Systems and Computers in Japan, Vol. 22, No. 14, 1991
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. 73-D-II, No. 9, September 1990, pp. 1485-1492

A Linear Algorithm for Motion of Rigid Objects Using Features of Parallel Lines and Optical Flow

Takuto Joko, *Associate Member* Koji Ito, Toshio Tsuji, and Mutsuhiro Terauchi, *Members*

Faculty of Engineering, Hiroshima University, Higashi-Hiroshima, Japan 724

SUMMARY

Various approaches have been proposed toward the problem of restoring three-dimensional (3-D) structures and motion of rigid bodies from image information. Ullman and Huang presented algorithms using point and line correspondences, in which they assume that the correspondence problems can be solved. Prazdny et al., on the other hand, presented an algorithm using optical flow, in which equations become nonlinear and thus the second derivative of velocity is required.

This paper proposes an algorithm which combines optical flow and edge information. First, considering segments consisting of edges in an image, we derive an equation for optical flow. Then, making use of parallelism of line segments, we show that 3-D motion can be restored by using linear equations. To apply the algorithm there must exist two pairs of parallel line segments on an object. This paper presents an algorithm for extracting these pairs of parallel line segments. Finally, we verify the effectiveness of the algorithm by simulation.

1. Introduction

The problem of restoring three-dimensional (3-D) motion and structure of an object using monocular image information under passive illumination is one of the important and interesting subjects in computer vision; however, it generally becomes an ill-posed problem since it restores from a 2-D image depth information that is degenerated by projection. Thus, in order to compensate missing information, we need to make and add certain assumptions or constraints to image data.

Two approaches toward motion restoration problem are classified [1-3]. One is an approach in which we find corresponding points in several consecutive images and after using them object motion is restored. The other makes use of velocity vector of motion (optical flow) at some instant which is projected onto an image.

The former ones are difficult depending on levels by which we extract features of images to take correspondence between images. Lines are more reliable in correspondence than points and faces. Ullman [4] obtained a fundamental algorithm for point correspondence, assuming a rigid object. That is, he showed that when only positional information is used, four points and three scenes suffice in orthographic projection, and five points and three scenes suffice in perspective projection to reconstruct 3-D structure and motion. Later, Tsai and Huang [5], using singular value decomposition of a matrix composed of intermediate parameters obtained from motion information, showed that motion can be restored from eight points and two scenes in perspective projection by deriving linear equations consisting of eight unknown variables.

ISSN0882-1666/91/0014-0070$7.50/0

Yen and Huang [6] are the first to notice restoration by line correspondence and proposed an algorithm based on straight line correspondence onto a unit sphere. Further, Liu and Huang [7] showed images of at least three scenes for motion restoration due to straight line correspondence and proposed a nonlinear algorithm using six lines and three scenes. Mitichi et al. [8] presented an algorithm based on invariance of angle formed by two lines on a rigid body, and Faugeras et al. [9] gave a method using Kalman filter. Since all of them find a solution by iteration, estimation of initial value is bottlenecked and it requires much computation time. Therefore, Liu and Huang [10] proposed a linear algorithm, which requires 13 straight lines and three scenes. Line correspondence requires more image data than point correspondence because, even if a direction of a line is determined in the 3-D space, the line possesses freedom to move along itself.

On the other hand, methods based on optical flow make use of space and time derivatives local on image planes. Higgins and Prazdny [11] derived a set of 12 nonlinear equations containing 11 unknown variables under the assumption that optical flow and faces composing a rigid body are smooth and showed that, if it is possible to solve them, motion of the rigid object can be restored. Note that this algorithm is difficult to obtain analytical solutions; moreover, it requires a second-order space derivative of optical flow and no solution is obtained if there is no translational velocity component parallel to the image plane. Therefore, attempts to simplify equations are done by assuming that faces within a field of vision can be approximated by a plane [13].

Thus far, motion restoration using straight line correspondence has been dealt with, but no study has been able to solve optical flow noticing line segments. Methods based on optical flow remain nonlinear problems.

Here, we propose an algorithm for restoring 3-D motion of an object from optical flow and edge (line segment) information under perspective projection. Assuming the existence of parallel line segments in the 3-D scene, we show that a nonlinear equation can be reduced to a linear equation by making use of the property. Although assuming that a pair of parallel line segments implies constraint of the problem, we think the attempt is useful since there is a good chance that such a scene exists. Further, almost no attempts have been made to solve optical flow by linear equations.

