TRACKING

TRACKING IN A COMPLEX VISUAL ENVIRONMENT

John (Yiannis) Aloimonos[1] Dimitris P. Tsakiris[2]

(1) Computer Vision Laboratory, Center for Automation Research and
Institute for Advanced Computer Studies and Computer Science Department
(2) Electrical Engineering Department and Systems Research Center
University of Maryland, College Park, MD 20742-3411

Abstract

We examine the tracking of 3-dimensional targets moving in a complex (e.g. highly textured) visual environment, which makes the application of methods relying on static segmentation and feature correspondence very problematic, even under specific assumptions about the trajectory of the target. From the system kinematics and optical flow formalism we derive a general criterion, the *Tracking Constraint*, which specifies the optimal camera reorientation as the one that minimizes the optical flow in the center of the camera in the sense of an appropriate metric. A correspondence-free scheme is devised which employs dynamic segmentation and linear features of the image in order to capture global information about the scene and bypass numerical differentiation of the image intensity.

1. INTRODUCTION

Visual tracking is an important problem in Computer Vision, not only because of its applications in areas such as aircraft and missile tracking, robot manipulation of objects, navigation, traffic monitoring, cloud tracking in meteorology, cell motion and tracking of moving parts of the body (e.g. the heart) in biomedicine, but also because it increases the robustness of Early Vision process solutions [1], allows more efficient use of our sensory and computational resources by focusing them on an area of interest and avoids blurring from the motion of the target by stabilizing it in the center of the camera.

The Vision Module of a tracking system consists of the hardware and software necessary for the processing of the visual information. It gets the analog image, digitizes it, creates an array of intensities for all points of the field of view and from this information locates the target and computes the camera positioning that will achieve tracking. Locating the target is the part of the tracking process referred to as the *Target Acquisition* phase and corresponds in the human oculomotor system to the fixation of a target using saccadic eye movements. The computation and execution of a smooth and continuous

camera motion that will keep the target foveated is called the *Tracking* phase and corresponds in the human visual system to the smooth pursuit eye movements. During this phase, the target tends to drift slowly away from the center of the image and, after some time, it may be lost. Then a new acquisition phase is required.

In the following we will assume that during the acquisition phase the target was brought to the center of the image plane and we will consider the problem of keeping the target foveated. We are mainly interested in the most general instance of the problem, involving a 3-D target moving in 3-dimensional space. Previous approaches [5], [6], [7], [8], [11], [13], [14], [16], [19], [20] to the problem are composed of three main steps: First, they perform segmentation of the image or of appropriately selected parts of it, in order to locate the target and extract the new position of some characteristic points of its image (centroid, feature points, etc.). Second, they match these new characteristic points with their corresponding ones in previous or current (in the binocular case) frames and thus extract information about the displacement of the target. Third, from this displacement they compute the necessary camera rotation that will achieve tracking.

In our approach we consider a general criterion (one not restricted to a specific target or visual environment) for 3-D tracking, the *Tracking Constraint*, which, if satisfied, guarantees tracking in the sense of keeping an initially foveated target stationary in the center of the camera. This formulates tracking as an Optimization problem and allows the use of powerful tools from Optimization Theory for the solution of the problem. We follow the continuous motion approach and use the concept of optical flow to formulate this constraint. The Tracking Constraint uses global information from entire areas of the image and does not require feature extraction or matching. Therefore it is expected to be more robust in the presence of noise than previous methods. To solve the problem in a correspondence-free manner (without computing the optical flow field), we assume that the shape of the target is known, so that we can estimate the kinematic parameters of its motion. Obviously, a requirement for knowledge of the shape in a general tracking scheme is by far less restrictive than an ad-hoc tracking scheme for a specific target or class of targets, since the latter may be used only for a specific target, while the former may be used for *any* tracking problem where the shape is known exactly or approximately.

Finally, and most importantly, the theory described here considers tracking in 3-D and suggests several research avenues on how tracking interacts with modules such as shape from "x", structure from motion, etc.; such interactions are necessary in our effort to integrate various vision modules [2], [12].

2. COMPUTATIONAL THEORY OF TRACKING

In this section we devise a model of the image formation process and the motion of the target, based on the system kinematics and the optical flow formalism. (For details see [3], [18].)

