Paper

A Pose Estimation Algorithm for Polyhedral Objects in X-Ray Image

Young Jun Roh* Non-member
Hyoungsuck Cho** Member

In an x-ray vision, identifying the pose, i.e. the position and orientation of objects from x-ray projection images is extremely important to monitor in real time and analyze mechanical parts which are invisible from outside. It is assumed here that the x-ray imaging conditions that include the relative coordinates of the x-ray source and the image plane are predetermined and the object geometry is known. In this situation, an x-ray image of an object at a given pose can be estimated computationally by using a priori known x-ray projection image model. It is based on the assumption that a pose of an object can be determined uniquely to a given x-ray projection image. Thus, once we have the numerical model of x-ray imaging process, x-ray image of the known object at any pose could be estimated. Then, among these estimated images, the best matched image could be searched and found. When adequate features in the images are available instead of the image itself, the problem becomes easier and simpler. In this work, we propose an efficient pose estimation algorithm for polyhedral objects whose image features consist of corner points and edge lines in their projection images. Based on the corner points and lines found in the images, the best-matched pose of a polyhedral object can be determined. To achieve this, we proposed an adequate and efficient image processing algorithm to extract the features of objects in x-ray images. The performance of the algorithm is discussed in detail including the limitations of the method. To evaluate the performance of the proposed method a series of simulation studies is carried out for various imaging conditions.

Keywords : Pose estimation, x-ray image, feature extraction.

1. Introduction

In machine vision area, position and oriantation of three dimensional (3D) objects have been rigorously applied in such industrial problems as process monitoring and control, assembly and PCB inspection. The techniques developed for pose estimation rely on the features identified on the surface of object[1] and the extreme contours of the object on the image[2].

In practice, there are many mechanical parts whose surface is specular, or sometimes they are occluded by the other parts and invisible from outside. In this case, camera vision system is not efficient and reliable to inspect or measure them. In these situations, x-ray system provides a good solution to overcome these limitations due to its penetrating characteristics in industrial field such as solder joint inspection system in PCB board and the battery of cellular phone inspection system. Although, it can be a good method to measure the shape or pose of objects ,until now, x-ray method has been mainly used for the inspection but not for the measurements in industrial field. For pose estimation of artificial knee implants in fluoroscopy images, Hoff and Komistek *et. al.* [3] propose a template matching technique for the 2D/2D matching. In this technique, the library of images is created in advance, which consists of views of the 3D models for the implant components rendered at different rotations. This technique needs a large size of library and may take a long matching time. Another approach using contours for the 3D/2D matching was proposed by Lavallee and Szeliski [2] to estimate the pose of free-form objects such as vertebra for spine surgery or skull for neurosurgery. Here, a free-form 3D object is modeled in advance by MRI method and a contour in x-ray image is matched to that of the 3D model.

In this paper, we have investigated a pose estimation of objects, with single x-ray image. For simplicity, only polyhedral objects are considered whose image features consist of corner points and edge lines in their projection images. Based on the corner points and lines found in the images, the best-matched pose of a polyhedral object is determined by a para-perspective model. To achieve this, we propose an adequate and efficient image processing algorithm to extract the features of objects in x-ray images. The performance of the algorithm is discussed in detail including the limitations of the method. To evaluate the performance of the proposed method a series of simulation studies are carried out and discussed.

2. X-Ray Imaging System

An x-ray imaging system for our investigation, is composed of the x-ray tube, x-ray image device and a rotating and tilting stage

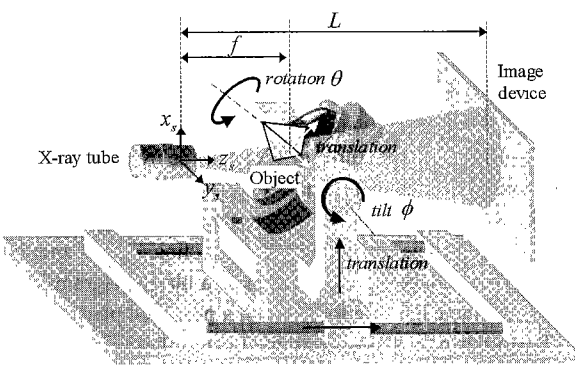


Fig. 1. An x-ray imaging system and its coordinates.

