International Journal of Computer Vision, 333–356 (1988)

Active Vision

JOHN (YIANNIS) ALOIMONOS and ISAAC WEISS
Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD 20742

AMIT BANDYOPADHYAY
Department of Computer Science, SUNY Stony Brook, Stony Brook, NY 11790

Abstract

We investigate several basic problems in vision under the assumption that the observer is active. An observer is called active when engaged in some kind of activity whose purpose is to control the geometric parameters of the sensory apparatus. The purpose of the activity is to manipulate the constraints underlying the observed phenomena in order to improve the quality of the perceptual results. For example a monocular observer that moves with a known or unknown motion or a binocular observer that can rotate his eyes and track environmental objects are just two examples of an observer that we call active. We prove that an active observer can solve basic vision problems in a much more efficient way than a passive one. Problems that are ill-posed and nonlinear for a passive observer become well-posed and linear for an active observer. In particular, the problems of shape from shading and depth computation, shape from contour, shape from texture, and structure from motion are shown to be much easier for an active observer than for a passive one. It has to be emphasized that correspondence is not used in our approach, i.e., active vision is not correspondence of features from multiple viewpoints. Finally, active vision here does not mean active sensing, and this paper introduces a general methodology, a general framework in which we believe low-level vision problems should be addressed.

1 Introduction and Motivation

Most past and present research in machine perception has involved analysis of passively sampled data (images). Human perception, however, is not passive, it is active. Perceptual activity is exploratory and searching. When humans see and understand, they actively look. In the process of looking, their eyes adjust to the level of illumination, focus on certain things, converge or diverge, and their heads move to obtain a better view of the scene.

It is very natural to ask why human observers operate in such a way, because certainly humans are very efficient in visual tasks. In other words, how does the fact that an observer is active affect the levels of a visual system (as described by Marr [36]), namely computational theory, representations, and processing algorithms and implementation. Does an active observer have any advantage over a passive observer, in any computational theoretic, algorithmic, or implementational way? In this research, we examine this question and we find that indeed an active observer has a great deal of advantage as far as the first two levels of the visual system are concerned (computational theory, algorithms). A natural way to examine this question is to study basic problems of vision whose solutions demonstrate visual abilities, avoiding in this way the potential philosophical snare of getting into a discussion of the vision problem in general.

Another motivation for examining active vision is the fact that passive vision has been shown to be very problematic. Almost every basic problem in passive machine perception is very

difficult, because it is ill-posed in the sense of Hadamard [22]. So, because there does not exist a unique solution, the problem has to be regularized by imposing additional constraints, which should be physically plausible. An example of such an additional constraint is some kind of smoothness of the unknown functions. There has been excellent research in regularizing early vision problems, originated in [41,42]. Even though the regularization paradigm is very attractive for its mathematical elegance and for being a legitimization of already published research [23,26, 27], it has some shortcomings, in the sense that it cannot deal with the full complexity of vision. One problem is the degree of smoothness required for the unknown function that has to be recovered; for example, some unrealistic results have been reported in surface interpolation, because depth discontinuities are smoothed too much. Research on regularization in the presence of discontinuities, while pioneering, is still premature [33,45,47]. Another problem is that standard regularization theory deals with linear problems and is based on quadratic stabilizers. In the case of nonquadratic functionals, standard regularization theory may be used; but the situation is problematic [37]. For nonquadratic functionals, the search space may have many local minima, and in this case only stochastic algorithms might have some success [32]. Aside from the fact that most passive vision problems are ill-posed, some well-posed problems are very unstable. That is to say, even if from the physical constraints the problem is shown to have a unique solution, finding this solution might be very difficult and unstable—in the sense that a small error in the input of the perceptual process can create catastrophic results in the output. An example of such a problem is the passive navigation problem [16,34,35,38,39,51], where retinal motions (retinal velocities or displacements) are used as the input. It is worth noting however that no theoretical results about the instability of this problem exist in the literature. All the findings depend on simulations [49] or case analyses [1]. In contrast, it will be shown that some problems that are ill-posed for a passive observer become well-posed for an active one, and problems that are unstable have a chance of becoming stable.