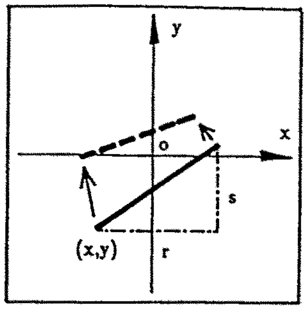


Fig. 1. Image plane.

2. Optical Flow and Line Segments

Optical flow represents velocity field in an image and we use perspective projection in this paper. Motion is represented by relative motion of an observer against an object. This makes it possible to express any motion using the same coordinate system.

Flow (velocity field) of a line segment can be represented by flow of both endpoints of the line segment. In this paper we call flow at an endpoint together with that at another endpoint relative to it "line segment flow."

Figure 1 shows line segment flow in an image plane. Let (x, y) be the coordinate of one endpoint and (r, s) be relative coordinate of the other endpoint. Then, we define line segment flow by their space derivatives and let them be u, v, μ, and λ. We have

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

2.1. Flow at endpoint

Figure 2 illustrates very small motion in the space in the Cartesian coordinate system $OXYZ$ such that the origin O becomes the momentary viewpoint. Let translational components of observer's motion be U, V, W and rotational components be A, B, C. Letting the coordinate of an endpoint P in the space be $P(X, Y, Z)$, the velocity components of P in very small motion are given by

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

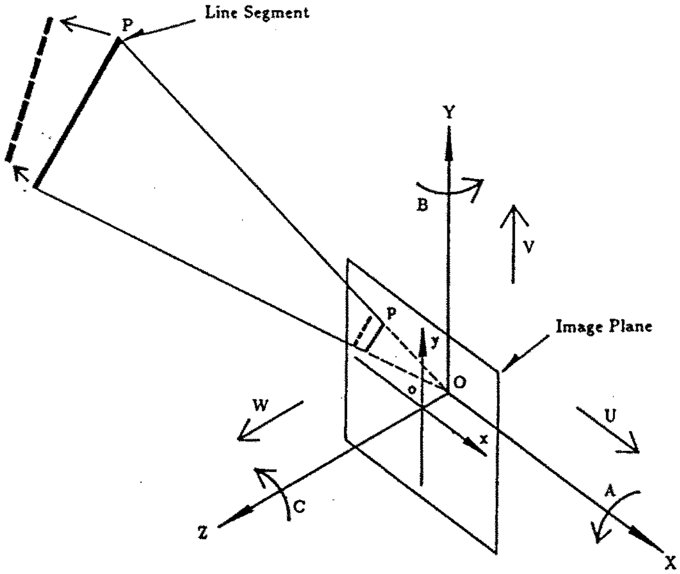


Fig. 2. Relationship between the space coordinate system and the image plane.

Representing a point into which $P(X, Y, Z)$ is projected on an image plane as $p(x, y)$, the relation between the two becomes

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

Then, from Eq. (1) the velocity component u, v at the endpoint $p(x, y)$ is

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

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

Here, we introduce the vanishing point (the point of motion at infinity)

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

Letting $\omega = W/Z$ and substituting it into Eqs. (4) and (5), we have [11]

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

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

The foregoing expression becomes a nonlinear equation with unknown parameters A, B, C, ω, x_0, and y_0 when we assume that x, y, u and v are obtained from image data.

2.2. Flow at relative endpoint

Next, we determine a Cartesian coordinate system $\hat{O}\hat{X}\hat{Y}\hat{Z}$ with an endpoint $P(X, Y, Z)$ of an endpoint of a line segment at the origin $\hat{O}$. Then, letting $M(m_1, m_2, m_3)$ be the other endpoint expressed by the coordinate system, M represents a directional vector of the line segment. The time derivative of M becomes

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

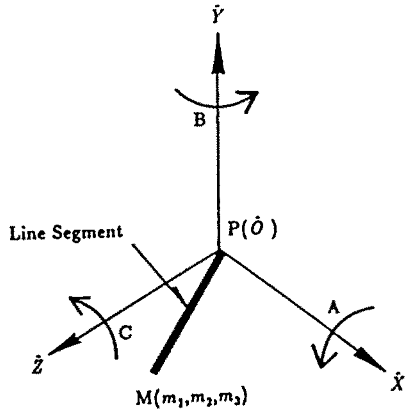


Fig. 3. The coordinate system relative to the point P.