Optical Flow is the velocity field generated on the image plane from the projection of objects moving in 3-D, from the motion of the observer with respect to the scene or from apparent motion, when a series of images gives the illusion of motion. The image intensity $s(x,y,t)$ at a point (x,y) of the image plane at time t is related to the instantaneous velocity $(u(x,y),v(x,y))$ at that point by the Optical Flow Constraint equation ([9],[17]):

$$s_x(x,y,t)u(x,y) + s_y(x,y,t)v(x,y) + s_t(x,y,t) = 0\,, \tag{1}$$

Let $\vec{T} = (T_1,T_2,T_3)$ and $\vec{\omega} = (\omega_1,\omega_2,\omega_3)$ be respectively the translational and angular velocities of the target at time t, relative to the camera coordinate system (Figure 1). The pinhole camera model and the system kinematics can be used to show that, under perspective projection, the optical flow field (u,v) induced at the point (x,y) of the image plane by the motion of the target, is:

$$u(x,y) = \frac{fT_1 - xT_3}{Z} - \omega_1\frac{xy}{f} + \omega_2\frac{(x^2+f^2)}{f} - \omega_3 y \tag{2}$$

$$v(x,y) = \frac{fT_2 - yT_3}{Z} - \omega_1\frac{(y^2+f^2)}{f} + \omega_2\frac{xy}{f} + \omega_3 x\,. \tag{3}$$

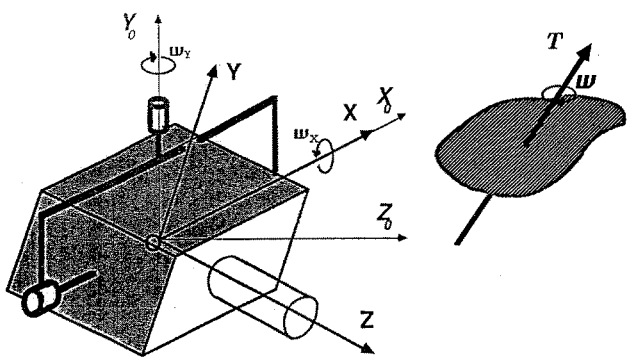

Figure 1.

Since the optical flow expresses the relative motion of the target and the camera, the problem of stabilizing the target at the origin of the image plane by a rotation (ω_x,ω_y) of the camera, can be expressed equivalently as making the optical flow field zero near the origin. When both the camera and the target are moving, the corresponding optical

flow field will be the superposition of two components: One induced by the target (u^t, v^t) and one induced by the camera motion (u^{cam}, v^{cam}). By observing that a rotation of the camera by (ω_x, ω_y) has the same effect on the image formation process as a rotation of the target by $(-\omega_x, -\omega_y)$, both components of the optical flow can be related to the corresponding kinematic parameters by the equ. (2) and (3).

The *tracking problem* can then be set as the specification of the camera rotation (ω_x, ω_y) that will minimize the optical flow field, in the sense of an appropriate metric $(\mathcal{L}_2)$, in a neighborhood $\mathcal{B}$ of the origin of the image plane :

$$\min_{\omega_x, \omega_y} \iint_{\mathcal{B}} (u^2(x,y) + v^2(x,y)) dx\, dy, \tag{4}$$

where $u = u^t + u^{cam}$ and $v = v^t + v^{cam}$. This is called the **Tracking Constraint**. The neighborhood $\mathcal{B}$ is specified by the dynamic segmentation process.

An **algorithm** for tracking the motion of an object moving in 3-D then suggests itself: *Step 1 :* Estimate (u^t, v^t). *Step 2 :* Solve the Tracking Constraint (eq. (4)) and derive the desired camera angular velocities that will achieve tracking.

Typically Step 1 involves the Optical Flow Constraint. If we assume that the optical flow (u^t, v^t) can be computed from the image, then tracking is achieved through minimization of (4) or, if (4) is combined with (1), tracking is achieved through solution of a penalized least-squares problem. The reader interested in this approach is referred to [18]. However, due to the fact that the computation of the optical flow field still remains a very hard problem especially for the case of complex visual environments involving discontinuities, we chose to seek an alternative approach. The next section describes our algorithm for tracking, which is based on the knowledge of the shape of the target, but does not assume knowledge of the optical flow field. Instead, it estimates the kinematic parameters $(\vec{T}, \vec{\omega})$ of the target motion and then it uses equ. (2) and (3) to compute (u^t, v^t) for the first step of the algorithm.

2.1. Estimation of the Three-Dimensional Motion

Suppose that an object exists in $\mathbb{R}^3$, with its surface given as a function $Z = g(X, Y)$ in the camera coordinate system $OXYZ$. The following analysis is done with respect to the camera coordinate frame.