We now set out to examine the advantages of an active observer with respect to the *computational theory* and *algorithms* levels of visual systems. In particular we show that:

1. The problem of shape from shading in the case of a passive observer has infinitely many solutions and additional assumptions are required to guarantee uniqueness. Furthermore, the stability of the developed algorithms is in question. In contrast, in the case of an active observer, the shape-from-shading problem is shown to have a unique solution. In particular the commonly used assumptions of smoothness of the visible surface and constant albedo become unnecessary. This makes it possible to deal with complex shapes having discontinuities of surface orientation and varying reflectance. The isotropic radiance constraint can be relaxed to deal with partly specular surfaces. Moreover, our method is not susceptible or prone to instabilities. Depth computation is addressed also.
2. The problem of shape from contour is a difficult one for a passive observer since assumptions have to be employed to obtain a unique solution. We show that in the case of an active observer the problem has a unique solution, which, moreover, can be found using linear equations.
3. The problem of shape from texture requires assumptions about the texture to make it solvable by a passive observer. We prove that an active observer can recover shape from texture without any assumptions and using linear equations.
4. Finally, the problem of structure from motion has been shown to be very unstable. We show that an active observer can recover structure from motion by using linear equations and obtaining a closed-form solution. In the case of a passive observer some closed-form solutions were obtained too, but they depended on the derivatives of the flow field. In contrast, our approach does not use local correspondence.

Table 1 compares the performance of a passive and an active observer in the solution of several basic problems. In the following sections we study the basic problems described in table 1. The case of the computation of optic flow will be

Table 1

Problem	Passive Observer	Active Observer
Shape for shading	Ill-posed problem. Needs to be regularized. Even then, unique solution is not guaranteed because of nonlinearity.	Well-posed problem. Unique solution. Linear equation used. Stability.
Shape from contour	Ill-posed problem. Has not been regularized up to now in the Tichonov sense. Solvable under restrictive assumptions.	Well-posed problem. Unique solution for both monocular or binocular observer.
Shape from texture	Ill-posed problem. Needs some assumption about the texture.	Well-posed problem. No assumption required.
Structure from motion	Well posed but unstable. Nonlinear constraints.	Well posed and stable. Quadratic constraints, simple solution methods, stability.
Optic flow (area based)	Ill posed. Needs to be regularized. The introduced smoothness might produce erroneous results.	Well-posed problem. Unique solution. Might be unstable.

described in subsequent publications. In all cases, we try to avoid solving the correspondence problem. Also, at this point it should be clear that active vision is not active sensing; it is just vision in which physical constraints are simplified because the observer can change state in an active way. Finally, it is worth noting at this point that the idea of the usefulness of active vision has been conjectured recently by Bajcsy [11].

2 Shape from Shading

2.1 Introduction

In the problem of recovering shape from shading, the input consists of the brightness at each point of an image, or images, and the desired output is the depth and/or the surface normal of the corresponding point on the visible surface. In principle, the depth map (the depth z as a function of the x, y coordinates) contains all the information about the surface; and the surface normals can be computed directly from the knowledge of the depth map. In practice, however, those depths cannot be derived with sufficient accuracy for calculating the normals and one would like to infer the normals directly from the image. In the following we briefly discuss the current approaches to solving the problem and their severe limitations, and then show how the active vision paradigm overcomes these limitations.

The simplest approach to the shape-from-shading problem involves using one image of the surface. To find a solution, the following assumptions are commonly used, none of which is particularly valid in a realistic situation:

1. The surface is smooth.
2. The surface reflectance characteristics, usually Lambertian, are the same throughout the surface.
3. The lighting, usually one-point light source, is the same throughout the surface.
4. The image is nearly noise-free.

Based on these assumptions, one can write a functional of the surface and its normals, which

should be minimal for the correct surface. The functional is in general a nonlinear function of a large number of unknowns (depth and normals at each point), so it is very hard to achieve convergence of the numerical optimization to the global minimum. Very good research along this line is described in Horn [25] and Ikeuchi and Horn [27]. But in this case, noise compounds the problem, creating instabilities. So these techniques have had only very limited success.