The relative positions r, s can be represented as follows using m_1, m_2, and m_3.

$$r=\frac{m_1-m_3x}{Z+m_3},\quad s=\frac{m_2-m_3y}{Z+m_3} \tag{10}$$

Then, from Eqs. (2), (9), and (10), μ and λ in Eq. (1) are

$$\begin{aligned}\mu&=\frac{\dot{m}_1-\dot{m}_3x-m_3\dot{x}}{Z+m_3}-\frac{(m_1-m_3x)(\dot{Z}+\dot{m}_3)}{(Z+m_3)^2}\\&=\frac{(-Bm_3+Cm_2)-(-Am_2+Bm_1)x-m_3u}{Z+m_3}\\&\quad-\frac{(m_1-m_3x)(-W-AY+BX-Am_2+Bm_1)}{(Z+m_3)^2}\end{aligned} \tag{11}$$

$$\begin{aligned}\lambda&=\frac{\dot{m}_2-\dot{m}_3y-m_3\dot{y}}{Z+m_3}-\frac{(m_2-m_3y)(\dot{Z}+\dot{m}_3)}{(Z+m_3)^2}\\&=\frac{(-Cm_1+Am_3)-(-Am_2+Bm_1)y-m_3v}{Z+m_3}\\&\quad-\frac{(m_2-m_3y)(-W-AY+BX-Am_2+Bm_1)}{(Z+m_3)^2}\end{aligned} \tag{12}$$

Here, setting $t = m_3/(Z + m_3)$, from Eq. (10) the directional vector $M(m_1, m_2, m_3)$ can be written as follows using r, s, and t:

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

Substituting it into Eqs. (11) and (12), μ and λ become

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

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

These are nonlinear equations with A, B, C, ω, and t as unknown parameters when x, y, r, s, u, v, μ, and λ are obtained from image data.

3. Algorithm for Estimating Motion Parameters

In Eqs. (7), (8), (14), and (15), ω and t are different depending on line segments since they are information on depth. That is, two among four equations for a line segment become effective. Thus, three line segments are needed to find five motion parameters A, B, C, x_0, and y_0.

However, Eqs. (7), (8), (14), and (15) are not easy to be solved in their forms since they are nonlinear equations. So we pay attention to the fact that unknown parameters contained in coefficients A, B, C, ω in Eqs. (14) and (15) is only t. If t can be found in any way, Eqs. (14) and (15) are reduced to linear equations. Here, assuming that there are several parallel line segments on a rigid object, we find t using its geometric properties.

3.1. Linear algorithm

When two line segments l_1 and l_2 are parallel, since directions of their directional vectors are the same, from Eq. (11) the following relation holds:

$$\begin{bmatrix}r_1+t_1x_1\\s_1+t_1y_1\\t_1\end{bmatrix}=\alpha\begin{bmatrix}r_2+t_2x_2\\s_2+t_2y_2\\t_2\end{bmatrix} \tag{16}$$

Eliminating α from this expression and solving it with respect to t_1 and t_2, we have

$$t_1 = \frac{r_2 s_1 - r_1 s_2}{s_2 x_1 - s_2 x_2 + r_2 y_2 - r_1 y_1}$$

$$t_2 = \frac{r_2 s_1 - r_1 s_2}{s_1 x_1 - s_1 x_2 + r_1 y_2 - r_1 y_1} \quad (17)$$

In this way we can compute t in the case in which two line segments are parallel in the space; and thus we can make Eqs. (14) and (15) linear. Since ω is different for each line segment, eliminating ω from Eqs. (14) and (15), we have

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

Equation (18) is a linear form with A, B, and C as unknown parameters and it is to be solved using three line segments.

However, if three line segments are all parallel, this equation is linearly dependent and thus it has no solution (proof omitted). Therefore, if we obtain a solution using this method, we need two pairs of parallel lines.

3.2. Detection of parallel line segments

To use Eq. (18) we need to be able to detect parallel line segments in the 3-D space from image data. When only positional information in one image plane is used, it is impossible to detect parallel line segments in the space. However, detection becomes possible by using velocity information.

This paper first assumes parallelism of two pairs of two line segments to find t and then using Eqs. (7), (8), (14), (15), and (18) we find A, B, C, x_0, and y_0. Next, we shall show that parallel line segments can be detected using redundancy of these expressions.