* Research Center, Mirae Co. Ltd.
#1, Osan-ri, Dongtan-myeon, Hwaseong-si, Gyeonggi-do, Korea 445-810
** Laboratory for Vision Systems and Machine Intelligence, KAIST
373-1, Kusong-dong, Yusong-gu, Daejon 305-701

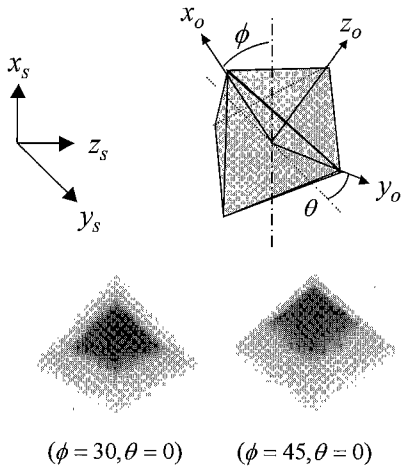


Fig. 2. X-ray images of a pyramid with different views.

to manipulate objects as shown in Figure 1. The stage has 6 degree of freedom with 4 translations (x, y, z, f), and tilting and rotating motion (ϕ, θ). In this configuration, the ratio between f and L becomes a magnification m onto the image plane.

Figure 2 shows the x-ray images of a pyramidal shape object obtained from different views which are achieved by manipulating the stage on which the object lies. Here, x, y, z and x_o, y_o, z_o indicate the world coordinates and object coordinates respectively.

3. X-Ray Image Processing Algorithm

Since the pose estimation in this work is based on the features such as edges and corner points in the images, it is important to extract exact features. There are a lot of image processing technique to extract image features in computer vision. However, most of them are suitable for conventional camera images and not applicable to x-ray images due to its inherent characteristics. This comes from the different imaging principles : images in camera vision are made by the reflected lights from the surface of an object. On the other hand, x-ray images are made by decayed ray through an object. In x-ray imaging method, x-ray intensity decays exponentially as it passes through an object. The amount of decay depends on initial intensity, penetrating length of x-ray and material properties such as density as represented in the following equation :

$$I = I_0 \cdot e^{-\mu x} \quad \cdots\cdots (1)$$

In this equation, I and I_o, respectively, are the intensity of x-ray before and after the penetration and μ is a decay constant of x-ray, and x is the penetrating length[4][5]. Thus, abrupt intensity change in an x-ray image by itself does not directly indicate edge information as in camera images. Also, x-ray images generally include a severe noise due to the scattered rays within the shield cabinet. In this section, we propose an efficient image processing method to extract the features of x-ray image by considering these characteristics. Since polyhedral objects are considered in this research, we will focus on extracting edges and corner points as image features. The procedure of extracting edges and corner points in x-ray images is represented as a block diagram in figure 3. Here, the line edges are found from a proposed plane fit method, then the corner points are determined using these lines, which are explained in the following sections.

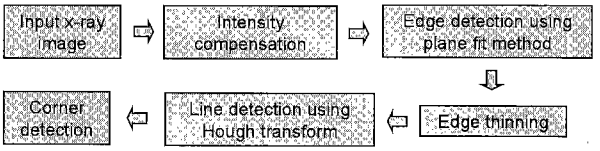
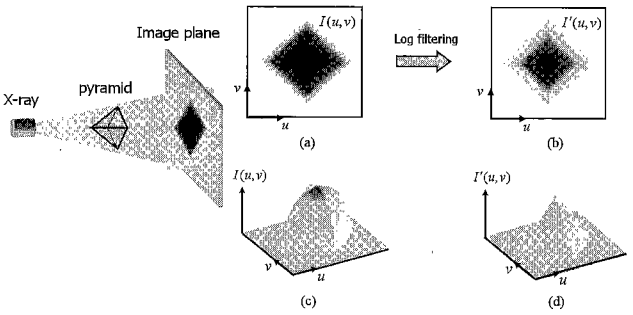


Fig. 3. The procedure of the proposed edge detection method.



(a) X-ray image of pyramidal shape, (b) Log filtered image,
(c) Topographic surface of (a), (d) Topographic surface of (b)

Fig. 4. Intensity compensation in x-ray image.