An improvement can be achieved by using two images of the surface. Combining information from the two images makes it easier to solve the minimization problem. In this case one does not need to consider the whole image at once during the minimization, as in the previous case, because the image can be decoupled into narrow strips along epipolar lines. Only points along a pair of such lines need be considered at the same time. However, to be able to take advantage of the two views, one must first find the correct correspondence between the two images. One can distinguish between two methods in using more than one image:

(a) The two cameras are very close to each other. In this case it is easy to establish a correspondence between points in the two images. Moreover, the equations one needs to solve are linear, because the small distance allows use of first-order Taylor expansion of the various functions involved. However, the small baseline between the cameras severely limits the accuracy of the method.
(b) The cameras are far apart. This leads to more accurate results, provided one can solve the correspondence problem. This problem has proven to be very difficult, and the techniques that deal with it are far from satisfactory. (For details see, e.g., Horn [24].)

The AV (Active Vision) paradigm has the advantages of both methods, without their shortcomings. First, having multiple viewpoints can solve the correspondence problem. In fact, it can be shown that three cameras are enough to resolve most of the correspondence ambiguities in the Lambertian case. The stability and reliability also increase. But multiple viewpoints are not enough by themselves to make a method work. This is because we again run up against the problem of nonlinear optimization, which, with so many variables, rarely converges to the global minimum without a very good initial guess. The key to the success of AV is the fusion of the long- and short-baseline methods. We have available images taken at short intervals, as well as images separated by long intervals, and we can use information from both.

The AV method can thus proceed in two stages:

1. A short-baseline stage, in which a succession of frames taken short distances apart is examined. This stage, being linear, is easily solvable, thereby providing initial estimates for the depths and surface normals.
2. A long-baseline stage. Now that an estimate exists, it can be used in several ways, as we shall see. In the Lambertian case we use it to establish correspondence between points seen from far-away viewpoints, while in the non-Lambertian case it is the initial guess in a nonlinear optimization procedure.

We shall show that with this method we can recover the geometry of the visible object at each individual point independently. We do not need the assumption mentioned before, of global optimization (maximum smoothness) of the whole object. Moreover, it turns out that in spite of the greater amount of data, our task is much easier than in the previous methods, since the recovery process can be done for each point separately, rather than having to deal with the image as a whole. All this is done in a stable and noise-resistant way.

2.2 Geometrical Preliminaries

We work in perspective projection and within a fixed Cartesian coordinate system x, y, z. For simplicity we assume that the camera moves with its optical axis remaining parallel to the z axis, and the lens is in the x, y plane. (Rotations will not get in the way of the basic principles.) The coordinates of the camera's lens center, x_c, y_c, are moving with a known motion, causing changes in the brightness of the image points. The coordinates of the image points are measured in the fixed coordinate system, regardless of the camera's

position, and are denoted by x_i, y_i (with $z_i = -1$, i.e., a focal length of 1). The coordinates of a point on the real object are denoted by x_o, y_o, z_o. One has the relation between the object, camera, and image coordinates:

$$x_i - x_c = -\frac{x_o - x_c}{z_o}$$
$$y_i - y_c = -\frac{y_o - y_c}{z_o} \quad (1)$$

A fixed camera forms a brightness function $E(x_i, y_i)$ in the image plane. This function changes when the camera moves (i.e., when x_c, y_c change), and the brightness now depends on four variables: $E(x_i, y_i; x_c, y_c)$. There are two possible ways to represent the change. (1) Measure the change at every point x_i, y_i of the image, i.e., find the partial derivatives of E with respect to x_i, y_i, and also with respect to x_c, y_c. This leads to optical flow-like methods. In the fluid mechanical analogy, this is a representation of the flow in the Euler coordinate system. (2) Follow a point with a given brightness along successive frames. This is analogous to following a particular element of the fluid along its path of motion and recording this element's coordinates and their derivatives. This is known as the Lagrangian system. The two methods are of course mathematically equiva- and the choice between them depends on convenience in a particular situation. For a Lambertian surface, the Lagrangian system has an obvious advantage, because an object point projects into image points of equal brightness in all images. Thus, following a point of a given brightness along successive frames means following the same object point. It is also preferable in the non-Lambertian case, in a modified form, as we shall see.

