A Robust Method for 3D Model Based Object Tracking

Masanobu Yamamoto
Department of Information Engineering, Niigata University
8050 Ikarashi 2-nocho, Niigata-city,950-21, Japan
e-mail: yamamoto@info.eng.niigata-u.ac.jp

Abstract

This paper describes a method to estimate pose and 3D motion parameters of an object from an image sequence under assumptions of monocular and perspective view as well as given 3D model of the object. We derive nonlinear equations which can simultaneously and directly estimate pose and 3D motion parameters of the object. Sufficient conditions are also given to obtain a unique solution to the nonlinear equations. The Newton method iteratively gives the solution to the system of nonlinear equations. Our method requires neither model to image matching nor image to image matching.

1 Introduction

Temporal-spatial gradient scheme enables us to recover 3D structure and motion of an object from an image sequence without correspondence between images. However, when the set of unknowns includes the depth information, it becomes difficult to obtain a unique solution since the number of unknowns usually exceeds the number of equations. The various constrains must be introduced to cope with this problem. When the surface of the object is planar [7][5] or quadric [6], when the depth is known [8][9][1], or when the motion is pure rotation or pure translation [1], the structure and/or motion can be recovered.

This paper concerns the following problem: The object has any surface structure, i.e. the scene is not limited to specific structures such as a planer or quadric world. The surface structure is known by a 3D model. However, the 3D model is deviated from the correct pose. Therefore, the model cannot give a correct depth information. This paper describes a direct method to correct the pose displacement of the model, and simultaneously estimate the 3D motion parameters from the image sequence.

The method is applied to a problem of tracking object based on 3D model. Our method requires neither model to image matching nor image to image matching, while the Lowe's method [3][4] requires model to image matching.

The next Section summaries the method of direct estimation of 3D motion parameters from an image sequence using a correct depth information. The 3rd Section presents an extended method which can revise the pose of the 3D model and simultaneously estimate 3D motion parameters. In the Section 4, we discuss the conditions to have an unique solution. The Section 5 is devoted to an application of our method to object tracking problem.

2 Direct Estimation of 3D motion

Let an object be rigid. We suppose that the pose and 3D shape of the object are given by a 3D model. Using a direct method [8][9][1], the 3D motion parameters of the object is obtained as follows.

Let a coordinate system of the scene be a Cartesian (x, y, z) one. Now, the displacement vector $\delta = (\delta x, \delta y, \delta z)^T$ of the point $\mathbf{p} = (x, y, z)^T$ on a rigid object can be presented as eq.(1) with the angular velocity vector $\mathbf{R} = (R_x, R_y, R_z)^T$ around the center of the object, $\mathbf{p}_c = (x_c, y_c, z_c)^T$, and the translation vector $\mathbf{T} = (T_x, T_y, T_z)^T$.

$$\delta = \mathbf{R} \times (\mathbf{p} - \mathbf{p}_c) + \mathbf{T} \tag{1}$$

We can observe the motion of object of which projection lies on a image plane, $z = 1$, with the origin of the coordinate as the projection center. An object point $\mathbf{p}$ is projected on the image point (X, Y).

$$\begin{cases} X = x/z \\ Y = y/z \end{cases} \tag{2}$$

By the total derivative of eq.(2), the image displacement vector $\mathbf{\Delta} = (\Delta X, \Delta Y)^T$ at the image point (X, Y) is related with the object displacement vector δ.

$$\mathbf{\Delta} = \frac{1}{z} A \delta \tag{3}$$

where

$$A = \begin{pmatrix} 1 & 0 & -X \\ 0 & 1 & -Y \end{pmatrix}$$

Substituting eq.(1) into eq.(3), the image displacement vector can be represented by the 3D motion parameters.

$$\mathbf{\Delta} = \frac{1}{z} A\mathbf{T} + (B - \frac{1}{z} B_c)\mathbf{R} \tag{4}$$

where

$$B = \begin{pmatrix} -XY & 1 + X^2 & -Y \\ -(1 + Y^2) & XY & X \end{pmatrix}$$

$$B_c = \begin{pmatrix} -y_c X & z_c + x_c X & -y_c \\ -(z_c + y_c Y) & x_c Y & x_c \end{pmatrix}$$

Let the intensity of the image at point (X, Y) at time t be $E(X, Y, t)$. The intensity is assumed remain constant so that

$$E(X + \Delta X, Y + \Delta Y, t + \Delta t) - E(X, Y, t) = 0. \tag{5}$$

Substituting $(\Delta X, \Delta Y)$ of eq.(4) into eq.(5), we can derive a nonlinear equation with motion parameters $\mathbf{T}, \mathbf{R}$ as unknowns. Solving the system of the nonlinear equations, we can get 3D motion parameters. However, it is difficult to find the solution to satisfy the system of equations because of noise. Instead minimizing the L_2 norm measure of the left hand side of eq.(5),

$$\sum_{X,Y} (E(X + \Delta X, Y + \Delta Y, t + \Delta t) - E(X, Y, t))^2 \tag{6}$$

we can get a least-square solution. Starting from an appropriate initial guess, the Newton method can give an optimal solution.

