Behavioral Visual Motion Analysis

Yiannis Aloimonos, Zoran Duriç, Cornelia Fermüller,
Liuqing Huang, Ehud Rivlin, and **Rajeev Sharma**
Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3411

Abstract

We propose here a new approach to addressing problems related to visual motion, namely the purposive approach [4]. Instead of considering the various visual motion tasks as applications of the general structure from motion module, we consider them as independent problems and we directly seek solutions for them. As a result we can achieve unique and robust solutions without having to compute optic flow and without requiring a full reconstruction of the visual space, because it is not needed for the tasks. In the course of the exposition, we present novel solutions to various important visual tasks related to motion, such as the problems of motion detection by a moving observer, passive navigation, relative-depth computation, 3-D motion estimation, and visual interception, using as input only the spatial and temporal derivatives of the image intensity function. It turns out that the spatiotemporal derivatives of the image (i.e. the so-called normal flow) do not seem to be capable of solving the general "structure from motion" problem. They are, however, sufficient to provide robust algorithms for the solution of many interesting visual tasks that do not require the full solution, but only part of it. The ability to create robust nontrivial behaviors suggests the possibility that visual perception could be studied as intelligent behavior. We point out some of the benefits and drawbacks of this paradigm that studies vision as a set of behaviors that recover the visible world partially, but well enough to carry out a task (purposive, animate or behavioral vision), and we contrast it to the traditional paradigm of treating vision as a general recovery problem.

1 Introduction and Motivation

The problem of structure from motion has attracted a lot of attention in the past several years [23, 30, 33] because of the general usefulness that a potential solution to this problem would have. Important navigational problems, such as detection of independently moving objects by a moving observer, passive navigation, obstacle detection, target pursuit, and many other problems related to robotics, teleconferencing, etc., would be simple applications of a structure from motion module. The problem has been formulated as follows: Given a sequence of images taken by a monocular observer (the observer and/or parts of the scene could be moving), to recover the shapes (and relative depths) of the objects in the scene, as well as the (relative) 3-D motions of independently moving bodies.

The problem has been formulated and usually treated as an aspect of the general task of recovering 3-D information from motion [25, 19]. The majority of the proposed solutions to date are based on the following modular and hierarchical approach:

1. First, one computes the optic flow on the image plane, i.e. the velocity with which every image point appears to be moving.[1]
2. Then segmentation of the flow field is performed and different moving objects are identified on the image plane. From the segmented optic flow one then computes the 3-D motion with which each visible surface is moving relative to the observer. (Assuming that an object moves rigidly, a monocular observer can only compute its direction of translation and its rotation, but not its speed.)
3. Finally, using the values of the optic flow along with the results of the previous step, one computes the surface normal at each point, or equivalently, the ratio Z_i/Z_j of the depths of any two points i and j.

The reason that most approaches have followed the above three-step approach is two-fold. The first is due to the formulation of the problem, which insists on recovering a complete relative depth map and accurate three-dimensional motion. The second is due to the fact that the constraints relating retinal motion to three-dimensional structure involve 3-D motion in a nonlinear manner that does not allow separability. For examples of such approaches, see [1, 34, 23, 30]. However, the past work in this paradigm, despite its mathematical elegance, is far from being useful in real-time navigational systems, and such techniques have found few or no prac-

[1] For clarity, we consider only the differential case. In the case of long range motion one computes discrete displacements, but the analysis remains essentially the same.

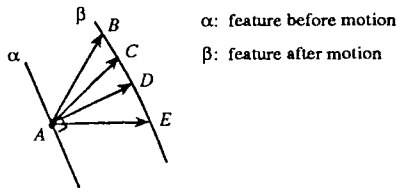


Figure 1: The aperture problem. Point A could have moved to B, C, D, E. However, whatever the value of the image motion vector is, its projection on the normal to α is always AD (known).

tical applications.[2] Consequently, this approach cannot be used yet to explain the ability of biological organisms to handle visual motion.

There exist many reasons for the limitations of the optic flow approach, related to all three steps listed above. To begin, the computation of optic flow is an ill-posed problem, i.e. unless we impose additional constraints, we cannot estimate it [19]. Such constraints, however, impose a relationship on the values of the flow field which is translated into an assumption about the scene in view (for example, smooth). Thus, even if we are capable of obtaining an algorithm that computes optic flow in a robust manner, the algorithm will work only for a restricted set of scenes. The only available constraint at every point (x, y) of the changing image $I(x, y, t)$ for the flow (u, v) is the constraint $I_x u + I_y v + I_t = 0$ [21], where the subscripts denote partial differentiation. This 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*. More graphically, it means that if a feature (for example, an edge segment) in the image moves to a new position, we don't know where every point of the segment moved to (see Figure 1); we only know the normal flow, i.e. the projection of the flow on the image gradient at that point.

A second reason has to do with the very essence of optic flow. An optic flow field is the vector field of apparent velocities that are associated with the variation of brightness on the image plane. Clearly, the scene is not involved in this definition. One would hope that optic flow would be equivalent to the so-called motion field [19], which is the (perspective) projection on the image plane of the three-dimensional velocity field associated with each point of the visible surfaces in the scene. However, the optic flow field and the motion field are not equal in general. Verri and Poggio [36] reported some general results in an attempt to quantify the difference between the optic flow and motion fields. Although we don't yet have necessary and sufficient conditions for the equality of the two fields, it is clear that they are equal only under specific sets of restrictive conditions.

A third reason is related to the second step of the existing algorithms for structure from motion. These algorithms attempt to first recover three-dimensional motion before they proceed to recover relative depth, and this problem of 3-D motion appears to be very sensitive in the presence of small amounts of noise in the input (flow or displacements) [31, 38, 1, 2].

In [31] several experiments as well as comparisons with various algorithms were made and the finding was that an average error of 1% to 2% in the input (retinal correspondence) can create an error of about 100% in the estimated parameters. An important question to ask then is what makes this problem unstable, and to seek ways to address any inherent instabilities that might arise. There is recent work towards this direction but difficult questions still remain.

But while theoretical research on the principles of structure from motion continues in its quest for optimal recovery, we can also follow an alternative approach. We can ask the following simple question: if we had a robust structure from motion module, what would we use it for? The answer of course lies in a taxonomy of visual tasks involving motion, i.e. navigational tasks. A few such *generic* navigational tasks are, for example, the following:

- *Detection of independently moving objects in the environment, by a moving observer.* This is a nontrivial task, as everything moves on an image obtained by a moving observer, thus making it hard to distinguish independent motion. Although many general schemes have been proposed for segmentation of a flow field into areas corresponding to differently moving objects, there are still problems in practical applications involving more than one independently moving object. Other approaches of interest are those that combine measurements of flow with some 3-D interpretation which can then be used for incremental improvements to segmentation in an iterative manner. However, no practical robust system for detecting independently moving objects in general environments and based on optic flow has been demonstrated to date.
- *Passive navigation.* Passive navigation is a term used to describe the processes by which a system can determine its motion with respect to the environment. This is important for kinetic stabilization which, in its simplest form, requires a system to maintain a fixed position and attitude in space in the presence of perturbing influences. More generally, stabilization can refer to any conditions placed on the motion parameters; for instance, the system might be required to translate without rotation. The two abilities are interrelated because stabilization is generally achieved by bringing the motion parameters to certain specified values. The capacity for passive navigation is prerequisite for any other navigational ability. In order to guide the system, some idea of the present motion and some method of setting it to known values must be available. In present robot systems the necessary information is often explicitly available as a result of a built-in coordinate system. For an autonomously moving system, however, there must be an active sensing capacity. It is possible to obtain the required in-

[2] Possible exceptions are photogrammetry and semiautonomous applications requiring a human operator.

formation mechanically as is done by the inertial guidance systems in guided missiles. However, the task can also be performed by visual means and it is this problem that we address here.

- *Obstacle avoidance.* Obstacle avoidance refers, simply, to the ability to utilize sensory information to maneuver in an environment containing physical objects without striking them. This can be considered a second-level ability. It requires some capacity for passive navigation, but little else, and could thus be considered the lowest level of active navigation. This task can be performed non-visually by range sensing methods, and it has been generally proposed that the problem be solved visually with a similar algorithm utilizing depth data from a scene reconstructed by the structure from motion module.
- *Avoidance of collision with a moving object.* A robust structure from motion module can detect the 3-D motion of a moving object, calculate its 3-D position with the aid of a binocular system and predict its three-dimensional trajectory. Thus, it can detect any possibilities for collision, by reconstructing the 3-D trajectory of the moving object.
- *Understanding of relative depth.* Visual motion provides a very rich amount of information about the relative depths of objects in the environment (which object is closer). Clearly, this is one of the outputs of the structure from motion module.
- *Visual pursuit.* A three dimensional visual pursuit system consists of an eye (camera(s)), a subject, an object and a mind. The mind uses information from the eye in order to control the movement of the subject so that it will collide with (intercept, catch) the object. Under the traditional paradigm of considering vision as a recovery problem, visual pursuit is just another application of the structure from motion module. In such a case, the camera would reconstruct the three dimensional positions and motions of the camera, the subject and the object and then this information would be utilized by a planning module to generate correct control of the subject.

Given the lack of success in developing a robust structure from motion module, it would seem reasonable to consider simpler problems. There are visual problems, such as the above, which do not require the full realization of the structure from motion capability, yet which are both nontrivial and possess the sort of environmental invariance that would give them general utility. To consider a few examples from biological navigation, the housefly can maneuver visually in three dimensions in a complex environment without striking obstacles; a number of bees and wasps can recognize and return to a particular location in their environment; and the frog can extend its tongue and catch flying insects. Human beings can also perform such tasks, but obviously they can be performed with far less computational equipment than humans possess. We propose here to consider, in the context of navigational tasks, some of the above problems, more specific and more restricted than the general structure from motion problem, with a view towards producing examples of visual systems that have the potential for robustness. This approach is termed purposive [4].

We show later that specific questions such as the ones above can be answered without having to go through the estimation of optic flow. The derivatives of the image intensity function are enough for the task. The approach taken in this paper calls for the solution of specific visual tasks, such as the ones above, in such a way that the solution does not have more power than it is supposed to have. For example, the procedure that provides relative depth is designed only for that purpose and cannot be used to solve the passive navigation problem, or the problem of 3-D motion estimation. Of course, if information about 3-D motion is known, it can be effectively utilized in the estimation of relative depth, but this is of no concern to us here. When building a system that can deal with visual motion problems, we can visualize it as consisting of many processes working in a cooperative manner to solve various problems. For example, the theories described in this paper could be used to design a process that computes relative depth from image measurements, independently of the process that computes 3-D motion. However, after a number of computational steps, when results about relative depth and 3-D motion become available from the two independent processes, they can be exchanged and the constraints relating them can be effectively utilized so that the results are as consistent as possible.

2 Qualitative Methods

Most visual navigation tasks, including the ones described above, have been considered to be subproblems for the reconstructive school. The connection is a natural one since most of these tasks involve shape and distance relationships between the system and the environment which can be expressed in terms of the quantitative idiom of the reconstructive school. This perception has tended to discourage explicit research on such specific problems by classifying them as special cases of an important general problem. It has also tended to obscure the fact that many of the operations necessary to implement specific visual tasks can be expressed in qualitative terms which are more aptly described in terms of the recognition idiom. Consider, for example, the problem of passive navigation. It is not necessary to know exactly how the system is moving with respect to the environment but only whether it is rotating or translating at all, and if so, in what direction forces must be applied to reduce the motion. In the case of obstacle avoidance, the most relevant information is not the exact distance in centimeters from the observer to each point in the environment, but whether the observer is on a collision course with a nearby obstacle and if so, in which direction it should move to avoid the danger of a crash. The common factor in these examples is that they do not require precise quantitative information and that in each case, the information necessary to carry out the task can be represented in a space having only a few degrees of freedom.

3 Organization of the Paper

We wish to develop the mathematics that will give rise to general solutions to the specific problems of detection of independent motion, passive navigation, relative depth estimation, obstacle avoidance, estimating whether an object is on a collision course with the observer, and visual pursuit using the derivatives of the image as input, as opposed to considering them as applications of the structure from motion module. Section 4 is devoted to the description of the input and Sections 5–9 describe general solutions to the specific tasks mentioned above. Finally, Section 10 is devoted to the presentation of some experimental results. It should be pointed out that here we are mostly interested in the theoretical principles behind these perceptual processes and the geometry of the normal flow. We seek solutions that have uniqueness properties using normal flow as input, since normal flow is well defined, while optic flow is not. Thus, we only present the computational theory behind each process. For various properties of the solutions of the individual problems, a theoretical error analysis and an extensive implementation, see [16, 18, 22, 29]. It will become clear that solving the abovementioned problems using normal flow (which contains less information than optic flow) becomes possible only through the employment of an active visual agent [5]. The reason is, of course, that some of the computational burden is transferred to the activity of the agent.