When we have two viewpoints, with either a short or long baseline, two matching systems of epipolar lines are created, one for each image. Unlike the single-image method, in which the image has to be treated as a whole, the two-camera method makes it possible to decouple the problem into reconstruction of shape along epipolar lines. In more detail, consider a plane containing the two lens-center points of the cameras. It intersects the two image planes in straight lines, namely two matching epipolar lines. A point on the epipolar line in one image corresponds to a point somewhere on the matching epipolar line in the other image (belonging to the same plane); thus we only have to find the correspondence of points along epipolar lines, which can offer a significant simplification.

In the following, we treat the Lambertian and non-Lambertian cases differently. The isotropy of the Lambertian reflectance function presents us with both a simplification and a difficulty. The simplification was mentioned above, namely the constant brightness in all images of a particular object point. The difficulty is that the parameters that influence the brightness, such as the surface orientation and the light direction, cannot be recovered by changing the point of view, as the reflectance function is independent of the point of view. To recover the orientation, we are forced to use spatial derivatives of the brightness. Happily, this makes the problem linear even for the long-baseline step (at least in our particular geometry).

We will now make the above arguments more formal. The image brightness is governed by the 'image irradiance equation,' which is easily generalized to our case:

$$E(x_i, y_i; x_c, y_c) = R(\hat{n}, \hat{v}, \vec{s}) \quad (2)$$

The left-hand side is obtained from measurements on the image at each camera location x_c, y_c. The right-hand side contains our assumptions about the light reflectance properties of the surface. This reflectance depends, in general, on the surface orientation $\hat{n}$, on the viewing direction $\hat{v}$, and on a vector containing a finite number of parameters, $\vec{s}$, which represents the light distribution and the surface intrinsic properties. The variables in R were separated in this way because $\hat{n}$ is the unknown that we are mainly interested in, and $\hat{v}$ is a parameter under our control, as the camera moves and changes the viewing direction. ($\hat{v}$ is the direction of the known vector $(x_i, y_i) - (x_c, y_c)$.) In view of the above, recovering shape from shading amounts to solving this image irradiance equation (for $\hat{n}$).

If the correspondence problem were solved, we could use the above relation for the same object point in different views, i.e., different $\hat{v}$s. We thus obtain a set of equations for the unknowns in R, in particular $\hat{n}$. Whether this set of equations is

degenerate depends on the particular R. For a typical R, $\hat{n}$ and $\hat{v}$ are coupled in a term of the form $\hat{n} \cdot \hat{v}$, so the equations are not degenerate, at least with respect to finding $\hat{n}$. For a Lambertian surface, however, the reflectance is independent of $\hat{v}$, and $\hat{n}$ cannot be found in this way.

2.3 Lambertian Surfaces

The Lambertian case is special in that the shape-reconstruction process can be decoupled not only along epipolar lines but also along isophotes. That is, when a contour of equal brightness moves in the image (with the movement of the camera), the new contour corresponds to the same set of object points as the old contour. This is because a Lambertian surface element is seen with the same brightness from every point of view. Thus, it is convenient to parametrize the image with a set of coordinates consisting of the epipolar lines and the isophotes. One can assign some labeling to the isophotes, which we denote by α, and a labeling to the epipolar lines, denoted by β. The particular labeling mapping is immaterial, as long as it is well behaved and increases monotonically (with respect to some spatial ordering of these coordinate lines). Such a well-behaved mapping is always possible at least at some neighborhood, which is what we need for taking derivatives. The coordinate lines (epipolar lines and isophotes) move and change their shape as the camera moves, but they keep their labels α, β. This is in accordance with the Lagrangian coordinate representation. The key advantage of the scheme is that an image point labeled α, β always corresponds to the same object point. The correspondence problem is then essentially solved. We can now attach the labels (α, β) to the object point (x_0, y_0, z_0) also. (Two views are needed to create epipolar lines. We can take one view as fixed, say at the beginning of the movement, while the other moves. We will not need the image from the fixed view. It only serves to create consistent systems of epipolar lines.)