3.1 Edge Detection Algorithm As indicated in Eq(1), intensity in an x-ray image can be thought as a function of penetrating length of the ray when it is assumed that the object is made of one material. Figure 4 (a) is a simulated x-ray image for a polyhedron, a pyramidal shape object. Since the ray transmits the longest length within the object in the peak points, the intensity for the ray in x-ray image is the lowest with decayed x-ray. Due to the exponentially decaying x-ray intensity as the penetrating length, the x-ray image of the pyramid object has an intensity distribution, which is represented by a topographic surface in figure 4 (c). Here, we can see that the surface of the intensity plot is distorted exponentially compared to the object surface. To detect the edges of a polygon in an x-ray image, this distorted intensity distribution needs to be linearized in advance. This is achieved by a log transform with normalization. The transformation, which is a process of converting the exponentially decaying intensity function to a linearized one the following equations.

$$L(u,v) = \log[I(u,v)] \quad \cdots\cdots (2)$$

$$I'(u,v) = \frac{L(u,v) - \min[L(u,v)]}{\max[L(u,v)] - \min[L(u,v)]} \times (2^n - 1) \quad \cdots\cdots (3)$$

where $I(u,v)$ is the intensity of a pixel at (u,v) in the image and n is the bit resolution of the image sensor. Figure 4 (d) shows a typical topographic surface plot of the log filtered image, which is a plot of intensities with respect to the image coordinates (u,v). Since the edges of the object are represented by straight lines in this linearized topographic surface, it makes easier to detect edges from this surface rather than from the original one.

The edges in an x-ray image of a polyhedral object are determined from the topographic surface of the log filtered image. Since the topographic surface represents the x-ray penetration distances through an object, the topographic surface of polyhedron

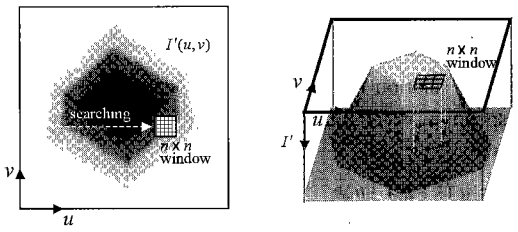


Fig. 5. Plane fit method on topographic surface.

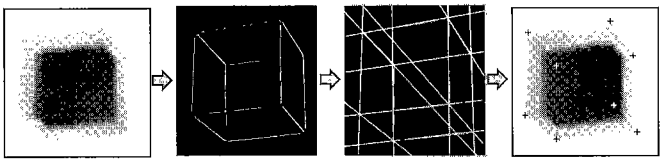


Fig. 6. Corner points detection procedure.

is composed of polygonal facets. This can be easily understood by intuition. And the edges of these polygonal facets of the topographic surface can be regarded as edges in the x-ray image. Based on that, we propose a plane fit method to determine the unclear edges in an x-ray image, actually by detecting the edges of the factes in the topographic surface. The algorithm scans whole of the topographic surface with a small mask (3x3 or 5x5 pixel area) as shown in Figure 5, and tries to find the best-fit palne within the mask area. When the mask is located on one of the planar facets, it is well fitted on the best-fit plane. On the other hand, the fitting result is not so good when the mask include edges where at least two planes meet together. Therefore, we can determine the edges where the plane fit result is bad.

The plane fit within a mask window is achieved by least square error method and based on the following plane equation with respect to image coordinates (u, v)

$$au + bv + c = I' \quad \cdots\cdots (4)$$

Here, a, b and c are the constants of the plane and I' is the height, intensity value, of a point (u,v) in the image. The square error e_p of the fit result is then defined as

$$e_p = \sum_{i=0}^{N-1} (I'_i - au_i - bv_i - c)^2 \quad \cdots\cdots (5)$$

where i is the index of a pixel within the mask and I'_i is the intensity of the point. From the least square expansion, the coefficients of the best fit plane for the mask are easily determined.

This plane fit error can be used as criterion of edge detection. Once we complete the plane fit and error evaluation over the whole image, we can get the error map image where the picks are located on the edges. Then we binarize the error map with the average value, which becomes the x-ray edge image of a polyhedral object. Since the edges obtained in this way have width of two or three pixels, we processed it with conventional edge-thining algorithm to reduce the data.

3.2 Corner Points Detection Algorithm The corner points can be the important features in an x-ray image of polyhedron objects. In this section, corner points detection algorithm in an edge image is proposed. Once an edge image of the given x-ray image is obtained from the proposed edge detection method, its straight lines can be extracted through Hough transform method. The corner points are then determined at the points where three or more lines intersect. Figure 6 shows an illustration of extracting corner points for a cube.