4 The Input

Our motivation is by now clear. We wish to avoid using optic flow as the input to visual motion tasks. On the other hand, we must utilize some description of the image motion. As such a description we choose the spatial and temporal derivatives $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}$ of the image intensity function $I(x, y, t)$. These quantities define the normal flow at every point, i.e. the projection of the optic flow on the direction of the gradient (I_x, I_y). Clearly, estimating the normal flow is much easier than estimating the actual optic flow. But how is normal flow related to the three-dimensional motion field? Is the normal optic flow field equal to the normal motion field, and under what conditions? This question was first investigated by Verri and Poggio [36]

Let $I(x, y, t)$ denote the image intensity, and consider the optic flow field $\vec{v} = (u, v)$ and the motion field $\vec{\bar{v}} = (\bar{u}, \bar{v})$ at a point (x, y) where the local (normalized) intensity gradient is $\vec{n} = (I_x, I_y)/\sqrt{I_x^2 + I_y^2}$. The normal motion field at point (x, y) is by definition

$$\begin{aligned} \bar{u}_n &= \vec{\bar{v}} \cdot \vec{n} && \text{or} \\ \bar{u}_n &= \frac{(I_x, I_y)}{\sqrt{I_x^2 + I_y^2}} \cdot \left(\frac{dx}{dt}, \frac{dy}{dt}\right) && \text{or} \\ \bar{u}_n &= \frac{\nabla I}{\|\nabla I\|} \cdot \left(\frac{dx}{dt}, \frac{dy}{dt}\right) && \text{or} \\ \bar{u}_n &= \frac{1}{\|\nabla I\|}\left(I_x \frac{dx}{dt} + I_y \frac{dy}{dt}\right) \end{aligned}$$

Similarly, the normal optic flow [21] is

$$u_n = -\frac{1}{\nabla I} I_t$$

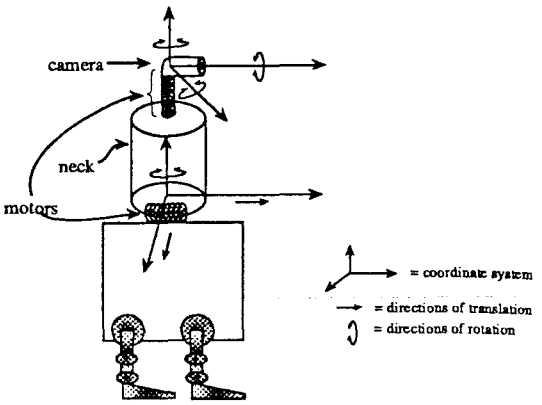


Figure 2: The active observer.

Thus $$\bar{u}_n - u_n = \frac{1}{\nabla I}\frac{dI}{dt}$$

From this equation it follows that if the change of intensity of an image patch during its motion $(\frac{dI}{dt})$ is small enough (which is a reasonable assumption) and the local intensity gradient has a high magnitude, then the normal optic flow and motion fields are approximately equal. Thus, provided that we measure normal flow in regions of high local intensity gradients, the normal flow measurements can safely be used for inferring 3-D structure and motion.

We are now ready to describe our solution to the various motion related tasks. Since the input to the perceptual process is the normal flow, and the normal flow field contains less information than the motion field, in order to solve various problems we need to transfer much of the computation to the activity of the observer [5]. A geometric model of the observer is given in Figure 2. Notice that the camera is resting on a platform ("neck") with six degrees of freedom (actually only one of the degrees is used) and the camera can rotate around its x and y axes (saccades).

5 Passive Navigation

5.1 A qualitative solution

The problem of passive navigation (kinetic stabilization) has attracted a lot of attention in the past ten years [13, 23, 24, 34, 31, 33] because of the generality of a potential solution. The problem has been formulated as follows: Given a sequence of images taken by a monocular observer undergoing unrestricted rigid motion in a stationary environment, to recover the 3-D motion of the observer. In particular, if (U, V, W) and (A, B, C) are the translation and rotation, respectively, comprising the general rigid motion of the observer, the problem is to recover the following five numbers: the direction of translation $(\frac{U}{W}, \frac{V}{W})$ and the rotation (A, B, C) (see Figure 3). The problem has thus been formulated as the general 3-D motion estimation problem (kinetic depth or structure from motion) and its solution would solve several other problems.

Consider a model for a monocular observer as in Figure 3. We assume that the observer moves forward. It

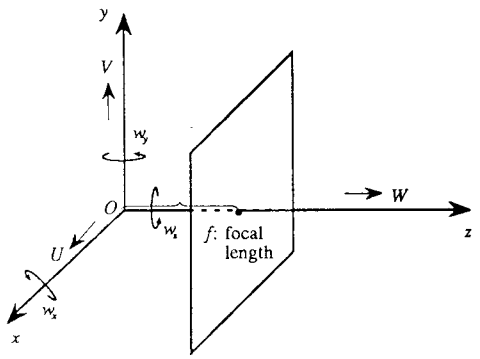


Figure 3: Geometric model of the observer.

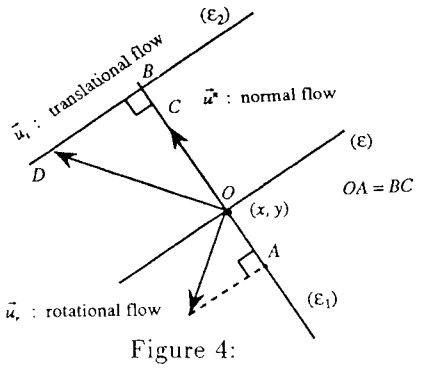


Figure 4:

should be noted that the observer is equipped with inertial sensors which provide the rotation $(A,\ B,\ C)$ of the observer at any time. As the observer moves in its environment, normal flow fields are computed in real time. Since optic flow due to rotation does not depend on depth but on image position $(x,\ y)$, we know (and can compute in real time) its value $(u^R,\ v^R)$ at every image point along with the normal flow.[3] That means that we know the geometrical locus of the optic flow due to translation (see Figure 4). Since the observer moves forward in a static scene, it is approaching anything in the scene and the flow is expanding. From Figure 4, it is clear that the focus of expansion (FOE) $(\frac{U}{W},\ \frac{V}{W})$ (when the gradient space of directions $(\frac{U}{W},\ \frac{V}{W})$ is superimposed with the image space) lies in the half plane defined by line ϵ. Clearly, at every point we obtain a constraint line which constrains the FOE to lie in a half plane. If the FOE lies on the image plane (i.e. the direction of translation is anywhere inside the sector $OABCD$ (Figure 5)) then the FOE is constrained to lie in an area on the image plane and thus it can be localized (see Figure 6). When the FOE does not lie inside the image, a closed area cannot be found, but the votes collected by the half planes indicate its general direction. In this case the observer, with a "saccade" (a rotation of the camera), can bring the FOE inside the image and localize it (Figure 7 explains the process).

[3] If computation of normal flow at some points is unreliable, we just don't compute normal flow there.

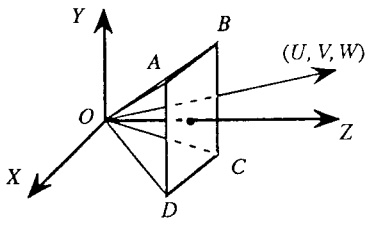
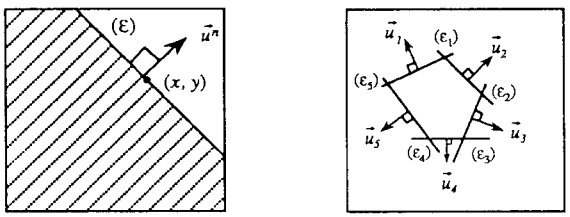


Figure 5: Consider the camera coordinate system. If the translation vector (U, V, W) is anywhere inside the solid $OABCD$ defined by the nodal point of the eye and the boundaries of the image, then the FOE is somewhere on the image.



(a) (b)

Figure 6: (a) From a measurement $\vec{u}$ of the normal flow due to translation at a point (x, y) of the image, every point of the image belonging to the half plane defined by ϵ that does not contain $\vec{u}$ is a candidate for the position of the focus of expansion, and collects one vote. The voting is done in parallel for every image measurement. (b) If the FOE lies within the image boundaries, then the area containing the highest number of votes is the area containing the FOE. Using only a few measurements can result in a large area. Using many measurements (all possible) results in a small area (in our experiments an area of at most three or four pixels).

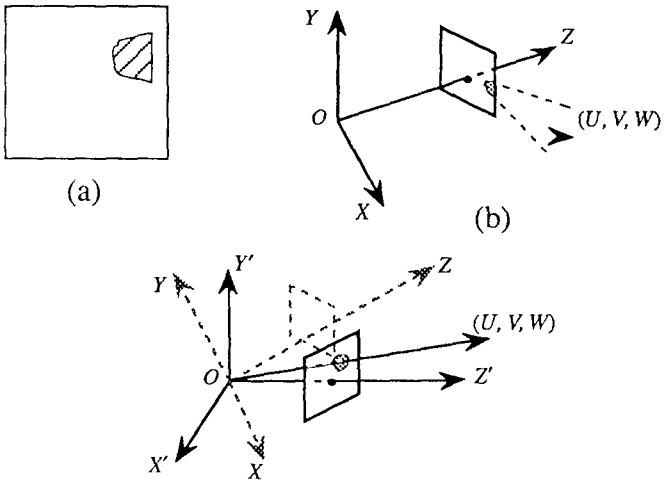


Figure 7: (a) If the area containing the highest number of votes has a piece of the image boundary as part of its boundary, then the FOE is outside the image plane (b). (b) The position of the area containing the highest number of votes indicates the general direction in which the translation vector lies. (c) The camera ("eye") rotates so that the area containing the highest number of votes becomes centered. With a rotation around the x and y axes only, the optical axis can be positioned anywhere in space. The process stops when the highest vote area is entirely inside the image.

5.2 The algorithm

We assume that the computation of the normal flow, the voting and the localization of the area containing the highest number of votes can happen in real time. In this paper we don't get involved with real time implementation issues as we wish to analyze the theoretical aspects of the technique. However, it is quite clear that computation of normal flow can happen in real time (there already exist chips performing edge detection). According to the literature on Hough transforms and connectionist networks [9], voting could also happen in real time. Let S denote the area with the highest number of votes. Let $L(S)$ be a Boolean function that is true when the intersection of S with the image boundary is the null set, and false otherwise. Then the following algorithm finds area S. We assume that the inertial sensors provide the rotation and thus we know the normal flow due to translation.

```
1.    begin {
2.            find area S
3.            repeat until L(S)
4.            { rotate camera around x, y axes
                so that the optical axis passes
                through the center of S (saccade)
5.            find area S
                }
              output S
              }
```

If the camera has a wide angle lens, then image points can represent many orientations, and only one saccade may be necessary. But if we have a small angle lens, then we may have to make more than one saccade.[4]

5.3 Improvement of the solution

It is clear that the technique just described provides as an answer an area on the image containing the FOE. How large or small this area can be depends on the distribution of surface markings and thus on the measured normal flow. If the FOE lies in a featureless area, the resulting area will not be small. For some applications the knowledge of area S might be enough to accomplish the task. We can, however, narrow down a more accurate solution, with S providing one constraint.

Assuming that inertial sensors provide us with the rotation, we can derotate the normal flow field. Thus, assuming a translational normal flow field $v_n(x, y)$, we have: $v_n = u \cdot n_x + v n_y$, where (u, v) is the optic flow and (n_x, n_y) the direction of the gradient at that point. Since we have derotated, the optic flow is

$$u = \frac{U - xW}{V}, v = \frac{V - yW}{Z}$$

and thus

$$v_n = -\frac{W}{Z}(xn_x + yn_y) + \frac{W}{Z}\left(n_x\frac{U}{W} + n_y\frac{V}{W}\right)$$

or

$$\frac{n_x}{v_n}\frac{U}{W} + \frac{n_y}{u_n}\frac{V}{W} - \frac{Z}{W} = \frac{n_x}{v_n}\cdot x + \frac{n_y}{u_n}\cdot y$$

This is a linear equation in the FOE $\left(\frac{U}{W}, \frac{V}{W}\right)$ and the time to collision with every scene point.

