ICARCV '90

INTERNATIONAL CONFERENCE ON AUTOMATION, ROBOTICS AND COMPUTER VISION

19-21 September 1990

PROCEEDINGS

ORGANISED BY

School of Electrical and Electronic Engineering
Nanyang Technological Institute, Singapore

The Institution of Engineers, Singapore

McGraw-Hill Book Co
Singapore New York St. Louis San Francisco Auckland Bogotá
Caracas Colorado Springs Hamburg Lisbon London Madrid Mexico
Milan Montreal New Delhi Oklahoma City Panama Paris San Juan
São Paulo Sydney Tokyo Toronto

A LINEAR ALGORITHM FOR MOTION ESTIMATION USING FEATURES OF OPTICAL FLOWS AND LINE SEGMENTS

Takuto JOKO, Koji ITO, Toshio TSUJI and Mutsuhiro TERAUCHI

Faculty of Engineering, Hiroshima University

Shitami, Saijo-cho, Higashi-Hiroshima 724, Japan

<Abstract> The estimation of the 3-D motion of a rigid object from a 2-D image sequence is one of the important problems in computer vision. The present paper proposes a new approach recovering 3-D motion using both the optical flow and line segment correspondences in perspective projection. First of all, a set of nonlinear equations are derived from optical flows of line segments in the image. It is then shown that when two pairs of parallel line segments in the scene are available, the nonlinear equations can be reduced to linear ones. The motion parameters are easily computed from the linear equations. In addition, the algorithm to detect the parallel lines in the scene is also given. Some simulation experiments will be carried out for the noise-free case, which confirms that the proposed algorithm is useful to estimate 3-D motion parameters from 2-D images.

INTRODUCTION

The estimation of the 3-D(three dimensional) motion of a rigid object from an 2-D image sequence is one of the important problems in computer vision. Two distinct approaches have been developed for the computation of motion parameters[1],[2]. One is based on extracting a set of two-dimensional features in the images corresponding to three-dimensional object features in the scene. The features might be points, lines, and/or curves. Motion parameters are then estimated from matching the features among image frames. The other approach is based on computing the two-dimensional field (i.e. optical flow) of instantaneous velocities of intensity values in the image plane.

In the feature-based method, it is assumed that image features have been extracted from each image and inter-frame correspondence has already been established between the features. Ullman[3] found one of the basic solutions from point correspondences based on assumption that the 3-D distance between two points remains the same after object motion. That is, under rigid body motion, it was shown that 3-D structure could be recovered uniquely from four points over three frames in orthographic projection, and also from five points over two frames in perspective projection. The procedures usually result in a system of nonlinear equations. Tsai and Huang[4] proposed a linear algorithm of 3-D motion estimation using eight point correspondences, which consists of two steps. First, a set of linear equations in eight unknown variables are derived. Then, these are solved from eight points over two frames in perspective projection.

Line correspondence is more reliable than point one. When lines are used as features, however, two frames are no longer sufficient and at least three frames are required. This is due to the fact that 3-D lines have an additional degree of freedom, i.e. a 3-D line can be moved along itself. Lie and Huang proposed a nonlinear algorithm using six line correspondences in three frames[5], and also a linear algorithm using thirteen line correspondences in three frames[6].

In the optical flow method, no correspondence between features over frames is required. The algorithm consists of two steps : 1) computing image plane velocities, and 2) recovering 3-D motion and structure from optical flow. A mathematical formulation of the second step was presented by Higgins and Prazdny[7]. They derived a set of twelve nonlinear equations in eleven unknown variables based on the assumptions that 1)the optical flow varies smoothly and 2) the surface of the rigid object is smooth. The algorithm requires the optical flow and its first- and second spatial derivatives. And furthermore, in order to reduce the complexity of the system of equations, the approaches are proposed restricting the nature of motion to be purely translatory or rotational[8], and/or restricting the imaged surface to be planar[2].

Now the present paper proposes a new approach recovering 3-D motion using both the optical flow and line segment correspondences in perspective projection. First of all, a set of nonlinear equations are derived from optical flows of line segments in the image. It is then shown that when two pairs of parallel line segments in the scene are available, the nonlinear equations can be reduced to linear ones. The motion parameters are easily computed from the linear equations. In addition, the algorithm to detect the parallel lines in the scene is also given. Some simulation experiments will be carried out for the noise-free case, which confirms that the proposed algorithm is useful to estimate 3-D motion parameters from 2-D images.

