Biol. Cybern. 67, 259–268 (1992)

Biological Cybernetics

Tracking facilitates 3-D motion estimation

Cornelia Fermüller [1,2] and Yiannis Aloimonos [1]

[1] Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD 20742-3411, USA
[2] Department for Pattern Recognition and Image Processing, Institute for Automation, Technical University Vienna, Treitlstrasse 3, A-1040 Vienna, Austria

Received November 14, 1991/Accepted in revised form February 28, 1992

Abstract. The recently emerging paradigm of Active Vision advocates studying visual problems in form of modules that are directly related to a visual task for observers that are active. Along these lines, we are arguing that in many cases when an object is moving in an unrestricted manner (translation and rotation) in the 3D world, we are just interested in the motion's translational components. For a monocular observer, using only the normal flow – the spatio-temporal derivatives of the image intensity function – we solve the problem of computing the direction of translation and the time to collision. We do not use optical flow since its computation is an ill-posed problem, and in the general case it is not the same as the motion field – the projection of 3D motion on the image plane. The basic idea of our motion parameter estimation strategy lies in the employment of fixation and tracking. Fixation simplifies much of the computation by placing the object at the center of the visual field, and the main advantage of tracking is the accumulation of information over time. We show how tracking is accomplished using normal flow measurements and use it for two different tasks in the solution process. First it serves as a tool to compensate for the lack of existence of an optical flow field and thus to estimate the translation parallel to the image plane; and second it gathers information about the motion component perpendicular to the image plane.

1 Introduction

For years visual motion interpretation has been approached through studying the "structure from motion" problem. The idea is to find methods of recovering the three-dimensional motion parameters and the structure of the objects in view from the dynamic imagery (Marr 1982; Ullman 1979; Koenderink and van Doorn 1975; Koenderink 1986).

Correspondence to: Y. Aloimonos

The way the problem has been addressed, was first to compute the exact position where each point in the image moved to. In cases of small motion the vector field that represents the change of every point in the image, the so called optical flow field, is computed from the spatio-temporal derivatives of the image intensity function. This requires the employment of additional constraints, such as smoothness. In cases, where the motion is considered large, features, such as points, line or contours in images taken at different time instances are corresponded. From the derived optical flow field or the correspondence between features the three-dimensional motion is then determined.

One can distinguish three phases in the evolution of research on the structure from motion problem. First, work dealt with the question of the existence of a solution, i.e. can we extract any information from a sequence of images about the structure and 3-D motion of the scene that cannot be found from a single image? Intensive research has been conducted in this field and several theoretical results have appeared that deal with questions such as: what can be recovered from a certain number of feature points in a given number of frames (Ullman 1979; Aloimonos and Brown 1989). Then the uniqueness aspects of the problem were studied. Nonlinear algorithms for the recovery of structure and motion from point (Longuet-Higgins and Prazdny 1980) or line correspondences and optic flow (Waxman et al. 1987) appeared increasingly in the literature. Such algorithms were based on iterative approximation techniques. So they lacked guaranteed convergence as well as a clear analytical formulations that would make a proof of uniqueness possible and allow other researchers to build upon them. Later "linear" algorithms and uniqueness proofs came out for points (Tsai and Huang 1984) and lines (Spetsakis and Aloimonos 1990), as well as flow (Adiv 1985); all were based on the same linearization technique. Although research in these lines has been accompanied by many experiments, none of the existing techniques could be used as a basic for an integrated system, working robustly in general environments.

The reasons for the lack of applicability to real world problems are due to the difficulty of estimating retinal correspondence, which is an ill-posed problem, the assumptions that have to be made to derive optical flow, and the sensitivity of 3D motion estimation to small changes in the data. Even optimal algorithms (Spetsakis and Aloimonos 1988) – optimal under the assumption of Gaussian noise – perform quite poorly in the presence of moderate noise. The efforts to remove these shortcomings gave birth to a new concept, Active Vision. The idea is to abandon the concept of recovering in any case all five possible motion parameters and the relative depth. If we consider simpler, specific problems (Aloimonos 1990) and allow the observer to be active, problems are easier to solve (Aloimonos et al. 1988) and a restriction to well defined input may be possible.

2 Active vision on well defined input

If we can "recover from a sequence of images the involved structure of the imaged scene and the relative three-dimensional motion", then various subsets of the computed parameters provide sufficient information to solve many practical problems, such as detection of independent motion, passive navigation, obstacle avoidance, prey catching, etc., as well as many other problems related to robotics and automation – hand-eye coordination, automatic docking, teleconferencing, etc. The difficulties posed from the structure from motion problem raise the idea to seek direct solutions to the above problems that don't presume complete recovery. If we can furthermore supply additional information to the solution finding task, we may solve problems that were originally considered as ill-posed, ill-conditioned and nonlinear. Additional information can be obtained by making the observer active and allow him therefore to manipulate and control certain parameters. This is the approach called for by the paradigm of Active Vision (Bajcsy 1985; Aloimonos et al. 1988). In their paper Aloimonos et al. discuss solutions to a few problems for an active observer, but they consider optical flow as input to their modules. We go one step further and restrict to just the spatio-temporal derivatives of the image intensity function. The question now is, what kind of activities enable us to solve certain visual problems?