Table 1 represents the errors of corner points in case of a cube for various values of the pose parameters, f and θ. From these simulations, the average error of the detected corner points is about 0.8 pixel, sub-pixel error, in a 256 by 256 image.

Table 1. The error of corner points detected for of a cube.

θ(°) \ f(mm)	35	40	45	50	60	average
15	0.75	0 76	0.75	0 77	0.90	0 78
16	0.96	0 84	0 74	0 71	0.93	0 84
17	0.55	0.67	0.75	0 84	0.89	0 74
18	0.76	0 95	0.67	0.80	1.07	0.85
19	0.93	0.79	0.71	0 77	1.12	0.86
20	0.67	0 58	0 79	0 71	0.72	0.70
21	0 71	0 72	0.82	0 71	0.84	0.76
22	0.89	0.67	0 58	0.86	0.79	0.76
23	0.86	0.96	0.58	0.87	0.75	0.80
24	0 74	0.77	0 73	0.70	0.82	0 75
25	0.77	0.95	0.89	0.49	0.66	0 75
average	0.78	0 79	0 73	0.75	0.86	0.78

4. Pose Estimation

In this section, we will present a method of estimating pose of polyhedral object from an x-ray image using the image feature information, which are the corner points and edge lines. It has been known that the pose of a three dimensional object is not determined uniquely by an analytical solution from just one image. Rather, the best matched pose for the given image can be solved iteratively based on the perspective imaging model.

4.1 Perspective Projection Model Figure 7 illustrates a perspective projection model of an x-ray imaging system. In this work, a normalized virtual plane of which the distance from x-ray source is unit length, is considered as an imaging plane as shown in the figure.

The objective is here to determine the relationships between the object coordinates and x-ray source coordinates, which is expressed by a homogeneous coordinate transformation as :

$$\begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad \cdots\cdots (6)$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ and $\mathbf{T} \in \mathbb{R}^{3\times1}$ are the rotation and translation matrices, respectively.

In this configuration, a point P_i and a line l_j on the object coordinates are projected onto a point Q_i and a line L_j respectively in the normalized image plane. The point Q_i in the imaging coordinates (u,v) is given by

$$u_i = \frac{\vec{i}\cdot\vec{P}_i + t_x}{\vec{k}\cdot\vec{P}_i + t_z} = \frac{\vec{I}\cdot\vec{P}_i + u_0}{\varepsilon_i + 1}$$
$$v_i = \frac{\vec{j}\cdot\vec{P}_i + t_y}{\vec{k}\cdot\vec{P}_i + t_z} = \frac{\vec{J}\cdot\vec{P}_i + v_0}{\varepsilon_i + 1} \quad \cdots\cdots (7)$$

$$\vec{I} = \vec{i}/t_z,\ \vec{J} = \vec{j}/t_z,\ \varepsilon_i = \vec{k}\cdot\vec{P}_i/t_z \quad \cdots\cdots (8)$$

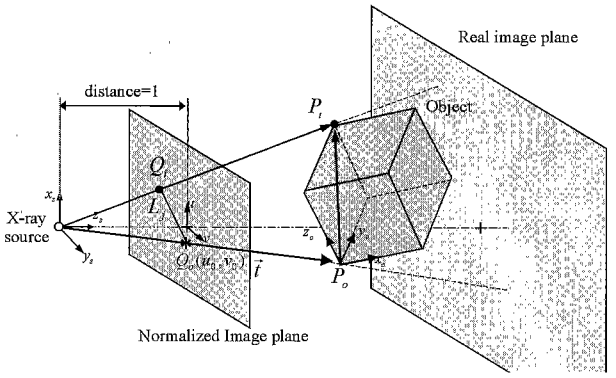


Fig. 7. The perspective projection model.