The initial guess is set as $\mathbf{T}^{(0)}, \mathbf{R}^{(0)}$. Assuming $\mathbf{T}$ and $\mathbf{R}$ are very small, $\mathbf{T}^{(0)} = \mathbf{0}, \mathbf{R}^{(0)} = \mathbf{0}$. The optimal solution is given by the iterative form

$$\mathbf{T}^{(n+1)} = \mathbf{T}^{(n)} + \Delta\mathbf{T} \tag{7}$$

$$\mathbf{R}^{(n+1)} = \mathbf{R}^{(n)} + \Delta\mathbf{R}. \tag{8}$$

The small corrections to $\mathbf{T}$ and $\mathbf{R}$

$$\Delta\mathbf{T} = (\Delta T_x, \Delta T_y, \Delta T_z)^T$$
$$\Delta\mathbf{R} = (\Delta R_x, \Delta R_y, \Delta R_z)^T$$

are obtained from the following system of linear equations.

$$\mathbf{J}(\mathbf{T}^{(n)}, \mathbf{R}^{(n)}) \begin{pmatrix} \Delta\mathbf{T} \\ \Delta\mathbf{R} \end{pmatrix} = \mathbf{e}^{(n)} \tag{9}$$

where $\mathbf{J}(\mathbf{T}, \mathbf{R})$ is a Jacobian matrix,

$$\mathbf{J}(\mathbf{T}, \mathbf{R}) = \begin{pmatrix} \vdots \\ \mathbf{J}_{X,Y}(\mathbf{T}, \mathbf{R}) \\ \vdots \end{pmatrix}$$

$$\mathbf{J}_{X,Y}(\mathbf{T}, \mathbf{R}) = \frac{\partial E(X + \Delta X, Y + \Delta Y, t + \Delta t)}{\partial(\mathbf{T}, \mathbf{R})}$$

$$\mathbf{e}^{(n)} = \begin{pmatrix} \vdots \\ e^{(n)}_{X,Y} \\ \vdots \end{pmatrix}$$

$$\begin{aligned} e^{(n)}_{X,Y} &= E(X, Y, t) - E(X + \Delta X^{(n)}, \\ &\quad Y + \Delta Y^{(n)}, t + \Delta t) \\ \mathbf{\Delta}^{(n)} &= (\Delta X^{(n)}, \Delta Y^{(n)})^T \\ &= \frac{1}{z} A\mathbf{T}^{(n)} + (B - \frac{1}{z} B_c)\mathbf{R}^{(n)} \end{aligned}$$

The $E(X+\Delta X, Y+\Delta Y, t+\Delta t) - E(X, Y, t)$ is a function of $\mathbf{\Delta}$, and the $\mathbf{\Delta}$ is a linear function of $\mathbf{T}$ and $\mathbf{R}$. Therefore, precomputing the Jacobian needs a differentiation of the combinatorial function. The linear equation of the system eq.(9) is written in details as follows:

$$\frac{1}{z}\mathbf{a} \cdot \Delta\mathbf{T} + (\mathbf{b} - \frac{1}{z}\mathbf{b}_c) \cdot \Delta\mathbf{R} = e^{(n)}_{X,Y} \tag{10}$$

where

$$\mathbf{a} = \begin{pmatrix} E_X \\ E_Y \\ -XE_X - YE_Y \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} -Y(XE_X + YE_Y) - E_Y \\ X(XE_X + YE_Y) + E_X \\ XE_Y - YE_X \end{pmatrix}$$

$$\mathbf{b}_c = \begin{pmatrix} -y_c(XE_X + YE_Y) - z_c E_Y \\ x_c(XE_X + YE_Y) + z_c E_X \\ x_c E_Y - y_c E_X \end{pmatrix}$$

$$(E_X, E_Y) = (\frac{\partial E}{\partial X}, \frac{\partial E}{\partial Y})$$

The system of linear equations (9) has a unique solution if and only if the coefficient matrix $\mathbf{J}(\mathbf{T}, \mathbf{R})$ is non-singular.

Consider the case in which the 3D model represents 3D shape of the object but does not have a correct pose. If so, the 3D information z is not correct, and the 3D motion parameters estimated by the method are not also correct. This affects an accuracy of the object tracking. The 3D model-based object tracking is established by estimating 3D motion parameters and moving the 3D model. However, when a noise affects the pose of the 3D model, the displacement between the 3D model and the object gradually grows up. Therefore, the method needs a mechanism to correct the displacement.

3 Correcting pose displacements of 3D model

Pose displacements of the 3D model consist of orientation and position displacements. The orientation displacement of the 3D model can be corrected by the rotation around the axis passing through the center of the object (x_c, y_c, z_c). The position displacement of the 3D model can be corrected by the translation. Supposing the displacements to be very small, the pose corrections can be represented by the vectors $\mathbf{Q} = (Q_x, Q_y, Q_z)^T$ and $\mathbf{S} = (S_x, S_y, S_z)^T$,which denote rotation and translation, respectively.