Here, we analyze the following problem that appears in various visual tasks, where response to object motion has to be generated and the translation of the moving object is the relevant factor: "Given an active observer viewing an object moving in a rigid manner (translation + rotation), recover the direction of the 3-D translation and the time to collision by using only the spatio-temporal derivatives of the image intensity function," Although this problem is not equivalent to "structure from motion", because it does not fully recover the 3-D motion, it is of importance in a variety of situations. If an object is rotating around itself and also translating in some direction, we are usually interested in its translation – for example in problems related to tracking, prey catching, interception, obstacle avoidance, etc.

We want to avoid using optical flow and use data that is derived from just the variations in the image intensity function as the input to the estimation of 3D-motion. As the only available constraint for the flow (u, v) of the time changing image $I(x, y, t)$ we accept the constraint $I_x u + I_y v + I_t = 0$ (Horn and Schunck 1981), where subscripts denote partial differentiation. This just means that we can only compute the projection of the flow on the gradient direction ($(I_x, I_y) \cdot (u, v) = -I_t$), i.e. the so-called normal flow. This equation, the optic flow constraint equation, is derived when assuming that the irradiance at time t at point $P(x, y)$ and at time $t + \delta t$ at point $P(z + \delta x, y + \delta y)$ are the same, or in other words

$$\mathrm{d}I/\mathrm{d}t = 0.$$

The input we use is the spatio-temporal variation in the brightness pattern, which is associated with the vector field of apparent velocities, the optical flow field. It is often considered to coincide with the motion field, the projection of the 3D-motion on the image plane. This fact is stated through the assumption $\mathrm{d}I/\mathrm{d}t = 0$, which says that the two fields are the same. However, the optic flow field and the motion field are not equal in general. Verri and Poggio (1987) reported some general results in an attempt to quantify the difference between them. In Fermüller and Aloimonos (1991) the difference between the normal components of these two fields is estimated by using a first-order Taylor series approximation for the spatio-temporal variation in the image intensity. If u_n denotes the normal flow value at point (x, y) and $\bar{u}_n$ the normal motion value at the same point, then the difference is given by:

$$\bar{u}_n - u_n = \frac{1}{\|\nabla I\|} \frac{\mathrm{d}I}{\mathrm{d}t}$$

This shows that the two fields are closer when the local image intensity gradient ∇I is high. Thus, if we measure normal flow only in regions where the intensity gradients are of high magnitude, we'll guarantee that the normal flow measurements can be used for inferring 3-D motion.

The idea of using the image gradients to directly estimate 3D motion without going through the intermediate stage of calculating the optic flow field first appeared in the work of Aloimonos and Brown (1984). They presented a complete solution for the case of pure rotation, whereas a detailed study of translational motion can be found in Horn and Weldon (1987) and Negahdaripour (1986). Finally, a hybrid technique appeared recently (Taalebi-Nezhaad 1990), using both optical flow and image gradients for addressing the 3-D motion estimation in the general case (rotation and translation). In this paper we address the general case, but we perform partial motion recovery. We estimate the direction of translation of a rigidly moving object using just normal flow.

3 Overview

We present a method for estimating the direction of translation and the time to collision for a monocular observer that has the capability of tracking. The observer derives from the image sequence the tracking movement of the observer's motor and uses these tracking parameters as input to the computation of the object's 3-D motion.

To begin with, the observer detects independent motion (Sharma and Aloimonos 1991) and fixates at the object, thus causing the optical axis to pass through the object. The translational direction of an object moving with translational parameters (U, V, W) and rotating with velocity $(\omega_z, \omega_y, \omega_x)$ is represented in the image plane by the point $(U/W, V/W)$, the Focus of Expansion (FOE). To give a graphical explanation: If we put the object at a distance equal to the focal length f in front of the nodal of the camera, the FOE represents the intersection point of the image plane and the motion trajectory which passes through the nodal point (see Fig. 1).

It has been argued that tracking is used in biological vision for the sake of simplifying the estimation of motion. Since we are studying computer vision for an active observer, our first question concerns the nature of the activities themselves. Therefore we should ask why one should go a roundabout way and derive intermediately the tracking movement. What do we gain from tracking?

Through tracking we can accumulate information over time and add therefore the parameter of time as additional component to the input information. Another advantage of tracking is, that since it is accomplished for a number of steps, the tracking parameters can be corrected sequentially (smoothed) and we don't rely on just one measurement.

The idea of using the tracking parameters for motion estimation was used before by Bandopadhay and Ballard (1991). They provide closed form solutions for the computation of the egomotion parameters for a binocular observer by employing the rotation angle and its first and second derivative (angular velocity and acceleration). In their paper they did not show how tracking was actually done, whereas we propose a complete solution: we first show how to compute the tracking parameters using normal flow and then how to use them for a 3-D motion estimation.

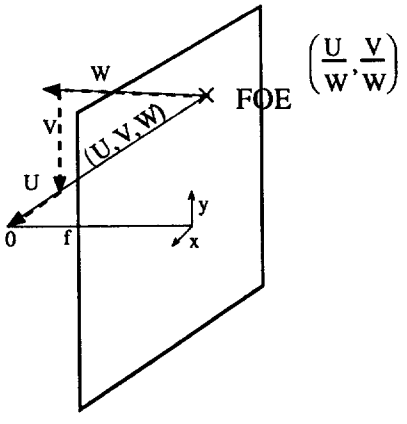

Fig. 1. FOE in the image plane

We are accomplishing the computation of the FOE and the time-to-collision through three modules, that involve the activities of fixation and tracking.

1. By fixating at an object point, which we consider to be the origin of the used coordinate system, we get image velocity at the center that represents the projection of parallel translation. We show how tracking can be used to derive the projection of parallel translation from just the spatio-temporal derivatives.