Theorem:

The optical flow (u, v), generated from the motion by $(\vec{T}, \vec{\omega})$ of a rigid target relative to our camera, is a linear function of the target motion parameters, of the form:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathcal{U} \begin{pmatrix} \vec{T} \\ \vec{\omega} \end{pmatrix} = \frac{\partial P}{\partial \vec{X}} (P_g^{-1}(x,y)) \vec{V}(P_g^{-1}(x,y)) \begin{pmatrix} \vec{T} \\ \vec{\omega} \end{pmatrix}. \tag{5}$$

where the matrix $\mathcal{U}$ contains known functions of the shape, the projection function and the retinal coordinates and where $\frac{\partial P}{\partial \vec{X}}(X,Y,Z)$ is the Jacobian matrix of the projection function $P(X,Y,Z)$, $P_g^{-1}(x,y)$ is the inverse projection mapping from the image to the target surface, which has shape $Z = g(X,Y)$, and $\vec{V}(X,Y,Z) = \begin{pmatrix} 1 & 0 & 0 & 0 & Z & -Y \\ 0 & 1 & 0 & -Z & 0 & X \\ 0 & 0 & 1 & Y & -X & 0 \end{pmatrix}$. For a proof of the theorem, a discussion of its consequences, and examples see [3].

In the sequel, the motion parameters $(\vec{T},\vec{\omega})$ will be denoted by $m_i, i = 1,\ldots,6$.

The problem now is to compute the motion parameters $m_i, i = 1,\ldots,6$ of (5), without actually computing the optical flow or the spatial derivatives of the image intensity function. For this purpose, we use the concept of *linear features*, which are sets of functionals over an area S of the image plane $F(t) = \{f_i(t) = \iint_S s(x,y,t))\mu_i(x,y)dx\,dy\,,\ i = 1,\ldots,n\}$, where $\mu_i(x,y)$ is called the measuring function of the ith linear feature. Consider the temporal derivative of the ith linear feature:

$$\dot{f}_i = \iint_S \frac{\partial s}{\partial t}\mu_i dx\,dy \overset{(1)and(5)}{=} -\sum_{k=1}^{6} m_k \iint_S \mu_i(u_{k_1}s_x + u_{k_2}s_y)dx\,dy\,,$$

where u_{ki} are the elements of the matrix $\mathcal{U}$ in (eq. 5). Defining $h_{ik} = -\iint_S \mu_i(u_{k_1}s_x + u_{k_2}s_y)dx\,dy$, we get $\dot{f}_i = \sum_{k=1}^{6} m_k h_{ik}$ and then

$$\mathbf{H}\vec{m} = \dot{\vec{f}}\,, \tag{6}$$

where $\mathbf{H}$ is the $n \times 6$ matrix of the coefficients h_{ik} and $\vec{m} = (m_1,\ldots,m_6)$. The vector of derivatives of linear features $\dot{\vec{f}}$ and the matrix $\mathbf{H}$ depend on measurable quantities. We can therefore compute them, solve the linear system (6) and obtain the motion parameter vector $\vec{m}$. Since the number n of linear features will normally be greater than 6, a least-squares solution will be used. It is well known that the system (6) has the unique minimal norm solution $\vec{m} = \mathbf{H}^{\dagger}\dot{\vec{f}}$, where $\mathbf{H}^{\dagger}$ is the Moore-Penrose inverse of $\mathbf{H}$.

In our target motion parameter estimation scheme no explicit calculation of the optical flow field is needed, as we have seen up to this point. In the next section we will see that neither is it needed for the tracking scheme developed there. Moreover, the only calculation involving the derivatives of the image intensity function s is done in order to obtain the parameters h_{ik}. Spatial differentiation of s is needed there, but, since numerical differentiation is an ill-posed problem, we tried to bypass it by using Green's Theorem on the boundary ∂S of the area S:

$$\iint_S [\frac{\partial}{\partial x}Q(x,y) - \frac{\partial}{\partial y}P(x,y)]dx\,dy = \int_{\partial S} P(x,y)dx + Q(x,y)dy.$$

Then, from the definition of h_{ik}:

$$h_{ik} = -\int_{\partial S} \mu_i(x,y) u_{k_1}(x,y) s(x,y)\, dy + \int_{\partial S} \mu_i(x,y) u_{k_2}(x,y) s(x,y)\, dx$$
$$+ \iint_S s(x,y) \tfrac{\partial}{\partial x}(\mu_i(x,y) u_{k_1}(x,y)) dx\, dy + \iint_S s(x,y) \tfrac{\partial}{\partial y}(\mu_i(x,y) u_{k_2}(x,y)) dx\, dy$$

Therefore, spatial differentiation of s was replaced by differentiation of μ_i and u_{ki}, which can be done analytically. Now, if we consider as S the projection of the target on the image plane, we can use the dynamic segmentation procedure to specify ∂S and then the above procedure will specify the kinematic parameters of the target.