By measuring the position in the image of the point labeled α, β as the camera moves—i.e., the functions $x_i(\alpha, \beta, x_c, y_c)$, $y_i(\alpha, \beta, x_c, y_c)$, and their derivatives—one can infer the depth and normal at the corresponding object point. As we shall see, the derivatives with respect to (x_c, y_c) are not needed except for finding an initial guess, but the derivatives with respect to α, β are the ones that enable us to recover the normal. Differentiating relation (2) we obtain

$$\frac{\partial x_i}{\partial \alpha} = -\frac{1}{z_o}\frac{\partial x_o}{\partial \alpha} + \frac{x_o - x_c}{z_o^2}\frac{\partial z_o}{\partial \alpha}$$
$$\frac{\partial y_i}{\partial \alpha} = -\frac{1}{z_o}\frac{\partial y_o}{\partial \alpha} + \frac{y_o - y_c}{z_o^2}\frac{\partial z_o}{\partial \alpha} \tag{3}$$

$$\frac{\partial x_i}{\partial \beta} = -\frac{1}{z_o}\frac{\partial x_o}{\partial \beta} + \frac{x_o - x_c}{z_o^2}\frac{\partial z_o}{\partial \beta}$$
$$\frac{\partial y_i}{\partial \beta} = -\frac{1}{z_o}\frac{\partial y_o}{\partial \beta} + \frac{y_o - y_c}{z_o^2}\frac{\partial z_o}{\partial \beta} \tag{4}$$

where the left-hand side quantities are measured from the image, and the right-hand side contains the unknowns.

Defining the two-dimensional vectors $\vec{x}_o = (x_o, y_o)$, $\vec{x}_i = (x_i, y_i)$, and $\vec{x}_c = (x_c, y_c)$, we substitute equations (1) in (3) and (4) to eliminate the $1/z_o^2$ factor and obtain

$$\frac{\partial \vec{x}_i}{\partial \alpha} = -\frac{1}{z_o}\frac{\partial \vec{x}_o}{\partial \alpha} - \frac{\vec{x}_i - \vec{x}_c}{z_o}\frac{\partial z_o}{\partial \alpha} \tag{5}$$

and a similar equation in β. Using this equation is the key idea in recovering the surface orientation in the Lambertian case. Its intuitive interpretation is that as the camera moves, the infinitesimal distances $\partial\alpha$, $\partial\beta$ between two object points do not change, but the corresponding $\partial\vec{x}_i$—i.e., the (geometrical) distances between the corresponding isophotes (or epipolars) in the image—do change. This change depends on the geometry of the object, as is seen in equation (5), and thus this geometry can be recovered.

For a short baseline, we calculate the change of the above derivatives caused by the camera's movement by differentiating with respect to $\vec{x}_c$:

$$\frac{\partial^2 \vec{x}_i}{\partial \alpha \, \partial \vec{x}_c} = -\left(\frac{\partial \vec{x}_i}{\partial \vec{x}_c} - \delta_{ij}\right)\frac{1}{z_o}\frac{\partial z_o}{\partial \alpha} \tag{6}$$

where δ_{ij} is the Kronecker delta, and the indexes i, j can take either of the two values x or y. We can multiply equations (1), (5), and (6) by z_o to obtain the linear system of equations:

$$(\vec{x}_i - \vec{x}_c)z_o + \vec{x}_o - \vec{x}_c = 0 \tag{7}$$

$$\frac{\partial \vec{x}_i}{\partial \alpha} z_o + \frac{\partial \vec{x}_o}{\partial \alpha} + (\vec{x}_i - \vec{x}_c) \frac{\partial z_o}{\partial \alpha} = 0 \qquad (8)$$

$$\frac{\partial^2 \vec{x}_i}{\partial \alpha \, \partial \vec{x}_c} z_o + \left(\frac{\partial \vec{x}_i}{\partial \vec{x}_c} - \delta_{ij} \right) \frac{\partial z_o}{\partial \alpha} = 0 \qquad (9)$$

The solution can now proceed in two steps:

Step 1: Solve the linear system of six equations [(7), (8), and (9)] for the six unknowns $\vec{x}_0$, z_0, $\partial \vec{x}_0 / \partial \alpha$, $\partial z_0 / \partial \alpha$. This is an inhomogeneous system of rank six (in general) and thus has a unique solution. The coefficients of the unknowns do not have to come from measurements at one point. (By 'measurements' we mean the functions $\vec{x}_i(\alpha, \beta)$ and their derivatives.) Rather, measurements can be taken from several points along the path of the point $\vec{x}_i(\alpha, \beta)$ and averaged. As the equations are linear, the averaged measurements can be substituted so that the system need be inverted only once. Thus we have obtained a good estimate of the unknowns.