2. In the next step, the output of the first module is used to acquire information about translation parallel to the optical axis. Again tracking is used, here as a tool for accumulating depth information over time.

3. In a third module we show that time to collision is related to the FOE and how to estimate it from the spatio-temporal information at the fixated point.

4 The choice of the coordinate system

Since the motion parameter (U, V, W) and $(\omega_x, \omega_y, \omega_z)$ are expressed relative to a coordinate system, the prediction of the object's position in subsequent frames is dependent on the choice of the coordinate system. The ideal place to put the origin of the coordinate system would be the mass center of the object (the "natural" system). Since the mass center is not known, different choices have to be made. Most commonly the camera's nodal point is chosen as the center of the coordinate system ("camera centered" coordinate system). Rotation is described around the nodal point. In the case of object motion this leads to different values for the motion parameters for each new frame, which is an unwelcome effect in the task of finding translational motion.

We therefore decided to attach the center of rotation to the object's point of intersection with the optical axis (an "object centered" coordinate system) (see Fig. 2). The active observer is free in its choice of the center and will therefore decide for a point belonging to a neighborhood of non-uniform brightness with distinguishable features.

This approach can be justified by the following argument: When choosing as fixated point the mass center of the object's image or a point in its near neighborhood, the resulting motion parameters are in many cases close to those of the natural system. In the natural coordinate system with center O_{natural} the velocity v at point P is due to the translational and the rotational component:

$$v = t_{\text{natural}} + \omega \times \overrightarrow{O_{\text{natural}}P}$$

and in the object centered coordinate system with center O_{object} the same velocity is expressed as

$$v = t_{\text{object}} + \omega \times \overrightarrow{O_{\text{object}}P}$$

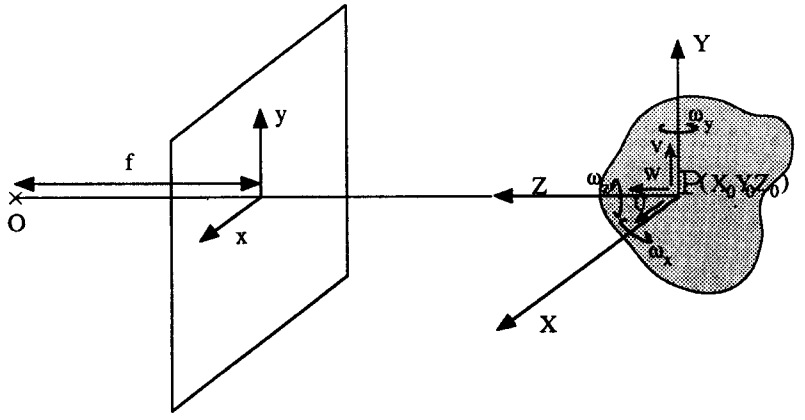

Fig. 2. Object centered coordinate system

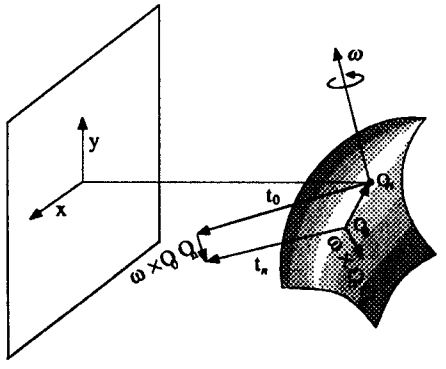

Fig. 3. The difference in translation between t_n in the natural system with center O_n and t_0 in the object centered system with center O_o is $\omega \times \overrightarrow{O_o O_n}$

Therefore the difference in translation between t_{natural} and t_{object} (see Fig. 3) is given by:

$$t_{\text{natural}} - t_{\text{object}} = \omega \times (\overrightarrow{O_{\text{object}} P} - \overrightarrow{O_{\text{natural}} P})$$
$$= \omega \times \overrightarrow{O_{\text{object}} O_{\text{natural}}}$$

The value becomes smaller as $\overrightarrow{O_{\text{object}} O_{\text{natural}}}$ decreases.

5 Tracking gives parallel translation

The first activity used in this approach is fixation. This action provides us with linear relations between the 3D- and the 2D-velocity-parameters. An object at distance Z in front of the camera moves in the 3D environment with translational velocity (U, V, W) and rotational velocity $(\omega_x, \omega_y, \omega_z)$. In an object centered coordinate system with center $P(X_0, Y_0, Z_0)$ under perspective projection the optical flow (u, v) is related to these parameters through the following equations:

$$\frac{dx}{dt} = u = \frac{Uf}{Z} - \frac{Wx}{Z} - \frac{xy\omega_1}{f} + \omega_2\left(\frac{x^2}{f} + \frac{f(Z - Z_0)}{Z}\right) - \omega_3 y$$

$$\frac{dy}{dt} = v = \frac{Vf}{Z} - \frac{Wy}{Z} - \omega_1\left(\frac{y^2}{f} + \frac{f(Z - Z_0)}{Z}\right) + \frac{\omega_2 xy}{f} + \omega_3 x$$

In a small area around the center x, y and $(Z - Z_0)/Z$ are close to zero. The optical flow components due to rotation and due to translation parallel to the optical axis converge to zero, and u becomes Uf/Z and v becomes Vf/Z.

The flow at the center of the image gives the projection of parallel translation, but only normal flow is available. We show that tracking can be used for the evaluation of optical flow in an iterative technique and prove the convergence of the method to the exact solution.