If we consider a small image patch P with Z_{av} the average depth of the scene points giving rise to the patch under consideration, then the above equation, for every point $(x_i, y_i) \in P$ with depth z_i, can be written as

$$\frac{n_x}{v_n}\frac{V}{W} + \frac{n_y}{v_n}\frac{V}{W} - \frac{Z_{\text{av}}}{W} = \frac{n_x}{v_n}x + \frac{n_y}{v_n}y + \left(\frac{Z_i}{W} - \frac{Z_{\text{av}}}{W}\right)$$

The expected value of the last term in the above equation is zero, and assuming that we can correctly compute (n_x, n_y) and v_n, equations

$$\frac{n_x}{v_n}\frac{U}{W} + \frac{n_y}{v_n}\frac{V}{W} - \frac{Z_{\text{av}}}{W} = \frac{n_x}{v_n}x + \frac{n_y}{v_n}y$$

at every point $(x, y) \in P$ constitute a linear system in the unknowns $\frac{U}{W}$, $\frac{V}{W}$ and $\frac{Z_{\text{av}}}{W}$. Solving such systems for several patches robustly provides the FOE and a hazard map (showing different time-to-collision values). The patches need at least three normal flow measurements, and so they can be quite small.

5.4 Analysis of the method

We have assumed that the inertial sensors will provide the observer with accurate information about rotation. Although expensive accelerometers can achieve very high accuracy, the same is not true for inexpensive inertial sensors and so we are bound to have some error. Thus we must assume that some unknown rotational part still exists and contributes to the value of the

[4] Up to this point the algorithm is similar to [20]. However, as will become clear later, it works even when rotation is present, while in [20] the solution works only for translational motion.

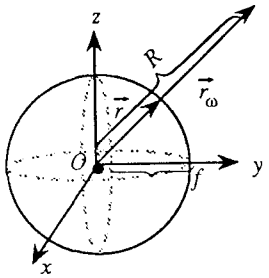


Figure 8:

normal flow. As a result, the method for finding the FOE (previous section) which is based on translational normal flow information (since we have "derotated") might be affected by the presence of some rotational flow. In this section, we study the effect of rotation (the error of the inertial sensor) on the technique for finding the FOE. At the same time *we provide a technique for bounding the FOE given a normal flow field containing both rotation and translation.*

In order to avoid artificial problems introduced by perspective distortions in the case of a planar retina and to simplify the formulas without loss of generality, we employ a spherical retina. Let a sphere with radius f and center O (Figure 8) represent the spherical retina (with O the nodal point of the eye) and a coordinate system $OXYZ$ attached to it. Let

$$\vec{r}_\omega = (X, Y, Z) \text{ be a world point}$$

and $\vec{r} = (x, y, z)$ be its image on the image plane.

Then $$\frac{\vec{r}}{f} = \frac{\vec{r}_\omega}{R},\ R = \|\vec{r}_\omega\| = \sqrt{\vec{r}_\omega \cdot \vec{r}_\omega}$$

In the sequel we derive expressions for optic (normal) flow in the new configuration.

If the velocity of the world point $\vec{r}_\omega$ is given by

$$\dot{\vec{r}}_\omega = -\vec{t} - \vec{\omega} \times \vec{r}_\omega$$

where $\vec{t}$ is translation $(\vec{t} = (U, V, W))$
$\vec{\omega}$ is rotation $(\vec{\omega} = (\omega_x, \omega_y, \omega_z))$

$$\text{then } \dot{\frac{\vec{r}}{f}} = \frac{\dot{\vec{r}}_\omega \cdot R - \vec{r}_\omega \cdot \dot{R}}{R^2}$$

$$\dot{R} = \frac{d}{dt}\left(\sqrt{\vec{r}_\omega \cdot \vec{r}_\omega}\right) = \frac{1}{2R}(\dot{\vec{r}}_\omega \cdot \vec{r}_\omega + \vec{r}_\omega \cdot \dot{\vec{r}}_\omega) = \frac{\dot{\vec{r}}_\omega \cdot \vec{r}_\omega}{R}$$

We have

$$\begin{aligned}\dot{\frac{\vec{r}}{f}} &= \frac{\dot{\vec{r}}_\omega}{R} - \frac{\vec{r}_\omega}{R^3}(\dot{\vec{r}}_\omega \cdot \vec{r}_\omega) \\ &= -\frac{\vec{t}}{R} - \frac{\vec{\omega} \times \vec{r}_\omega}{R} - \frac{\vec{r}_\omega}{R} \cdot \frac{1}{R^2}\left((-\vec{t} - \vec{\omega} \times \vec{r}_\omega) \cdot \vec{r}_\omega\right) \\ &= -\frac{\vec{t}}{R} - \frac{\vec{\omega} \times \vec{r}}{f} + \frac{\vec{r}}{f} \cdot \frac{1}{R}(\vec{t} \cdot \frac{\vec{r}}{f})\end{aligned}$$

or

$$\begin{aligned}\dot{\vec{r}} &= -\frac{\vec{t}f}{R} - \vec{\omega} \times \vec{r} + \frac{\vec{r}}{Rf}(\vec{t} \cdot \vec{r}) \qquad (1) \\ &= \frac{1}{R}\left[-\vec{t}f + \frac{\vec{r}}{f}(\vec{t} \cdot \vec{r})\right] - \vec{\omega} \times \vec{r}\end{aligned}$$

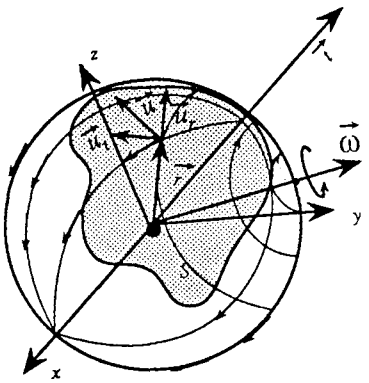


Figure 9:

Thus, the translational flow is

$$\vec{u}_t = \frac{1}{R}\left[-\vec{t}f + \frac{\vec{r}(\vec{t} \cdot \vec{r})}{f}\right]$$

while the rotational flow is given by

$$\vec{u}_R = -\vec{\omega} \times \vec{r}$$

Without loss of generality we can set $f = 1$.

At this point we define two quantities that will be of use later. They are $\tau = \frac{R}{\|\vec{t}\|}$, which we term time to collision, and $k = \frac{\|\vec{\omega}\|}{\|\vec{t}\|}R = \|\vec{\omega}\|\tau$, which represents the effective ratio of rotation and translation.

The geometry of the spherical projection is then given in Figure 9. It has been shown [28] that a full (360°) visual field simplifies motion analysis. However, what we usually have is just a piece of the surface of the sphere (due to a limited field of view). Assume then that the image (the part that we see) is projected on the surface patch S. Obviously, voting for the estimation of the FOE can be performed for all points on S.

5.4.1 Principles of voting

Consider

$\vec{r}_i = (x, y, z)$, a point in S,
$\vec{n}_i = (n_x, n_y, n_z)$, the image gradient direction at point $\vec{r}_i$,
$\dot{\vec{r}}_i = \vec{u}_i = (u_x, u_y, u_z)$, the flow at point $\vec{r}_i$, and
$\vec{u}_i^n = (\vec{n}_i \cdot \vec{u}_i) \cdot \vec{n}_i$, the normal flow at $\vec{r}_i$.

Then (see Figure 10) if $\vec{r} = (x, y, z)$ is a point in S, a feature point $\vec{r}_i$ will vote for $\vec{r}$ being the FOE (direction of translation) iff $\vec{u}_i^n(\vec{r} - \vec{r}_i) < 0$ (see Figure 10).

If $V[\vec{r}]$ represents the number of votes collected at point $\vec{r}$, then it is easy to see that

$$V[\vec{r}] = \sum_{\vec{r}_i \in S} U\left[\vec{u}_i^n(\vec{r}_i - \vec{r})\right]$$

where $U(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$ (Heaviside function)

Let $S' = \left\{\vec{r} | \forall \vec{r'} \in S,\ V[\vec{r}] \geq V[\vec{r'}]\right\}$ be the set of points that have acquired the maximum number of votes. There are two cases:

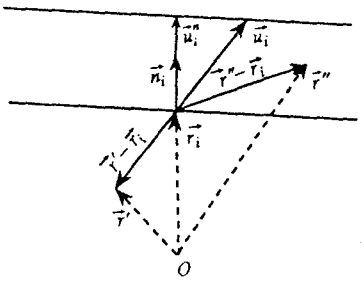


Figure 10:

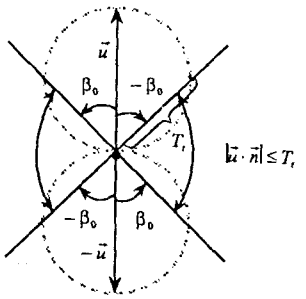


Figure 11:

Case 1: S' does not intersect the border of S, in which case the FOE is in S'.

Case 2: S' touches the border of S, in which case the FOE could be outside S.

It should be clear that if there is no rotation, then S' will always contain the FOE or give the direction of the FOE—i.e. the direction towards which we need to rotate. The size of S' depends on the distribution of features.

We can investigate the performance of the voting scheme in the presence of rotation. In particular we can ask how large area S is when rotation is present. It has been shown that this depends on the angle θ_ω between the direction of translation and the axis of rotation as well as on the rotation-to-translation ratio k. In particular, θ_ω distorts area S' and k enlarges it as it grows. The interested reader can consult [16].

5.4.2 Correctness of voting

The normal flow (as well as the actual flow) is very small in the region close to the FOE, and in the directions close to orthogonal to the directions of the flow. Consequently, even when only translation is present, in order to avoid inaccuracies that might arise in the estimated direction of the normal flow—numerical manipulation of very small quantities is unstable—we are going to discard any normal flow whose magnitude is less than some threshold T_t. Later, it will turn out that choosing this threshold greatly facilitates the geometrical analysis of the technique. Considering an actual flow $\vec{u}$ at a point A (see Figure 11) we can compute the locus of gradient directions $\vec{n}$ along which the normal flow (i.e. the projection of $\vec{u}$ on $\vec{n}$) is bigger than the threshold T_t. In Figure 11 they are all directions inside angle BAC defined by $\beta_0 = \arccos\frac{T_t}{\|\vec{u}\|}$ for $\frac{T_t}{\|\vec{u}\|} \leq 1$, or there are no such directions for $\frac{T_t}{\|\vec{u}\|} > 1$.

We now develop a condition that needs to be satisfied in order for voting at a point to be correct in the presence of rotation.

Voting will clearly be correct only if the direction of the translational normal flow is the same as the direction of the actual normal flow, that is when

$$(\vec{n}\cdot\vec{u}_t)(\vec{n}\cdot\vec{u}) > 0 \qquad (2)$$

In addition, since we consider only normal flows greater than threshold, we need

$$|\vec{n}\cdot\vec{u}| > T_t \qquad (3)$$

Inequality (2) becomes

$$\begin{aligned}(\vec{n}\cdot\vec{u}_t)(\vec{n}\cdot\vec{u}) &= (\vec{n}\cdot\vec{u}_t)(\vec{n}\cdot\vec{u}_t+\vec{n}\cdot\vec{u}_R) = \\ &= (\vec{n}\cdot\vec{u}_t)^2+(\vec{n}\cdot\vec{u}_t)(\vec{n}\cdot\vec{u}_R) > 0\end{aligned} \qquad (4)$$

So, if we set $|\vec{n}\cdot\vec{u}_R| = T_t$, then there are two possibilities: either $|\vec{n}\cdot\vec{u}|$ is below the threshold, in which case it is of no interest to voting, or the sign of $\vec{n}\cdot\vec{u}$ is the same as the sign of $\vec{n}\cdot\vec{u}_t$. In other words, if we can set the threshold equal to the maximum value of the normal rotational flow, *then our voting will always be correct.* But at point $\vec{r}$ of the sphere the rotational flow is

$$\begin{aligned}|\vec{n}\cdot\vec{u}_R| &\leq \|\vec{n}\|\cdot\|\vec{u}_R\| = \|\vec{u}_R\| = \|\vec{\omega}\times\vec{r}\| = \\ &= \|\vec{\omega}\|\cdot\|\vec{r}\|\cdot|\sin(\angle\vec{\omega},\vec{r})| \leq \|\vec{\omega}\|\end{aligned}$$

Thus if we choose $T_t = \|\vec{\omega}\|$, then the sign of $\vec{n}\cdot\vec{u}$ (actual normal flow) is equal to the sign of $\vec{u}_t\cdot\vec{n}$ (translational normal flow) for any normal flow of magnitude greater than T_t.

