searches,'' *J. Amer. Soc. Information Sci.*, vol. 31, pp. 240-247, 1985.

[9] D. A. Buell, ''An analysis of some fuzzy subset applications to information retrieval systems,'' *Fuzzy Sets, Syst.*, vol. 7, pp. 35-43, 1982.

[10] R. Goetschel and W. Voxman, ''Optimal clustering in graphs with clustered edges,'' *Information Sci.*, vol. 38, no. 1, pp. 13-20, Jan. 1987.

[11] N. Gouvernet, ''Approximate reasoning in medical genetics,'' in *Fuzzy Sets and Possibility*. New York: Pergamon, 1982, pp. 524-530.

[12] H. Heaps, *Information Retrieval and Theoretical Aspects*. New York: Academic, 1978.

[13] A. Kandel, *Fuzzy Mathematical Techniques With Applications*. Reading, MA: Addison-Wesley, 1986.

[14] P. Kantor, ''The logic of weighted queries,'' *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, Dec. 1981.

[15] S. Kleene and C. Kleene, ''On a notation for ordinal numbers,'' *J. Symbol. Log.*, vol. 3, no. 4, pp. 150-155, 1938.

[16] D. H. Kraft and D. A. Buell, ''Fuzzy sets and generalized Boolean retrieval systems,'' *Int. J. Man-Machine Studies*, vol. 19, pp. 45-56, 1983.

[17] F. Lancaster, *Vocabulary Control For Information Retrieval*, 2nd ed. Arlington, VA: Information Retrieval Press, 1986.

[18] F. Lancaster and E. Fayen, *Information Retrieval On-Line*, Melville, LA, 1973.

[19] J. Lukasiewicz, ''Many valued systems of propositional logic,'' in *Plish Logic*, S. McCall, Ed. New York: Oxford Univ. Press, 1930, 1957.

[20] M. Markel, *Finding And Using Information*. (chapter 4). New York: St. Martin's Press, 1984, ch. 45, p. 38.

[21] K. Menger, ''Statistical metrics,'' *Proc. Nat. Acad. Sci. USA*, vol. 28, pp. 2001-2003, 1942.

[22] M. Mizumoto, ''Some Methods of Fuzzy Reasoning,'' in *Advances in Fuzzy Set Theory and Applications*. Amsterdam, The Netherlands: North-Holland, 1979, pp. 117-136.

[23] ——, ''Fuzzy inference using max composition in the compositional rule of inference,'' in *Approximate Reasoning In Decision Analysis*, Gupta and Sanchez, Eds. Amsterdam, The Netherlands: North-Holland, 1982, pp. 67-76.

[24] C. V. Negoita, ''On the application of the fuzzy set separation theorem for automatic classification in information retrieval systems,'' *Information Sci.*, vol. 5, pp. 279-286, 1973.

[25] W. Pedrycz, ''Applications of fuzzy relational equations for methods of reasoning in presence of fuzzy data,'' *Fuzzy Sets, Syst.*, vol. 16, pp. 163-176, 1985.

[26] T. Radecki, ''Fuzzy set theoretical approach to document retrieval,'' *Information Process., Manage.*, vol. 15, pp. 247-259, 1979.

[27] ——, ''On the inclusiveness of information retrieval systems with documents indexed by weighted descriptors,'' *Fuzzy Sets, Syst.*, vol. 5, pp. 159-176, 1981.

[28] ——, ''Foundations Of Fuzzy Information Retrieval,'' in *Management Decision Support Systems Using Fuzzy Sets and Possibility Theory*, J. Kacprzyk and R. R. Yager, Eds. Cologne: Verlag TUV Rheinland, 1985, pp. 70-82.

[29] H. Raiffa, *Decision Analysis Introductory Lectures on Choices Under Uncertainty*. Reading, MA: Addison-Wesley, 1968.

[30] H. Reichenbach, ''Wahrscheinlichkeitslogic,'' *Erkenntnis*, vol. 5, pp. 37-43, 1935-1936.

[31] J. S. Ro, ''An evaluation of the applicability of ranking algorithms to improve the effectiveness of full-text retrieval. I. On the effectiveness of full-text retrieval,'' *J. Amer. Soc. Information Sci.*, vol. 39, no. 2, pp. 73-78, Mar. 1988.

[32] G. Salton, *Introduction To Modern Information Retrieval*. New York: McGraw-Hill, 1983.

[33] G. Salton and M. E. Lesk, ''Computer evaluation of indexing and text processing,'' *J. Assoc. Comput. Mach.*, vol. 25, pp. 8-36, 1968.

[34] K. Schmucker, *Fuzzy Sets, Natural Language Computations, And Risk Analysis*. Rockville, MD: Computer Science Press, 1984, pp. 5-19.

[35] C. Stafill and B. Kahle, ''Parallel free-text search on the connection machine system,'' *ACM Commun.*, vol. 29, no. 12, Dec. 1986.

[36] C. Suen and Q. Wang, ''Analysis and design of decision tree based on entropy reduction and its application to large character set recognition,'' *IEEE Trans. Pattern Analy. Machine Intell.*, vol. PAMI-6, pp. 406 417, July 1984.

[37] ——, ''Large tree classifier with heuristic search and global training,'' *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, Jan. 1987.

[38] D. Swanson, ''Searching natural language text by computer,'' *Science*, vol. 132, pp. 1099-1104, 1960.

[39] Q. Takle, ''STAIRS search strategy: Ideas and opinions,'' *Online Rev.*, vol. 4, pp. 163-168, June 1980.

[40] R. Tong and P. Bonissone, ''A linguistic approach to decision making with fuzzy sets,'' *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, Nov. 1980.

[41] R. Tong and D. Shapiro, ''Experimental investigation of uncertainty in a rule-based system for information retrieval,'' *Int. J. Man-Machine Studies*, vol. 22, pp. 265-282, 1985.

[42] ——, ''RUBRIC: An environment for full-text information retrieval,'' in *Proc. 8th Int. ACM Conf. R & D Information Retrieval*, Montreal, Canada, June 1985.

[43] R. R. Yager, ''Linguistic models and fuzzy truths,'' *Int. J. Man-Machine Studies*, vol. 10, pp. 483-494, 1978.

[44] ——, ''A logical bibliographic searcher: An application of fuzzy sets,'' *IEEE Trans. Syst., Man, Cybern.*, vol. 10, pp. 51-53, 1980.

[45] ——, ''Weighted queries in information retrieval systems,'' *J. Amer. Soc. Information Sci.*, vol. 38, no. 1, pp. 23-24, 1987.

[46] L. A. Zadeh, ''Fuzzy sets,'' *Information, Control*, vol. 8, pp. 338-353, 1965.

[47] ——, ''Outline of a new approach to the analysis of complex systems and decision processes,'' *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 28-44, 1973.

[48] ——, ''The concept of a linguistic variable and its application to approximate reasoning,'' *Information Sci.*, vol. 8, pp. 199-249 and pp. 301-357, 1975.