The problem of current optical flow algorithms is that additional constraints are imposed. Constraints that impose a relationship on the values of the flow field are usually used, and this amounts in assumptions, such as smoothness, about the scene in view. This basic problem is overcome by providing the observer with activity. The computation is thus transferred to the active observer, who has the ability to iteratively adjust his motion through his control mechanism to the given situation.

In cases, where the dominant motion of the object is translation towards the observer, the resulting optical flow vectors are emanating from a point which lies inside the object's image. The coordinates of this point, the FOE, are consequently close to zero. Otherwise the optical flow pattern is due to vectors that are about parallel and have about the same magnitude. Typical normal flow patterns for both cases are shown in Fig. 4.

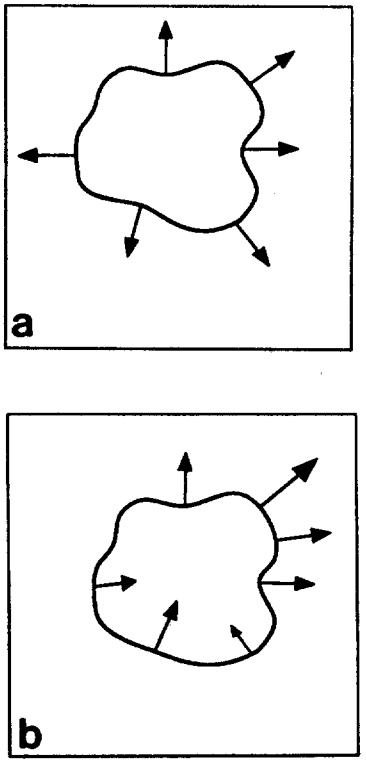

Fig. 4. **a** Normal flow vectors emanating from a point inside the object. **b** Normal flow vectors, when the translational component parallel to the image plane is not much larger than the component perpendicular to the plane

For these cases, where the FOE lies inside the object, the normal flow vectors are mainly due to translation, because the rotational components near the object center are very small. Therefore a simple technique using only the direction of the normal flow measurements can be applied. Given the normal flow vector at a point, we know that the FOE lies in the half-plane, which is separated from the one containing the normal flow vector through the greylevel edge. Considering every available normal flow measurement

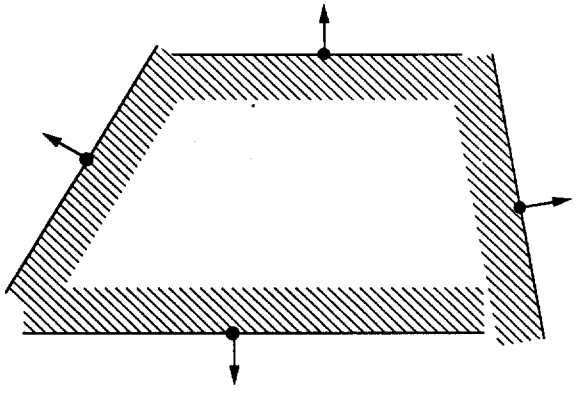

Fig. 5. Each available normal flow measurement constrains the possible location of the FOE

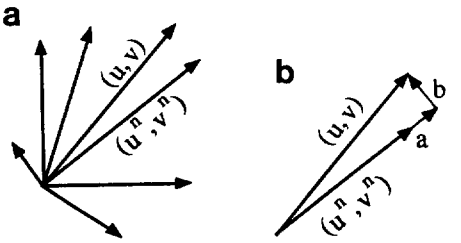

Fig. 6. **a** Normal flow vectors in different directions. **b** The new flow vector (resulting from object motion and tracking) is due to **a**, the error in magnitude, and due to **b**, the error in direction

will narrow the possible location of the FOE to a small area (see Fig. 5) (see also Horn and Weldon 1987; Aloimonos 1990). When dealing with such normal flow patterns, it would make no sense to use the method introduced in this paper; we are concerned here with the more complicated case as displayed in Fig. 4b.

Let us compute the normal flow in a set of directions in a small area around the origin (fixation point). The normal flow is the projection of the optical flow on the gradient direction. The largest of the normal flow values in the different directions is therefore the one closest to the optical flow. Let us call this normal flow vector the "maximum normal flow" and denote it by (u^n, v^n) (see Fig. 6a). We take it as an approximation to the correct optic flow and use it to track the fixated point. The purpose of tracking is to correct for the error in the approximation. In order to keep a point with optical flow (u, v) in the center of the image the observer has to perform a movement that produces the same value of optical flow in the opposite direction. The way our observer accomplishes this task is by rotating the camera around the nodal point about the x- and y-axis. While the observer is moving it takes the next image and computes again the normal flow vectors. If the maximum normal flow was equal to the optical flow, a new optical flow (due to object motion and egomotion) of zero will be achieved.

Usually, however, the maximum normal flow and the optical flow are not equal; they will differ in magnitude and/or in direction. An error in magnitude results in a flow vector in the direction of maximum normal flow, and an error in direction creates a flow vector perpendicular to it (see Fig. 6b). The actual error is usually in both magnitude and direction. Thus the new flow vector is a vector sum of the two components. Again it can be approximated by the largest normal flow vector measurement. The new measured normal flow is used as a feedback value to correct the optical flow and the tracking parameters; the new normal flow vector is added to the maximum normal flow vector computed in the first step. Proceeding by applying the same technique to the successive estimated errors will result in an accurate estimate of the actual flow after a few iterations. The proof of convergence to the exact solution follows:

We use here a simplified model to explain tracking. The change of the local coordinate system during tracking and the fact, that the object is coming closer, is not considered. Since, for the purpose of optical flow estimation the number of tracking steps is small, the error originating from this model is not essential. Concerning a specific application, the algorithm will stop when the computed error is smaller than a given threshold, which will cover model errors.