OPTICAL FLOW AND LINE SEGMENTS

The optical flow represents the two-dimensional field of instantaneous velocities of intensity values in the image plane. Then the flow of line segment, which is called "line flow", is represented by the instantaneous velocities of both the end points on the line segment.

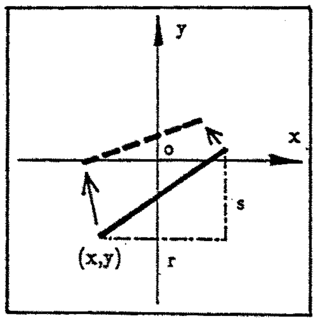


Fig.1 Image plane.

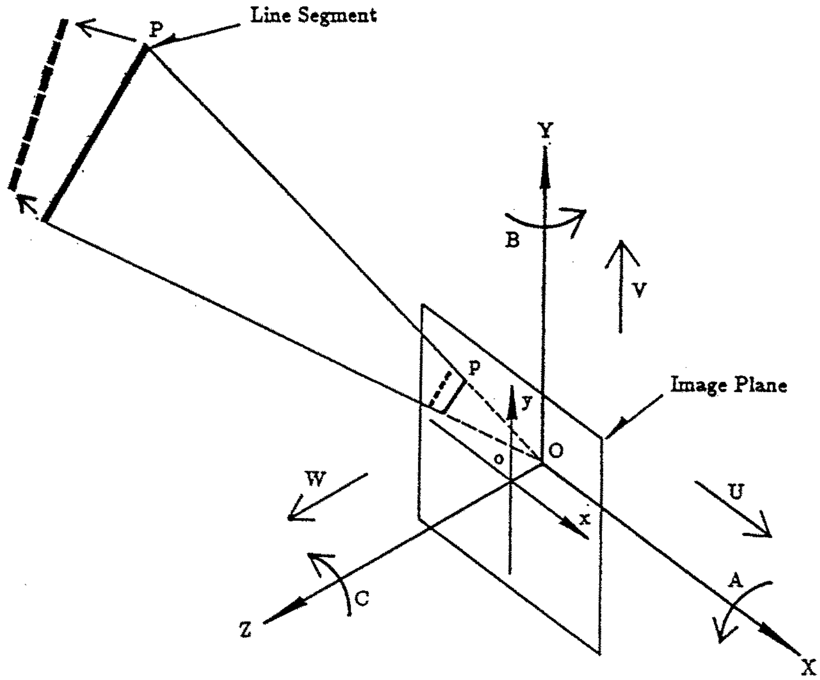


Fig.2 Relationship between the space coordinate system and the image space.

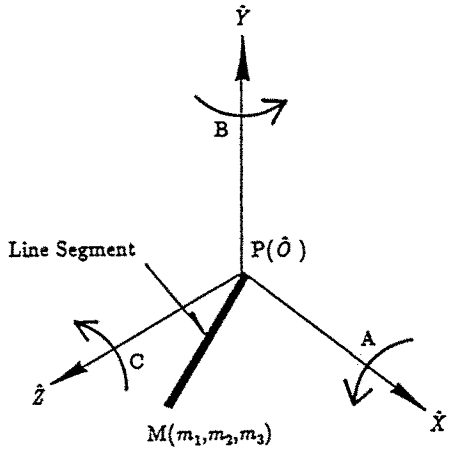


Fig.3 The coordinates system relative to the point P.

*Fig.*1 shows a line flow on the image plane. Let's denote an end point by (x,y) and another end point by (r,s). Then, (r,s) is represented by the relative coordinates. The line flow is defined by these time derivatives, which are given by

$$u = \dot{x},\ v = \dot{y},\ \mu = \dot{r},\ \lambda = \dot{s}. \tag{1}$$

As shown in *Fig.*2, let $OXYZ$ be a Cartesian coordinate system that is fixed with respect to the eye, OZ being the line of sight. The image plane is normal to the Z axis and is at unit distance from the origin. The x and y axes are parallel to the X and Y axes, respectively. Let (U,V,W) be the translational velocity of $OXYZ$ relative to the scene, and let (A,B,C) be its angular velocity. Then if (X,Y,Z) are the coordinates of the end point P of line segment in the scene, the instantaneous velocity of P is given by

$$\begin{aligned} \dot{X} &= -U - BZ + CY\ , \\ \dot{Y} &= -V - CX + AZ\ , \\ \dot{Z} &= -W - AY + BX\ . \end{aligned} \tag{2}$$