Step 2: The accuracy of the previous step may not be good, as we used a derivative—equation (9)—with respect to the camera position x_c. This amounts to using a short-baseline stereo technique. For better accuracy we can use the first four equations, (7) and (8), at two far away points (or camera locations) $\vec{x}_c$ and $\vec{x}'_c$. This will result in eight equations, two of which are superfluous. This step needs the establishment of a correspondence between the two views. Since we already know the location of the object points roughly from the first step, applied to both images with $\vec{x}_c$ and $\vec{x}'_c$, any correspondence ambiguities are resolved. Without the first step, correspondence ambiguities could arise from having several points with the same brightness along one epipolar. (Alternatively to the first step, we could simply trace the point (α, β) from frame to frame, but then we may need continuous, uninterrupted tracing.) Now the longer baseline will significantly improve the accuracy and stability. Equation (9), involving the derivatives with respect to $\vec{x}_c$, is no longer part of the system. Since we still have a linear system of equations, we can again make use of averaged measurements with different base points, and invert a system of equations with the coefficients calculated from the averaged measurements.

The above steps will yield the location of the object point x_o, y_o, z_o and one tangent on the surface, namely $\partial x_o / \partial \alpha, \partial y_o / \partial \alpha, \partial z_o / \partial \alpha$. A similar procedure using the β parameter will give another tangent. Once the tangents of the surface at $\vec{x}_o, z_o$ are known, the normal can immediately be calculated by their vector product

$$\hat{n} = A^{-1/2} \left(\frac{\partial x_o}{\partial \alpha}, \frac{\partial y_o}{\partial \alpha}, \frac{\partial z_o}{\partial \alpha} \right) \times \left(\frac{\partial x_o}{\partial \beta}, \frac{\partial y_o}{\partial \beta}, \frac{\partial z_o}{\partial \beta} \right)$$

where A is the scalar product of the above tangent vectors.

A few points are worth noting:

1. At no point did we need to know either the albedo (the surface reflectance) or the light characteristics. They can both vary from point to point on the surface without affecting the calculation. Thus, when a change in the brightness is detected, we are able to tell whether it results from a change in the geometry of the surface or from the reflectance or lighting (and can calculate the local geometry). This is unlike other theories.
2. The surface is not necessarily smooth, as is assumed in most shape reconstruction theories. If one of the derivatives used is too large, we simply label this point as a discontinuity and apply our method to other points in the neighborhood.
3. The tangents have not been derived from the depth map, which would have been quite inaccurate. They were independent variables in a system of equations that determined both the depth and the tangents. It would be of interest to carry out an error analysis of the results.
4. The equations for both the short and long baseline turned out to be linear, which frees us from the recurring hard problems of nonlinear optimization.
5. Using the brightness function at several viewpoints was not enough to recover the surface normal, and we needed its derivatives $\partial \vec{x}_i / \partial \alpha$, $\partial \vec{x}_i / \partial \beta$. The infinitesimals $\partial \alpha$, $\partial \beta$ do not change from one image to another, but $\partial \vec{x}_i$ changes in a way dependent on the normal,

which can thus be recovered. We will not need the derivatives in the non-Lambertian case (except at the first stage).

6. The formalism is applicable to any contours, not necessarily isophotes. Many contours that are marked on the object can be detected on an image by means other than changes in shading. For instance, contours can be formed by changes in texture or color. If the change is continuous, such as a gradual color change, we can draw contours on which some property such as color remains constant. We can then label the contours in the same way we labeled the isophotes and apply the above formalism without change. We can thus find the position and orientation of the visible surface elements. If the change that produced a contour was a sharp discontinuity, the contour is isolated. In this case we cannot measure the derivative of the contour's property along epipolar lines, the way we measured the derivative of the isophote's brightness. We can still measure the other derivative, i.e., of the epipolar line along the contour. (Formally, we can differentiate with respect to β, but not α). This is enough to find the position and direction of the contour element—by solving equations (7), (8) and (9) with β. This line element lies on a visible surface element, and its direction is the direction of one tangent to that surface. In this case, then, without further information, the surface element's orientation can be determined only up to rotation around the contour element.