In each iteration step we are computing an approximation to difference between the observer's egomotion and the object motion. Considering the possible sources of error we have to show that the approximation error will become zero.

Deviations of the chosen maximum normal flow from the optical flow value are due to the following reasons.

- Deviations covered through the model:
  The fact that normal flow measurements are computed in a finite number of directions causes an error in direction of up to half the size of the interval's size between two normal flow measurements. If measurements in n directions are performed the maximum error y is bounded by: $y < \pi/n$.
- Deviations coming from simplifications and discrete computations:
  In the evaluation of flow measurements the parts linear and quadratic in x, y, and $Z - Z_0$ are ignored. Furthermore each measurement in one direction is computed as the average of the normal flow values in a range y of directions. These reasons may cause errors in magnitude as well as direction, and a different vector than the closest normal flow vector may be chosen.
- General errors occuring in normal flow computation:
  Sensor noise in normal flow measurements and the numerical computation of the derivatives of the image intensity function can influence the magnitude and the direction of the estimated value.

Let v be the magnitude of the actual optical flow. The error sources give rise to specifying the error in magnitude, x, in percentage of the actual value. x_i is the magnitude error in the maximum normal flow measurement in step i and y_i is the angle between the maximum normal flow vector and the optical flow vector, where $x_i < x$ and $y_i < y$. Therefore the difference between the optical flow and the first measurement of maximum normal flow is given by

$$diff_1 = \begin{pmatrix} vx_1 \cos y_1 \\ v \sin y_1 \end{pmatrix},$$

where the x-axis is aligned with the maximum normal flow vector (see Fig. 7). The square of its magnitude is computed as:

$$\|diff_1\|^2 = v^2 x_1^2 \cos^2 y_1 + v^2 \sin^2 y_1$$

The second normal flow vector, if measured from the direction of the maximum normal flow vector derived in the second step is given by

$$diff_2 = \begin{pmatrix} \|diff_1\| x_2 \cos y_2 \\ \|diff_1\| \sin y_2 \end{pmatrix},$$

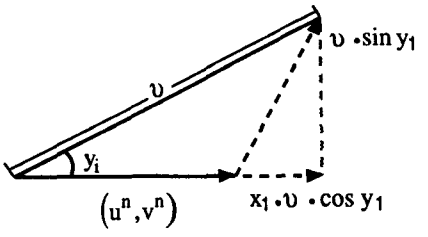

Fig. 7. Error between optical flow vector and maximum normal flow vector

and the square of its magnitude is therefore

$$\|diff_2\|^2 = x_1^2 x_2^2 v^2 \cos^2 y_1 \cos^2 y_2 + x_1^2 v^2 \cos^2 y_1 \sin^2 y_2 + v^2 \sin^2 y_1 \sin^2 y_2 + x_2^2 v^2 \sin^2 y_1 \cos^2 y_2$$

In general, if we denote by $\{a, b\}$ the fact that either a or b has to be chosen, then $\|diff_n\|^2$ can be expressed as

$$\|diff_n\|^2 = v^2 \sum_{\text{all permutations}} \prod_{i=1}^{n} \{x_i^2 \cos y_i^2, \sin y_i^2\}$$

Since $x_i < 1$ and $\sin y_i < 1$ it follows that $\Pi_i (x_i^2 \cos y_i^2, \sin y_i^2\}$ and thus the whole term converges to zero. Therefore we have shown the convergence of the approximation value to the actual optical flow value for the "simplified tracking model".

6 Estimating the FOE using tracking

When continuing with tracking over time, as an object comes closer and the value of Z becomes smaller, the optical flow value increases. In order to track correctly and adjust to the increasing magnitude of the optical flow value the tracking parameters have to be changed, too. From the change of the tracking parameters the change in Z can be derived. If tracking is accomplished by rotation with a certain angular velocity, this just means, that the change in depth is derived from the angular acceleration. In the sequel we show the relation between image motion and tracking movement and explain the computation of the tracking parameters, which have to be changed in every step. We explain the exact process of tracking for a geometric setting consisting of a camera that is allowed to rotate around two fixed axes: X- and Y-. These axes coincide with the local coordination system of the image plane at the beginning of the tracking process.

We describe rotation by an angle ϕ around an axis, which is given by its directional cosines n_1, n_2, n_3, where $n_1^2 + n_2^2 + n_3^2 = 1$. The transformation of a point P with coordinates (X, Y, Z) before and (X', Y', Z') after motion is described through the linear relation:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

where the transformation matrix R is of the following form:

$$\begin{pmatrix} n_1^2+(1-n_1^2)\cos\phi & n_1n_2(1-\cos\phi)-n_3\sin\phi & n_1n_3(1-\cos\phi)+n_2\sin\phi \\ n_1n_2(1-\cos\phi)+n_2\sin\phi & n_2^2+(1-n_2^2)\cos\phi & n_2n_3(1-\cos\phi)-n_1\sin\phi \\ n_1n_3(1-\cos\phi)-n_2\sin\phi & n_2n_3(1-\cos\phi)+n_1\sin\phi & n_3^2+(1-n_3^2)\cos\phi \end{pmatrix} \equiv \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} = R$$