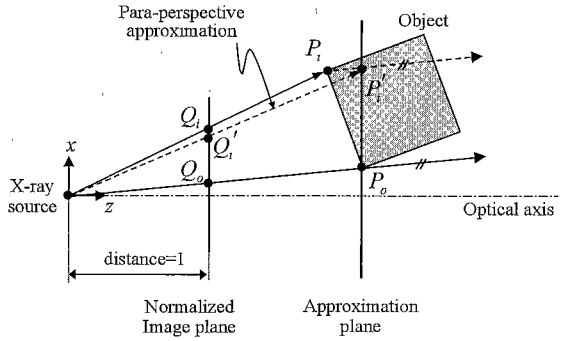


Fig. 8. Para-perspective projection.

where $\vec{P}_i$ is a position vector of a point P_i, $\vec{i},\vec{j},\vec{k}$ are the unit vectors of x,y,z axes, respectively for the object coordinates and $(u_0, v_0) = (t_x/t_z, t_y/t_z)$.
And a line l_j on the object coordinates, is also imaged as a line L_j in virtual plane,

$$l_j : \vec{P_j} = \overrightarrow{W_j} + \lambda \cdot \overrightarrow{V_j} \quad \lambda \in \mathbb{R}^1 \qquad \cdots\cdots (9)$$

where $\overrightarrow{V_j} \in \mathbb{R}^{3\times1}$ is a directional vector of the line l_j and $\overrightarrow{W_j} \in \mathbb{R}^{3\times1}$ is a point on the line

$$L_j : a_j(\vec{I}\cdot\vec{P_i} + u_0) + b_j(\vec{J}\cdot\vec{P_i} + v_0) + c_j(1+\varepsilon_i) = 0 .$$

4.2 Para-Perspective Projection Model Pose estimation is performed by using a para-perspective iterative method that was introduced by Dornaika and Garcia [8]. Para-perspective approximation is explained in 2D space for convenience illustrated in Figure 8.

Here, it is approximated that an object point P_i is imaged at a point Q_i' rather than its real projection point Q_i on the normalized image plane. As shown in the figure, the projection follows a parallel ray approximation, where all the points on the object coordinates lie in the approximated plane via rays parallel to the ray passing through the origin of the object coordinates. This para-perspective model is mathematically achieved by a first-order approximation of $1/(1+\varepsilon_i)$ in the perspective model:

$$\frac{1}{1+\varepsilon_i} \approx 1-\varepsilon_i \quad , i \in \{1...n\} . \qquad \cdots\cdots (10)$$

By applying this approximation to equations (7), we obtain a para-perspective approximated projection $Q_i^p(u_i^p, v_i^p)$:

$$\begin{aligned} u_i^p &= (\vec{I}\cdot\vec{P_i} + u_0)(1-\varepsilon_i) \approx \vec{I}\cdot\vec{P_i} + u_0 - u_0\varepsilon_i \\ v_i^p &= (\vec{J}\cdot\vec{P_i} + v_0)(1-\varepsilon_i) \approx \vec{J}\cdot\vec{P_i} + v_0 - v_0\varepsilon_i . \end{aligned} \qquad \cdots\cdots (11)$$

Then, the relation between the para-perspective (x_i^p, y_i^p) and the perspective projection (x, y) of P_i in the normalized plane is written by :

$$\begin{aligned} u_i^p &= u_i(1+\varepsilon_i) - u_0\varepsilon_i \\ v_i^p &= v_i(1+\varepsilon_i) - v_0\varepsilon_i . \end{aligned} \qquad \cdots\cdots (12)$$

The para-perspective equation (12) is rewritten if we inclnde the parameters of the object pose :

$$\begin{aligned} u_i^p - u_0 &= \frac{\vec{i} - u_0\vec{k}}{t_z}\vec{P_i} \\ v_i^p - v_0 &= \frac{\vec{j} - v_0\vec{k}}{t_z}\vec{P_i} . \end{aligned} \qquad \cdots\cdots (13)$$

By substituting equations (7) and (11) in equations (13), we obtain

$$\begin{aligned} (u_i - u_0)(1+\varepsilon_i) &= \vec{P_i}\cdot\vec{I_p} \\ (v_i - v_0)(1+\varepsilon_i) &= \vec{P_i}\cdot\vec{J_p} \end{aligned} \qquad \cdots\cdots (14)$$

where $\vec{I_p} = \frac{\vec{i} - u_0\vec{k}}{t_z}$ and $\vec{J_p} = \frac{\vec{j} - v_0\vec{k}}{t_z}$.