In the perspective projection, the image of $P(X,Y,Z)$ is formed by drawing a line from it to the origin $O(0,0,0)$ which intersects the image plane at (x,y). Then (x,y) are represented by

$$x = \frac{X}{Z},\quad y = \frac{Y}{Z}\ . \tag{3}$$

From this the instantaneous image velocity (u,v) of the end point $p(x,y)$ can be written as

$$\begin{aligned} u &= \frac{\dot{X}}{Z} - X\frac{\dot{Z}}{Z^2} \\ &= (-\frac{U}{Z} - B + Cy) - x(-\frac{W}{Z} - Ay + Bx)\ , \end{aligned} \tag{4}$$

$$\begin{aligned} v &= \frac{\dot{Y}}{Z} - Y\frac{\dot{Z}}{Z^2} \\ &= (-\frac{V}{Z} - Cx + A) - y(-\frac{W}{Z} - Ay + Bx)\ . \end{aligned} \tag{5}$$

Further, introducing the "vanishing point"

$$x_0 = \frac{U}{W},\quad y_0 = \frac{V}{W} \tag{6}$$

and substituting (6) and $\omega = W/Z$ into (4) and (5), we obtain

$$u = xyA - (x^2+1)B + yC + (x - x_0)\omega\ , \tag{7}$$

$$v = (y^2+1)A - xyB - xC + (y - y_0)\omega\ . \tag{8}$$

When x, y, u and v are obtained from the image data, the above (7) and (8) comprise nonlinear equations in six unknowns A, B, C, ω, x_0, y_0

Next, as shown in *Fig.*3, let's define a Cartesian coordinate system $\hat{O}\hat{X}\hat{Y}\hat{Z}$ which is fixed with respect to the end point $P(X,Y,Z)$. Then let's denote another end point by $M(m_1,m_2,m_3)$ which is represented by the coordinate system $\hat{O}\hat{X}\hat{Y}\hat{Z}$. Note that M represents the direction vector of the line segment. The instantaneous velocity of M is given by

$$\begin{aligned} \dot{m}_1 &= -Bm_3 + Cm_2\ , \\ \dot{m}_2 &= -Cm_1 + Am_3\ , \\ \dot{m}_3 &= -Am_2 + Bm_1\ . \end{aligned} \tag{9}$$

Further, using m_1, m_2 and m_3, the relative coordinates (r,s) can be written as follows.

$$r = \frac{m_1 - m_3 x}{Z + m_3} , \quad u = \frac{m_2 - m_3 y}{Z + m_3} . \tag{10}$$

Then, from (2), (9) and (10), μ and λ of (1) may be written in the forms.

$$\begin{aligned} \mu &= \frac{\dot{m}_1 - \dot{m}_3 x - m_3 \dot{x}}{Z + m_3} - \frac{(m_1 - m_3 x)(\dot{Z} + \dot{m}_3)}{(Z + m_3)^2} \\ &= \frac{(-Bm_3 + Cm_2) - (-Am_2 + Bm_1)x - m_3 u}{Z + m_3} \\ &\quad - \frac{(m_1 - m_3 x)(-W - AY + BX - Am_2 + Bm_1)}{(Z + m_3)^2} , \end{aligned} \tag{11}$$

$$\begin{aligned} \lambda &= \frac{\dot{m}_2 - \dot{m}_3 y - m_3 \dot{y}}{Z + m_3} - \frac{(m_2 - m_3 y)(\dot{Z} + \dot{m}_3)}{(Z + m_3)^2} \\ &= \frac{(-Cm_1 + Am_3) - (-Am_2 + Bm_1)y - m_3 v}{Z + m_3} \\ &\quad - \frac{(m_2 - m_3 y)(-W - AY + BX - Am_2 + Bm_1)}{(Z + m_3)^2} . \end{aligned} \tag{12}$$

Let $t = m_3 / (Z + m_3)$. Then from (10), the direction vector $M(m_1, m_2, m_3)$ can be written as follows.

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = (Z + m_3) \begin{bmatrix} r + tx \\ s + ty \\ t \end{bmatrix} . \tag{13}$$

Substituting (13) into (11) and (12), we have

$$\begin{aligned} \mu &= (ry + sx + txy + rs)A \\ &\quad - (2rx + t + tx^2 + r^2)B \\ &\quad + (s + ty)C + (1 - t)r\omega - ut , \end{aligned} \tag{14}$$

$$\begin{aligned} \lambda &= (2sy + t + ty^2 + s^2)A \\ &\quad - (ry + sx + txy + sr)B \\ &\quad - (r + tx)C + (1 - t)s\omega - vt . \end{aligned} \tag{15}$$

When x, y, r, s, u, v, μ and λ are obtained from the image data, the above (14) and (15) comprise nonlinear equations in five unknowns A, B, C, ω and t.