5.4.3 The case of dominant rotation

Although the technique described in this paper was derived to solve the problem of kinetic stabilization it turns out that it has general applicability. It can be modified to handle the case of dominant rotation with translation.

Consider a pattern of optic flow in the case of pure rotation. On a spherical retina the optic flow will correspond to vectors tangent to the circles around the axis of rotation $\vec{\omega}$. The point at which the axis of rotation passes through the image will be called the AOR. If there is circular optic flow in the image (due to pure rotation) the center of all the circles is the AOR. If we take an arbitrary optic flow vector $\vec{u}_R$ at the point $\vec{r}_i$ then we can say that a point $\vec{r}$ is a candidate for the AOR if

$$(\vec{r}_i\times\vec{u}_R)\vec{r} < 0.$$

This inequality expresses the fact that the feature point and the flow vector at the point span the plane p which cuts the sphere in two hemispheres where one contains all possible candidate points for the AOR (and all of them satisfy the previous inequality). Furthermore, all possible positions of the AOR lie on the great circle which is normal (on the sphere) to the great circle which is the intersection of the plane p and the image sphere. In

other words if we replace $\vec{u}_R$ with the normal flow $\vec{u}_R^n$ the inequality will still hold.

Very similar reasoning applies in the case of a flat retina (perspective projection). Given an optic flow (u, v) at the feature point (x_i, y_i) all possible candidate points for the AOR are on the right of the line passing through (x_i, y_i) and parallel to (u, v). Furthermore, they all lie on the line normal to (u, v) and originating at (x_i, y_i). In other words candidate points (x, y) for the AOR satisfy the inequality

$$((u, v, 0) \times (x - x_i, y - y_i, 0))(0, 0, 1) < 0.$$

This inequality indicates that the z component of the vector product of the optic flow vector and the difference of the candidate AOR point and the feature point must be negative. As in the case of a spherical retina this holds even when the optic flow (u, v) is replaced by the normal flow (u^n, v^n). As was done in the case of translation, voting can be performed. Points with maximum votes are candidates for the AOR. If a minimum is sought then the opposite direction will be found. If the area is closed then the AOR is localized as before; otherwise its general direction will be indicated by the area with maximum votes.

An analysis (on a spherical retina) similar to the one performed for the case of dominant translation can be performed again. This time, however, the threshold should be set to $T_t = \tau = \frac{R}{\|\vec{t}\|}$ (time to collision). If the magnitude of the normal flow is greater then T_t then it must have the same sign (and direction) as rotational normal flow.

When $\vec{\omega}$ and $\vec{t}$ are parallel the angular radius of the uncertainty region is equal to θ_{r_0} where $\cot\theta_{r_0} = \frac{\|\vec{\omega}\|}{\|\vec{t}\|} R$. The difference in the angular radii of the uncertainty areas around the FOE and the AOR is that the tangent is replaced by the cotangent. When $\theta_\omega > 0$ the uncertainty area around the AOR changes shape in a similar manner as the uncertainty area around the FOE. It extends in the direction $\vec{\omega} \times \vec{t}$ with the growth of θ_ω and gets closer to the AOR in the opposite direction.

6 Active Detection of Independent Motion

Among the more significant papers devoted exclusively to detecting moving objects is [32] by Thompson and Pong. It recognizes the difficulty of motion detection using only visual information in the form of optic flow, and considers additional constraints that may have to be applied for motion detection, e.g. knowledge about camera motion, moving object tracking, and information about scene depth. Though it presents a good discussion of the various trade-offs involved, all techniques proposed still depend on the computation of the optic flow.

The two approaches that are closest to the technique described here (emphasizing qualitative techniques for particular situations) are [17] and [27]. In [17] Bhanu et al. identify a fuzzy FOE (see also [14]) and propose a rule-based qualitative analysis of the motion of scene points. However, this requires point correspondences

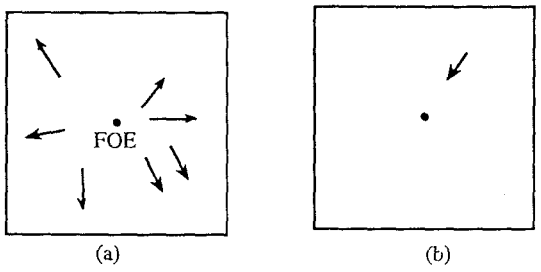


Figure 12: (a) If the observer translates along its optical axis, then the normal flow field has the property that it points away from the origin (FOE) at every point. This normal flow field is as expected and it does not signify independent motion (although it might exist). (b) There exist values of the normal flow that do not point away from the FOE. They are not as expected and thus signify independent motion.

that are difficult to obtain in general and involves considerable "high-level" (and hence expensive) reasoning, which would seem to be inappropriate for the relatively "low-level" task of motion detection. In [27] Nelson gives motion detection techniques based on normal flow and pattern recognition that can be used in situations when the observer motion is specific, and when the object motion changes rapidly in comparison with the changes in camera motion (termed "animate vision"; see also [10]).

The basis of the technique described here lies in deviations from expectations. If the observer moves in a stationary environment then he/she expects to receive a normal flow field that obeys some properties (see Figure 12). If there exist independently moving visible objects in the scene then some of these properties will not hold in parts of the normal flow image; these unexpected "anomalies" signify the existence of independent motion.[5] However, it is possible that the normal flow field appears as expected while there still exists independent motion. In the sequel we will examine the problem in more detail.

The motion field and hence the optic flow is due to the motion of the observer (inducing a flow $\vec{u}^{\text{eg}}$) and the motion of independent objects in view, inducing a flow $\vec{u}^{\text{ind}}$. Then the normal flow at every point is: $v_n = u_n^{\text{eg}} + u_n^{\text{ind}}$, where u_n^{eg}, u_n^{ind} are the normal components of $\vec{u}^{\text{eg}}$ and $\vec{u}^{\text{ind}}$ respectively.

We consider the case where the motion of the observer is translation (if there is rotation, the observer's inertial sensors can provide it; then we can derotate the normal flow field and thus consider only translation). Also, the previous algorithm (Section 5) provides the FOE (or an area containing it). To simplify the exposition we first assume that the FOE is a known point but we can easily generalize to the case where the FOE lies in an area S.

To make the image acquisition active, we assume that the camera can be given very small translation, whose

[5] This principle of deviations from expectations and anomalies is very powerful and can be used in many other situations.

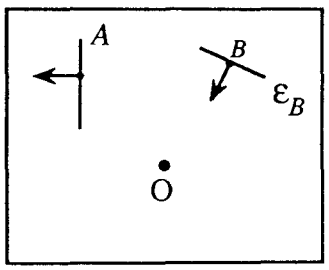


Figure 13:

net result is a momentary shift of the FOE in the image. We also assume that this can be done in a controlled manner, so that the FOE can be moved to a desired position(with a given accuracy). These small shifts are called *jitters*.

The engineering basis for using the "jitter" is that the shift in the FOE helps in motion detection (on the basis of purely geometric considerations, to be discussed), while the fact that it causes only momentary and controlled displacement about an equilibrium position eliminates the need for point correspondence. We assume that the motors responsible for moving the camera have the dynamic control capability needed for producing the jitter. We believe that with a suitably designed camera system this should be possible [6]. We do not concern ourselves here with the details of the motor control involved but consider instead only the effect of the resulting shifts of the FOE. Note that the "jitter" does not effect the dominant motion of the observer (i.e. that of the mobile platform). Thus its effect on the image flow is only "additive". That is, the image flow pattern due to the egomotion is modified by the addition of the flow due to the jitter at a point. This constrains the nature of the changes to the flow that can be brought about, as will be explained later.

6.1 The computational theory

Consider Figure 13, which represents the normal flow at two points A, B with O being the FOE. Clearly, if the normal flow points towards the FOE (i.e. the FOE lies in the half plane defined by the line normal to the flow), then this particular point (B) is moving independently of the sensor. If, however, the normal flow points away from the FOE (as in A), this could be due to egomotion or to a combination of egomotion and independent motion. Thus further constraints need to be applied to always be able to detect independent motion. At this juncture additional information from the image sequence could be used, for example, the value of u_n, but in accordance with our goal of devising a strategy that uses only the "sign" of u_n, we have to define some additional activity that may make motion detection possible. It is easy to see that the following conditions are necessary and sufficient for detecting independent motion at a point for a particular position of the FOE.

(a) $\vec{u}_n^{\text{ind}}$ points toward the FOE.

(b) The length of $\vec{u}_n^{\text{eg}}$ is less than the length of $\vec{u}_n^{\text{ind}}$.

In general, the conditions above will not be satisfied at every point in the image. The only "tool" that we allow ourselves at this point is shifting of the FOE by small "jitters" as explained earlier. The question we ask then is, what *exploratory action* (a sequence of shifts of the FOE) will guarantee motion detection at all points of the image. One condition that any exploratory activity guaranteeing "completeness" will have to satisfy is established by the following observation: If $O_1, O_2, \ldots, O_k$ is a sequence of new FOE locations (formed by an exploratory action), and the convex hull of the set of points $\{O_1 \ldots O_k\}$ encloses the entire region of interest, then complete detection of independent motion is guaranteed. This constitutes a necessary condition for the completeness of detection.

We can also observe that if the region of interest is the entire rectangular image, and the FOE is shifted at least to the four corners of the image, then the necessary condition for guaranteeing detection is satisfied. Up to now, we were mostly concerned with condition (a). Before we establish conditions under which both conditions (a) and (b) are satisfied, let us consider condition (b). If it is violated, then the length of $\vec{u}_n^{\text{eg}}$ is larger than the length of $\vec{u}_n^{\text{ind}}$, or $\|\vec{u}_n^{\text{eg}}\| > \|\vec{u}_n^{\text{ind}}\|$. Any exploratory action (since we have no control over $\vec{u}_n^{\text{ind}}$) would attempt to decrease $\vec{u}_n^{\text{eg}}$. The following two exploratory activities attempt to satisfy condition (b).

If the point of interest is point $A(x_A, y_A)$ then we can either move the FOE close to A, or decrease the angle between the line connecting the FOE to point A and the gradient of the image at point A. The first action decreases the flow due to egomotion (egomotion flow at the FOE is zero) and the second action decreases the normal flow due to egomotion.

6.2 The algorithm

For a typical robot task, detecting the motions of objects that are small, distant, or slow is not very important. On the other hand, detecting the motions of objects that are large, close, or fast may be critical for the robot, and any useful motion detection strategy should guarantee the detection of such motions.[7]

[6] After all, human eyes are perpetually active and can be moved very efficiently with the help of extensive groups of muscles. Any artificial system that purports to emulate human performance—at least in achieving navigational goals (for example)—should have similar "active" capabilities [10].

[7] For example, for safe navigation a mobile robot needs to detect any sharp changes in nearby objects that are large enough to be important (e.g. another robot or a human that may move across its path), while other moving objects may not be of immediate interest if they are distant (e.g. they will not affect the robot's planned path) or if they are too small (e.g. a fly). Of course this may represent only a typical scenario; under other circumstances or for other missions all motion may be critically important, and it would be justified to pay the cost, which consists of increased exploratory activity (careful scanning of the scene) or a decrease in the overall speed of the robot. An analogy from the biological world can easily be made. When a deer or other animal senses danger it slows down or even stops completely and looks around care-

There is obviously a trade-off involved between the activity required and the parameters that describe the sensitivity of motion detection under different conditions. Computation of normal flow proceeds in real time. Normal flows pointing toward the FOE are classified as moving independently. Additional activity by the observer (moving the FOE at least in the four image boundaries) may uncover additional independently moving points.

If we consider a point $A(x, y)$ on the image plane, then the length of the flow $\vec{u}^{\text{eg}}$ is $\|\vec{u}^{\text{eg}}\| = \frac{W}{Z} \cdot r$, where r is the distance of A from the FOE. Assume further that the detection paradigm is such that for the point of interest A at least one position of the FOE lies within distance d from A. That is $r < d$ for point A. The length of the corresponding egomotion flow is thus $\|\vec{u}^{\text{eg}}\| < \frac{W}{Z} \cdot d$ and consequently the normal egomotion component obeys

$$\|u_n^{\text{eg}}\| < \frac{W}{Z} \cdot d$$

For independent motion to be detected, we need both conditions (a) and (b) to be satisfied. One way to guarantee that (a) will be satisfied is to move the FOE to a new position so that A will be inside the segment defined by the two FOE positions. For condition (b) to be satisfied we need (worst case) that

$$\begin{array}{ll} & \|\vec{u}_n^{\text{ind}}\| > \|\vec{u}_n^{\text{eg}}\| \\ \text{so that} & \|\vec{u}_n^{\text{ind}}\| > \frac{W}{Z} d \\ \text{or} & d < \|\vec{u}_n^{\text{ind}}\| \frac{Z}{W} \end{array}$$

If the above inequality is satisfied, then we are guaranteed to detect independent motion at point A.