In this manner, a line on the normalized plane L_j in equation(9) can also be rewritten as

$$\begin{aligned} a_j\overrightarrow{W_j}\cdot\vec{I_p} + b_j\overrightarrow{W_j}\cdot\vec{J_p} + (a_ju_0 + b_jv_0 + c_j)(1+\eta_j) &= 0 \\ a_j\overrightarrow{V_j}\cdot\vec{I_p} + b_j\overrightarrow{V_j}\cdot\vec{J_p} + (a_ju_0 + b_jv_0 + c_j)\xi_j &= 0 \end{aligned} \qquad \cdots\cdots (15)$$

where $\eta_j = \vec{k}\cdot\overrightarrow{W_j}/t_z$, $\xi_j = \vec{k}\cdot\overrightarrow{V_j}/t_z$.

Therefore, two constraint equations to determine the pose transformation vectors $\vec{I}_p, \vec{J}_p$ are provided by equation (14) on each point feature, and another two constraint equations can be prepared in equation (15) from a line feature in an image if it is available. In the case of a polyhedral object, a number of corner points and lines are available as the features in determining the vectors $\vec{I}_p, \vec{J}_p$. When n corner points and m lines are detected in the image, $2(n+m)$ equations are available and is written as a matrix form :

$$\mathbf{G}\cdot\begin{bmatrix}\vec{I}_p \\ \vec{J}_p\end{bmatrix} = \mathbf{C} \quad ; \mathbf{G} \in \mathbb{R}^{2(m+n)\times6}, \ \mathbf{C} \in \mathbb{R}^{2(m+n)\times1} \qquad \cdots\cdots (16)$$

where the matrices $\mathbf{G}$ and $\mathbf{C}$ are expressed as equation (17) using the image features :

$$\mathbf{G} = \begin{bmatrix} P_1^T & 0^T \\ 0^T & P_1^T \\ \vdots & \vdots \\ P_m^T & 0^T \\ 0^T & P_m^T \\ a_1W_1^T & b_1W_1^T \\ a_1V_1^T & b_1V_1^T \\ \vdots & \vdots \\ a_nW_n^T & b_nW_n^T \\ a_nV_n^T & b_nV_n^T \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} (u_1 - u_0)(1+\varepsilon_1) \\ (v_1 - v_0)(1+\varepsilon_1) \\ \vdots \\ (u_m - u_0)(1+\varepsilon_m) \\ (v_m - v_0)(1+\varepsilon_m) \\ -(a_1u_0 + b_1v_0 + c_1)(1+\eta_1) \\ -(a_1u_0 + b_1v_0 + c_1)\xi_1 \\ \vdots \\ -(a_nu_0 + b_nv_0 + c_n)(1+\eta_n) \\ -(a_nu_0 + b_nv_0 + c_n)\xi_n \end{bmatrix} . \qquad \cdots\cdots (17)$$

Finally, the pose of an object is estimated by solving the equation (16) and determining the transformation vector $[\vec{I}_p, \vec{J}_p]^T$. The estimation is performed iteratively, since the equation (16) comes from a simplified para-perspective model and the equation itself includes estimation error.

The procedure of the iteration is summarized as :

Step 1. Para-perspective estimation :

Initialize para-perspective parameters

$\varepsilon_i = 0$, $\eta_j = 0$, $\xi_j = 0$; $i = 1,2,...,n$ and $j = 1,2,...,m$.

Step 2. Calculate $[\vec{I}_p \ \vec{J}_p]^T = (\mathbf{GG})^{-1}\mathbf{G}^T\mathbf{C}$.

Step 3. Calculate pose parameters $\mathbf{R},\mathbf{T}$ using $[\vec{I}_p, \vec{J}_p]^T$

$$\mathbf{T} = [t_x, t_y, t_z]^T$$

$$; \quad t_z = \frac{1}{2}\left(\frac{\sqrt{1+u_0^2}}{\|\vec{I}_p\|} + \frac{\sqrt{1+v_0^2}}{\|\vec{J}_p\|}\right), \quad t_x = u_0 t_z \quad \text{and} \quad t_y = v_0 t_z$$

$$\mathbf{R} = [\vec{i} \quad \vec{j} \quad \vec{k}]^T$$

$$; \quad \vec{k} = t_z^2 \cdot \vec{I}_p \times \vec{J}_p + t_z \cdot (v_0 \vec{I}_p \times \vec{k} - u_0 \vec{J}_p \times \vec{k})$$

$$\vec{i} = t_z \vec{I}_p + u_0 \vec{k} \ , \quad \vec{j} = t_z \vec{J}_p + v_0 \vec{k} \ .$$