The problem correcting pose displacements as well as estimating 3D motion parameters is stated as follows.

Given a sequence of images $E(X, Y, t)$ and 3D information of the object from incorrect 3D model, let estimate 3D motion parameters $\mathbf{T}, \mathbf{R}$ and pose corrections $\mathbf{S}, \mathbf{Q}$.

In order to solve the problem, let derive an equation including pose corrections $\mathbf{S}, \mathbf{Q}$ and 3D motion parameters as unknowns. We suppose that a depth z map is given by

$$z = f(x, y) \tag{11}$$

which is obtained from the incorrect model. When the corrections $\mathbf{S}, \mathbf{Q}$ are given, the correct depth map is obtained as follows. Correcting the displacements of orientation and position, a point $\mathbf{p} = (x, y, z)^T$ on the incorrect 3D model moves to $\mathbf{p}' = (x', y', z')^T$

$$\mathbf{p}' = \mathbf{p} + \mathbf{Q} \times (\mathbf{p} - \mathbf{p}_c) + \mathbf{S}. \tag{12}$$

Solving eq.(12) with respect to x, y, z, and neglecting the products of the components of $\mathbf{Q}, \mathbf{S}$ because these products can be regarded as very small second and more order terms, the $\mathbf{p}$ is approximated by

$$\mathbf{p} = \mathbf{p}' - \mathbf{Q} \times (\mathbf{p}' - \mathbf{p}_c) - \mathbf{S} \tag{13}$$

Substituting eq.(13) into eq.(11) and rewriting terms x', y', z' with x, y, z, respectively, we obtain

$$\begin{aligned} z + Q_y(x - x_c) - Q_x(y - y_c) - S_z = \\ F(x + Q_z(y - y_c) - Q_y(z - z_c) - S_x, \\ y - Q_z(x - x_c) + Q_x(z - z_c) - S_y). \end{aligned} \tag{14}$$

Solving eq.(14) with respect to z, the correct depth map

$$z = g(x, y, \mathbf{S}, \mathbf{Q}) \tag{15}$$

is obtained. If the pose displacements are not corrected,

$$g(x, y, 0, 0) = f(x, y).$$

The eq.(15) represents a depth map on the scene coordinates (x, y). A depth map on the image coordinates (X, Y) may be convenient rather than one on the scene coordinates, since we can observe the scene by the image coordinates. A domain transformation from the scene coordinates to the image ones is represented as follows. Substituting $x = Xz, y = Yz$ from the eq.(2) into the eq.(11) , and solving the result with respect to z gives a depth map on the image coordinates,

$$z = F(X, Y). \tag{16}$$

Similarly, substituting $x = Xz, y = Yz$ into eq.(14) gives

$$\begin{aligned} z + Q_y(Xz - x_c) - Q_x(Yz - y_c) - S_z = \\ f(Xz + Q_z(Yz - y_c) - Q_y(z - z_c) - S_x, \\ Yz - Q_z(Xz - x_c) + Q_x(z - z_c) - S_y). \end{aligned} \tag{17}$$

Solving the eq.(17) with respect to z gives

$$z = G(X, Y, \mathbf{S}, \mathbf{Q}) \tag{18}$$

If the pose displacements of the 3D model is not corrected,

$$G(X, Y, 0, 0) = F(X, Y) \tag{19}$$

It is, in general, difficult to obtain the transformation between the both coordinate systems by an analytical approach. In practice we use, however, a geometrical modeler [2], namely SOLVER, which can give

(1) depth maps, $g(x, y, \mathbf{S}, \mathbf{Q})$, $G(X, Y, \mathbf{S}, \mathbf{Q})$,

(2) images of the model of the objects, and

(3) orientation map (g_x, g_y) on the model surface.

The $(\Delta X, \Delta Y)$ of the eq.(5) given in the previous Section is represented by the eq.(4). The z in the eq.(4) is a function of $\mathbf{S}, \mathbf{Q}$ as shown by the eq.(18). Therefore, the constraint equations (5) derived from image points are a system of nonlinear equations with $\mathbf{T}, \mathbf{S}$ and $\mathbf{S}, \mathbf{Q}$ as unknowns.

$$E(X + \Delta X, Y + \Delta Y, t + \Delta t) - E(X, Y, t) = 0$$

where

$$\begin{aligned} \mathbf{\Delta} &= \frac{1}{z} A\mathbf{T} + (B - \frac{1}{z} B_c)\mathbf{R} \\ z &= G(X, Y, \mathbf{S}, \mathbf{Q}) \end{aligned}$$

The least-square solution to (6) is obtained from the Newton iteration. The initial guess of the motion parameters is set as the motion parameters calculated from the incorrect depth map $z = F(X, Y)$ by the method described in the previous Section. The initial guess of the pose corrections is set as $\mathbf{S}^{(0)} = \mathbf{0}$ and $\mathbf{Q}^{(0)} = \mathbf{0}$ since these displacements are regarded as very small.