ESTIMATION OF MOTION PARAMETERS

Since ω and t in (7), (8), (14) and (15) are related to the relative depth, they changes depending on the line segments. Although we can obtain four equations from a line segment, they are not independent. Only two of them are independent. Therefore, three line segments are required to solve A, B, C, x_0 and y_0.

Even if three independent line segments are available, however, it is not easy to solve the system of nonlinear equations (7), (8), (14) and (15). Note that the unknown parameter in the coefficients A, B, C and ω of (14) and (15) is only t. If the unknown parameter t could be found, (14) and (15) would immediately give a set of linear equations. We assume here that there exist some parallel line segments in the scene. Then a separate computation can be carried out on the parameter t utilizing the geometrical structure of parallel line segments.

Linear algorithm

When a couple of line segments l_1 and l_2 are parallel, the direction vector of each line segment is same. From (13), then, we have

$$\begin{bmatrix} r_1 + t_1 x_1 \\ s_1 + t_1 y_1 \\ t_1 \end{bmatrix} = \alpha \begin{bmatrix} r_2 + t_2 x_2 \\ s_2 + t_2 y_2 \\ t_2 \end{bmatrix} . \tag{16}$$

Canceling α gives

$$\begin{aligned} t_1 &= \frac{r_2 s_1 - r_1 s_2}{s_2 x_1 - s_2 x_2 + r_2 y_2 - r_1 y_1} , \\ t_1 &= \frac{r_2 s_1 - r_1 s_2}{s_1 x_1 - s_1 x_2 + r_1 y_2 - r_1 y_1} . \end{aligned} \tag{17}$$

It is thus seen that the unknown t can be computed separately when a couple of line segments are parallel.

Canceling ω in (14) and (15), we have

$$\begin{aligned} s\mu - r\lambda &= (s^2 x + tsxy - try^2 - rsy - tr)A \\ &\quad + (r^2 y + trxy - tsx^2 - rsx - ts)B \\ &\quad + (s^2 + r^2 + trx + tsy)C - tsu + trv . \end{aligned} \tag{18}$$

Since (18) is a linear equation in the unknowns A, B and C, we can solve it using three different line segments. When all of them are parallel, however, the equations becomes linear dependent in terms of A, B and C. Therefore, we need two pairs of parallel line segments to obtain the solution.

Detection algorithm of parallel line segments

In order to utilize (18), the parallel line segments in the three dimensional scene have to be found from the image data. When only the position data are available in the image, it is impossible to specify the parallel line segments in the scene. If the instantaneous velocity in the image is available, however, we can find the parallel line segments as shown in *Fig.4*.

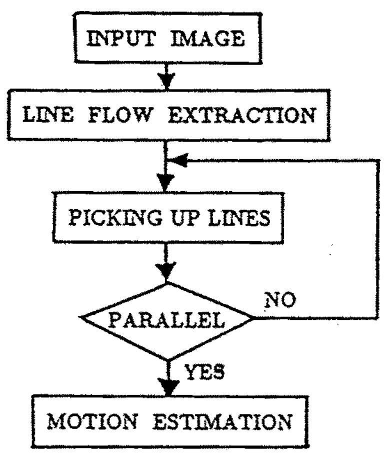


Fig.4 The flow chart of processing.

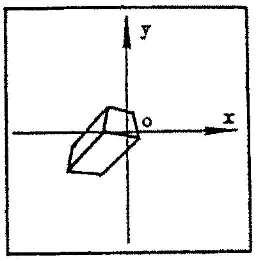
Table.1 Detections of two pairs of parallel line segments.

U = 0.6 , V = 0.8 , W = 0.2

	True	Estimation Values					
		a	b	c	d	e	f
A (rad/sec)	0.01	0.01033	0.01049	-0.7205	0.008731	0.004048	-0.000226
B (rad/sec)	0.01	0.01008	0.009285	-0.7207	0.008480	0.01256	-0.01715
C (rad/sec)	0.01	0.01013	0.009926	-0.5380	0.008925	0.05625	0.02514
x_0(average)	3.0	2.976	3.007	3.789	2.395	3.072	5.087
y_0(average)	4.0	3.986	4.025	-5.953	3.078	0.4053	2.671
x_0(variation)		0.0447	0.0685	1.885	0.4423	0.5401	0.7680
y_0(variation)		0.04725	0.1042	1.490	0.4773	13.62	0.7827

Table.2 Two pairs of parallel line segments.