First, we take two pairs of two line segments and find t for each line segment. Once A, B, and C are found by Eq. (18) and ω by Eqs. (14) and (15), Eqs. (7) and (8) become expressions in which only x_0 and y_0 are unknown and thus four different values are obtained from four line segments. If two pairs of line segments selected are not parallel, these values should not coincide with each other. We use this for checking parallelism.

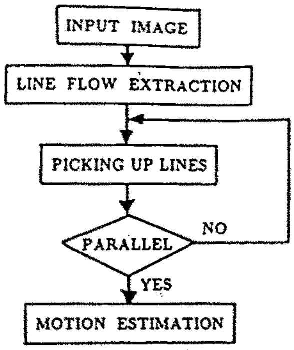


Fig. 4. The flow of the processing.

In other words, by implementing such a check for every combinatin of two pairs of line segments in an image plane we can detect pairs of parallel line segments.

4. Simulation

We implemented computer simulation to verify the effectiveness of the proposed motion parameter estimation algorithm. The data used were space coordinates of endpoints of line segments and we made translation and rotation about the origin and obtained line segment flow by perspective projection.

A flow of the algorithm is shown in Fig. 4. First, we derive two pairs of line segments from an image plane. Next, we estimate motion parameters using these line segments. Then, implementing parallelism check, if both pairs are parallel, we determine motion parameters.

4.1. Parallelism check

We first verify the validity of the method for parallelism check presented in Sect. 3.2. Three cases for geometric relationship between two line segments in the space are used:

(1) they are parallel;

(2) they are not parallel but lie on the same plane; and

(3) they are twisted positions.

Table 1. Parallelism check

$U=0.6,\ V=0.8,\ W=0.2$

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

Using this method we can correctly estimate only in the case in which two pairs of line segments are both parallel. Either one of the two pairs of line segments must not be in the relationship of cases (2) or (3).

Table 1 shows the results when we classify two pairs of line segments, depending on which of (1), (2) or (3) in the relationship they are related to, and estimate parameters using Eq. (18). Classification of line segments is shown:

a: two pairs of parallel line segments;

b: two pairs of parallel lines lie in the same plane; and

c: pair of parallel line segments and pair of line segments which are not parallel but lie on the same plane;

d: pair of parallel line segments and pair of line segments in twisted positions;

e: two pairs of line segments which are not parallel but lie on the same plane; and

f: two pairs of line segments in twisted positions.

We also obtained four different values of x_0 and y_0 by Eqs. (7) and (8) and compared their variation coefficients (standard deviation/average). We gave the data at locations of the four line segments in the space and unified the Z-coordinate of the one endpoint of each line segment to be 10 and let the lengths be 10 - 40. Table 2 shows data of two pairs of parallel line segments as an example of given line segments.

Tables 1a and b show the results when we found t using parallelism of line segments and got correct estimated values. Tables 1c - f assume parallelism for nonparallel line segments and thus correct estimated values are not obtained. For checking parallelism, variation coefficients for x_0 and y_0 are used. When two pairs of parallel line segments are used, variation coefficients for x_0 and y_0 are both about 0.1, even in the case in which four line segments lie in the same plane as in b. On the other hand, when they are not parallel, they are greater than or equal to 0.4, even in the case in which one pair is parallel and the other is in twisted positions as in d. Effectiveness of the check for parallelism of pairs of line segments using Eqs. (7) and (8) follows from this result.

4.2. Computational precision

If two pairs of parallel line segments are detected successfully, we can estimate motion parameters using Eq. (18). Note that no effective method has been found other than one of finding optical flow (velocity field) by difference between two images. Therefore, as motion among images becomes larger, it becomes harder to

Table 2. Two pairs of parallel line segments

	X	Y	Z		X	Y	Z
Line segment 1	20	14	10	—	32	26	19
Line segment 2	8	−2	10	—	40	30	34
Line segment 3	18	27	10	—	30	24	15
Line segment 4	−3	−3	10	—	21	−9	20

regard the difference as approximated derivative and thus it seems that estimation precision goes down. Then we investigated relation between magnitude of motion parameters and estimation precision using data of two pairs of parallel line segments shown in Table 2.