The iteration is expressed by

$$\mathbf{T}^{(n+1)} = \mathbf{T}^{(n)} + \Delta\mathbf{T} \quad (20)$$

$$\mathbf{R}^{(n+1)} = \mathbf{R}^{(n)} + \Delta\mathbf{R} \quad (21)$$

$$\mathbf{S}^{(n+1)} = \mathbf{S}^{(n)} + \Delta\mathbf{S} \quad (22)$$

$$\mathbf{Q}^{(n+1)} = \mathbf{Q}^{(n)} + \Delta\mathbf{Q} \quad (23)$$

The set of $\Delta\mathbf{T}$, $\Delta\mathbf{R}$ and small corrections to $\mathbf{S}, \mathbf{Q}$

$$\Delta\mathbf{S} = (\Delta S_x, \Delta S_y, \Delta S_z)^T$$
$$\Delta\mathbf{Q} = (\Delta Q_x, \Delta Q_y, \Delta Q_z)^T$$

is obtained from solving the system of linear equations

$$\mathbf{J}(\mathbf{T}^{(n)}, \mathbf{R}^{(n)}, \mathbf{S}^{(n)}, \mathbf{Q}^{(n)}) \begin{pmatrix} \Delta\mathbf{T} \\ \Delta\mathbf{R} \\ \Delta\mathbf{S} \\ \Delta\mathbf{Q} \end{pmatrix} = \mathbf{e}^{(n)} \quad (24)$$

where $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$ is a Jacobian matrix

$$\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q}) = \begin{pmatrix} \vdots \\ \mathbf{J}_{X,Y}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q}) \\ \vdots \end{pmatrix}$$

$$\mathbf{J}_{X,Y}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q}) = \frac{\partial E(X + \Delta X, Y + \Delta Y, t + \Delta t)}{\partial(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})}.$$

$$\begin{aligned} \mathbf{\Delta}^{(n)} &= (\Delta X^{(n)}, \Delta Y^{(n)})^T \\ &= \frac{1}{z^{(n)}} A\mathbf{T}^{(n)} + (B - \frac{1}{z^{(n)}} B_c)\mathbf{R}^{(n)} \\ z^{(n)} &= G(X, Y, \mathbf{S}^{(n)}, \mathbf{Q}^{(n)}) \end{aligned}$$

Components of the Jacobian matrix, $\mathbf{J}_{X,Y}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$, include a partial derivative of z with respect to $\mathbf{S}, \mathbf{Q}$. The partial derivatives are derived from partially differentiating the eq.(17) with respect to $\mathbf{S}, \mathbf{Q}$. Therefore, the linear equation of the system (24) is described in detail

$$\frac{1}{z^{(n)}} \mathbf{a} \cdot \Delta\mathbf{T} + (\mathbf{b} - \frac{1}{z^{(n)}} \mathbf{b}_c) \cdot \Delta\mathbf{R} + \frac{\mathbf{a} \cdot \mathbf{T}^{(n)} - \mathbf{b}_c \cdot \mathbf{R}^{(n)}}{(z^{(n)})^2} (\mathbf{c} \cdot \Delta\mathbf{S} - (\mathbf{d} - \mathbf{d}_c) \cdot \Delta\mathbf{Q}) = e^{(n)}_{X,Y}$$

where (25)

$$\mathbf{a} = \begin{pmatrix} E_X \\ E_Y \\ -XE_X - YE_Y \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} -Y(XE_X + YE_Y) - E_Y \\ X(XE_X + YE_Y) + E_X \\ XE_Y - YE_X \end{pmatrix}$$

$$\mathbf{b}_c = \begin{pmatrix} -y_c(XE_X + YE_Y) - z_c E_Y \\ x_c(XE_X + YE_Y) + z_c E_X \\ x_c E_Y - y_c E_X \end{pmatrix}$$

$$\mathbf{c} = \frac{1}{h} \begin{pmatrix} f_x \\ f_y \\ -1 \end{pmatrix}$$

$$\mathbf{d} = \frac{z^{(n)}}{h} \begin{pmatrix} Y + f_y \\ -X - f_x \\ Yf_x - Xf_y \end{pmatrix}$$

$$\mathbf{d}_c = \frac{1}{h} \begin{pmatrix} y_c + z_c f_y \\ -x_c - z_c f_x \\ y_c f_x - x_c f_y \end{pmatrix}$$

$$\begin{aligned} h &= 1 - Xf_x - Yf_y - (Y + f_y)Q^{(n)}_x \\ &\quad + (X + f_x)Q^n_y + (Xf_y - Yf_x)Q^{(n)}_z \end{aligned}$$