Since the image coordinates (x, y) are related to the 3D-coordinates through: $x = Xf/Z$ and $y = Yf/Z$, we get the following equations:

$$x' = \frac{(r_1x + r_2y + r_3f)f}{(r_7x + r_8y + r_9f)}$$

$$y' = \frac{(r_4x + r_5y + r_6f)f}{(r_7x + r_8y + r_9f)}$$

In order to compensate for the image motion (u, v) of the point P_0, which moves from $(0, 0)$ to (u, v) at one time unit the camera has to be rotated by ϕ, n_1, and n_2, where

$$u = n_2 f \tan\phi$$

$$v = -n_1 f \tan\phi$$

Taking at the center of the image the flow measurements (u, v) at the beginning of the tracking process at time t_1, and assuming that the object doesn't change its distance Z_1 to the camera, we can conclude that during a time interval Δt an image flow $(u\,\Delta t, v\,\Delta t)$ would be measured. The tracking motion necessary for compensation is given by

$$\frac{Uf}{Z_1} = n_2 \tan\phi.$$

But at time t_2 the object has moved to distance Z_2 and we measure a rotation

$$\frac{Uf}{Z_2} = n_2' \tan\phi'$$

Figure 8 shows the relationship between the 3D motion and the tracking parameter. Since $Z_2 - Z_1 = W\,\Delta t$, the change in the reciprocal of the rotation angle is proportional to W/U, because

$$\frac{1}{n_2' \tan\phi'} - \frac{1}{n_2 \tan\phi} = \frac{Z_2 - Z_1}{Uf\,\Delta t} = \frac{W\,\Delta T}{Uf\,\Delta t}$$

and the FOE $(U/W, V/W)$ can be computed as

$$\frac{U}{W} = 1 \Big/ f\left(\frac{1}{n_2' \tan\phi'} - \frac{1}{n_2 \tan\phi}\right) = 1 \Big/ \left(\frac{1}{n_2' \tan\phi'} - \frac{f}{u\,\Delta t}\right)$$

and

$$\frac{V}{W} = 1 \Big/ f\left(\frac{1}{-n_1' \tan\phi'} - \frac{f}{v\,\Delta t}\right).$$

It remains to be explained how tracking is actually pursued, since we are facing the problem of a constantly changing local coordinate system. The interested reader can consult Appendix A, which is devoted to the tracking parameter computation.

7 Estimating the time to collision

If the values of the motion parameters don't change over the tracking time the value Z/W, the time to collision, expresses the time left until the object will hit the infinitely large image plane. A relationship between

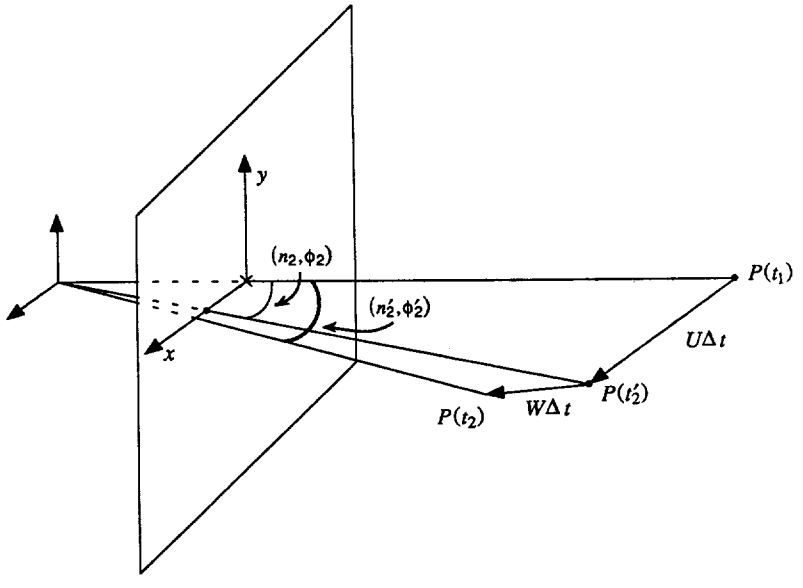

Fig. 8. From the optical flow value, which is due only to translation parallel to the image plane, a translation of P from $P(t_1)$ to $P(t_2)$ is inferred, and therefore the tracking parameters (n_2, ϕ) are expected. But actually the point has moved to $P(t_2)$ and a rotation described by $(n_{2'}, \phi')$ is measured

FOE and time to collision is inherent in the scalar product of the optical flow vector $(\mathbf{u}, \mathbf{v})$ with the vectors in gradient direction $(\boldsymbol{\alpha}, \boldsymbol{\beta})$:

$$\begin{pmatrix} u \\ v \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \|v^n\|$$

For the pixels in the center, for which we ignore the linear and quadratic parts in x, y and $(Z - Z_0)/Z$ in the relation between optical flow and 3D-parameters we get the relationship:

$$\frac{Uf}{Z}\alpha + \frac{Vf}{Z}\beta = \|v^n\|$$

$$\frac{Uf}{W}\alpha + \frac{Vf}{W}\beta = \|v^n\| \frac{Z}{W}$$

Since we know the FOE, we can compute the time to collision from this relationship, by measuring the normal flow value in all directions of the set and by solving an overdetermined system of linear equations by minimizing the least square error.