In the above inequality, d basically represents the cost involved in detecting motion using an exploratory strategy that guarantees detection. Obviously the cost of the exploration decreases (i.e. d increases) when the time to collision with the environment is small (large depth, small W). On the other hand, if $\|\vec{u}\|$ is the smallest retinal motion (due to independent 3-D motion) that can be detected, then

$$d = \|\vec{u}\| \cdot \frac{Z}{W}.$$

If $\|\vec{u}^{\text{ind}}\| < \|\vec{u}\|$, then there is no guarantee of detection.

This formalizes the earlier intuitive discussion and also indicates a way to control the performance of the motion detection strategy. At any stage a higher precision (lower $\|\vec{u}\|$) can be achieved without changing the exploratory action (parameterized by d), but by decreasing the dominant speed of the robot.

When the purpose of motion detection is to serve as an early warning system to detect independently moving objects in the scene it is not necessary to guarantee the detection of all moving (feature) points. The detection of a few moving points (that satisfy some criteria, to eliminate "false alarms") should suffice since it can trigger a more detailed analysis (perhaps over a narrower region), again depending on the task at hand. We show how the cost of motion detection may be dramatically reduced when the requirement is to guarantee detection of a compact moving object of at least a minimum projected size. In particular, the cost of the exploratory activity can be linked to the minimum (image) diameter of the objects of interest.

As discussed earlier, it may be reasonable to assume that the boundary of the image of a compact object in the scene forms a closed contour. In particular, this implies that all the points on the boundary of the object are features, and would be successful candidates for our motion detection paradigm provided the projected motion $\vec{u}^{\text{ind}}$ is sufficiently large ($\|\vec{u}^{\text{ind}}\| > \|\vec{u}\|$). We define the diameter ϵ of an arbitrary object as the diameter of the largest circle that can be inscribed in the closed contour that forms the boundary of the projected image of the object. Now it becomes clear that any exploratory paradigm that "covers" the image so that it guarantees the detection of all points distance ϵ apart will guarantee the detection of at least a few points on an object having features. The points are guaranteed to be detected by an appropriate sequence of FOE shifts as discussed earlier. Moreover, because the boundaries of objects are locally smooth, the points thus detected will be clustered together, so that it may be possible to eliminate false alarms arising due to various noise sources that result in isolated points appearing to have independent motions. In practice, the presence of larger features on the objects, e.g. lines separating regions, would make the detection even easier. Thus the effort in the exploratory activity can be reduced when the objects of interest have image diameters greater than some threshold and when there need be no guarantee of detecting objects having projected diameters below that threshold. This is most appropriate when the purpose of the robot is such that larger and nearer objects are more interesting than smaller and farther objects, as may be true for many typical robot tasks such as safe navigation in a dynamic environment. However, at any stage the precision of the detection can be increased by decreasing the diameter and threshold using closer FOE shifts in the exploratory action.

7 Estimating 3-D Motion

Assume an observer imaging an object moving in an unrestricted rigid manner. The motion of the object can be described as the sum of a rotation plus a translation. We can choose a point through which the rotation axis passes; this gives a unique rotation and translation describing the rigid motion (in general there are infinitely many combinations of rotations and translations describing the same rigid motion). In many visual tasks we are only interested in the translation of the moving object and we need no information about how it rotates around itself. This section describes how we can estimate the direction of the object's translation without being able to recover the rotation using a technique based on normal flow. Assume that the object is translating with velocity $\mathbf{V} = (U, V, W)$ and rotating with angular velocity $\Omega = (A, B, C)$ around a point $P = (X_0, Y_0, Z_0)$ on the

fully, alert to the slightest movement (and may "jump" even for a falling leaf), whereas normally it is less sensitive to the motions around it. It would be desirable to equip a robot with a similar mechanism for motion detection that would have a variable level of sensitivity.

object. Point P is on the object and its exact choice will be made clear later.

Point P is visible in the image ($p = (x_0, y_0)$) and we attach a coordinate system onto the object at point P with axes parallel to the observer's coordinate system. We express the motion of the object in this "object-based" coordinate system. The velocity of a point Q on the object is

$$\mathbf{V} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \Omega X \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}$$

Then the normal flow v_n along direction (n_x, n_y) at point (x, y) is $v_n = un_x + vn_y$

where (u, v) is the motion field. Expressing (u, v) in terms of 3-D motion we get

$$\begin{aligned} v_n = \frac{W}{Z}\left(n_x\frac{U}{W} + n_y\frac{V}{W}\right) - \frac{W}{Z}(n_x x + n_y y) \\ -A(y - y_0)(xn_x + yn_y) + \\ +B(x - x_0)(xn_x + yn_y) + \\ +C[(x - x_0)n_y - (y - y_0)n_x] + \\ +\frac{Z - Z_0}{Z}(Bn_x - An_y), \end{aligned}$$

or $$\frac{n_x}{v_n}\frac{U}{W} + \frac{n_y}{v_n}\frac{V}{W} - K\frac{Z}{W} = \frac{n_x}{v_n}x + \frac{n_y}{v_n}\cdot y$$

where

$$\begin{aligned} k = 1 \;&+\; A(y - y_0)\left(x\frac{n_x}{v_n} + y\frac{n_y}{v_n}\right) \\ &-\; B(x - x_0)\left(x\frac{n_x}{v_n} + y\frac{n_y}{v_n}\right) \\ &-\; C(x - x_0)\frac{n_y}{v_n} \\ &+\; C(y - y_0)\frac{n_x}{v_n} \\ &-\; \frac{Z - Z_0}{Z}\left(B\frac{n_x}{v_n} - A\frac{n_y}{v_n}\right) \end{aligned}$$

Consider a small patch of the image around point $p = (x_0, y_0)$ and let us assume that the average depth there is Z_{av}. If we add the quantity $k\left(\frac{Z}{W}\right) - \frac{Z_{\text{av}}}{W}$ to both sides of the above equation, we get

$$\frac{n_x}{v_n}\frac{U}{W} + \frac{n_y}{v_n}\frac{V}{W} - \frac{Z_{\text{av}}}{W} = \frac{n_x}{v_n}x + \frac{n_y}{v_n}y + \left(k\frac{Z}{W} - \frac{Z_{\text{av}}}{W}\right)$$

One can verify that the mean of the last term in the equation above is zero (assuming that the mean of x is x_0 and of y, y_0).

We can thus consider several linear equations:

$$\frac{n_x}{v_n}\frac{U}{W} + \frac{n_y}{v_n}\frac{V}{W} - \frac{Z_{\text{av}}}{W} = \frac{n_x}{v_n}x + \frac{n_y}{v_n}y$$

in the neighborhood around P. Solution of the system provides the FOE.

The reader must have realized that it was the choice of the coordinate system in which we expressed the motion that allowed us to isolate the translational part of the problem. Since P is the center of rotation, the rotational flow at point $p = (x_0, y_0)$ is zero. In other words the above equation is exact at point (x_0, y_0) and approximate in its neighborhood. The error terms, however, have zero mean. This provides the potential for robust estimation. The time to collision is also estimated.

It is, however, clear that the technique for addressing the passive navigation problem (Section 5) cannot be used for the 3-D motion of an object estimation problem (while they are both the same problem if considered as general recovery problems). For example, voting for the values of normal flow produced by the motion of an object can provide a very large solution area.

8 Obstacle Avoidance—Relative Depth

One of the most elementary forms of navigation is obstacle avoidance by a moving, compact sensor. It is a prerequisite, however, for many more complex abilities since any system performing a more complicated task must avoid obstacles in the process. Obstacle avoidance is thus one specific problem for which a general solution is highly desirable. In this context, a general solution refers to a system that works effectively in a wide range of real environments. This implies, among other things, that the system performance does not depend upon artificial constraints on the nature of objects in the environment such as assuming planar or smoothly curved surfaces, rigid or unmoving objects, mathematically uniform textures, and so forth.

The concept of "obstacleness" is a relative one. When we move about in our environment, every object might represent a potential obstacle, depending on its position and our direction of motion, and depending on its size. In addition, time plays an important role. When we move towards a building, the building itself represents a potential obstacle if our intent was to go beyond it. In other words, an object represents an obstacle if the observer is on a collision course with it, its size is comparable to the observer's size and the time to collision is smaller than some value which depends on the particular aspects of the problem under consideration.

Thus, we consider the problem of obstacle avoidance as synonymous to the problem of computing the times to collision to different parts of the scene, or finding relative depth at places of interest. This section is devoted to computing time to collision and relative depth from normal flow. The technique of the previous section will be used. We consider the most general case, where an object in view is moving rigidly (rotation plus translation).

8.1 Computing time to collision for a moving object

Recalling the last equation of the previous section, which is exact at the position $p = (x_0, y_0)$, we have

$$v_n = \frac{W}{Z}\left(n_x\frac{U}{W} + n_y\frac{V}{W}\right) - \frac{W}{Z}(x_0 n_x + y_0 n_y)$$

with all terms defined as previously. If $\frac{U}{W}, \frac{V}{W}$ is already computed, the quantity $\frac{Z}{W}$ is directly available.

8.2 Computing relative depth

Assume two objects A and B moving in a rigid manner, while an active observer has the task of finding which one is closer. The camera is active and can undergo a short abrupt motion along its optical axis. Let us assume that at time t_1 the camera is stationary and then it moves with velocity W_c at time t_2. Assuming that the velocities of the two objects remain unchanged during the time interval $[T_1, t_2]$, we obtain (as in Section 7) for times t_1 and t_2

$$\frac{Z^A_{av}}{W_A} = A_1 \qquad \frac{Z^B_{av}}{W_B} = B_1$$
$$\frac{A^A_{av}+W_A dt}{W_A+W_C} = A_2 \qquad \frac{Z_{av}+W_B dt}{W_B+W_C} = B_2$$

where A_1, B_1, A_2, B_2 are known.

From these equations, assuming that dt is very small, we obtain

$$\frac{Z^A_{av}}{Z^B_{av}} = \frac{A_1 A_2 (B_2 - B_1)}{B_1 B_2 (A_2 - A_1)},$$

and hence relative depth.

9 Visual Pursuit [8]

In a general three dimensional visual pursuit system, we find an agent, whose motion is under the control of our system; a camera, which is used to generate useful visual information to control the agent; and an object, which may be moving. If we could find the three dimensional positions and motion parameters [19] [25] of the camera, the object, and the agent, it would be a simple arithmetic problem to predict and guide the collision of the agent with the object. However, we shall show here that it is not necessary to recover these parameters.

When we try to solve the visual pursuit problem through 3D recovery, we estimate much more than we actually need in order to perform this generic visual task. Taking a purposive viewpoint we develop a robust, qualitative solution to the problem that does not require correspondence or full 3D recovery.

There are two general cases of the pursuit problem. The camera can be mounted separately from the agent and the object, or the camera can be mounted on the agent or the object. In a situation where a human agent pursues a flying ball, both of these problems are involved. The "camera" is mounted on the agent (the human's body) which is intended to collide with the object. When the human is sufficiently close to the ball, the "camera", which is mounted on the head, is independent of the agent (the hand, possibly carrying a tool such as a bat), and the hand is to collide with the ball. Thus the solution of both problems would provide a theoretical basis for an integrated mobile "hunting" system, or for a baseball player!

From a mathematical viewpoint it is equivalent whether the camera is mounted on the agent or the camera is mounted on the object. In such systems the collision is solely determined by the relative motion of the object and the agent, and it is equivalent whether we are controlling the motion of the entity that the camera is mounted on, or the entity that is moving separately from the camera. However, they may have different applications. An example of the case when the camera is mounted on the agent is an airplane that is attacking a target. An example of the case when the camera is mounted on the object is a camera that is guiding a plane to land near the camera.

Let us assume a Cartesian coordinate system with its origin at the focus of the camera, with the z-axis pointing towards the general direction of the agent and the object, such that both the object and the agent are in the full view of the camera.

Assume that the agent is located at $(X_s, Y_s, Z_s)^T$ with a velocity of $(V_{xs}, V_{ys}, V_{zs})^T$, and the object is located at $(X_o, Y_o, Z_o)^T$ with a velocity of $(V_{xo}, V_{yo}, V_{zo})^T$. If the agent or the object is also rotating at the time, we can choose the rotation axis to go through visible points on the surface of the agent or the object, chosen such that the rotation parameters are irrelevant in the prediction and guidance of a collision. However, for simplicity we assume that the motion is instantaneously translational. In the general case the analysis remains essentially the same, but the formulae become more complicated.