[49] ——, in *Fuzzy Automate And Decision Process*, Gupta, Ed. New York: North-Holland, 1977.

[50] ——, ''A theory of approximate reasoning,'' *Machine Intelligence*, vol. 9. New York: Elsevier, 1979, pp. 149-194.

[51] ——, ''The role of fuzzy logic in the management of uncertainty in expert systems,'' in *Fuzzy Sets and Systems*, pp. 199-277, 1983.

[52] R. Zenner, ''A new approach to information retrieval systems using fuzzy expressions,'' in *Fuzzy Sets and Systems*, vol. 17, pp. 9-22, 1985.

Egomotion Determination Through an Intelligent Gaze Control Strategy

Matthew J. Barth and Saburo Tsuji

***Abstract*—We present a computationally inexpensive method that rapidly and robustly determines both the translational direction and rotational component of motion through the use of an active vision sensor. Our method employs an intelligent gaze control strategy, where an active camera first *fixates* on an item in the environment while simultaneously measuring motion parallax. The camera then rapidly *saccades* to a different fixation point, based on this measure. The algorithm iteratively seeks out fixation points that are closer to the translational direction of motion, rapidly converging so that the camera always points in the instantaneous direction of motion. At that point, the tracking motion of the camera is equal but opposite in sign to the mobile entity's rotational component of motion.**

Manuscript received January 18, 1992; revised February 11, 1993. This work was supported by the Japan Society for the Promotion of Science (JSPS).

M. J. Barth is with the College of Engineering, University of California, Riverside, Riverside, CA 92521.

S. Tsuji is with the Department of Control Engineering, Osaka University, Toyonaka, Osaka 560, Japan.

IEEE Log Number 9209682.

0018-9472/93$03.00 © 1993 IEEE

I. Introduction

Egomotion determination is important to any mobile entity (e.g., humans, robots) when moving within and interacting with its environment. When the egomotion parameters of direction and rotation are known, short-term navigational control can be accomplished and tasks such as determining environmental depth and structure are possible. A rich source of egomotion information is provided through visual stimulus when an observer moves in space. By observing the environment, one can follow desired paths and avoid collisions with obstacles.

Much work in visual motion analysis has concentrated on recovering general 3-D motion and scene structure from passively acquired 2-D image sequences (review given in [1]). For several reasons, this is a difficult problem to solve, particularly when the motion and scene are arbitrarily complex. While techniques have been proposed that yield the right answer for noiseless data, it has been shown that perturbing the noiseless data will result in a solution quite different from the one that is correct (e.g., [2]). These techniques are also computationally complex, making them inadequate for real-time processing.

Since determining general 3-D motion via vision is problematic, much research has been done concerning only transitional motion [3], [4]. When a visual sensor translates without rotation, all of the flow vectors in the time-varying images emanate from a single point known as the focus of expansion (FOE). This point can be determined from the intersection of the measured flow vectors, and can then be used to determine the translational motion vector of the visual sensor. Many algorithms have been developed that use the FOE, such as determining depth to scene points via the time-to-adjacency relationship [4], [5]. In short, the FOE can provide reliable motion information as long as the motion is purely translational. However, if even a small amount of rotation occurs during translational motion, the accuracy of the FOE (if one can be found) deteriorates. Effects of rotation on the FOE have been given elsewhere, the general conclusion being that translational motion determined from an FOE is highly inaccurate when rotation occurs [4].

In order to robustly determine both the translational direction and rotational component of motion in a rapid fashion, we present a computationally inexpensive method that exploits the constraints imposed by actively controlling the visual sensor's motion independently of the observer's general motion. Due to the advent of sophisticated controllable visual hardware, much emphasis has been placed on the "active" or animate vision paradigm in computer vision research (principally described in [6]–[10]). As part of the active vision paradigm, it is possible to control the gaze direction in order to fixate on items in the environment while the observer moves. Fixating on a particular point in the environment while moving induces a sensation of visual depth referred to as motion parallax. During motion while fixating, our method first measures motion parallax near the line of sight. Based on this measure, the method then performs "saccadic" movement, i.e., rapid movement of the camera in order to select a new fixation point. The algorithm operates in an iterative fashion, repeatedly selecting and tracking fixation points, followed by saccading in the direction of the observer's motion. The algorithm quickly converges so that the gaze direction coincides with the instantaneous translational direction. At that point, the tracking motion of the camera is equivalent to the negative of the rotational components of motion. This algorithm operates continuously, always keeping the gaze aligned with the forward motion, providing accurate egomotion parameters. A beneficial side effect of this method is that obstacle detection will have a higher degree of success, since the gaze is always aligned with the instantaneous forward direction of motion.

Our strategy is very similar to the *optokinetic nystagmus* behavior in humans. This reflex in human vision consists of two parts. First, as we move in our environment, our eyes tend to select a point in the scene and track that point as we move forward. Through the use of our field-holding reflex, we are able to fixate rather accurately. As the fixation point reaches a point where mechanically we cannot continue tracking, we perform a saccadic movement that moves the head/eyes rapidly in the direction opposite of the original tracking motion. A new fixation point is then selected and tracked as before. Humans perform this behavior regardless of whether the path is linear or curvilinear. It was suggested by Cutting [11] that motion parallax is used by humans during fixation for wayfinding. We expound on this idea, and apply it to egomotion determination using an intelligent gaze control strategy. Experiments are carried out both in simulation and in the real world, giving results that are close to the actual motion parameters.

II. Preliminary Mathematics

Consider a *motion* reference frame attached to a moving body as it moves through a stationary environment. In this reference frame we consider a coordinate system whose z-axis is always aligned with the instantaneous direction of translation, as shown in Fig. 1. We then consider a point **P** whose coordinates in space (with respect to the motion reference frame) are $\mathbf{r} = (X, Y, Z)$. If the motion of the body has instantaneous translation given by $\mathbf{t} = (0, 0, W)$[1] and instantaneous rotation given by $\omega = (A, B, C)$, then the velocity vector of point **P** is given as (e.g., from [12]):

$$\mathbf{V} = -\mathbf{t} - \omega \times \mathbf{r} \quad \text{or} \quad \dot{X} = -BZ + CY,$$
$$\dot{Y} = -CX + AZ, \quad \dot{Z} = -W - AY + BX \tag{1}$$

These velocity equations are more easily generalized in spherical coordinates, so we note in Fig. 1 that

$$\theta = \tan^{-1} \frac{\sqrt{X^2 + Y^2}}{Z}, \quad \phi = \tan^{-1} \frac{Y}{X}, \quad L_r = \sqrt{X^2 + Y^2 + Z^2} \tag{2}$$

where L_r is the length of vector **r**.