2.2. Tracking

Using eq. (2), (3), (5), and the target motion parameters m_i that we just computed, the Tracking Constraint can be brought to the form:

$$J(\omega_x, \omega_y) = \alpha\omega_x^2 + \beta\omega_y^2 + \gamma\omega_x + \delta\omega_y + \epsilon\omega_x\omega_y + \zeta\,,$$

where α, $\beta, \ldots, \zeta$ are known functions of the data.

Obviously $\alpha, \beta \geq 0$. Experimental results show that they are not simultaneously zero and that $4\alpha\beta - \epsilon^2 \neq 0$. It is easy to prove that under those conditions the Tracking Constraint has a unique minimizer: $\omega_x^* = \frac{\delta\epsilon - 2\beta\gamma}{4\alpha\beta - \epsilon^2}$ and $\omega_y^* = \frac{\gamma\epsilon - 2\alpha\delta}{4\alpha\beta - \epsilon^2}$, which is the desired camera rotation that achieves tracking.

3. EXPERIMENTAL RESULTS

3.1. Synthetic experiments

The previous tracking algorithm was implemented on SGI IRIS graphics workstations for a textured planar target moving in front of a randomly textured background (Figure 2).

In this paper only the diagram of the distance oP_W versus time is presented as a measure of the quality of tracking. (P_w is the projection on the image plane of the "center" W of the target; o is the origin of the image plane coordinate system.) The simulation of the tracking process consists of a sequence of a target motion followed by a camera tracking action at each time instant. Therefore, motion at even time instants on the diagrams corresponds to target motion, while motion at odd time instants corresponds to camera tracking motion.

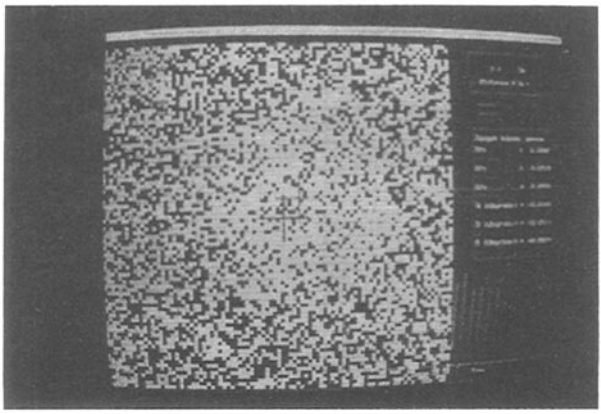

Figure 2.

Very good results were obtained for this tracking scheme fed with "perfect" estimates of the target motion (accurate values for the direction of translation and rotation) for various general motions of the target (Figure 3). The results degenerate "gracefully" as the quality of estimates decreases due to very fast motion of the target or noisy images. A sequence of "noisy" target motion estimates was generated in order to test the performance of the algorithm under inaccurate motion estimates. The algorithm performs well with 30% to 60% noise in target motion estimation (Figure 4).

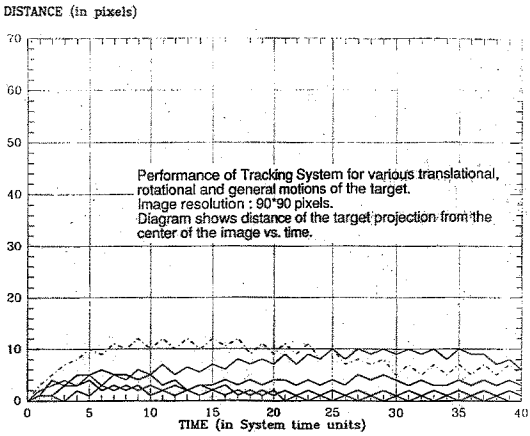

Figure 3.

