$. The objective function being minimized in our approach is

$$E = \min_{R,t} \sum_m||(v_mv_m^T-I)(\boldsymbol{R}P_m+\boldsymbol{t}-c_m)||^2. \tag{16}$$

To find the optimal solution ($\boldsymbol{R^*}$, $\boldsymbol{t^*}$) of (16), we adopt the iterative-closest point (ICP) principle introduced by Besl et al. [1]. The ICP algorithm always converges monotonically to a minimum value of a mean-square distance metric, and the rate of convergence is more rapid than that of generic nonlinear optimization methods (such as the Gauss-Newton method). Although it does not guarantee that the global minimum can always be found, it does suggest that the global minimum (or a very approximate local minimum) can be obtained from a very board range of initial guesses. The ICP algorithm has also been adopted for solving pose estimation problem for the perspective case [13]. Lu et al. [9] have recently proposed a method that is very similar to the ICP algorithm for solving the P*n*P problem as well.

The principle of the ICP algorithm is the iteration of the following two stages. (1) Find the closest point in the corresponding line for each 3D point. (2) Find a rigid transformation that transforms the 3D points to their closest points in a least-squared-error manner.

Algorithm 2: {The same variables defined in Algorithm 1 are used}

Step 0. Let ($\boldsymbol{R_0}$, $\boldsymbol{t_0}$) be the initial rigid transformation estimated using Algorithm 1.

Step 1. Compute $P^*_m= \boldsymbol{R_0}P_m+\boldsymbol{t_0}$ for all $m = 1, ...,n$.

Step 2. For each point P^*_m, find its closest point, P'_m, in l_m. That is, $P'_m = vv^T(P^*_m-c)+c$ for all $m = 1, ...,n$.

Step 3. Find the rigid transformation that minimizes the sum of squared distances between P^*_m and P'_m, $m = 1, ...,n$. That is, find ($\boldsymbol{R_{new}}$, $\boldsymbol{t_{new}}$) that minimizes $\sum_m||\boldsymbol{R_{new}}P^*_m + \boldsymbol{t_{new}} - P'_m||^2$.

Step 4. If $\boldsymbol{R_{new}}$ is close enough to the identity matrix and $\boldsymbol{t_{new}}$ is also close enough to the zero vector, then **stop**. Else, compose ($\boldsymbol{R_{new}}$, $\boldsymbol{t_{new}}$) and ($\boldsymbol{R_0}$, $\boldsymbol{t_0}$) by $\boldsymbol{R_0} \leftarrow \boldsymbol{R_{new}R_0}$, $\boldsymbol{t_0} \leftarrow \boldsymbol{t_{new}}+\boldsymbol{R_{new}t_0}$, and go to *Step 1*.

In *Step 3* of Algorithm 2, the least-squared-error transformation between two sets of 3D points has a closed-form solution, which can be solved via singular value decomposition [1].

5. Experimental Results

An omni-directional camera composed of a lens and a curved mirror is used in our experiment, as shown in Fig. 3(a). The reflection curve of this camera is designed to maximize the average image resolutions in a range of viewing angles, but not deliberated to satisfy the single view-point constraint. Hence, such an imaging device is a non-perspective GID but with higher image resolutions and better point-spread properties than those designed to satisfy the single view-point constraint. Such a camera is thus more suitable for robot guidance and ego-motion estimation. The intrinsic model of this GID, $L(\cdot,\cdot)$, has been investigated in the manufacturing process. That is, for all $(i,j)\in D_I$, the corresponding ray, $L(i,j)$, w.r.t. a selected CCS of the GID is known. Nevertheless, even if the intrinsic model of the GID has not been investigated during the manufacturing process, it can also be calibrated via some other methods (e.g., the one introduced in [5]).

In practice, the quantization error, image correspondence error, and the calibration error (of the intrinsic model) can all generate errors to the estimated poses. To verify the accuracy of our method, two such omni-directional GIDs were used and thus a non-perspective stereo pair was formed. We put this stereo setup in an indoor environment, and some 3D points (totally 38 points) in this environment were measured in advance and employed for pose estimation, as shown in Fig. 3(b). After using the method introduced above, the poses of both imaging devices were estimated respectively. Hence, a calibrated stereo pair of omni-directional cameras was constructed, which could help us compute the 3D coordinate of any other point in this environment if its corresponding image points had been identified in both images. In this way, we computed the coordinates of some 3D points in this environment and used them to verify the accuracy of the poses estimated with our method. The left part of Table I lists the errors measured for some right angles, while the right part of Table I lists the errors measured for some length ratios, where line 0 shown in Fig. 4 serves as the unit length. The 3D reconstruction results show that our method is very accurate.