8 Experimental results

We tested the method on synthetic imagery by using the graphics package Swivel. This way we were able to simulate object motion as well as camera rotation. In order to analyze the robustness of the method, we evaluated the accuracy of the normal flow values in the center of the images. At every point we determined v_{act}, the projection of the known optical flow value on the gradient direction computed there. The error (*err*) in the normal flow values was defined as standardized difference between v_{act} and the normal flow value, v_{meas} ($err = (v_{act} - v_{meas})/v_{act}$ %). This way we computed an average error of 76.14% and a standard deviation of 179.64% for the motion sequence at the beginning of the tracking process. This constitutes a large error and is comparable to errors appearing in noisy real imagery.

Fig. 9. First image in the sequence used for tracking

The object displayed in Fig. 9 moves in the direction $U/W = 4$ and $V/W = 2$, with an image motion at the center of $u = 0.004$ and $v = 0.002$ focal units, and we tracked it over a sequence of 100 images. Concerning the implementational details, we computed normal flow measurements in 10 directions in an area of 9×9 pixels at the center of the image. When testing the first module, with which parallel translation is estimated, we used a threshold of 0.0002 focal units. The method converges very quickly, usually after 2 to 3 iterations. We added rotation of growing magnitude to the object motion, and it turned out that the algorithm converges for this set-up even for relatively large rotations. (The object was 25 units away from the camera and moved with translational velocity of $U = 0.1$, $V = 0.05$, $W = 0.025$ units per time and the method converges for rotations of up to 0.3° per time unit around the x- y- and z-axis.) Some graphical representations are given: Figure 10 shows for the case of no rotation the three normal flow fields that were computed in the 9×9 pixels large area, before convergence was achieved. In Figure 11 two maximum normal flow vector sequences are displayed (a: for no rotation, b: for rotation $\omega_x = 0.1°$, $\omega_z = 0.1°$). Using the estimates of parallel translation from this module and continuing with tracking over 100 steps resulted in FOE values of less than 15% error (e.g., for the case of no rotation we computed an FOE of $U/W = 4.21$ and $V/W = 1.79$). With these experiments we demonstrated that our technique can tolerate a large amount of noise in the input (normal flow) and still be successful.

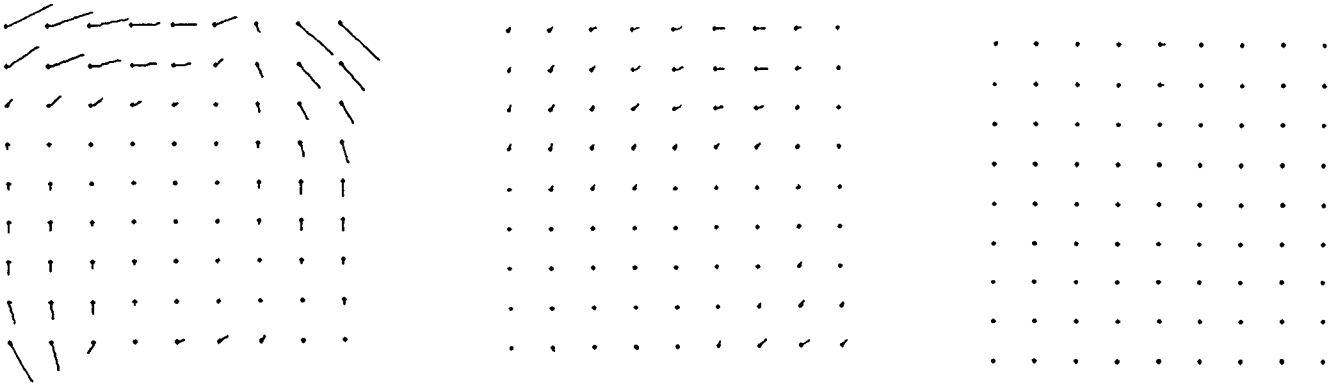

Fig. 10. Normal flow fields for a tracking sequence

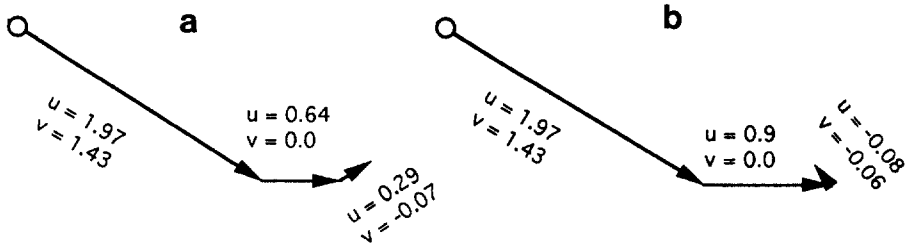

Fig. 11. Maximum normal flow vectors for **a** no rotation and **b** for rotation $\omega_x = 0.1°/\Delta t$, $\omega_z = 0.1°/\Delta t$

9 Conclusions

It has been argued before by psychologists that biological organisms use tracking in the motion estimation process. In this paper we have exploited the advantages of the tracking activity to solve for a monocular observer the problem of computing a moving object's translational direction and its time-to-collision. We have presented a complete solution to this task by showing how tracking can be pursued when only normal flow measurements are used and how these parameters are of use in the 3-D motion parameter decoding strategy. The presented technique consists of three subtasks. First tracking is used in combination with fixation to estimate the motion components parallel to the image plane, and second tracking serves to compute the perpendicular translational components and to estimate the FOE. The output of these modules is employed then to estimate the time-to-collision.