	X	Y	Z		X	Y	Z
line 1	20	14	10	-	32	26	19
line 2	8	-2	10	-	40	30	34
line 3	18	27	10	-	30	24	15
line 4	-3	-3	10	-	21	-9	20



First, two pairs of a couple of line segments are picked up in the image. Assuming that each pair is parallel, the parameter t is computed from (17). Then, since A, B and C are obtained from (18) and ω from (14) and (15), (7) and (8) become the equations in the unknowns x_0 and y_0. Accordingly, we can obtain four different values of x_0 and y_0 from four line segments respectively. If the line segments which have been chosen are really parallel, these values should be same. Consequently, we can extract the parallel line segments from variations in these values.

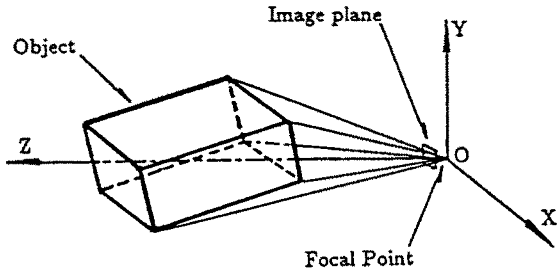


Fig.5 A rectangular parallelepiped in the scene.

SIMULATION

Detection of parallel line segments

Generally, a couple of line segments are able to take three kind of geometrical situations as follows.

i) in parallel

ii) not in parallel but on the plane

iii) in twisted positions

In this algorithm, it is possible to estimate the motion parameters only when there exist two pairs of parallel line segments in the scene. *Table.*1 shows the motion parameters which were estimated in various cases. a)~f) in Table 1 denote those that follow.

a): two pairs of parallel line segments [i) and i)].

b): two pairs of parallel line segments which are on the same plane [i) and i)].

c): parallel line segments and a pair of line segments which are on the same plane [i) and ii)].

d): parallel line segments and a pair of line segments which are in a twisted position [i) and iii)].

e): four line segments which are on the same plane [ii) and ii)].

f): two pairs of line segments which are in twisted positions [iii) and iii)].

Then four different x_0 and y_0 were computed from (7) and (9) and those variations(standard deviation / average) are also shown in *Table.*1. *Table.*2 shows the position data of two pairs of parallel line segments as an example. Z coordinate of end-points of every line segment is at z=10, and the length is 10~40.

We can not estimate the correct motion parameters under the situations of c)~f) at *Table.*1, because they does not lie in parallel. But we can exclude to adopt these estimation results based on the variations of x_0 and y_0. When two pairs of parallel line segments were used, even if all of them were on the same plane, the variations of both x_0 and y_0 are less than 0.1. On the other hand, the variations of c) ~f) are larger than 0.4. From this, it is possible to check whether the line segments which have been picked up are really in parallel.

Application to rectangular parallelepiped

Simulation experiments were carried out for the noise-free case. In the simulations, we suppose that the rectangular parallelepiped with textured surfaces consisting of line segments(edges) is moving in front of the camera. *Fig.*5 shows the simulation model. The camera is located at the origin of the 3D world coordinates. The image plane is parallel with $X-Y$ plane, and the center of the image is at

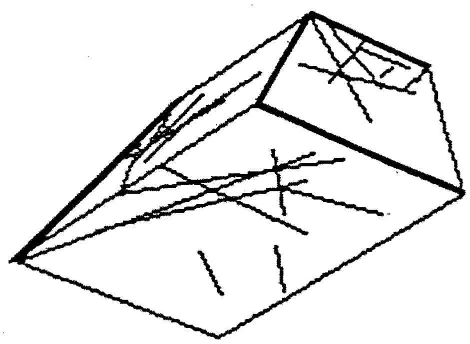


Fig.6 An example of detected parallel lines.

(0,0,1). Then the view angle of the camera is about 60°. The 3D object is located such that one of apexes is at the origin of the world coordinates and three edges coincide with X, Y and Z axes respectively. Then it was moved first by rotations (10°,10°,40°) around X, Y and Z axes through the origin, then followed by translations (−10,10,60) along with X, Y and Z axes.