In addition, since a stereo pair is formed, the correspondence of a point selected in one image should lie in a curve in the other image, as illustrated in Fig. 5. It is called the epi-polar line in the perspective case, and is referred to as the *matching curve* here. Fig. 6(a) shows some points selected in one image. If no errors occurr, their associated matching curves should pass through the corresponding points in the other image. Fig. 6(b) shows the matching curves of the points shown in 6(a). As can be seen, these matching curves all pass through the corresponding points in a close manner.

6. Summary

In this paper, we have proposed a method for pose estimations of generalized imaging devices. Since the imaging devices considered in our framework may not be perspective, their pose estimation problem is referred to as the NP*n*P problem in this paper. First, we investigated the case when n=3 and presented how to get its exact solutions. Some particular useful special cases, such as the linear pushbroom and partially parallel camera models, have also been investigated and they were shown

TABLE I. Errors for some right angles and length ratios.

	Real angle	Estimated angle	Error	Real ratio	Estimated ratio	Error
Case 1	90	89.9967	0.36*e-4	0.7550	0.7384	0.021987
Case 2	90	90.0017	0.20*e-4	0.8926	0.9005	0.008851
Case 3	90	90.0070	0.78*e-4	0.0624	0.0623	0.001603
Case 4	90	90.0138	1.54*e-4	0.1812	0.1809	0.001655
Case 5	90	90.0164	1.83*e-4	0.0990	0.1043	0.053535

to have analytic solutions. We observed that the solutions of the NP3P problem can be served as an initial estimate for obtaining an approximated solution for the NP*n*P problem, and a random-selection strategy was developed to identify a better triple of 3D/2D correspondences for getting this initial estimate. In addition, to increase the efficiency of the initial-estimation stage, a perspective camera model was also proposed and used for approximating a GID. Finally, the iterated-closest point (ICP) principle was adopted for refinement the pose initially estimated.

Although a non-perspective imaging device was used in our experiment, the proposed method can be applied not only to non-perspective imaging devices, but also perspective ones. Our approach thus provides a generally effective way for pose estimation of general imaging devices. Experimental results have also shown that our method is quite accurate.

Acknowledge: This work was supported in part by National Science Council, R.O.C. under Grants NSC 90-2213-E-001-033 and 90-2213-E-001-014.

References

[1] K. S. Arun, T. S. Huang and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Trans. Pat. Ana. & Mach. Int., Vol. 9, No. 5, 1987.*

[2] P. J. Besl and N. D. McKay, "A Method for Registration of 3D Shapes," *IEEE Trans. Pat. Ana. & Mach. Int., Vol. 14, No. 5, 1992.*

[3] D. F. Dementhon and L. S. Davis, "Model-Based Object Pose in 25 Line of Code," *Int. J. Computer Vision*, 15, pp. 123-141, 1995.

[4] M. A. Fischler and R. C. Bolls, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Iimage Analysis and Automated Cartography," *Commun. ACM, Vol 24, 1981.*

[5] M. D. Grossberg and S. K. Nayar, "A General Imaging Model and A Method for Finding Its Parameters," *Proc. IEEE Int. Conf. Computer Vision, July 2001.*

[6] R. Gupta and R. Hartley, "Linear Pushbroom Cameras," *IEEE Trans. Pat. Ana. & Mach. Int.*, Vol. 19, No. 9, pp. 963-975, 1997.

[7] R. M. Haralick, C. N. Lee, K. Ottenberg and M. Nolle, "Analysis and Solutions of the Three Point Perspective Pose Estimation Problem," *Proc. IEEE Int. Conf. Computer Vision &*

Pattern Recognition, 1991.

[8] D. G. Lowe, "Robust Model-based Motion Tracking Through the Integration of Search and Estimation," *Int. J. Computer Vision*, 8:2, pp. 113-122, 1992.

[9] C. P. Lu, G. D. Hager and E. Mjolsness, "Fast and Globally Convergent Pose Estimation from Video Images," *IEEE Trans. Pat. Ana. & Mach. Int., Vol. 22, No.6, 2000.*

[10] K. Miyamoto, "Fish Eyes Lens," *J. Opt. Soc. Am.,* Vol. 54, pp. 1061-1061, 1994.

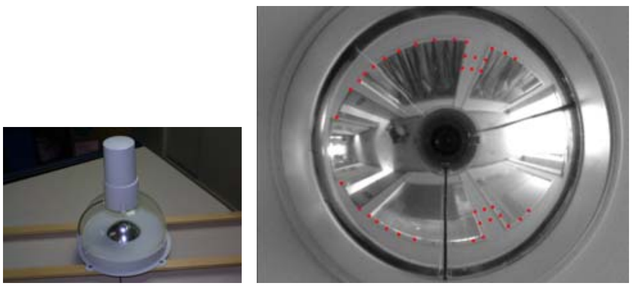
[11] P. Rademacher and G. Bishop, "Multiple-center-of-projection Images," *Proc. SIGGRAPH'98*, July 1998.

[12] R. Swaminathan, M. D. Grossberg and S. K. Nayar, "Caustics of Catadioptric Cameras," *Proc. IEEE Int. Conf. Computer Vision*, *July 2001*.