Figure 5 shows the relationship between the motion parameter and estimation error: (a) is angular velocity around the Z axis; (b) is velocity U along the X axis; and (c) expresses the relationship between velocity W in the direction of the Z axis and estimation error. Generally, it is impossible to estimate absolute translational velocity only from data on image planes. Therefore, we unify Z coordinate values of the endpoints of four line segments and standardize by the value; u_0, v_0, and ω are standardized values of UI, V, and W and we have

$$u_0=\frac{U}{Z},\ v_0=\frac{V}{Z},\ \omega=\frac{W}{Z} \tag{19}$$

In Figs. 5(a) and (b) error increases as motion becomes larger, but in Fig. 5(c) it decreases once and then increases. If we want to suppress the error to within 1 percent, the rotational component A should be less than or equal to about 0.004 rad and u_0 should be within 0.12. However, ω has large error and it is hard to suppress it within 1 percent. If we want to suppress it within 7 percent, it must be no greater than about 0.07.

When we deal with real images, since we cannot find absolute coordinate values of an object, it is impossible in practice to standardize by a point on an object. Therefore, for translational component we evaluate an error considering the vanishing point (x_0, y_0) that is the ratio with depth component. Figure 6 shows how translational velocity ω affects the estimation error of the vanishing point x_0. The error increases when ω is too small and also when it is too large. The precision is worse even when ω is small because the depth component enters the denominator of the vanishing point (x_0, y_0). To suppress the error of x_0 within 5 percent, $\omega(=W/Z)$ must be greater than 0.015 and less than 0.08. Thus, we should appropriately settle sampling interval between images so that the ω value of each line segment does not become too small. Note that since rotational component is not affected by W, precision becomes better as motion between images becomes smaller.

4.3. Application to rectangular parallelepiped

As an example of a rigid body containing parallel line segments, we implemented estimation of motion parameter using a rectangular parallelepiped having texture. Here by texture we mean pixels on an object surface detected as line segments (edges). Figure 7 shows the conceptual model of this simulation. The image plane is parallel to the X-Y planes and is the plane such that the image center is (0, 0, 1). The size of the image is assumed to be a square of 1 × 1. Then, the visual angle is about 60°. When making 3-D data, we made a rectangular parallelepiped such that the origin is one vertex and three edges coincide with X, Y, and Z axes and edge lengths are (20, 30, 50). Next, we rotated it in its role, pitch, and yaw each by (10, 10, 40) degrees and translated it by (-10, 10, 60) in the directions of X, Y, and Z axes.

Figure 8 shows a result of projection of an object onto an image plane. Bold lines represent two pairs of parallel lines detected by parallelism check using scattering of estimated values. The variation coefficients of x_0 and y_0 estimated by these line segments are both less than or equal to 0.1.

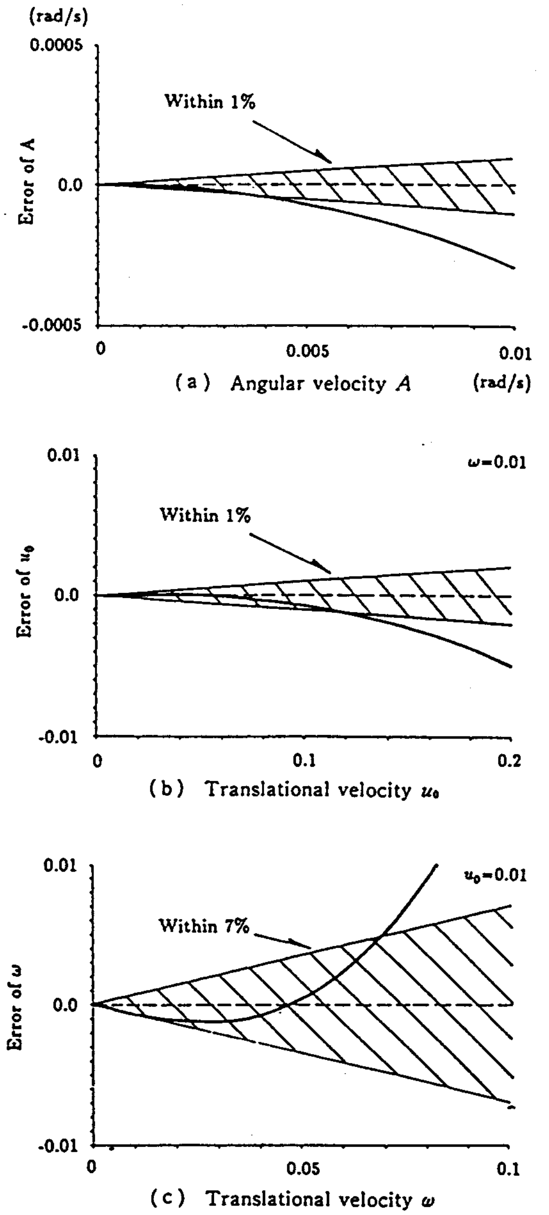


Fig. 5. The relationship between the motion parameter and its error.