Step 4. Update the para-perspective parameters

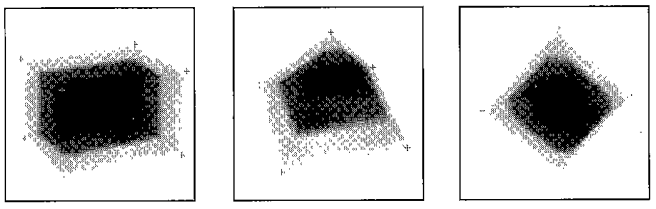
$$\varepsilon_i = \frac{\vec{k} \cdot \vec{P}_i}{t_z}, \quad \eta_j = \frac{\vec{k} \cdot \vec{W}_j}{t_z}, \quad \xi_j = \frac{\vec{k} \cdot \vec{V}_j}{t_z} \ .$$

Step 5. Go to step 2 and continue the procedure until there is no meaningful change in the para-perspective parameters.

5. Simulation Results

A series of simulations was performed for three different sample objects, a cube, a pyramid and an octahedral object of which x-ray images are shown in Figure 9.

Table 2 represents the simulation results for these three objects, where the line features are mainly used in the estimation and corner points are partially used. The simulations are conducted on 55 different poses for each object, and the averaged errors of the results are listed in the table. From the results, it can be pointed out that as the number of features increases in the estimation, the estimation error is decreased. In most cases, the estimation is finished within 3 iterations. However, there is an important fact that the number of iteration for convergence is related to the distance between an object and x-ray source due to the para-perspective approximation error. Since the para-perspective model is a parallel ray approximation, thus has a large approximation error when an object is located close to the x-ray source. Figure 10 shows the trend on how the number of iterations is reduced according to the distance between an object and x-ray source. For convenience, the distance is normalized with the representative size of an object. As shown in this figure, the iteration number for a successful estimation is reduced from 19 to 3 as the normalized distance varies from 0.8 to 2. When the normalized distance is less than 0.8 no convergence has been achieved.



(a) cube (b) pyramid (c) octahedron

Fig. 9. The simulation images of polyhedral objects.

Table 2. The error of pose estimation.

(a) result of a cube

	paraperspective iteration		
No. of correspondences	Rot. Error (deg.)	Trans. Error (%)	No. of iterations
12 lines + 1 point	0.33	0.45	2
12 lines + 2 points	0.31	0.32	2
12 lines + 3 points	0.31	0.28	3
12 lines + 4 points	0.28	0.25	2
12 lines + 5 points	0.25	0.20	2
12 lines + 6 points	0.23	0.20	2
12 lines + 7 points	0.21	0.20	2
12 lines + 8 points	0.21	0.19	2
average	0.27	0.26	2

(b) result of a pyramid

	paraperspective iteration		
No. of correspondences	Rot. Error (deg.)	Trans. Error (%)	No. of iterations
8 lines + 1 point	0.48	0.29	3
8 lines + 2 points	0.52	0.45	2
8 lines + 3 points	0.53	0.38	2
8 lines + 4 points	0.51	0.29	2
8 lines + 5 points	0.43	0.26	2
average	0.50	0.33	2

(c) result of an octahedral object

	paraperspective iteration		
No. of correspondences	Rot. Error (deg.)	Trans. Error (%)	No. of iterations
12 lines + 1 point	0.54	0.49	2
12 lines + 2 points	0.52	0.50	2
12 lines + 3 points	0.51	0.41	2
12 lines + 4 points	0.48	0.41	2
12 lines + 5 points	0.44	0.43	2
12 lines + 6 points	0.43	0.40	2
average	0.49	0.44	2

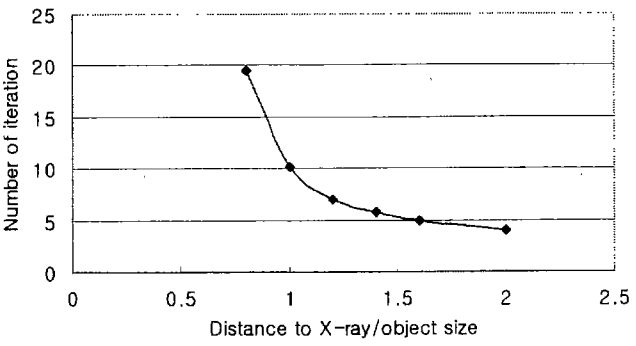


Fig. 10. Iteration numbers v.s. normalized distance.