We are primarily interested in the angular velocity component of **P** *away* from the instantaneous direction of motion, i.e., $\dot{\theta}$. By differentiating above, we obtain

$$\dot{\theta} = \frac{XZ}{L_r^2 \sqrt{X^2 + Y^2}} \dot{X} + \frac{YZ}{L_r^2 \sqrt{X^2 + Y^2}} \dot{Y} - \frac{X^2 + Y^2}{L_r^2 \sqrt{X^2 + Y^2}} \dot{Z}. \tag{3}$$

Also from Fig. 1, we see that

$$X = L_r \sin\theta \cos\phi, \quad Y = L_r \sin\theta \cos\phi, \quad Z = L_r \cos\theta \tag{4}$$

and substituting into (3) we get

$$\dot{\theta} = \frac{\cos\theta \cos\phi}{L_r} \dot{X} + \frac{\cos\theta \sin\phi}{L_r} \dot{Y} - \frac{\sin\theta}{L_r} \dot{Z}. \tag{5}$$

Further, if we substitute (1) and (4) into (5) we get

$$\dot{\theta} = \frac{W \sin\theta}{L_r} + A \sin\phi - B \cos\phi. \tag{6}$$

[1] Note that the first two terms are zero since the z-axis is always aligned with the translational direction.

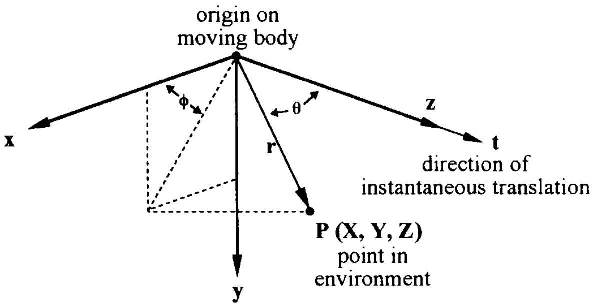


Fig. 1. Motion reference frame with z-axis aligned with the direction of instantaneous translation.

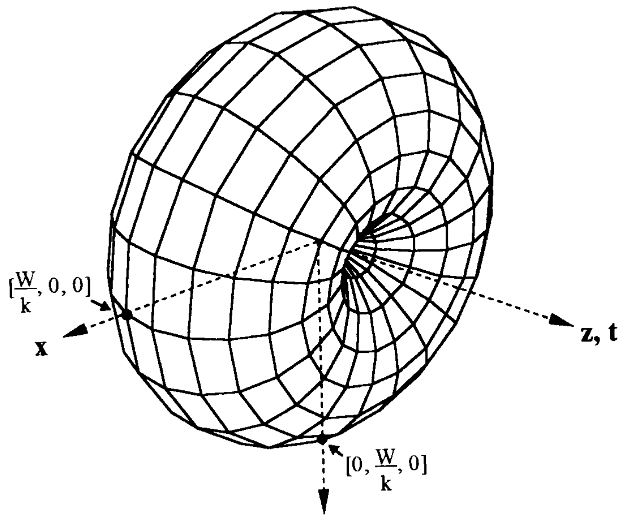


Fig. 2. Three-dimensional surface given by (7) when $A = B = 0$ (torrus with an inside radius of zero).

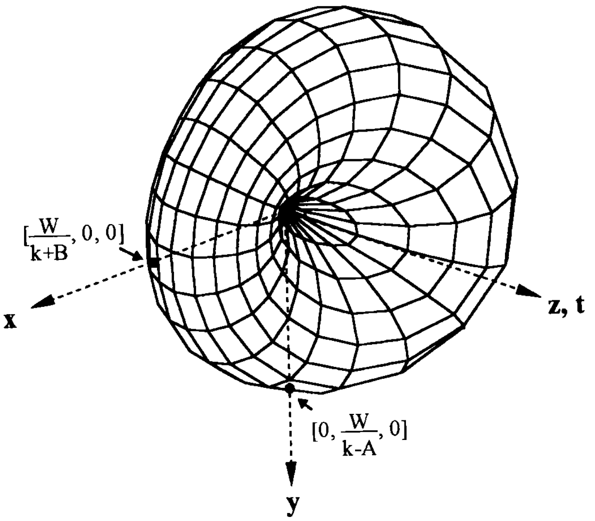


Fig. 3. Three-dimensional surface given by (7) when $A = 1$, $B = 2$, and $k = 5$ (deformed torrus with an inside radius of zero).

We now want to consider all the points that have equal angular velocity away from the instantaneous translational direction. We set $\dot{\theta}$ constant in (6), and solve for L_r as a function of θ, ϕ, and the motion parameters to get

$$L_r = \frac{W \sin \theta}{k - A \sin \phi + B \cos \phi} \qquad (7)$$

where k is a constant. When there is no rotation component of motion (i.e., $A = B = 0$), the solution to this equation is shown in Fig. 2. We see that this surface corresponding to constant angular velocity away from the translational direction is a torrus with an inside radius of zero.

Consider now a camera placed at the center of the motion reference frame so that its focal origin coincides with the coordinate system's origin. If the camera can pivot around its focal origin while maintaining a relatively small field of view, we can consider the imaging domain to be spherical.[2] Then, the optical flow on the imaging surface (when measured as angular displacements) will directly correspond to the angular velocities of the points in space. Therefore, the surface described by (7) also represents the angular optical flow (with respect to the translational axis) of the imaged points on a spherical image domain. Because the surface corresponds to points of equivalent angular velocity away from the instantaneous axis of translation, we can also state that the optical flow of these points are equivalent, and therefore we refer to this surface as an iso-flow surface.

When there is a rotational component of motion, the iso-flow surface becomes a "deformed" torrus,[3] depending on the values of A and B. An example of such an iso-flow surface is shown in Fig. 3. The inside diameter of this deformed torrus is still zero and any cross section cut by a perpendicular plane that contains the origin (i.e., a plane given when ϕ is constant) will show circular surface boundaries.

When we vary k in (7), we obtain a set of nested toroids that all intersect at the origin. It is difficult to illustrate these toroids in three dimensions, so we shall now consider a perpendicular cross section of these iso-flow surfaces, i.e., an intersection of the iso-flow surfaces with a plane given when ϕ is constant. If we choose $\phi = 0$, the intersecting plane will be the x-z plane. In this plane, we see that the 3-D surfaces project as nested circles that are tangent to the instantaneous direction of motion, as shown in Fig. 4. The optical flow of the points in this plane depend on the values of k and the rotational component of motion,[4] given by B. *Regardless of the amount of rotation, the circular shape of these iso-flow contours remains the same* (note that these two-dimensional nested circles have been mathematically derived in other work concerning optical flow, e.g., [11]).

Motion without Fixation

In Fig. 4(a), we consider the case when the observer's motion is purely translational. We will distinguish now between positive and

[2] An "active" vision system, based on a camera that can pan and tilt, closely approximates a visual sensor with a spherical imaging domain [15].

[3] Note that rotation around the translational axis, i.e., C, has no effect on the shape of the surface.

[4] If we consider the y-z cross-sectional plane ($\phi = \pi/2$), the optical flow values will depend on the rotation A; when considering any other cross-sectional plane, the values of optical flow depend on some combination of A and B.