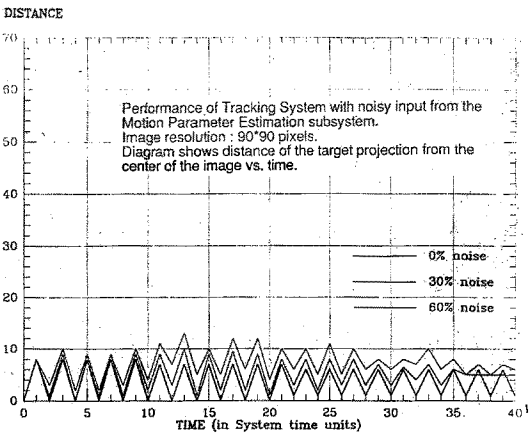

Figure 4.

Errors are introduced in this tracking scheme from discretization effects, which will affect mainly the temporal derivatives of the linear features, numerical instabilities in the least-squares solution of the linear system $\mathbf{H}\vec{m} = \vec{f}$ and the coarse resolution of the image

used in simulations.

Several sets of linear features were considered before establishing experimentally that the best 3-D motion parameter estimation is obtained with moments of the image, which have the advantage that they are easily implementable in hardware, therefore the above scheme can be used for real-time applications. (See [4] for a theoretical justification of this choice.) Moreover, the Connection Machine can be used for the real-time computation of the matrix $\mathbf{H}$ and the vector of linear features $\vec{f}$, which are the computational bottlenecks of the above algorithm.

3.2. A robotic implementation

In order to demonstrate the feasibility of this approach, we implemented the tracking algorithm in a robotic environment built in our laboratory. The task was the following: a programmable Mitsubishi robot arm held a soft drink can which it could move towards any direction against a textured background. On the hand, a 6-dof American Merlin robot arm equipped with a Sony DC-37 CCD camera was used to emulate the tracking system. The task then was to have the American Merlin robot arm track the can held by the first robot, which could move towards any direction (see Figure 5, which shows part of the "seeing" arm on the left "watching" the arm that holds the object to be tracked). Figure 6 shows the image of the object to be tracked as seen by the tracker.

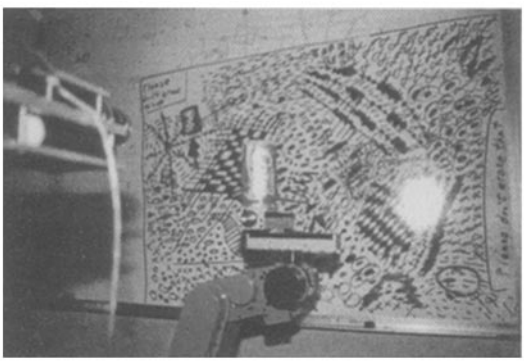

Figure 5.

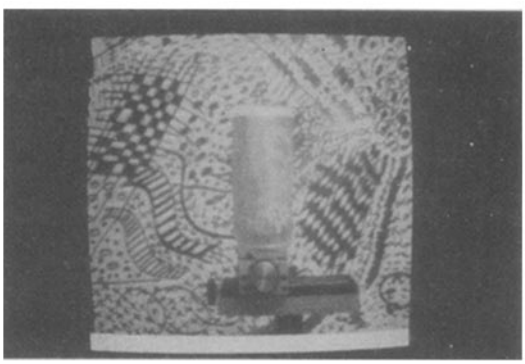

Figure 6.

Because the analysis of the images taken by the tracker had to be done on a VAX 785 and the primitive image processing operations on a VICOM image processor, all linked through a network, a real time implementation was impossible due to the heavy communication bottleneck. For this reason, we chose to present results from the experiments in the same format as in the case of the synthetic simulations. Again, the vertical axis of

the plots represents the distance oP_w as before, and the horizontal axis represents time and motion at even time instants on the diagrams corresponds to target motion, while motion at odd time instants corresponds to camera tracking motion. Figure 7 shows the results from three experiments. In the first (solid line), the actual motion was translation along the x-axis and under the assumption that the object in view was planar (instead of cylindrical) in order to examine the deviation of the results as a function of deviation from the local shape assumption. Clearly the results are not sensitive. The next two experiments (dotted and dashed lines) show results under horizontal motion (dotted) and motion in the $x - z$ plane (dashed). It is quite clear that the results are very satisfactory.