2.4 Non-Lambertian Surfaces

When the reflectance function has a nonisotropic component, the reconstruction becomes more difficult. The isophotes at different viewpoints no longer correspond to the same object point. Thus the simple Lagrangian formalism described above has to be modified. Additionally, more unknowns are added to the problem, as there are more parameters in the reflectance function, including the relative strengths of the nonisotropic components. (They enter the vector $\vec{s}$ in equation (2).) This reflectance function is in general nonlinear. We think that the Lagrange formalism, namely following points on the changing image, is still preferable to the Euler formalism used in most optical-flow theories (measuring changes at a fixed point on the image) because it enables us to deal with an object point (and its neighborhood) separately from other points, while a fixed point on the image corresponds to different object points during the motion of the camera.

Because simply following the isophotes is not useful as in the Lambertian case, the image brightness E has to be taken into account explicitly. It is useful to work in a five-dimensional vector space V, whose components are $E, \vec{x}_i, \vec{x}_c$. In the fluid mechanical analogy, the subspace spanned by the first three components, $E, \vec{x}_i$, is somewhat analogous to a 'phase space,' while the $\vec{x}_c$ represents temporal dimensions. Thus, for one viewpoint (fixed $\vec{x}_c$), the image is a 2D surface in the above 3D subspace. It is simply the surface described by brightness function $E(x_i, y_i)$. As the camera moves along some (known) path $\vec{x}_c(\gamma)$ in the $\vec{x}_c$ dimensions, it creates a one-parameter family of such 2D surfaces, so that we have a 3D surface in the 5D space. This surface is our data, measured from the images. We shall call it the 'brightness surface,' to distinguish it from the visible surface.

Consider again the image irradiance equation

$$E(x_i, y_i; x_c, y_c) = R(\hat{n}, \hat{v}, \vec{s}) \tag{2}$$

where the viewing direction $\hat{v}$ depends on $\vec{x}_i$, $\vec{x}_c$:

$$\hat{v} = \frac{\vec{x}_i - \vec{x}_c}{|\vec{x}_i - \vec{x}_c|}$$

This equation has to be solved for each object point. We assume that the functional form of R is the same at each point, with different parameters. The 3D brightness surface appears on the left-hand side of this equation. To solve the equation, we need to represent the right-hand side too. This can be done by representing each visible surface element (around an object point) as a trajectory in the 5D space. For such an element, all the unknowns $\hat{n}, \vec{s}, \vec{x}_o, z_o$ are fixed. The camera moves along the path $\vec{x}_c(\gamma)$, causing a simultaneous move $\vec{x}_i(\gamma)$ in the image coordinates, in accord-

ance with the perspective geometry equation (1). The light reflected by the element in the viewing direction also changes. Using the reflectance function R, we can compute $R(\gamma)$. Now we can plot a trajectory in the 5D space, with R being plotted in the E dimension. We obtain the curve

$$\Gamma(\gamma) = (\vec{x}_c(\gamma), \vec{x}_i(\gamma), R(\gamma))$$

In summary, we have a trajectory $\Gamma(\gamma)$ for every set of unknown parameters pertaining to one visible surface element.

If such a surface element lies on our visible object, then its trajectory in the 5D space V must lie on the 3D brightness surface (representing E in that space). This is because of the equality $E = R$, i.e., the reflectance calculated for the surface element has to match the measured image brightness, in every image observed.

Since the visible object is made of such elements, the 3D brightness surface created by the object in the 5D space is made up of such individual trajectories. We can distinguish between the different trajectories by their starting point. Looking at the first image on the camera's path, each point $\vec{x}_i$ belongs to one surface element, and can be used to label the corresponding trajectory.

The solution of the image irradiance equation $E = R$ for each surface element is now reduced to finding a set of unknowns $\hat{n}, \vec{x}_o, z_o, \vec{s}$ for which the element's trajectory in the 5D space lies entirely on the birghtness surface (and has a given starting point). Correspondence is not an issue here. A trajectory in V immediately defines a correspondence between successive images. Stated differently, the correspondence problem has been merged into the trajectory-matching problem.