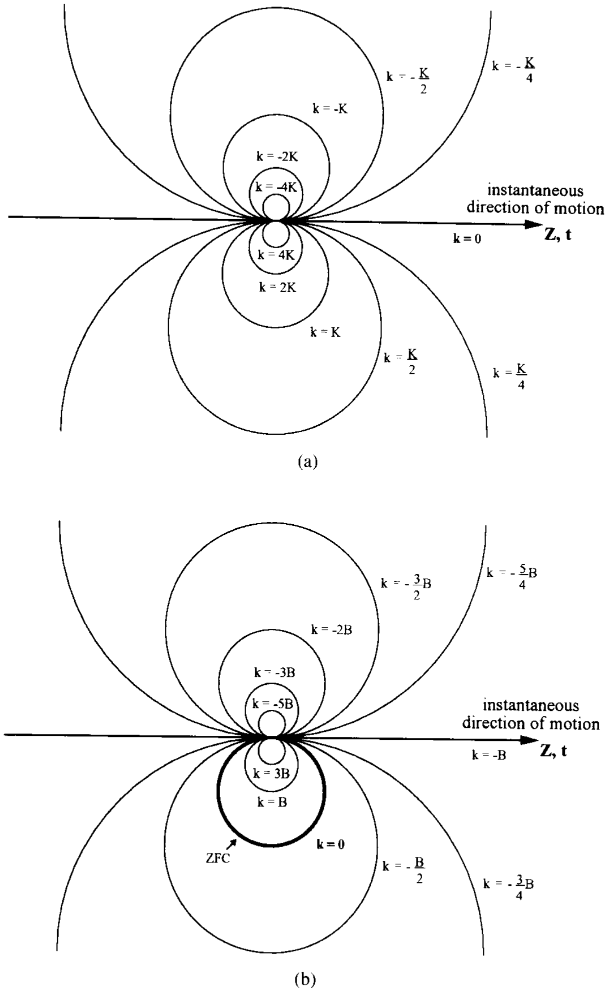


Fig. 4. (a) Iso-flow contours while gazing straight ahead for translation only in the x-z plane; $t = (U, 0, W)$ (note K is some constant). (b) Iso-flow contours while gazing straight ahead for translation and rotation in the x-z plane; $t = (U, 0, W)$, $w = (0, B, 0)$.

negative optical flow with respect to the direction of motion (z-axis). When gazing straight ahead (along the z-axis), optical flow on one side will flow to the left, and the optical flow on the other side will flow to the right. If we consider optical flow positive in the direction of the x-axis, then the optical flow on the left side of the z-axis will be negative, and on the right, positive.

We see that when $k = 0$, i.e., when the optical flow is zero, all of the environmental points with zero optical flow lie on a straight line pointed in the direction of translational motion. This makes sense from an FOE standpoint, since this line will intersect the imaging surface at a single point, i.e., the FOE (or focus of contraction (FOC) if viewing away from the forward direction).

In Fig. 4(b), the observer undergoes both translational and rotational motion. We now see that all of the points with zero optical flow lie on a circle, which has been referred to in other work as the zero flow circle (ZFC) [13]. From this we can see that since the points on the ZFC will project on to the imaging surface at many different points, a single FOE point will not exist. It is also important to note that all of the points inside the ZFC will have optical flow values opposite in sign from the optical flow values outside the ZFC. Also, points that fall on the line of instantaneous translation will have optical flow equal but opposite in sign from the rotational component of motion (in this case, $-B$).

Motion with Fixation

Consider now an observer moving while fixating on a particular point in the environment. The observer maintains the image of the fixation point in the center of view, and thus the optical flow of the corresponding image point will be zero. Keep in mind, however, that the fixation point will be one of the many points that has zero optical flow. As described above, the points that have zero optical flow lie on a circle, the ZFC. Therefore, the shape of this iso-flow contours will remain the same during fixation, however, the values of optical flow along the contours change according to the location of the fixation point. The contour that contains the fixation point will have zero optical flow and the contours inside the ZFC will have optical flow opposite in sign from that of the contours outside the ZFC.

In order to maintain fixation, the observer must add a component of rotation to the visual sensor. Humans do this by rotating their eyes and/or head while under going general motion. In a sense, the observer "forces" the iso-flow contour containing the fixation point to have zero optical flow. If the observer's general motion is only translational and the observer is fixating on an object in the environment, then the final motion of the visual sensor has both translational and rotational components. If the observer is moving both translationally and rotationally while simultaneously fixating, then the final motion of the visual sensor contains a rotational component composed of both the original rotation and the rotation required in order to keep the fixation point at the view center.

III. Motion Parallax

Motion parallax (also called kinetic depth) is the sensation of visual depth obtained by a moving observer while fixating on a point in the visual scene. Objects in front of the fixation point move in the direction opposite of the observer movement, while objects behind the fixation point move in the same direction.[5] Motion parallax has been researched extensively by psychologists studying animate vision since motion parallax is used as one of the cues for depth perception (e.g., [14], [15], [16]). Although motion parallax is useful in determining *relative* depth, it cannot be used to determine *absolute* depth.

A general case of motion parallax is illustrated in Fig. 5. We consider an observer at two positions along a line of translation, position p_1 and p_2. The observer maintains his gaze on the fixation point during motion, therefore the optical axis of the visual sensor always intersects point Z. We also consider a point A and a point B that lie in front and behind the fixation point Z, respectively. The image motion of point A is given by angle α, and of point B by angle β. What we wish to show is how α and β change as the angle θ formed by the line of sight through the fixation point and the line of translation changes. If we define Z_1 and Z_2 as the distances from the fixation point to the observer at positions p_1 and p_2, respectively, then we can write Z_2 in terms of Z_1 using the Law of Cosines:

$$Z_2 = \sqrt{d^2 + Z_1^2 - 2dZ_1 \cos\theta}. \qquad (8)$$

[5] This is clearly understood from the previous section since the optical flow inside the ZFC has opposite sign from the optical flow outside the ZFC.

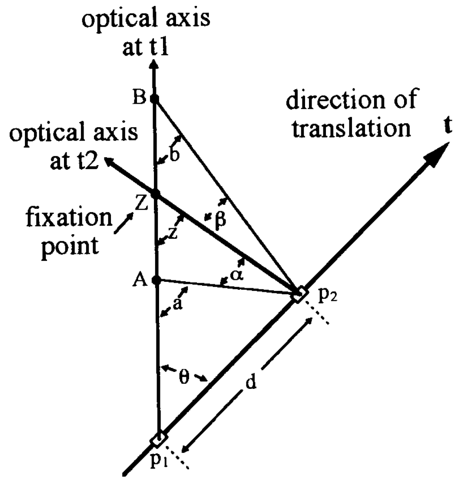


Fig. 5. Motion parallax given by angles α and β.