The agent and the object will collide after time t provided that

$$t = \frac{X_s - X_o}{V_{xo} - V_{xs}} = \frac{Y_s - Y_o}{V_{yo} - V_{ys}} = \frac{Z_s - Z_o}{V_{zo} - V_{zs}} > 0 \qquad (5)$$

If the projection of the agent (i.e. a point of it) on the image plane is (x_s, y_s), and the projection of the object is (x_o, y_o), assuming unit focal length and perspective projection, we have

$$x_s = \frac{X_s}{Z_s} \qquad (6)$$
$$y_s = \frac{Y_s}{Z_s} \qquad (7)$$
$$x_o = \frac{X_o}{Z_o} \qquad (8)$$
$$y_o = \frac{Y_o}{Z_o} \qquad (9)$$
$$v_{xs} = \frac{V_{xs}}{Z_s} - x_s \frac{V_{zs}}{Z_s} \qquad (10)$$
$$v_{ys} = \frac{V_{ys}}{Z_s} - y_s \frac{V_{zs}}{Z_s} \qquad (11)$$
$$v_{xo} = \frac{V_{xo}}{Z_o} - x_o \frac{V_{zo}}{Z_o} \qquad (12)$$
$$v_{yo} = \frac{V_{yo}}{Z_o} - y_o \frac{V_{zo}}{Z_o} \qquad (13)$$

where (v_{xs}, v_{ys}), (v_{xo}, v_{yo}) is the flow produced by the agent and the object at points (x_s, y_s) and (x_o, y_o), respectively. Combining (6–9) with (5), we obtain the following relation for the prediction of collision:

$$t = \frac{x_s Z_s - x_o Z_o}{V_{xo} - V_{xs}} \qquad (14)$$
$$= \frac{y_s Z_s - y_o Z_o}{V_{yo} - V_{ys}} \qquad (15)$$
$$= \frac{Z_s - Z_o}{V_{zo} - V_{zs}} \qquad (16)$$

[8] This section demonstrates that depth recovery is not necessary for motion coordination problems.

$$> \quad 0 \tag{17}$$

We call (14–17) the *Visual Constraints of Collision*. The visual pursuit problem is solved if we can guide the system to satisfy these constraints. Using the processes of Sections 7 and 8 we can estimate the locomotive intrinsics (i.e. the direction of translation and the time of collision). In what follows, we solve the visual pursuit problem in the case when the camera is mounted on the object, using only the signs of the three locomotive intrinsics, and then we present a solution in the case when the camera is mounted separately to supervise the agent, using the locomotive intrinsics, relative depth, and the direction of motion.

9.1 Camera mounted on the object

The problem is equivalent whether the camera is mounted on the agent or on the object. For simplicity here we assume that the camera is mounted on the object and that we can control the velocity of the agent. We choose a Cartesian coordinate system with its origin at the focus of the camera, and with its z-axis pointing towards the general direction of the agent, such that the agent is in the full view of the camera.

As the camera is mounted on the object, the coordinates of the object on the image plane are zero, as is its velocity. We have $(X_o, Y_o, Z_o)^T = 0$ and $(V_{xo}, V_{yo}, V_{zo})^T = 0$, as well as $(x_o, y_o) = 0$ and $(v_{xo}, v_{yo}) = 0$. In the following, when we write Z_o and Z_s, we always mean $E(Z_o)$ and $E(Z_s)$ (i.e. the average depth around the neighborhood), unless otherwise specified. Thus (14–17) can be simplified to

$$t = -\frac{x_s Z_s}{V_{xs}} \tag{18}$$

$$= -\frac{y_s Z_s}{V_{ys}} \tag{19}$$

$$= -\frac{Z_s}{V_{zs}} \tag{20}$$

$$> \quad 0 \tag{21}$$

From (18) and (20) we have $x_s = V_{xs}/V_{zs}$. From (19) and (20) we have $y_s = V_{ys}/V_{zs}$. Thus if we draw a line from the origin through the focus of expansion $(V_{xs}/V_{zs}, V_{ys}/V_{zs})$, or the first two locomotive intrinsics, on the image plane, we have a set of all the points that will collide with the origin. In order to collide the agent with the object, we should control the motion of the agent so that the focus of expansion lies inside the image of the agent. The third locomotive intrinsic Z_s/V_{zs} is the negative of the time to collision (see (21)). Note that since $t > 0$, the third locomotive intrinsic should be negative for the collision to occur. In this case, $V_{zs} < 0$, that is the agent should be coming closer to the camera.

Since we have an active camera, for simplicity we can rotate the camera such that the image of the agent will be in the center. The agent will collide with the object if we can keep the focus of expansion at the origin and keep an expanding pattern of normal flows.

If the focus of expansion is not at the origin, we can devise a control strategy to guide the focus of expansion towards the origin of the image plane according to the signs of the three locomotive intrinsics, indicating whether the velocity of the agent needs to be increased or decreased at any time instant.

- If $Z/V_z < 0$ and $V_{xs}/V_{zs} = V_{ys}/V_{zs} = 0$, a collision will occur.
- If $Z/V_z = 0$, a collision has occurred;
- If $Z/V_{zs} > 0$, the agent is going away. Decrease V_{zs} and
  - If $V_{xs}/V_{zs} = 0$, do not change V_{xs};
  - If $V_{xs}/V_{zs} > 0$, decrease V_{xs};
  - If $V_{xs}/V_{zs} < 0$, increase V_{xs};
  - If $V_{ys}/V_{zs} = 0$, do not change V_{ys};
  - If $V_{ys}/V_{zs} > 0$, decrease V_{ys};
  - If $V_{ys}/V_{zs} < 0$, increase V_{ys};
- If $Z/V_{zs} < 0$, the agent is coming closer. Do not change V_{zs} and
  - If $V_{xs}/V_{zs} = 0$, do not change V_{xs};
  - If $V_{xs}/V_{zs} > 0$, increase V_{xs};
  - If $V_{xs}/V_{zs} < 0$, decrease V_{xs};
  - If $V_{ys}/V_{zs} = 0$, do not change V_{ys};
  - If $V_{ys}/V_{zs} > 0$, increase V_{ys};
  - If $V_{ys}/V_{zs} < 0$, decrease V_{ys}.

This constitutes a qualitative paradigm for colliding the agent with the object when the camera is mounted on the object. We only use the sign of the three locomotive intrinsics. We can predict the collision, and if a collision will not occur we qualitatively control the velocity of the agent towards a state such that the agent will collide with the object.

9.2 Camera mounted separately

When the camera is mounted separately, the camera may be stationary or in motion relative to the world coordinate system. But for simplicity, we choose a coordinate system with its origin at the focus of the camera and its z-axis pointing towards the general direction of the agent and the object such that both the agent and the object are in full view of the camera. In this coordinate system, the camera is stationary, and velocity is measured relative to the camera.

9.2.1 Object coming towards the camera

The special case when the object is coming towards the camera may need to be handled differently. If we can correctly identify cases when the object is coming towards the camera, we may want to move the camera away from the pathway of the object and then proceed as usual, or when the object is small and is not destructive, we may just put the agent in front of the camera to receive the object.

We can detect whether the object is coming towards the camera using the analysis in the previous section. When the object is coming towards the camera, the focus of expansion of the object lies inside the image of the object and the third locomotive intrinsic is negative.

If we send the agent to the front of the camera, we have the case studied in the last section. It can be determined

from the time to collision of the agent and the object whether the agent is moving fast enough to intercept the object.

In the following general analysis, we assume that the object is not coming towards the camera, but moving in any other direction.

9.2.2 General case of camera mounted separately

From (14–15), we obtain

$$((x_s V_{yo} - y_s V_{xo}) - (x_s V_{ys} - y_s V_{xs}))Z_s - ((x_o V_{yo} - y_o V_{xo}) - (x_o V_{ys} - y_o V_{xs}))Z_o = 0 \quad (22)$$

Note that if (x_s, y_s), $(V_{xs}/V_{zs}, V_{ys}/V_{zs})$, (x_o, y_o), and $(V_{xo}/V_{zo}, V_{yo}/V_{zo})$ are a group of parallel vectors, (22) will be satisfied. Thus in the general case of a separately mounted camera, we first obtain the direction of motion of the object; then we rotate the z-axis of the camera such that it will be in the direction of the object. Then we can move the agent in the direction parallel to the direction of motion of the object. This is a group of sufficient conditions for the collision of the agent and the object when we have good control of the original position of the agent.

To satisfy (16–17), we need to find the time to collision t and make it equal to (16). Combining (10–13) with (14–17), we obtain

$$t = \frac{x_s - x_o \frac{Z_o}{Z_s}}{(v_{xo} + x_o \frac{V_{zo}}{Z_o})\frac{Z_o}{Z_s} - (v_{xs} + x_s \frac{V_{zs}}{Z_s})} \quad (23)$$

$$= \frac{y_s - y_o \frac{Z_o}{Z_s}}{(v_{yo} + y_o \frac{V_{zo}}{Z_o})\frac{Z_o}{Z_s} - (v_{ys} + y_s \frac{V_{zs}}{Z_s})} \quad (24)$$

$$= \frac{1 - \frac{Z_o}{Z_s}}{\frac{V_{zo}}{Z_o} \cdot \frac{Z_o}{Z_s} - \frac{V_{zs}}{Z_s}} \quad (25)$$

$$> 0 \quad (26)$$

If we can find a point on the agent and a point on the object which have the same normal direction (n_x, n_y), from (23–24) we find the time to collision as follows:

$$t = \frac{n_x x_s + n_y y_s - (n_x x_o + n_y y_o)\frac{Z_o}{Z_s}}{(v_{no} + (n_x x_o + n_y y_o)\frac{V_{zo}}{Z_o})\frac{Z_o}{Z_s} - (v_{ns} + (n_x x_s + n_y y_s)\frac{V_{zs}}{Z_s})} \quad (27)$$

Similar equations can be obtained if we can find two normal directions from the agent and the object which are perpendicular. Combining with (25), we have

$$\frac{V_{zs}}{Z_s} = \frac{V_{zo}}{Z_o} - \frac{(\frac{Z_s}{Z_o} - 1)(v_{no}\frac{Z_o}{Z_s} - v_{ns})}{n_x(x_s - x_o) + n_y(y_s - y_o)} \quad (28)$$

Thus, control of the agent is achieved by varying V_{xs}/V_{ys} in order to satisfy (22) and V_{zs} in order to satisfy (26) and (28). According to these equations, we can devise a system for qualitative control of the motion of the agent, so that the agent will collide with the object. This scheme can be accomplished through six sequential phases as follows. The first three phases are devised to satisfy (22). The next two phases are devised to satisfy (26) and (28). The last phase tests to see if the agent will collide with the object without further control of the agent.

1. Rotating the camera. Through the point (x_o, y_s) draw a line in the image plane with direction $(V_{xo}/V_{zo}, V_{yo}/V_{zo})$.
   - If the line goes through the origin, proceed to the next phase;
   - If the origin is on the lower left portion of the image plane, rotate the camera up and to the right;
   - If the origin is on the upper right portion of the image plane, rotate the camera downward and to the left;
2. Position the agent.
   - If the line drawn above goes through the agent, proceed to the next phase;
   - If the agent is on the lower left portion of the image plane, move the agent up and to the right;
   - If the agent is on the upper right portion of the image plane, move the agent downward and to the left;
3. Move the agent parallel to the image plane. Change the velocity of the agent, such that $V_{xs}/V_{ys} = x_s/y_s$.
   - If the agent is on the left of the object, V_{xs} and V_{ys} should be increased;
   - If the agent is on the right of the object, V_{xs} and V_{ys} should be decreased;
   - If the agent and the object collide on the image plane, do not change V_{xs} and V_{ys};
4. Positive time to collision. Proceed to the next phase after:
   - If $Z_o/Z_s = 1$, do not change V_{zs} at present;
   - If $Z_o/Z_s > 1$, adjust V_{zs} such that
   $$\frac{V_{zs}}{Z_s} > \frac{Z_o}{Z_s} \cdot \frac{V_{zo}}{Z_o};$$
   - If $Z_o/Z_s < 1$, adjust V_{zs} such that
   $$\frac{V_{zs}}{Z_s} < \frac{Z_o}{Z_s} \cdot \frac{V_{zo}}{Z_o};$$
5. Move the agent perpendicular to the image plane.
   - If (28) is satisfied, proceed to the next phase.
   - If V_{zs}/Z_s is larger in (28), decrease V_{zs};
   - If V_{zs}/Z_s is smaller in (28), increase V_{zs};
6. Predicting collision. The agent will collide with the object if the following conditions are met, or otherwise repeat from phase 1:
   $$\frac{x_o}{y_o} = \frac{\frac{V_{xo}}{V_{zo}}}{\frac{V_{yo}}{V_{zo}}} = \frac{x_s}{y_s} = \frac{\frac{V_{xs}}{V_{zs}}}{\frac{V_{ys}}{V_{zs}}}$$
   $$(\frac{Z_o}{Z_s} - 1)(\frac{V_{zs}}{Z_s} - \frac{Z_o}{Z_s} \cdot \frac{V_{zo}}{Z_o}) > 0$$
   and
   $$\frac{V_{zs}}{Z_s} = \frac{V_{zo}}{Z_o} - \frac{(\frac{Z_s}{Z_o} - 1)(v_{no}\frac{Z_o}{Z_s} - v_{ns})}{n_x(x_s - x_o) + n_y(y_s - y_o)}$$

In summary, in the case when the camera is mounted on the the object or the agent, we have devised a qualitative strategy to predict and guide the collision of the agent and the object. We only used the signs of the three locomotive intrinsics (FOE and time to contact) to qualitatively control the velocity of the agent such that the visual constraints of collision will be satisfied.