*Fig.*6 shows the object projected on the image plane. The heavy line indicates two pairs of parallel line segments which were extracted based on the variations of the estimates of x_0 and y_0. Then those variations were less than 0.1. *Fig.*7 shows the estimation results of the motion parameters when the object was rotated and translated successively. a) is real and estimated values of angular velocity A, and b) is those of vanishing point $x_0\,(=U/W)$.

CONCLUSION

In this paper, we have presented a linear algorithm for motion estimation based on line segment flows and parallel line segments. The algorithm needs four line segments correspondences over two frames and the detection of two pairs of parallel line segments. We may obtain line segment correspondences by combining the optical flow with the edge extraction. That is, the velocity field along the line segment changes monotonously and has different direction vectors depending on the line segment. Therefore, it is possible to obtain the line segment flows by forming the optical flows on the image into groups corresponding to each line segment(edge). Further, the nonlinear equations could be reduced to linear ones by assuming two pairs of parallel line segments in the scene. Since we can often find the scenes which include parallel line segments such as box, column, texture of surface etc., this does not mean to restrict the range of application. Finally, it should be noted that since the optical flow requires the first spatial derivatives, the computation results depend on the magnitude of the translation and rotation between image frames. We have to select the adequate sampling period between image frames.

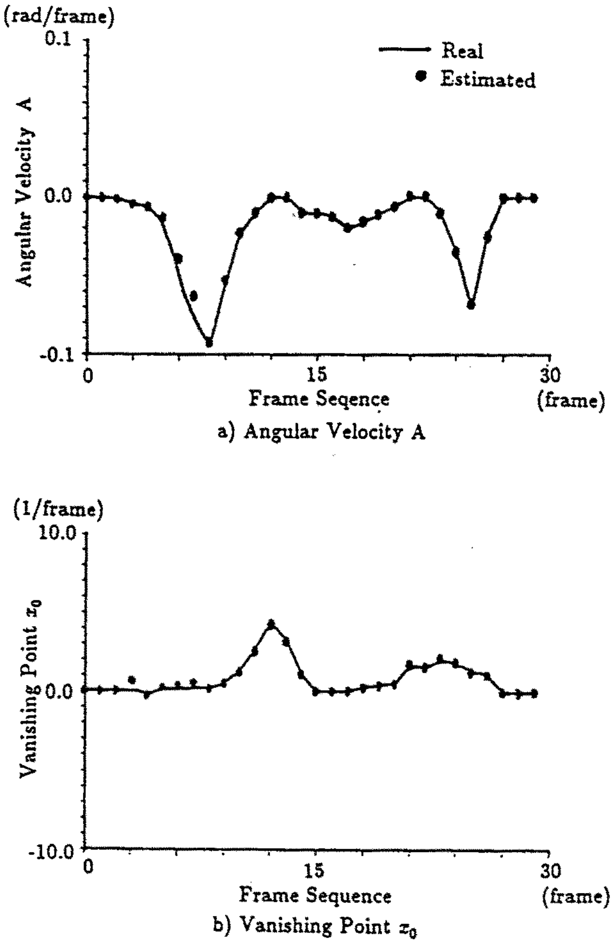


Fig.7 Estimation results of the motion parameters when a rectangular parallelepiped was rotated and translated continuously.

REFERENCES

[1]J.K.Aggarwal and N.Nandhakumar : "On the computation of motion from sequence of images - a review", Proc.IEEE, **76** , 8, pp.917-935(1988).

[2]M.Subbaro : "Interpretation of Visual Motion - A Computational Study", Morgan Kafuman Pub (1988).

[3]S.Ullman : "The interpretation of visual motion", The MIT Press (1979).

[4]R.Y.Tsai and T.S.Huang : "Uniqueness and estimation of three-dimensional motion parameters of rigid object with curved surface" , *IEEE Trans. Pattern Anal. and Machine Intell.*, **PAMI-6** , 1, pp.13-27(1984).

[5]Y.Liu and T.S.Huang : "Estimation of rigid body motion using straight line correspondence", *Comput. Vision, Graphics and Image Process.*, **43** , pp.37-52(1988).

[6]Y.Liu and T.S.Huang : "A linear algorithm for motion estimation using straight line correspondence", *Comput. Vision, Graphics and Image Process.*, 44 , pp.35-57(1988).

[7]H.C.Longuet-Higgins and K.Prazdny : "The interpretation of moving retinal image" , *Proc. Roy. Soc. London*, B-208, pp385-397(1980).

[8]T.D.Williams : "Depth from camera motion in a real world scene" , *IEEE Trans. Pattern Anal. and Machine Intell.* ,**PAMI-2** , 6, pp.511-516(1980).