From the Law of Sines, we know that

$$\frac{\sin\theta}{Z_2} = \frac{\sin z}{d}. \tag{9}$$

Combining these two equations, we can express angle z in terms of d, θ, and Z_1:

$$z = \sin^{-1}\left[\frac{d\sin\theta}{\sqrt{d^2 + Z_1^2 - 2dZ_1\cos\theta}}\right]. \tag{10}$$

In a similar fashion, we can determine equations for the angles a and b:

$$a = \sin^{-1}\left[\frac{d\sin\theta}{\sqrt{d^2 + A_1^2 - 2dA_1\cos\theta}}\right] \tag{11}$$

$$b = \sin^{-1}\left[\frac{d\sin\theta}{\sqrt{d^2 + B_1^2 - 2dB_1\cos\theta}}\right]. \tag{12}$$

Finally, to determine α and β, we see that

$$\alpha = a - z, \quad \beta = z - b. \tag{13}$$

We set the values of A_1, Z_1, B_1, and d constant, and plot α, β, and their difference $\alpha - \beta$ for θ ranging from 180° to 0°, shown in Fig. 6. From this we can verify that the motion parallax of points A and B given by α and β is zero when $\theta = 0°$ or 180°, i.e., when the line of sight is directed along the line of translation. Further, both α and β go through a maximum near 90°. The functions $\alpha(\theta)$ and $\beta(\theta)$ act very much like the sine function, and indeed, when d is relatively small compared to A_1, Z_1, and B_1, $\alpha(\theta)$ and $\beta(\theta)$ can be approximated by

$$\alpha(\theta) = \sin^{-1}\left[\frac{Z - A}{AZ}\right] d\sin\theta; \qquad d \ll A, Z, B \tag{14}$$

$$\beta(\theta) = \sin^{-1}\left[\frac{B - Z}{BZ}\right] d\sin\theta; \qquad d \ll A, Z, B. \tag{15}$$

It is important to note that the similarity between (14) and (15) with (7), the equation that describes an iso-flow surface in three dimensions. When the rotational components A and B are zero, (7) consists of the sine of θ multiplied by a constant; (14) and (15) have the same form. If we plot motion parallax given by (14) or (15) in three dimensions, we see that the surface is the same as depicted in Fig. 2. This makes sense, since motion parallax is independent of rotational motion. We see that motion parallax is zero along the axis of instantaneous translational motion and maximal when gazing perpendicular to the translational axis.

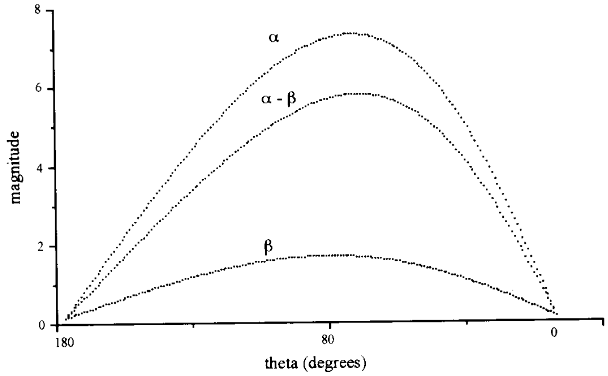


Fig. 6. $\alpha(\theta)$, $\beta(\theta)$ and $\alpha - \beta$ versus θ from (13) ($A = 5$, $Z = 13$, $B = 20$, $d = 1$).

Another key point in Fig. 6 and (14) and (15) is that the amplitude of α is greater than the amplitude of β for all values of θ. That is, points nearer to the observer than the fixation point will flow in the direction opposite to the observer motion, and because they are closer, they will have greater magnitude than the flow of points behind the fixation point. Cutting suggested that humans use this information for wayfinding [11]. By grouping the flow vectors moving in one direction from a group flowing in the opposite direction and by calculating the average magnitude of flow of each group, it is possible to determine the direction of observer motion. If a group of flow vectors has greater magnitude than the other group, then the observer movement is in the direction opposite to the motion of the group of greater magnitude. However, there are cases where this concept fails. If the closer group of points are close to the fixation point, and the farther group of points are far from the fixation point, then the magnitude of the closer group shall be smaller than the distant group. This problem can be rectified through an intelligent choice of fixation points. By choosing a fixation point so that there are ample scene points between the observer and fixation point, and more specifically, that there are points closer to the observer than the halfway mark to the fixation point, then the problem is eliminated.

IV. Egomotion Determination Algorithm

General Strategy

In order to determine the direction and rotational components of general motion for a mobile entity such as a mobile robot, we propose a method where the visual sensor's motion is actively controlled. Our method makes use of two different kinds of motion for the camera. We first choose a point in the environment and *fixate* on it while we move a short distance. During fixation, we measure the motion parallax that occurs from feature points that lie on or near the line of sight. Based on the measure of motion parallax, we then *saccade* the camera, selecting a new view direction. A new fixation point is then selected, and the process repeats. The algorithm converges so that the visual sensor is always pointed along the direction of instantaneous translation, where no motion parallax occurs.

The assumptions of this method are as follows.

1) The mobile entity's motion to be measured must change slowly enough so that the saccadic algorithm is allowed to converge.
2) The environment must be sufficiently rich in objects, so that when choosing fixation points, other object points exist that are closer than half the distance between the visual sensor and the fixation point. This assures that the average motion of objects behind the fixation point moves less than the average motion of objects in front of the fixation point. This can be satisfied through an intelligent choice of fixation points, as is described in the next section.

The steps of the algorithm are as follows.

1) Regardless of where the camera is pointed, we choose a fixation point in the center of the field of view.
2) The fixation point is tracked as the mobile entity moves for a fixed distance. During tracking, we measure the flow of scene points that are imaged near the fixation point. We group the flow vectors into a positive-moving group and a negative-moving group, based on the directions of flow. We then calculate the average magnitude of flow within each group. We refer to the resulting value of the positive-moving group as $|F|_+$ and the value of the negative-moving group as $|F|_-$.
3) A new view direction is calculated, given by

$$\theta_{i+1} = \theta_i - K(|F|_+ - |F|_-) \tag{16}$$

where θ_i is the previous view direction and K is a control constant that determines how quickly the algorithm converges to choosing a fixation point along the instantaneous direction of motion. The value of K depends on the length of the tracking step, the focal length, and a limit determining how close objects may come to the visual sensor (the choice of K is discussed later when we consider convergence and stability).
4) The visual sensor saccades to the new view direction, and the process repeats from step 1).

As the choice of fixation points approaches the line of translation, the $K(|F|_+ - |F|_-)$ term in (16) goes to zero, and the selected view direction will converge so that the visual sensor always points along the line of translation. While tracking fixation points that lie on the line of instantaneous translation, the tracking velocity will be equal but opposite in sign to the mobile entity's rotational velocity. This fact is confirmed in Fig. 4(b), where we see that the optical flow value of the line of instantaneous translation is $-B$ in the case when we consider motion in the x-z plane. We thus know both the direction and rotation of our mobile entity.