In the case when the camera is mounted separately to supervise the agent to collide with the object, we have devised a set of sufficient conditions to satisfy the visual constraints of collision. This set of sufficient conditions can be reached by a qualitative scheme of control without any exact 3D depth or velocity information of the agent and the object.

Our method can also be used to control the collision even when the object is rotating in addition to having instantaneous translational motion.

10 Recapitulation and Experiments

We have presented solutions to several problems related to visual motion using normal flow as the input. Although we have not solved the general structure from motion (sfm) problem using normal flow, we have presented solutions to various important problems that are simple applications of the sfm module. The robustness of the proposed algorithms relies heavily on the robustness of the computation of normal flow, i.e. spatiotemporal derivatives of the image intensity function. But even without using any elaborate schemes for computing the normal flow (after all, some of the techniques presented only require its sign) we have performed several experiments. We report here a few of them:

(a) **Egomotion estimation**

We have performed several experiments with both synthetic and real image sequences in order to demonstrate the stability of our method. From experiments on real images it was found that in the case of pure translation or pure rotation the method computes the focus of expansion or the axis of rotation very robustly. In the case of general motion it was found from experiments on synthetic data that the behavior of the method is as predicted by our theoretical analysis (see [16]).

Figure 14 shows one of the images from a dense sequence collected in our laboratory using an Merlin American Robot arm that translated while acquiring images with the camera it carried (a Sony miniature TV camera). Figure 15 shows the last frame in the sequence and Figure 16 shows the first frame with the solution area (where the FOE lies), which agrees with the ground truth. Figures 17 and 18 show the first and last frames in a sequence of images collected through a rotation of the sensor and provided by the University of Massachusetts for the Workshop. Figure 19 shows the first frame of the sequence with the solution area for the AOR.

(b) **Detection of independent motion**

Figure 20 shows the experimental setting for testing the algorithm for motion detection from a translating, active camera. The CCD camera is mounted on a slide to simulate pure translation, and can be given small rotations around a revolving platform to simulate the exploratory activity. The model board simulates an out-

Figure 14:

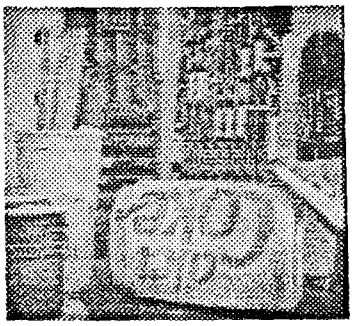


Figure 15:

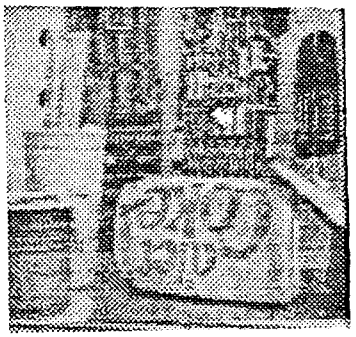


Figure 16:

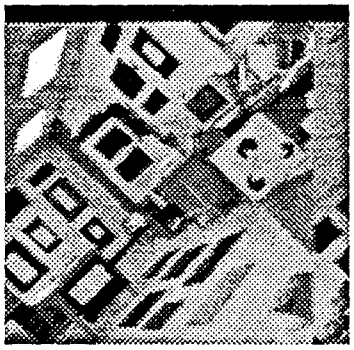


Figure 17:

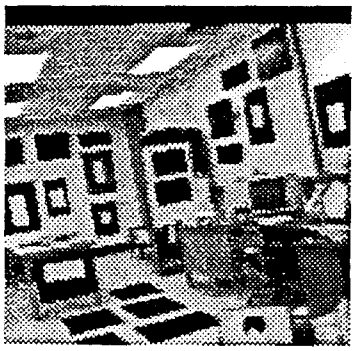


Figure 18:

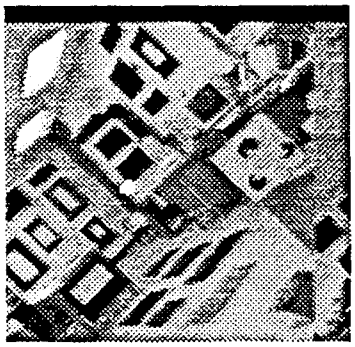


Figure 19:

door scene. The image sequence is captured by Data Translation QuickCapture on a Macintosh IIci. Figures 21(a)–21(d) show the results of the experiment: (a) is a sequence of closely sampled images taken from a moving and active camera; (b) shows the output of the motion detection algorithm without any exploratory activity; (c) shows the output after four shifts of the FOE as part of a simple exploratory activity; and (d) shows the motion detection output (dark) overlaid on the image (light).

(c) Relative Depth

We have performed several experiments on both synthetic and real data in order to test the feasibility and stability of our approach. We report here some experiments on real data. The setup for our experimental work with real images consists of a CCD camera mounted on a slide so that it can purely translate along its optical axis. The camera is viewing a scene consisting of a toy ("Mrs. Potatohead") and a toy robot arm (Radio Shack). The arm (carrying the "vision" of Mrs. Potatohead) is initially placed closer to the camera. Figures 22 and 23 are taken with the camera stationary and the arm moving toward Mrs. Potatohead. Figure 24 shows the normal flow produced from the motion of the arm, using the straightforward gradient technique [19]. Figure 25 is taken after the camera has moved forward and Figure 26 shows the normal flow produced.

Using the algorithms in Sections 5 and 7, we estimated the relative depth of the toy and the arm. We computed the quantity Z/V_c, where Z is the depth of a point and V_c is the speed of the camera, and considered the median value for the arm and the toy. It was found that this

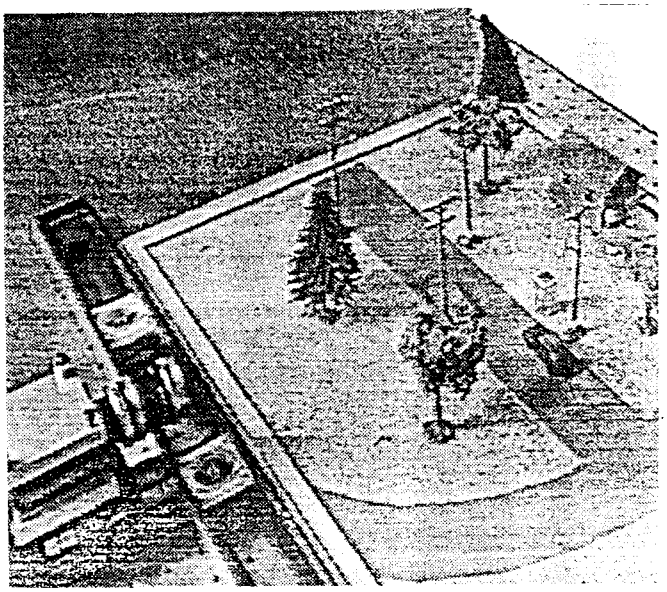


Figure 20:

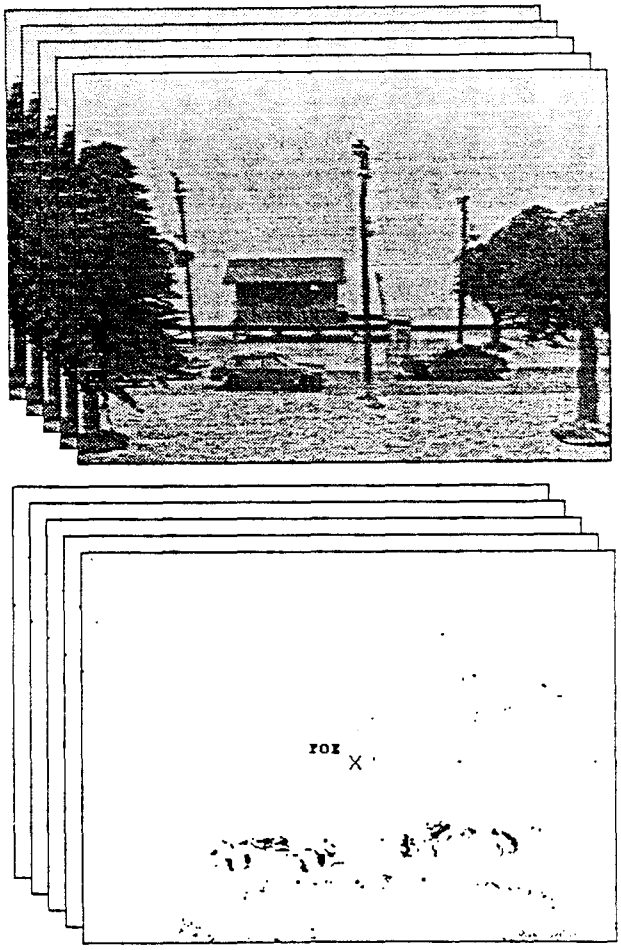
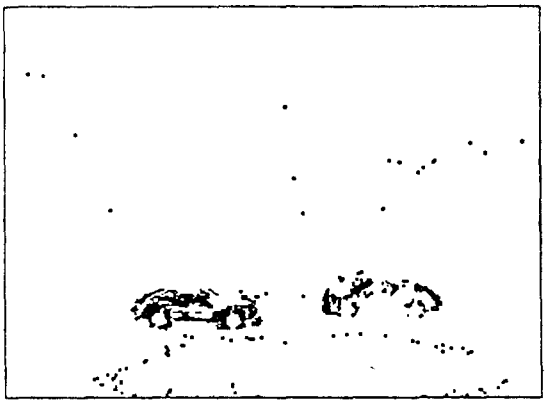


Figure 21: (a) and (b)



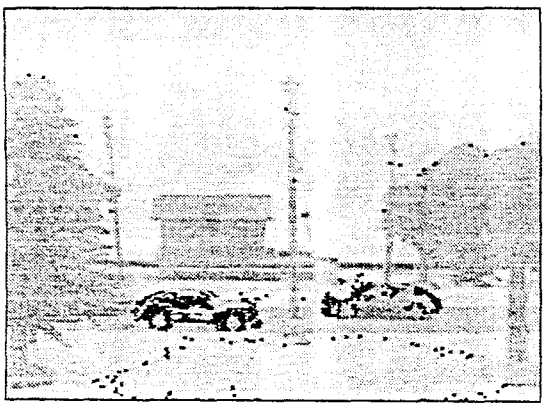


Figure 21: (c) and (d)

Figure 22:

Figure 23:

Figure 24:

Figure 25:

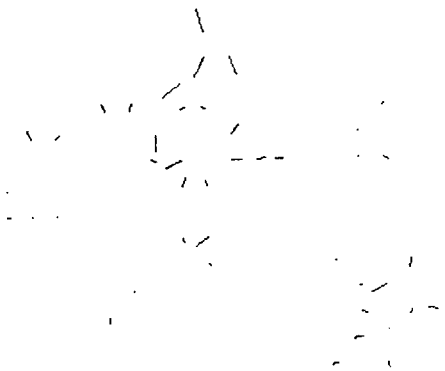


Figure 26:

value was 7.553544 for the arm and 9.118339 for the toy, which agrees with the ground truth.

We performed the same experiment with the arm at the same distance as the toy. We found that the value of the median relative depth (Z/V_c) was 10.230856 for the arm and 10.145772 for the toy, which again agrees with the ground truth.