The surface gradient (f_x, f_y) on the 3D model is calculated at

$$\begin{aligned} x &= Xz^{(n)} + Q^{(n)}_z(Yz^{(n)} - y_c) \\ &\quad - Q^{(n)}_y(z^{(n)} - z_c) - S^{(n)}_x \\ y &= Yz^{(n)} - Q^{(n)}_z(Xz^{(n)} - x_c) \\ &\quad + Q^{(n)}_x(z^{(n)} - z_c) - S^{(n)}_y \end{aligned}$$

Calibrating the pose of the 3D model by $\Delta\mathbf{S}, \Delta\mathbf{Q}$ at every iteration, the $\mathbf{c}, \mathbf{d}, \mathbf{d}_c, h$ among the coefficients of eq.(25) may be simplified. The $n+1$th iteration is calculated based on the revised 3D model after the nth iteration. The Jacobian matrix can be updated by setting $\mathbf{S}^{(n)} = \mathbf{0}, \mathbf{Q}^{(n)} = \mathbf{0}$ since the 3D model has been revised. The surface gradient (f_x, f_y) on the incorrect 3D model can be replaced by a surface gradient (g_x, g_y) on the revised 3D model. The $\mathbf{c}, \mathbf{d}, \mathbf{d}_c, h$ may be simplified as

$$\mathbf{c} = \frac{1}{h} \begin{pmatrix} g_x \\ g_y \\ -1 \end{pmatrix}$$

$$\mathbf{d} = \frac{z^{(n)}}{h}\begin{pmatrix} Y + g_y \\ -X - g_x \\ Yg_x - Xg_y \end{pmatrix}$$

$$\mathbf{d}_c = \frac{1}{h}\begin{pmatrix} y_c + z_c g_y \\ -x_c - z_c g_x \\ y_c g_x - x_c g_y \end{pmatrix}$$

$$h = 1 - Xg_x - Yg_y.$$

The system of linear equations (24) has a unique solution if and only if the coefficient matrix $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$ is non-singular.

$$[\mathbf{J}^T\mathbf{J}]^{-1}\mathbf{J}^T\mathbf{e}^{(n)}$$

The iterative procedure of eqs.(21)~(23) stops when the amount of (6) does not decrease. Then, the 3D model is posed in the correct orientation and position, the 3D motion parameters are obtained.

4 Uniqueness of Solution

The uniqueness of estimating the motion parameters and the pose corrections of the 3D model depends on the 3D shape, surface pattern and motion of the object. The parameters of motion and pose corrections can be uniquely determined if and only if the coefficient matrix $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$ of eq.(24) is non-singular. The following three cases do not have a unique solution.

Case 1. If $\mathbf{J}(\mathbf{T}, \mathbf{R})$ is singular, we can not get a unique solution. The $\mathbf{J}(\mathbf{T}, \mathbf{R})$ is a minor matrix of $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$. Therefore, if the $\mathbf{J}(\mathbf{T}, \mathbf{R})$ is singular, then the $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$ is also singular. The $\mathbf{J}(\mathbf{T}, \mathbf{R})$ is a coefficient matrix of eq.(9) which can estimate only 3D motion parameters as described in Section 2. Typical examples belonging the case are the Barber's pole and the bucket without surface pattern [9].

Case 2. If the $\mathbf{J}(\mathbf{S}, \mathbf{Q})$ is singular, we can not have a unique solution. The $\mathbf{J}(\mathbf{S}, \mathbf{Q})$ is a minor matrix of the $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$. Therefore, if the $\mathbf{J}(\mathbf{S}, \mathbf{Q})$ is singular, the $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$ is also singular. The $\mathbf{J}(\mathbf{S}, \mathbf{Q})$ is the same coefficient matrix as one of the estimation equation which determines the motion parameters from range image sequence [10]. The typical examples of the case are a plane, sphere, body of rotation and etc..

Case 3. When we interpret the motion of the object as a motion stereo of the camera, if the length of the baseline of the motion stereo is zero, then the $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$ is singular. Eq.(1) can be rewritten in

$$\delta = \mathbf{R} \times \mathbf{p} + \mathbf{t}$$

$$\mathbf{t} = \begin{pmatrix} T_x + y_c R_z - z_c R_y \\ T_y - x_c R_z + z_c R_x \\ T_z + x_c R_y - y_c R_x \end{pmatrix}$$

The $-\mathbf{t}$ is the base line of the camera motion stereo. The following relation is established.

$$\mathbf{a} \cdot \mathbf{t} = \mathbf{a} \cdot \mathbf{T} - \mathbf{b}_c \cdot \mathbf{R}$$

The right hand side of the equation, $\mathbf{a} \cdot \mathbf{T} - \mathbf{b}_c \cdot \mathbf{R}$, imposes on any coefficients of $\mathbf{\Delta S}$ and $\mathbf{\Delta Q}$ in the eq.(25) , i.e. any elements of the 7th to the 12th rows of $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$. Therefore, if the base line $-\mathbf{t}$ equals to $\mathbf{0}$, then $\mathbf{J}(\mathbf{T}, \mathbf{R}, \mathbf{S}, \mathbf{Q})$ is singular.