Convergence and Stability

Considering the $|F|_+$ and $|F|_-$ values as single points in front and behind the fixation point, we can model the $(|F|_+ - |F|_-)$ term in (16) simply as $\alpha - \beta$, where α and β are given in (14) and (15). Therefore, (16) takes the form

$$\theta_{i+1} = \theta_i - K_1 \sin \theta_i \tag{17}$$

where K_1 is a cumulative constant made up of the constant K in (16), along with the terms A, Z, B, and d from (14) and (15). It can be shown that (17) converges to zero when $K_1 < 90$ for all initial values of θ in the range from $-\pi$ to π. As an example, the convergence is shown when $K_1 = 30$ in Fig. 7. Note the convergence is slower when θ is close to 180°. Even though at 180° the gaze direction is along the line of instantaneous translation, the algorithm will tend to diverge away from that point due to any small disturbing rotational value.

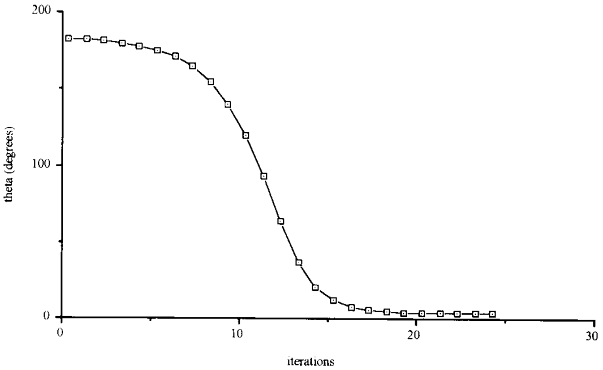


Fig. 7. Convergence for (17) with $K_1 = 30$.

Since we do not know *a priori* the values A, Z, and B for each selected fixation point, we cannot immediately choose an optimum constant K in (16) for convergence. However, we can assure convergence and stability by choosing K with knowledge of the maximum possible length of any flow vector used in determining $|F|_+$ or $|F|_-$. Knowing the geometry of the mobile entity and the attached camera, it is possible to determine how close environmental objects can get to the visual sensor. By knowing the focal length and the closest possible distance of an object in terms of the translational motion during the tracking period (with corresponding movement d), we can determine the maximum possible flow that object will have when imaged. We consider the extreme case when the fixation point is at infinity and we are viewing perpendicular to the direction of motion. K then can be chosen so the saccadic algorithm will always converge:

$$K < \frac{90°}{\text{maxflow}} \tag{18}$$

where maxflow is the maximum largest flow of any object. K should be chosen close to this value for rapid convergence, but should remain less than the value in order to assure stability.

V. Experiments and Results

In order to show the effectiveness of our method, we have carried out both simulated and real-world experiments. In our experiments, we used a mobile robot with an attached active vision system consisting of a video camera that can both pan and tilt. Because our robot only traveled on a flat floor, our motion was limited to motion in the x-z plane, with the rotational components $A = C = 0$. In order to assure the assumption of having objects in between the fixation point and the camera, we directed the view slightly downward. Since there are usually points seen on the ground that are much closer than a chosen fixation point, the assumption was satisfied.

Simulation

The movement of a robot with an active camera was simulated on a Sun workstation in various environments such as a flat floor with objects, an approaching wall, and objects randomly placed in 3-D space. Both linear and curvilinear paths with different degrees of rotation were simulated. In the simulation, the focal length of the camera was set to 300 pixels and the image plane consisted of a 512 × 512 array. The robot moved in one-step units, and the

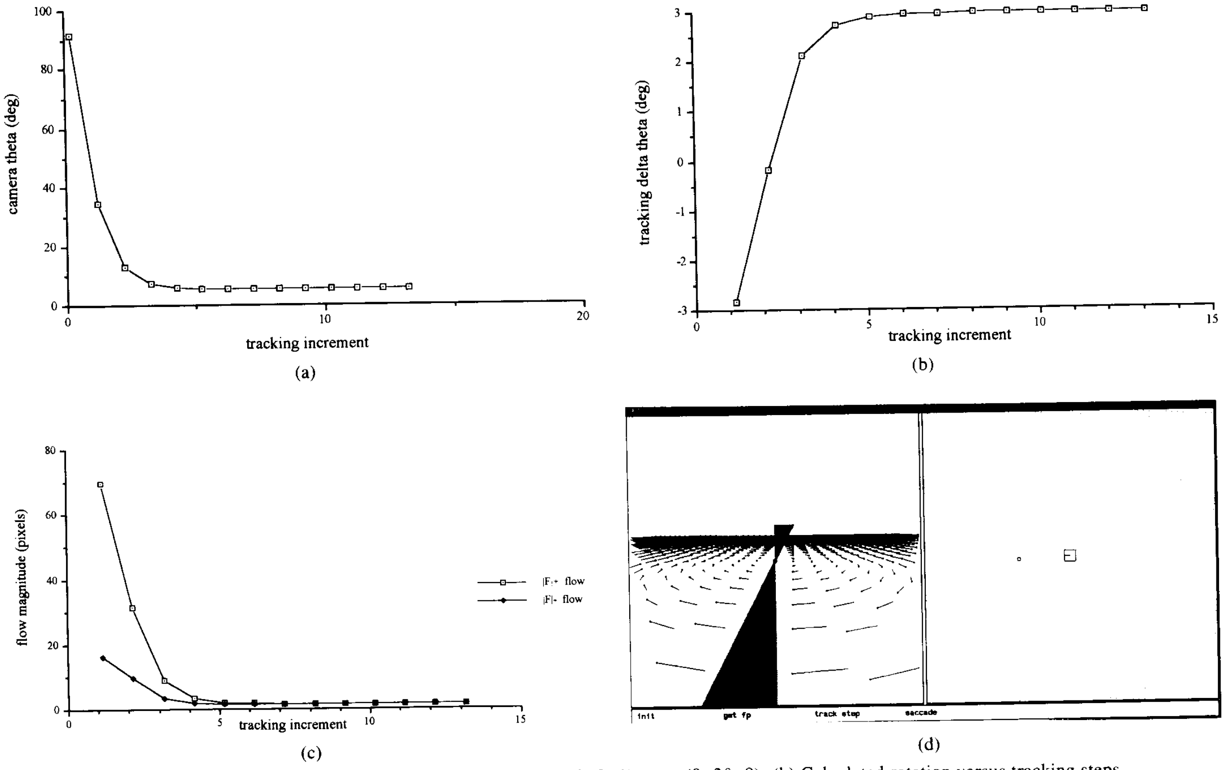


Fig. 8. (a) Angle of camera versus tracking steps for $t = (0, 0, 1)$, $\omega = (0, 3°, 0)$. (b) Calculated rotation versus tracking steps for $t = (0, 0, 1)$, $\omega = (0, 3°, 0)$. (c) Positive-moving flow $|F|_+$ and negative-moving flow $|F|_-$ versus tracking steps for $t = (0, 0, 1)$, $\omega = (0, 3°, 0)$. (d) Flow simulation as the robot moves and tracks a fixation point. The camera is oriented 90° to the direction of translation and the robot rotates 3° counterclockwise. The left frame shows the flow vestors and the motion parallax of the center vertical line. The right frame shows the robot and camera position in the environment.

camera's focal center was 2 units above the floor. In the flat floor simulation, the camera tilt angle ϕ was set to a constant 11° down from the horizontal robot axis; therefore the fixation point generally was selected 10 units away. We allowed no imaged object point to come closer than 2.5 units to the robot, therefore, the maximum flow of an object point was no greater than 85 pixels. For this reason, K was chosen to be equal to one in the simulation.