A theoretical analysis of the tracking algorithm of the first module has been pursued and the convergence of the method has been proved. Furthermore the exact computation of the tracking parameters, under consideration of the change of local coordinate systems is given. Experimental studies have been conducted on synthetic imagery and we achieved very good results. The method was developed mainly for cases, where the translational components perpendicular to the image plane are not much larger than the ones parallel to the plane. Otherwise the flow pattern in the object's image will consist of vectors emanating from a point. This point, the FOE, can then be estimated with a qualitative technique (Aloimonos and Brown 1984; Aloimonos 1990).

Appendix A. Computation of tracking parameters

In this section we describe the computation of the tracking parameters for the second module. Unlike Sect. 5, where a "simplified model" was used, we take here the change of the local coordinate system into account and show the necessary parameter transformations between the coordinate systems.

In the first module the projection of parallel translation at the beginning of the tracking process has been computed as described in Sect. 5. From these measurements we derived the rotational parameters $\phi 1$, $n_{1,1}$ and $n_{2,1}$ necessary to track for one time interval. When continuing with tracking, we have to consider that through the rotation of the image plane the local coordinate system attached to it changes also. At each tracking step, in the current local coordinate system an optical flow emerges that is due to the change in the Z-distance. The rotation necessary to compensate for this value has to be computed and is added to the old rotation. The summation of rotational vectors is justified, since we are adding a very small vector.

The computation of the rotation vector from normal flow is done in the following way: We compute in the new system the normal flow vectors in different directions and take the maximum value. This vector spans from (0, 0) to (u_n, v_n). In order to compensate for this vector by rotation around the fixed X- and Y-axes, the point (0, 0) and the point (u_n, v_n) are transformed back to the old system through (1)

$$x_{\text{old}} = \frac{(r_1 x_{\text{new}} + r_2 y_{\text{new}} + r_3 f) f}{(r_7 x_{\text{new}} + r_8 y_{\text{new}} + r_9 f)}$$

and

$$y_{\text{old}} = \frac{(r_4 x_{\text{new}} + r_5 y_{\text{new}} + r_6 f) f}{(r_7 x_{\text{new}} + r_8 y_{\text{new}} + r_9 f)} \quad (1)$$

The same formula can be applied to compute from the coordinates the necessary rotation to transform one point into the other.

Acknowledgements. This research was supported by the National Science Foundation under a Presidential Young Investigator Award to Y. Aloimonos, Alliant Systems Inc. and Texas Instruments Inc. and the Österreichisches Bundesministerium für Wissenschaft und Forschung and the Österreichische Bundeskammer der Gewerblichen Wirtschaft.

References

Adiv G (1985) Determining 3D-motion and structure from optical flow generated by serveral moving objects. IEEE Trans Pattern Anal Machine Intell PAMI-7:384–401

Aloimonos JY (1990) Purposive and qualitative active vision. Proc. DARPA Image Understanding Workshop, pp 816–828

Aloimonos J, Brown CM (1984) Direct processing of curvilinear sensor motion from a sequence of perspective images. In Proc Workshop on Computer Vision: Representation and Control, pp 72–77

Aloimonos J, Brown CM (1989) On the kinetic depth effect. Biol Cybern 60:445–455

Aloimonos J, Weiss I, Bandopadhay A (1988) Active vision. Int J Comput Vision 2:333–356

Bajcsy R (1985) Active perceptive vs. passive perception. In Proc. IEEE Workshop on Computer Vision, pp 55–59

Bandopadhay A, Ballard DH (1991) Egomotion perception using visual tracking. Comput Intell 7:39–47

Fermüller C, Aloimonos Y (1991) Estimating 3-D Motion from Image Gradients. Technical Report CAR-TR-564, Center for Automation Research, University of Maryland

Horn B, Schunck B (1981) Determining optical flow. Artif Intell 17:185–203

Horn BKP, Weldon J (1987) Computational-efficient methods for recovering translational motion. In Proc International Conference on Computer Vision, pp 2–11

Koenderink J (1986) Optic flow. Vision Res 26:161–180

Koenderink J, van Doorn A (1975) Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. Opt Acta 22:773–791

Longuet-Higgins HC, Prazdny K (1980) The interpretation of a moving retinal image. Proc R Soc London B 208:385–397

Marr D (1982) Vision. Freeman, San Francisco

Negahdaripour S (1986) Direct passive navigation. PhD thesis, Department of Mechanical Engineering, MIT, Cambridge, Mass

Sharma R, Aloimonos J (1991) Robust detection of independent motion: an active and purposive solution. Technical Report CAR-TR-534, Center for Automation Research, University of Maryland

Spetsakis ME, Aloimonos J (1990) Structure from motion using line correspondences. Int J Comput Vision 4:171–183

Spetsakis ME, Aloimonos J (1988) Optimal computing of structure from motion using point correspondence. In Proc International Conference on Computer Vision, pp 449–453

Taalebi-Nezhaad MA (1990) Direct recovery of motion and shape in the general case by fixation. In Proc DARPA Image Understanding Workshop, pp 284–291

Tsai RY, Huang TS (1984) Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. IEEE Trans Pattern Anal Machine Intell PAMI-6:13–27

Ullman S (1979) The interpretation of visual motion. MIT Press, Cambridge, Mass

Verri A, Poggio T (1987) Against quantitative optic flow. Proc IEEE International Conference on Computer Vision

Waxman, AM Kamgar-Parsi B, Subbarao (1987) Closed-form solution to image flow equations for 3D structure and motion. Int J Comput Vision 1: 239–258