[13] P. Wunsch and G. Hirzinger, "Registration of CAD-Models to Images by Iterative Perspective Matching," *Proc. Int. Conf. Pattern Recognition*, pp. 78-83, 1996.

[14] Y. Yagi, S. Kaato, and T. Tsuji, "Collision Avoidance Using Omnidirectional Image Scnsor (COPIS)," *Proc. IEEE Int. Conf. Robotics & Automation*, pp. 910-915, Apr. 1991.

[15] J. S. C. Yuan, "A General Photogrammetric Method for Determining Object Position and Orientation," *IEEE Trans. Robotics and Automation*, Vol. 5, pp. 129-142, 1989.



(a) (b)

Figure 3. (a) The adopted omni-directional camera. (b) An image captured with the camera shown in (a), and the red points are the 3D points used for pose estimation.

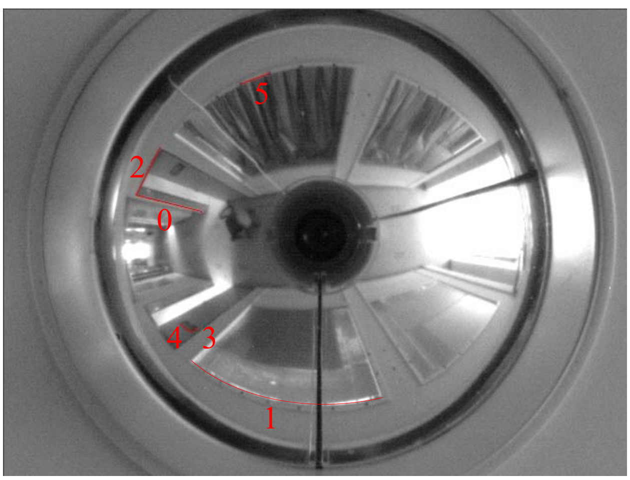


Figure 4. The lines used for the length-ratio results, where line 0 is the unit length.

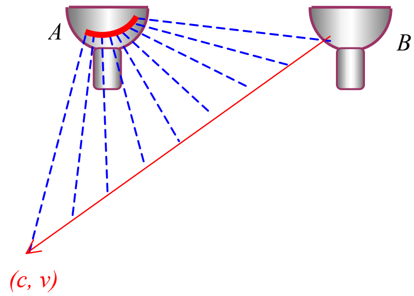
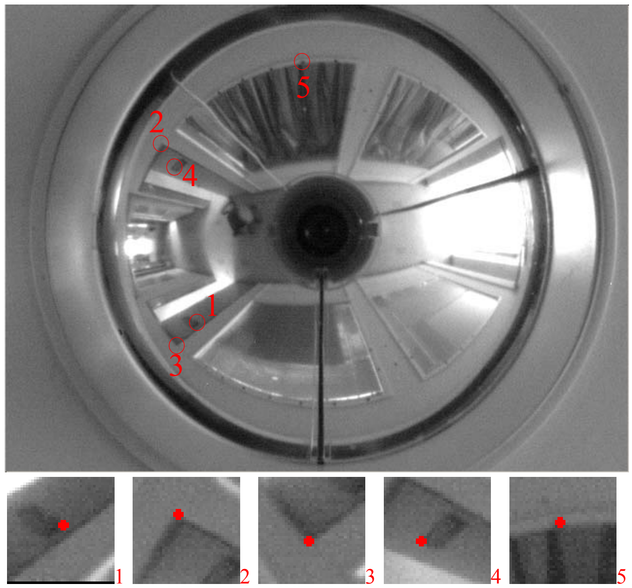
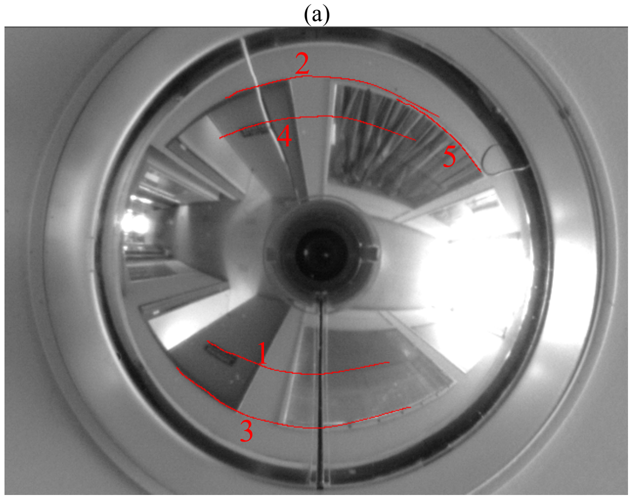


Figure 5. Illustration of the matching curve.



(a)



(b)

Figure 6. (a) Five points selected in one image. (b) The matching curves of these points in the other image.