The algorithm was executed using many different rotational values, including the case of zero rotation resulting in pure translation. In all cases the algorithm converged to within 10 percent of the final correct rotational value in 7 steps or less. We show the flat floor case where the robot rotates 3° counterclockwise for each unit of forward translation in Fig. 8. The initial camera angle starts at 90° to the left of the direction of motion. Fig. 8(a) gives the camera angle relative to the instantaneous direction of translation after each saccadic step. We see that the algorithm rapidly converges to a final camera angle slightly less than 2°. Therefore, during the tracking step, the camera's z-axis crosses the line of translation, resulting in equal angles at the start and end of tacking. Fig. 8(b) shows the tracking delta theta value, which should be equal to the robot rotation component after convergence. We see that convergence occurs after 5 steps, with a final value of 2.9°. Fig. 8(c) shows the values of $|F|_+$ and $|F|_-$ for each algorithm iteration. We see that the $|F|_+$ dominates until convergence when they are roughly equal. Fig. 8(d) shows the simulated view from the camera with the associated motion parallax when the camera tracks at the initial angle of 90°.

Real World

We have carried out the same algorithm in real-world experiments. An active camera was attached to a robot, which moved along the perimeter of a circle whose radius was 430 mm. The camera only rotated around the Y-axis, and had a focal length of 1240 pixels. The camera was tilted down so that the fixation points were chosen on the floor at around 1.5 m. Objects were not allowed to come close enough to the camera so as to generate flow larger than 90 pixels. Therefore, we set $K = 1$ in the saccadic control equation so that the convergence of the algorithm was guaranteed.

The robot moved 5° for each tracking step (38 mm translation). We had initially pointed the camera at −100° from the direction of translation. The flow of objects points near the center vertical line was measured and $|F|_+$ and $|F|_-$ values were determined. Fig. 9 shows the scene observed during the step when the camera was oriented at −85° from the direction of translation. Also in the figure are the flow vectors associated with object points. The fixation point is near the center of the image, identified by a small circle. We see that the near objects have large flow moving right and far objects have small flow moving left.

The results of the convergence are shown in Fig. 10. We see that the algorithm converged to a measure of robot rotation of 4.7°. The measured rotation remained within ±0.3° of this value. In Fig. 10(c), we see the relative values of $|F|_+$ and $|F|_-$. At the fourth step, we can see that there was a very close object so as to

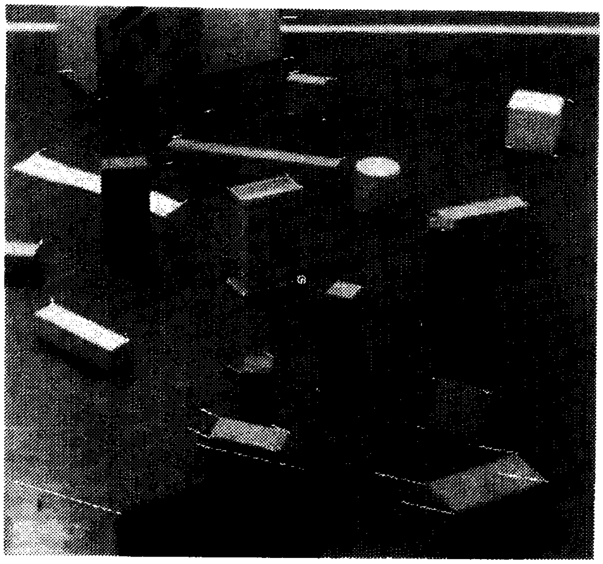
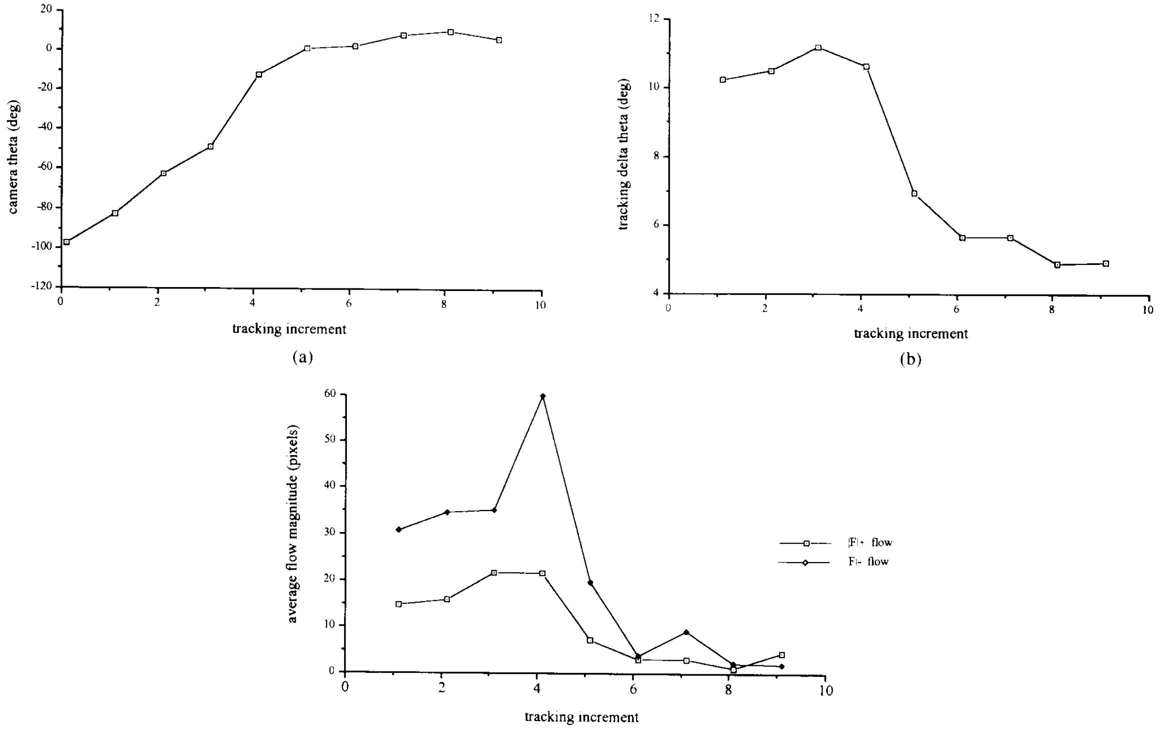


Fig. 9. View of the scene with flow vectors when the camera is oriented at −85° with robot rotation at 5°. The fixation point is seen in the middle of the image indicated by a small circle.