There is a case having no unique solution besides the above 3 cases. Let us consider the simplest case where all motion parameters except T_x are zero, also all pose parameters except S_x are zero. Let us assume that the 3D model consists of two planes, $z = f_{x_i}x + z_0, (i = 1, 2)$. The ΔT_x and ΔS_x are obtained from the system of linear equations

$$\begin{pmatrix} \frac{E_{X_1}}{z_1} & \frac{E_{X_1}T_x f_{x_1}}{z_1 z_0} \\ \frac{E_{X_2}}{z_2} & \frac{E_{X_2}T_x f_{x_2}}{z_2 z_0} \end{pmatrix} \begin{pmatrix} \Delta T_x \\ \Delta S_x \end{pmatrix} = \begin{pmatrix} e_{X_1} \\ e_{X_2} \end{pmatrix}. \tag{26}$$

The determinant of the coefficient matrix is

$$\frac{E_{X_1}E_{X_2}T_x(f_{x_2} - f_{x_1})}{z_0 z_1 z_2}.$$

When $E_{X_i} = 0$ from the case 1, $f_{x_i} = 0$ from the case 2, or $T_x = 0$ from the case 3, the determinant equals zero. In the case of $f_{x_1} = f_{x_2}$ except the 3 cases, the determinant also equals zero, therefore there is no unique solution.

5 Experiments

An experimental result is described, where an object can be tracked by estimating 3D motion parameters and correcting the pose displacements of the 3D model. The target for tracking in this experiment is a ball which is hanging from the ceiling using a string.

The pendulum, the ball with a string, swings in the plane perpendicular to the camera axis. The ball itself is rotating around the string. The total number of frames in the image sequence is 100. The first frame of the image sequence is depicted in Fig.1. This case corresponds to the **Case 2** in the

Figure 1: A ball as tracking a target at the 1st frame.

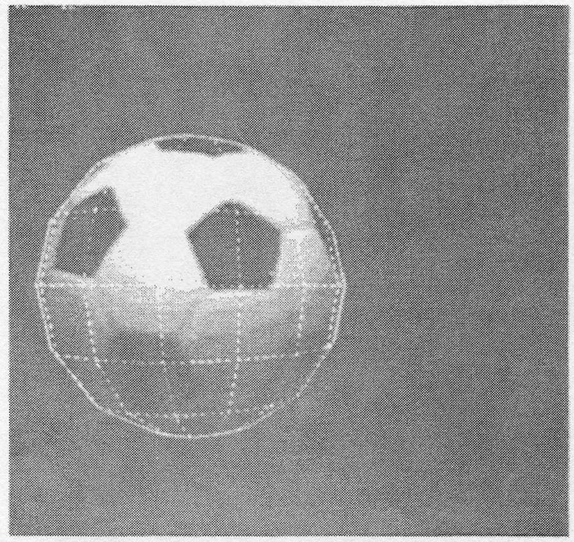

Figure 2: 3D model of the object at starting frame.

previous Section, since the ball is a sphere. Therefore, the orientation displacement of the model can not be uniquely corrected. We wish to correct only position displacement. However, the estimation of z component of the position displacement falls in ill-condition. Consequently, we correct only x and y components, i.e. S_x and S_y of the position displacement, and estimate 3D motion parameters, i.e. T_x, T_y, T_z, R_x, R_y and R_z.

At the beginning of tracking, the 3D model of the ball is manually overlapped on the image of the object. The manual adjustment involves the pose displacement of the model. The 3D model is depicted in the Fig.2. The tracking results without correcting pose displacements are shown Figs.3, 4. The Fig.3 shows the 3D model at every 10 frames overlapped on the first frame. The Fig.4 shows the 3D model and the object image at the 100th frame. It is clear to increase the displacement between the object and the 3D model.

Let us show the effect correcting the pose displacement. As soon as the tracking starts, the displacement is corrected. The Fig.5 shows the 3D model at every 10 frames overlapped on the first frame. Figs.6 and 7 show tracking results at the 64th and 100th frames, respectively. It seems to be good fitting between the model and object.

How much displacement of the position can our method correct ? We made an experiment of correcting the pose displacement and estimating motion parameters from the 1st and 2nd frames. Fig.8 shows the 3D model before correcting pose displacement. The model is the ball's radius away from the exact position. Fig.9 shows the 3D model at every iteration in the Newton method up to 7 iterations. Making the experiments under various positional displacements, it becomes clear to be able to perfectly correct the displacement within a half of radius of the ball.

6 Concluding

We presents a method which can directly correct the pose displacement of the model and estimate 3D motion parameters assuming that 3D shape of the object is given as a 3D model. Using this model-based method, it is possible to track the object without matching model to image and correspondence between successive images.