Figure 9 shows examples of estimation when various motion parameters are given in sequence. The true values and estimated values of angular velocity A are shown in (a) and those of the vanishing point x_0 are in (b). The horizontal axis is the frame sequence and the

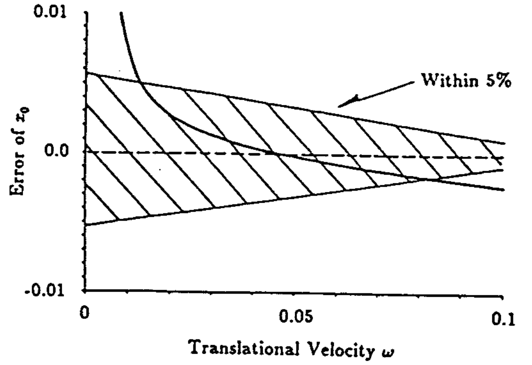


Fig. 6. The relationship between the translational velocity and the error of the vanishing point x_0.

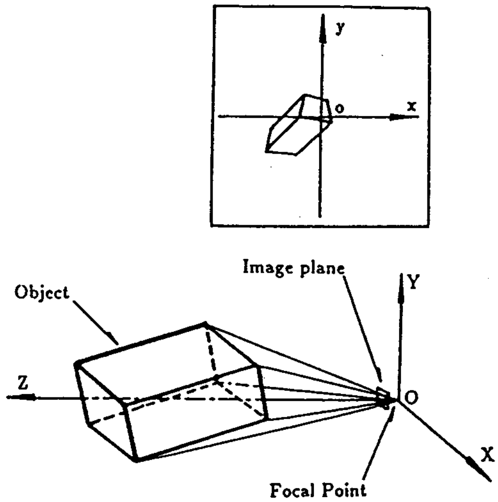


Fig. 7. Simulation model.

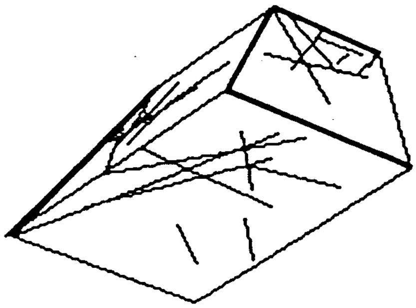
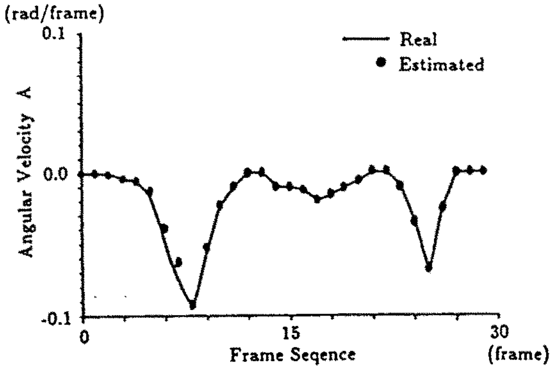
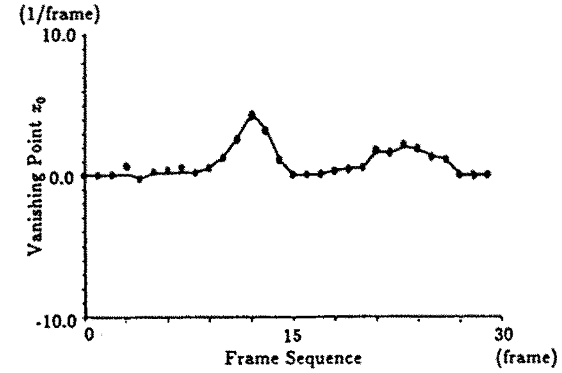


Fig. 8. An example of selected parallel lines.



(a) Angular velocity A



(b) Vanishing point x_0

Fig. 9. Simulation result.

vertical axes are angular velocity and translational velocity between frames.