(a)

(b)

(c)

Fig. 10. (a) Angle of camera versus tracking steps for t = (0, 0, 1), ω = (0, 5°, 0). (b) Calculated rotation versus tracking steps for t = (0, 0, 1), ω = (0, 5°, 0). (c) Positive-moving flow $|F|_+$ and negative-moving flow $|F|_-$ versus tracking steps for t = (0, 0, 1), ω = (0, 5°, 0).

give a high $|F|_-$ value. The resulting 6 percent error is due to an accumulation of error in the measure of the tracking delta theta, tracking error of the fixation point, and error in the measure of the flow vectors in determining $|F|_+$ and $|F|_-$.

VI. Discussion and Conclusions

We have developed a computationally inexpensive methodology for determining the egomotion parameters of a mobile entity by using an active camera that tracks fixation points and subsequently performs saccadic motion. Through the use of controlled saccadic motion, the rotational and translational values of motion can be determined after a few iterations of the algorithm. If the motion of the mobile entity changes smoothly along its trajectory, the algorithm will "track" the instantaneous direction of motion, and will provide continuous egomotion values for any type of path. It is important to note that our method simultaneously provides rotation and translation information, as opposed to methods that first determine rotation and then derotate an image in order to find the FOE and thus the direction of translation.

Although experiments were carried out using a mobile robot that moved on a flat floor (essentially 2-D motion), the algorithm is applicable to full 3-D motion. As was shown in Sections II and III, motion parallax occurs when the fixation point falls in any cross-sectional plane perpendicular to the instantaneous direction of translation (i.e., a plane when ϕ is constant). Therefore, the algorithm will converge to the proper direction of translation and will provide rotational information with respect to the visual sensor's coordinate basis. Also, note that this method does not assume that the forward pointing axis of the moving entity coincides with the instantaneous direction of motion. Such an assumption can be made for vehicles with conventional forward wheel steering with no wheel slippage. In this case, a similar but somewhat simpler algorithm can be used when the camera's rotational angle is known with respect to the forward direction of the vehicle [17].

In our experimentation, the limiting factor for calculating the egomotion parameters in real time is the optical flow determination. In order to determine the motion parallax around the line-of-sight, we tracked a small number of feature points (<20) over several frames. This was performed on a Sun 4 workstation with a standard framegrabber and each flow determination step required 10 to 20 s. The evaluation of the saccadic control equation requires only a single multiply and two additions, which can be considered computationally inexpensive when compared to evaluating the nonlinear equations of standard 3-D motion and structure determination techniques used on optical flow. Furthermore, we have encountered a small percentage of error in our experiments. This error could be reduced by improving the angular resolution of our active camera. Also, the algorithm depends greatly on accurate tracking and accurate flow measurements, so errors in the tracking and flow algorithms need to be minimized.

Acknowledgment

This work was carried out during a 17-month research visit at Osaka University by the first author. Many thanks go to the members of the Tsuji Laboratory in the Department of Control Engineering, in particular, to Dr. Hiroshi Ishiguro.

References

[1] H.-H. Nagel, "Images sequences—Ten (octal) years—From phenomenology towards a theoretical foundation," in *Proc. Int. Conf. Pattern Recogn.*, Paris, France, 1986.

[2] S. Ullman, "Recent computational studies in the interpretation of structure from motion," in *Human and Machine Vision*, J. Beck, B. Hope, and A. Rosenfeld, Eds. New York: Academic, 1983.

[3] D. Lawton, "Processing translational motion sequences," *Comput. Graphics, Image Process.*, vol. 22, pp. 116-144, 1983.

[4] R. Dutta, R. Manmatha, E. Riseman, and M. Snyder, "Issues in extracting motion parameters and depth from approximate translational motion," in *Proc. DARPA Image Understanding Workshop*, Cambridge, MA, 1988.

[5] D. Ballard and C. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[6] R. Bajcsy, "Active perception," *Proc. IEEE*, vol. 76, no. 8, pp. 996-1005, 1988.

[7] P. J. Burt, "Smart Sensing within a Pyramid Vision Machine," *Proc. IEEE*, vol. 76, no. 8, pp. 1006-1015, 1988.

[8] R. Bajcsy, "Perception with feedback," in *Proc. DARPA Image Understanding Workshop*, Cambridge, MA, 1988.

[9] D. Ballard, "Reference frames for animate vision," in *Proc. Int. Joint Conf. Artificial Intell.*, Detroit, MI, 1989.

[10] M. Swain and M. Stricker, Eds., "Promising directions in active vision," in *Proc. Nat. Sci. Foundat. Active Vision Workshop*, Chicago, IL, 1991.

[11] J. E. Cutting, *Perception with an Eye for Motion*. Cambridge, MA: MIT Press, 1986.

[12] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.

[13] D. Raviv and M. Herman, "Towards an understanding of camera fixation," in *Proc. IEEE Int. Conf. Robot., Automat.*, Cincinnati, OH, 1990.

[14] H. von Helmholtz, *Physiological Optics*, vol. 3, (Optic. Soc. Amer., 1924. New York: Dover, 1964, p. 295.

[15] J. J. Gibson, *The Perception of the Visual World*. Cambridge, MA: Riverside Press, 1950.

[16] J. J. Koenderink and A. J. Van Doorn, "Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer," *Optica Acta*, vol. 22, no. 9, pp. 773-791, 1975.

[17] M. Barth, H. Ishiguro, and S. Tsuji, "Computationally inexpensive egomotion determination for a mobile robot using an active camera," in *Proc. IEEE Int. Conf. Robot. Automat.*, Sacramento, CA, 1991.

An Invariant Pattern Recognition Machine Using a Modified ART Architecture

Narayan Srinivasa and Musa Jouaneh

Abstract—**A novel invariant pattern recognition machine is proposed based on a modified ART architecture. Invariance is achieved by adding a new layer called F_3, beyond the F_2 layer in the ART architecture. The design of the weight connections between the nodes of the F_2 layer and the cells of the F_3 layer are similar to the invariance net. Computer simulations show that the model is not only invariant to translations and rotations of 2-D binary images but also noise-tolerant to these transformed images.**

I. Introduction

The problem of invariant pattern recognition has interested researchers for a long time. Casasent *et al.* [1] developed a model based on optical correlations to achieve invariance to position, rotation, and scaling of images. Cavanagh [2] proposed a model for size and position invariance in the visual cortex of the brain. Fukushima [3] and Fukushima and Miyake [4] developed a model based on several hierarchically organized processing stages to gradually free image processing from its spatial coordinates. Higher order threshold logic units were used for invariant image processing by Maxwell *et al.* [5]. Szu [6] used holographic coordinate

Manuscript received July 18, 1992; revised February 11, 1993.

N. Srinivasa is with the Department of Mechanical Engineering, University of Florida, Gainesville, FL 32611.

M. Jouaneh is with the Department of Mechanical Engineering, University of Rhode Island, Kingston, RI 02881.

IEEE Log Number 9209683.