More experiments need to be done in various objects. The method may converge a false solution among mutiple ones to the system of nonlinear equations, also may not stably get a unique solution in case of ill-condition. The next study will discuss conditions, to get a correct solution, relating to the structure and surface pattern of the object, charrcteristics of camera, and the relation between camera and object.

References

[1] B.K.P.Horn and E.J.Weldon Jr., Direct methods for recovering motion. *International Journal of Computer Vision*, vol.2, pp.51-76, 1988.

[2] K.Koshikawa and Y.Shirai, A 3-D modeler for vision research, *ICAR*, pp.185-190, 1985.

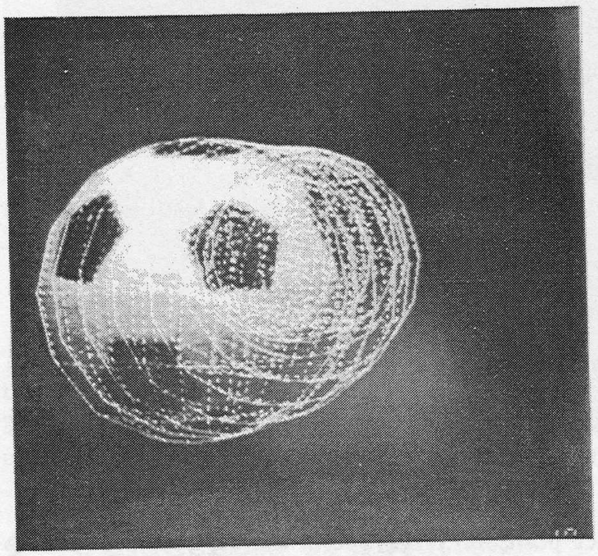

Figure 3: The 3D model at every 10 frames without correcting pose displacement.

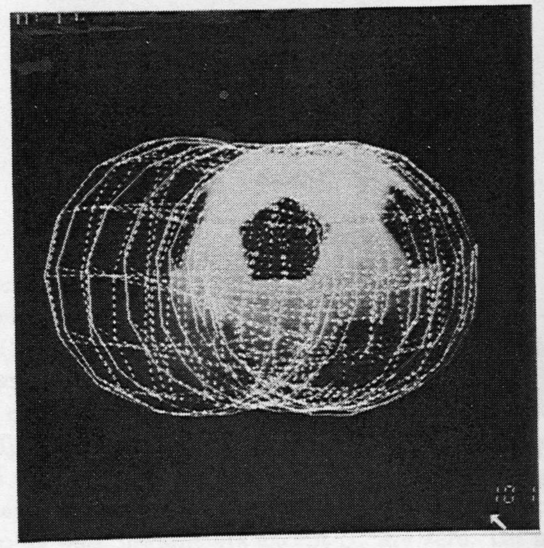

Figure 5: 3D model at every 10 frames by correcting pose displacement.

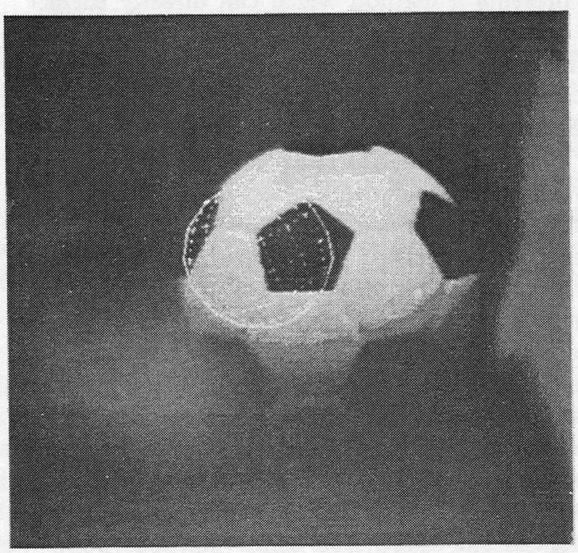

Figure 4: The 3D model at 100th frame without correcting pose displacement.

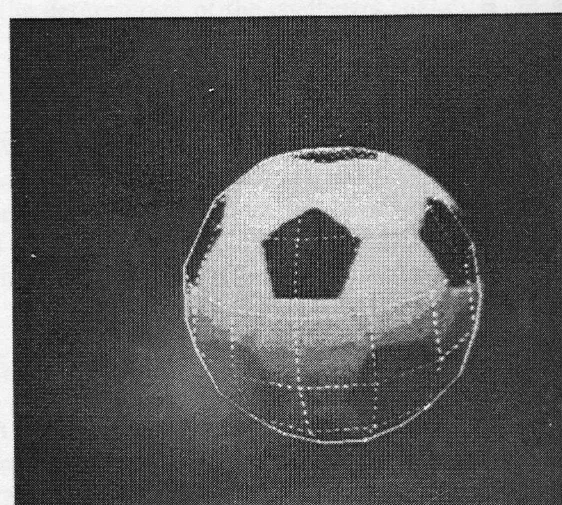

Figure 6: 3D model at 64th frame by correcting pose displacement.