6. Conclusions

In this paper, an estimation of the pose of a polyhedral object using an x-ray image has been studied. The method is based on the image features of a polyhedral object such as corner points and line edges. To detect features efficiently in x-ray images we proposed a series of image processing methods, which includes an edge detection algorithm named as a plane fit method. From the simulations, the corner points are detected successfully with the averaged error about 0.8 pixel.

From the features of lines and corner points detected in x-ray image, the object pose was estimated iteratively using a para-perspective approximation method. The estimations are performed on images of various poses. The error is found to be less than 0.5(degree) and 0.44(%) in rotation and translation, respectively. The iteration for the successful estimation was related to the distance between an object and x-ray source due to the para-perspective approximation error. Once the normalized distance is beyond 0.8, in our simulations, the iterative estimation converged successfully with sufficiently small errors. In this research, only polyhedral objects were considered for the convenience of defining image features. However, it needs to be expanded to arbitrary shapes including curved surfaces such as a cylinder, a cone and so on.

(Manuscript received Jur. 28, 2003, rivised Nov. 7, 2003)

References

(1) S. Lavallee and R. Szeliske : "Recovering the Position and Orientation of Free-Form Objects from Image Contours Using 3D Distance Maps", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 17, No 4, pp. 378-390 (1995)

(2) D. G. Lowe : "Fitting parameterized three-dimensional models to images", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 13, No. 5, pp. 441-450 (1996)

(3) W. A. Hoff, R. D. Komistek, D. A. Dennis, S. Walker, E. Northcut, and K. Spargo : "Pose Estimation of Artificial Knee Implants in Fluoroscopy Images Using A Template Matching Technique", *IEEE*, pp181-186 (1996)

(4) Editors : "Computed tomography details casting defects", Advanced Materials & Process, pp. 54-66 (1990-11)

(5) Y. Ohta, K. Maenobu, and T. Sakai : "Obtaining surface orientation from texels under perspective projection", *Proc. of the 7th IJCAI* (1981)

(6) R. Jain, R. Kasturi, and B. G. Schunck : "Machine Vision", McGraw-Hill, pp. 140-157 (1995)

(7) D. H. Ballard and C. M. Brown : "Computer Vision", Prentice-Hall, pp. 83-84 (1982)

(8) F. Dornaika and C. Garcia : "Pose Estimation using Point and Line Correspondences", Real-Time Images 5, pp. 215-230 (1999)

(9) Y. K. Ryu and H. S. Cho : "Visual inspection scheme for use in optical solder joint inspection system", IEEE ICRA '96, Mennesota, USA (1996)

(10) Y. J. Roh : "The Analysis and Design of x-ray Cross Sectional Imaging System Using Digital Tomosynthesis", M.S. Thesis, KAIST (1997)

(11) T. S. Kang, K. W. Ko, W. S. Park, Y. J. Roh, and H. S. Cho : "PCB joint inspection using x-ray", tech. report LCA-9802-V-3/8, LCA, KAIST (1998)

(12) T. S. Kang, K. W. Ko, W. S. Park, Y. J. Roh, and H. S. Cho : "PCB joint inspection using x-ray", tech. report LCA-9612-V-2/5, LCA, KAIST (1996)

Young Jun Roh (Non-member) was born in Busan, Korea, on June 18, 1971. He received a Ph.D. degree from Mechanical Engineering at Korea Advanced Institute of Science and Technology. He received his M.S. degree in the same institute and B.S. in Mechanical Engineering at YonSei University, Seoul. His research interests include automated visual inspection and 3D reconstruction and measurements.

Hyungsuck Cho (Member) was received his B.S. degree from Seoul National University, Korea in 1971, his M.S. degree from North-western University in 1973, Evanston, IL, USA. and his Ph.D. degree from the University of California at Berkeley, CA,USA. in 1977. Since 1978 he has been a professor of Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST). His research interests are focused on environment perception and recognition for mobile robots, machine vision, pattern classification, and and machine intelligence. Now, he serves on editorial boards of international journals; IEEE Industrial Electronics, Journal of Robotic Systems, Robotica, Control Engineering Practice (IFAC) and etc.