5. Conclusions

We have presented an algorithm for finding motion parameters of a rigid body in the 3-D space using the properties of line segments flow and parallel line segments. In the future we will attempt to improve estimation precision of continuous motion by adding an assumption that motion parameters change smoothly and to improve our algorithm to accommodate the case in which motion between frames is large.

To apply the algorithm presented in this paper to real images, we need to find line segment flow, which can be obtained by combining a method concerning extraction of optical flow with one of edge extraction. That is, as shown in Fig. 1, velocity field on a line segment changes monotonically and has a different directional vector for each line segment. Therefore, we think that line segment flow can be obtained by grouping after making optical flow (velocity field) obtained in an image correspond to line segments (edges).

Although we described estimation of motion parameters in this paper, the problem of reconstructing a 3-D structure follows. Using the algorithm, a directional vector of each line segment can be obtained, and it is easy to check whether they exist in the same plane in a geometrical sense. Thus, it seems to be effective for reconstruction of surface structures.

REFERENCES

1. D. Marr. VISION: A Computational Investigation into the Human Representation and Processing of Visual Information. W. H. Freeman and Company (1982).
2. Tadahiro Kitahashi (Ed.). Special issue on image understanding. J. of the Japanese Society for Artificial Intelligence, Vol. 4, 1, pp. 2-51 (1989).
3. 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).
4. S. Ullman. The Interpretation of Visual Motion. MIT Press (1979).
5. 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).
6. B. L. Yen and T. S. Huang. Determining 3-D motion and structure of a rigid body using straight line correspondence. NATO ASI Series, F2, pp. 371-394 (1983).
7. Y. Liu and T. S. Huang. Estimation of rigid body motion using straight line correspondence. Comput. Vision, Graphics, and Image Proc., 43, pp. 37-52 (1988).
8. A. Mitichi, S. Seida, and J. K. Aggarwal. Line-based computation of structure invariance. Proc. IEEE Computer Society Workshop on Motion, pp. 175-180 (1986).
9. O. D. Faugeras, F. Lustman, and G. Toscani. Motion and structure from point and line matches. Proc. 1st Int. Conf. Computer Vision, pp. 25-34 (1987).

10. 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).
11. H. C. Longuet-Higgins and K. Prazdny. The interpretation of moving retinal image. Proc. Roy. Soc. London, B-208, pp. 385-397 (1980).
12. T. W. Williams. Depth from camera motion in a real world scene. IEEE Trans. Pattern Anal. and Machine Intell., PAMI-2, 6, pp. 511-516 (1980).
13. M. Subbara. Interpretation of Visual Motion—A Computational Study. Morgan Kaufman Pub. (1988).

AUTHORS (from left to right)




Takuto Joko graduated from Second Group (Electrical Engineering Group), Faculty of Engineering, Hiroshima University, in 1988. Currently, he is in the Master's program of the graduate school of the University. He is engaged in research on computer vision, somatological engineering, etc. He is a student member of the Society of Instrumental and Control Engineers.

Koji Ito obtained a Master's degree from Nagoya University in 1969, and he also has a Dr. of Eng. degree. In 1970 he joined Automatic Control Laboratory of the faculty as a Research Associate. Since 1979 he has been an Associate Professor in Second Group (Electrical Engineering Group), Faculty of Engineering, Hiroshima University. His research interests include man-machine interface. He is a member of the Society of Instrumental and Control Engineers; Institute of Electrical Engineers of Japan; Japan Robot Society; Japan ME Society; and IEEE.

Toshio Tsuji obtained a Master's degree in System Engineering from Hiroshima University in 1985. He also has a Dr. of Eng. degree. Since 1985 he has been a Research Associate on the Faculty of Engineering of the University. He is engaged in research on somatological engineering, robot engineering, and man-machine systems. He is a member of the Society of Instrumental and Control Engineers; Institute of Electrical Engineers of Japan; Japan Robot Society; The Japan Ergonomics Research Society; and IEEE.

Mutsuhiro Terauchi graduated from the Department of Mechanical Engineering, Faculty of Engineering, Kansai University, in 1976, obtained a Master's degree in 1984, and in 1989 obtained the necessary units in the doctoral course of the graduate school of Hiroshima University. In that year he became a Research Associate of the University. Currently he is engaged in research on computer vision and image understanding. He is a member of AVIRG; IEEE; Information Processing Society of Japan; The Japan Ergonomics Research Society; and Japan Society for Mechanical Engineering.