11 Purposive, Behavioral, Active Vision

Vision has been studied, for the most part, as a general recovery problem, i.e. its goal has been to reconstruct an accurate representation of the visible world and its properties, for example, to recover boundaries, shape from texture, shading, motion, etc. Following this point of view, we consider the "brain"—or any intelligent system possessing vision—as consisting of vision and everything else (planning, reasoning, memory, etc.). In other words, we view the role of vision as that of creating a central database which stores accurate 3-D information about the scene. Then other cognitive processes (such as planning, for example) can access this database, extract whatever information they need and modify it to suit their needs. This central database is created by visual modules—such as the sfm module—that have been integrated in some way [3].

But if the analysis in this paper is valid, it demonstrates that we can solve many interesting problems, without creating a very accurate or full representation of the scene and its properties. Clearly, when a problem is simpler and more restricted, it is easier to solve. However, these simpler problems (in our case, simpler than the general sfm problem)—namely, passive navigation, motion detection, 3-D translation estimation, obstacle avoidance, relative depth, visual interception—are quite important and not very specific. They are generic in the sense that they have environmental invariance. In other words, developing such visual motion capabilities constitutes theoretical research. The fact that we may be able to robustly solve many less general problems—which, of course, cannot replace the reconstructive modules–demonstrates that we are capable of building machines that robustly achieve various behaviors. By putting such behaviors together, can we achieve "intelligent systems"? If this is possible, it provides an alternative way to study perception. A few publications over the past few years [6, 8, 11, 12, 15, 26, 35] have supported such an approach, which has acquired various names such as purposive, task-based, behavioral, active, animate, utilitarian, etc. In this section we attempt to describe the paradigm in more detail and we point out its drawbacks as well as its potential usefulness.

11.1 An attempt to formalize

With the realization that behavioral vision has as its goal the development of robust, non-primitive behaviors displayed by a robotic agent, we should be able to formalize the concept of behavior and the concept of an agent. At the same time we need to be able to provide a formal way of generating new behaviors and a calculus of behaviors or purposes.

If there is a similarity of this approach to old ideas of goal-based vision—where systems using knowledge at all levels, including domain-specific knowledge, were built and it turned out that many corners had to be cut and many oversimplified assumptions had to be made—it exists only in spirit. *An intelligent agent (observer) is a system that has a set of goals or purposes, at all times. To pursue these goals, it has to exhibit a set of behaviors.* Not all agents have the same purposes; some are more sophisticated than others and they display different behaviors.

It would be hard to give a general definition for an agent (or such a definition would be so general that it wouldn't be useful at an engineering level). We are surrounded by agents. They are basically entities that *interact* with the world around them and *act* appropriately in each situation. As they act and sense, they display behaviors and fulfill purposes.

Coming back to the basic question of formalizing behaviors, we realize that there is a very rich set of them. Some are primitive, others more sophisticated and others quite complex. In such situations, it is nice to be able to start from primitives, that is a set of behaviors from which all others can be constructed. But it is not at all clear which behaviors are the primitive ones.

To avoid a potential philosophical snare, we sidestep the question and we ask: how can we formalize behaviors, and then generate new and more complex ones from old ones and from learning?

A behavior is a sequence of perceptual events and actions whose task is to accomplish a goal. Visual input is received in a continuous manner and various processes (such as those described here and others) work together in order to recognize perceptual events and take appropriate action (an action could be a motion (navigation, manipulation), or a change of an internal state of the agent displaying the behavior). The problem is then to control such a system. It must be emphasized that the processes performing the visual analysis in order to recognize the perceptual events perform only partial recovery of the world, i.e. to accomplish some behaviors we do not need an accurate and full scene representation.

In abstract terms, a behavior of an agent is a system broadly known as discrete event process [7]. However, despite numerous results in the literature, there is at the present time apparently no unifying theory for the control of discrete event processes. Nor is it very clear what such a theory should accomplish. Numerous approaches to the modeling of discrete processes have appeared in the literature (Boolean models, Petri nets, formal languages, temporal logic, port automata, and flow networks).

An interesting model proposed recently [37] treats the controlled set of processes as the generator of a formal language (an automaton taking various actions) and studies how the recognizer of a specific (target) language (another machine recognizing perceptual events) may be employed as a controller, incorporating the desired closed-loop system behavior, and it is shown how to construct such a controller under some assumptions. It is also shown how, given two such controllable systems

(let's say behaviors B_1 and B_2), to create the shuffle operation $B_1 \| B_2$, so that we can create more complex behaviors from existing ones. However, the main conclusion of such control theoretic work may be paraphrased by saying that "supervisors must be modeled on the task to be accomplished". In other words, there does not appear to be a general universal way to accomplish behaviors (control them) or make new ones from old ones; it appears that the problem depends on what has to be accomplished.

11.2 Object recognition

Although it is not hard to see how to study navigation in this paradigm of behavioral vision, it might seem hard to apply this point of view to recognizing objects. What would it mean to have behaviors that recognize objects?

This difficulty can be easily avoided by attempting to solve an easier problem, namely that of recognizing the function of an object[9] (there may exist many functions for a single object) that is required to accomplish the behavior under consideration (an agent always executes a behavior). So, recognition can be considered in the context of an agent performing it in an environment, while executing a behavior.

An object can fulfill a function, suit a purpose. If the agent recognizes this, it has recognized the object. In fact, it has not recognized an object in the sense that it can name it as a human would, but it has recognized it "well enough" to act on it (for example, use it, avoid it, eat it, mate with it, etc.). But in most cases, deducing an object's purpose with regard to the current behavior can be done by testing the existence of some perceptual properties of the image of the object. Usually, to find out if an object can fulfill a function we need to perform various *partial* recovery tasks. Thus, without reconstructing the world fully, we can recognize many objects to the degree that we can utilize them (examples: big and moving closer (danger), man-made, graspable, movable, of certain size, with a concavity (cup), etc.).

Although such an approach does not address all aspects of object recognition, it seems to be well suited to the design of robots.

12 Conclusions

We have presented the foundations behind a set of processes that interpret visual motion in a purposive manner. We showed that an active observer can solve a series of important problems through the use of the derivatives of the image intensity function. In particular, we presented direct solutions for the problems of kinetic stabilization (passive navigation), detection of independent motion, obstacle avoidance, relative depth and 3-D motion (translation) computation and visual interception. Although the abovementioned problems are applications of the general structure from motion problem, we addressed them as independent problems in their own right and produced solutions that depend on data which can be measured.

[9] I.e. not recognizing the object but finding out enough information about it to utilize it.

The possibility that important behaviors can be realized by the cooperation of processes that recognize perceptual events without having to create a full representation of the outside world suggests that vision can be studied as a part of a system that has purposes which translate into behaviors. This point of view opens several interesting research areas, all related to the development of intelligent visual behaviors. We have pointed out various possible formalizations for this approach, as well as the associated problems.

Research in this paradigm will become more interdisciplinary with time, since the basic premise is that vision should not be studied in isolation but as a part of an intelligent system. New questions about control arise, and the integration of vision with planning, manipulation, memory and learning will provide interesting research avenues.

Whether this behavioral vision paradigm is the natural evolution of the field is still questionable. This will certainly depend on the results that are generated. Behavioral vision addresses a normative question (what should be), i.e. how should we best design robots for a set of tasks. Reconstructive vision addresses a theoretical question (what could be), i.e. what range of possible mechanisms could exist in vision systems. The empirical question (what is), i.e. how actual biological systems are designed, is addressed by other communities(psychology, neuroanatomy, etc.), while the normative and theoretical questions are studied by computer vision. And although these three questions do not necessarily have the same answers, they are closely related.

Acknowledgements: Thanks to Ruzena Bajcsy for numerous discussions and for suggesting that an observer could be modeled as a discrete event dynamic system, and to Barbara Burnett for her help in preparing the paper and doing the artwork.

This work would not have been possible without the support of DARPA, NSF, Alliant Techsystems, Inc. and Texas Instruments, Inc.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. PAMI*, 7:384–401, 1985.

[2] G. Adiv. Inherent ambiguities in recovering 3d motion and structure from a noisy flow field. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 70–77, 1985.

[3] J. Aloimonos and D. Shulman. *Integration of Visual Modules: An Extension of the Marr Paradigm*. Academic Press, Boston, 1989.

[4] Y. Aloimonos. Purposive and qualitative active vision. In *Proc. DARPA Image Understanding Workshop*, pages 816–828, 1990.

[5] Y. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *Int'l. J. Comp. Vision*, 2:333–356, 1988.

[6] M.A. Arbib. Perceptual structures and distributed motor control. In V.B. Brooks, editor, *Handbook of*

Physiology: The Nervous System II. Motor Control, pages 1449–1480, 1981.

[7] R. Bajcsy. Personal communication, 1991.

[8] R. Bajcsy and P. Allen. Sensing strategies. In *Proc. U.S.-France Robotics Workshop*, 1984.

[9] D.H. Ballard. Parameter networks. *Artificial Intelligence*, 22:235–267, 1984.

[10] D.H. Ballard. Reference frames for animate vision. In *Proc. Int'l. Joint Conference on Artificial Intelligence*, pages 635–1641, 1989.

[11] R.A. Brooks. Achieving artificial intelligence through building robots. Technical Report TR 899, M.I.T., Cambridge, MA, 1986.

[12] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE J. Robotics Automation*, 2:14–23, 1986.

[13] A. Bruss and B.K.P. Horn. Passive navigation. *Computer Vision, Graphics, Image Processing*, 21:3–21, 1983.

[14] W. Burger and B. Bhanu. On computing a 'fuzzy' focus of expansion for autonomous navigation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 563–568, 1989.

[15] C.H. Chen and A.C. Kak. A robot vision system for recognizing 3-d objects in low-order polynomial time. *IEEE Trans. Systems, Man, Cybernetics, Special Issue on Computer Vision*, 1989.

[16] Z. Duriç and Y. Aloimonos. Passive navigation: An active and purposive solution. Technical Report CAR-TR-560, Center for Automation Research, University of Maryland, College Park, MD, 1991.

[17] B. Bhanu et al. Qualitative target motion detection and tracking. In *Proc. DARPA Image Understanding Workshop*, pages 370–398, 1989.

[18] C. Fermüller and Y. Aloimonos. Estimating 3-d motion from image gradients. Technical Report CAR-TR-554, Center for Automation Research, University of Maryland, College Park, MD, 1991.

[19] B.K.P. Horn. *Robot Vision*. McGraw Hill, New York, 1986.

[20] B.K.P. Horn and E.J. Weldon Jr. Computationally-efficient methods of recovering translational motion. In *Proc. International Conference on Computer Vision*, pages 1–11, 1987.

[21] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[22] L. Huang and Y. Aloimonos. Relative depth from motion using normal flow: An active and purposive solution. Technical Report CAR-TR-535, Center for Automation Research, University of Maryland, College Park, MD, 1991.

[23] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[24] H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. Royal Soc. London B*, 208:385–397, 1984.

[25] D. Marr. *Vision*. W.H. Freeman, San Francisco, 1982.

[26] H.P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. IJCAI-77*, page 584, 1977.

[27] R.C. Nelson. Qualitative detection of motion by a moving observer. In *Proc. DARPA Image Understanding Workshop*, pages 329–338, 1990.

[28] R.C. Nelson and J. Aloimonos. Finding motion parameters from spherical flow fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58:261–273, 1988.

[29] R. Sharma and Y. Aloimonos. Robust detection of independent motion: An active and purposive solution. Technical Report CAR-TR-534, Center for Automation Research, University of Maryland, College Park, MD, 1991.

[30] M.E. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *Int'l. J. Computer Vision*, 4:171–183, 1990.

[31] M.E. Spetsakis and Y. Aloimonos. Optimal computing of structure from motion using point correspondences in two frames. In *Proc. International Conference on Computer Vision*, pages 449–453, 1988.

[32] W.B. Thompson and T-C. Pong. Detecting moving objects. *Int'l. J. Computer Vision*, 4:39–57, 1990.

[33] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of three dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. PAMI*, 6:13–27, 1984.

[34] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, Cambridge, MA, 1979.

[35] S. Ullman. Visual routines. *Cognition*, 18:97–157, 1984.

[36] A. Verri and T. Poggio. Against quantitative optic flow. In *Proc. International Conference on Computer Vision*, pages 171–180, 1987.

[37] W.M. Wonham and P.J. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J. Control Optimization*, 25:637–659, 1987.

[38] G.S. Young and R. Chellappa. 3-d motion estimation using a sequence of noisy stereo images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 710–716, 1988.