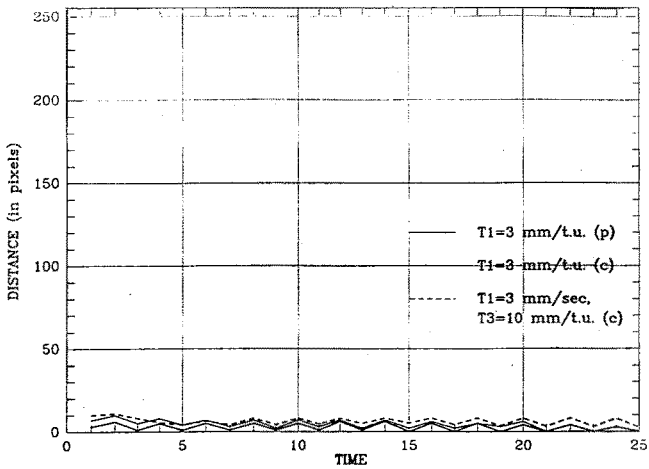

Figure 7.

Acknowledgements

The support of the Defense Advanced Research Projects Agency and the U.S. Army Engineer Topographic Laboratory under Contract DACA76-89-C-0019 is gratefully acknowledged, as is the help of Barbara Burnett in preparing this paper.

REFERENCES

1. J. Aloimonos, I. Weiss and A. Bandopadhay, "Active vision", *Int'l. J. of Comp. Vis.*, 333–356, 1988.
2. J. Aloimonos and D. Shulman, *Integration of Visual Modules: An Extension of the Marr Paradigm*, Academic Press, Boston, 1989.
3. J. Aloimonos and D. Tsakiris, "On the Mathematics of Visual Tracking", Center for Automation Research TR-2102, September 1988.
4. S. Amari, "Feature Spaces which Admit and Detect Invariant Signal Transformations", *Proc. 4th Intl. Joint Conf. on Pattern Recognition*, pp. 452–456, Tokyo, 1978.

5. P. Bouthemy and A. Benveniste, "Modelling of Atmospheric Disturbances in Meteorological Pictures", *IEEE Trans. PAMI*, PAMI-6, 5, 1984.
6. J.J.Clark and N.J.Ferrier "Control of Visual Attention in Mobile Robots", Proc. 1989 IEEE Conf. on Robotics and Automation.
7. K. Cornog, "Smooth Pursuit and Fixation for Robot Vision", M.S. Thesis, MIT AI Laboratory, 1985.
8. A.L. Gilbert, M.K. Giles, G.M. Flachs, R.B. Rogers and Y.H. U, "A Real-Time Video Tracking System", *IEEE Trans. PAMI*, PAMI–2, 1, 1980.
9. B.K.P. Horn and B.G. Schunk, "Determining Optical Flow", *Artificial Intelligence*, 17, pp. 185–203, 1981.
10. J. Loncaric, F. Decommarmond, J. Bartusek, Y.Pati, D. Tsakiris, R. Yang "Modular Dextrous Hand", Systems Research Center, TR 89-31, 1989.
11. S. Nagalia, "Real-Time Acquisition of Objects in Motion", Center for Automation Research, CS-TR-1398, 1984.
12. T. Poggio, E.B. Gamble, and J.J. Little, "Parallel integration of visual modules", *Science* **242**, 436–440, 1989.
13. S.A. Rajala, A.N. Riddle and W.L. Snyder, "Application of the One-Dimensional Fourier Transform for Tracking Moving Objects in Noisy Environments", *CVGIP*, 21, pp. 280–293, 1983.
14. J.W. Roach and J.K. Aggarwal, "Computer Tracking of Objects Moving in Space", *IEEE Trans. PAMI*, PAMI-1, 2, 1979.
15. A. Rosenfeld and A. Kak, *Digital Picture Processing*, Academic Press, 1982.
16. R.J. Schalkoff and E.S. McVey, "A Model and Tracking Algorithm for a Class of Video Targets", *IEEE Trans. PAMI*, PAMI-4, 1, 1982.
17. B.G. Schunk, "The Image Flow Constraint Equation", *CVGIP*, 35, pp. 20–46, 1986.
18. D.P. Tsakiris, "Visual Tracking Strategies", Systems Research Center, TR 88-66, 1988.
19. T.Y.Young and S.Gunasekaran "A regional approach to tracking 3-D motion in an image sequence", in *Advances in Computer Vision and Image Processing*, Vol. 3, pp. 63–99, JAI Press Inc., 1988.
20. B. Wilcox, D.B. Gennery, B. Bon and T. Litwin, "Real-Time Model-Based Vision System for Object Acquisition and Tracking", *SPIE*, Vol. 754, 1987.