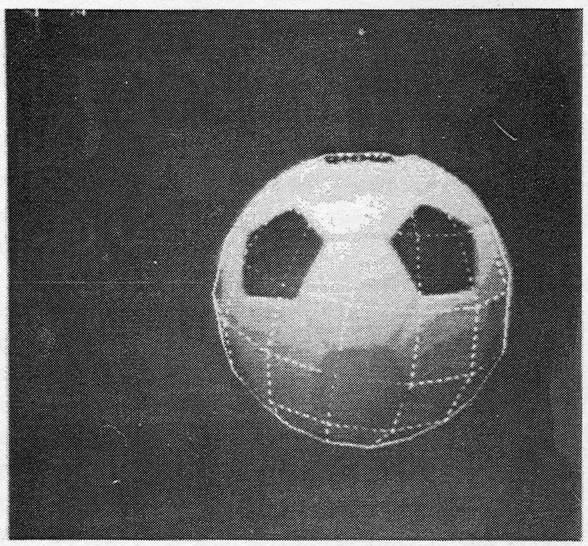

Figure 7: 3D model at 100th frame by correcting pose displacement.

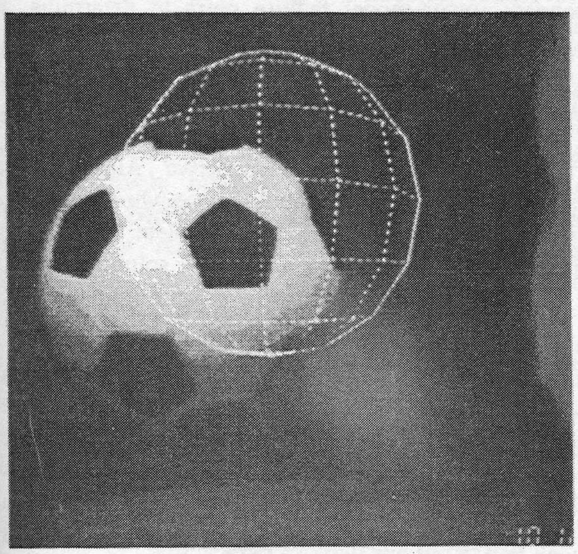

Figure 8: Initial position of the 3D model.

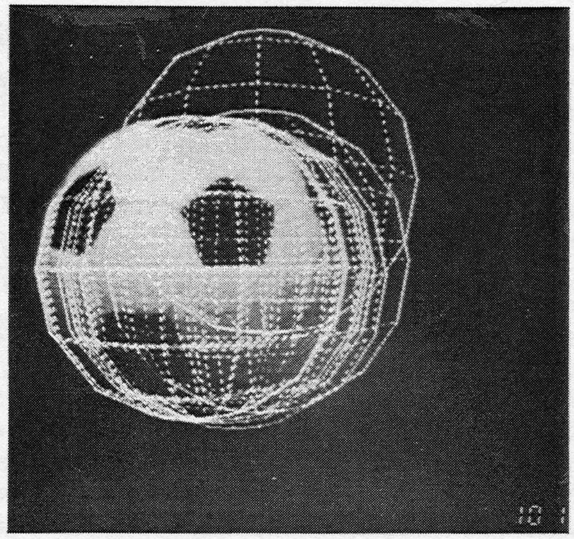

Figure 9: Correcting the pose of the 3D model at every iteration.

[3] D.G.Lowe, Integrated treatment of matching and measurement errors for robust model-based motion tracking, Proc. of *3rd ICCV*, Osaka, pp.436-440, 1990

[4] D.G.Lowe, Fitting parameterized three-dimensional models to images, *IEEE*,vol.PAMI-13,no.5,pp.450, 1991

[5] S.Negahdaripour and B.K.P.Horn, Direct passive navigation, *IEEE*,vol.PAMI-9, no.1, pp.168-176, 1987.

[6] S.Negahdaripour, Direct methods for structure from motion, Ph.D Thesis , Mechanical Engineering, MIT, 1986

[7] R.Y.Tsai and T.S.Huang, Estimating three-dimensional motion parameters of a rigid planar patch. *IEEE*, vol.ASSP-29, no.6, pp.1147-1152, 1981.

[8] M.Yamamoto, Direct estimation of three-dimensional motion parameters from image sequence and depth, *IECE Japan*, vol.J68-D, no.4, pp.562-569, 1985.

[9] M.Yamamoto, A general aperture problem for direct estimation of 3-D motion parameters. *IEEE*, vol.PAMI-11, no.5, pp.528-536, 1989.

[10] M.Yamamoto, P.Boulanger, J.-A.Beraldin, M.Rioux and J.Domey, Direct estimation of deformable motion parameters from range image sequence, Proc. 3rd ICCV, Osaka, pp.460-464, 1990.