So, by following trajectories that are generated by single object points, we have been able to decouple the reconstruction problem and solve separately for small sets of parameters, belonging to each object point, rather than dealing with the image as a whole. The usual theories, in contrast, lead to a large set of coupled equations involving all the image points. This trajectory-following method can be associated, in a way, with the Lagrange system of fluid mechanics.

Having clarified the theoretical vision aspect of the problem, we have now to deal with the more practical problem of fitting a trajectory to a surface. First, how do we define fitting? One way is to demand that the equation $E = R$ be satisfied at several points along the trajectory. Thus we obtain a set of equations, one from each such point, for the set of unknowns. If the number of points is at least as large as the number of unknowns, a solution can be found, in a generic case. If it is larger, the reliability improves. This is similar to the situation described in section 2. As noted there, the system is not degenerate because the geometrical unknowns, $(\hat{n}, \vec{x}_o, z_o)$, are coupled to the viewing direction $\hat{v}$. The other parameters in R that have some coupling to the viewing direction will also be recovered. For instance, having a single light source, we may find its direction, and find the product of the intensity with the surface albedo, but the latter cannot be separated in this method.

In practice, because of noise and other inaccuracies, it may be impossible to find such a solution. Thus it is preferable to turn the problem into one of optimization. Unlike theories such as regularization, this optimization is not a result of some additional assumptions such as smoothness, which are not needed here. It is simply a way to make the tolerance limit of the fitting less stringent.

One functional we can optimize is the sum of distances from each trajectory point to the nearest brightness surface point. For simplicity, we measure the distance in the E dimension only, taking the coordinates $\vec{x}_c$, $\vec{x}_i$ to be the same for both the trajectory and the surface. Thus, we seek to minimize the functional

$$\int_{\Gamma(\gamma)} \{R[\vec{x}_i(\vec{x}_o, z_o), \vec{x}_c \hat{n}, \vec{s}] - E(\vec{x}_i, \vec{x}_c)\}^2 \, d\gamma$$

over all possible values of the unknowns $\vec{x}_o, z_o, \hat{n}$, $\vec{s}$, with the constraint of passing through a given starting point. The unknowns enter the problem through both their explicit appearance in the integrand and their determination of the tracjectory $\Gamma(\gamma)$. (Recall that $\vec{x}_i$ is determined by equation (1), which contains the unknown position $\vec{x}_o$, z_o of the object point.)

Although the number of unknowns is small, we may still face difficulties resulting from the nonlinear nature of the problem. Our strategy to deal with that is similar to the one we used in the

Lambertian case. In step 1, we use a very short trajectory and linearize our expressions, using a Taylor expansion of R. Altneratively, higher derivatives can used instead of several close points along the trajectory. The solution of the linear equations provides a good initial guess for step 2.

Step 2 is the nonlinear optimziation described above. The small number of unknowns and the good initial guess make the task quite easy.

As in the Lambertian case, no use has been made of a smoothness assumption. Furthermore, the position in space, as well as the orientation, have been recovered for each object point separately. The other parameters in the reflectance function that are coupled to the viewing direction are also recovered. The multiple viewpoints make the result reliable and stable. For the Lambertian case, this method has a certain degeneracy, as $\hat{n}$ and $\vec{s}$ cannot be separated solely by moving the camera, as noted before.

2.5 Summary and Conclusions

In evaluating the AV paradigm for recovering shape from shading, two major benefits arise:

1. More viewpoints give us more information, and allow us to dispose of the restrictive assumptions used in previous research and to increase the stability with respect to noise and other errors.
2. One would think that with more images the task of processing the given information will be harder than for one or two viewpoints. In fact, just the opposite has happened, because we have been able to decouple the handling of the individual object points, and solve the problem in small, separate parts.

These advantages together allow us to infer the geometrical parameters as well as reflectance-function parameters at each point individually. This means that we recover the true shape of the object, rather than a smoothed and 'optimized' version of it, as other theories do. This also resolves the perennial problem of whether a change in the image brightness is caused by a change in the object's geometry, or by other light-reflectance factors.

3 Shape from Contour

Here we study the problem of the detection of shape from contour by an active observer, and we compare the performance of this new active scheme with the passive approach. We consider the contour to be planar. Work on nonplanar contours can be found in [8].

3.1 Introduction

The human perceiver is able to derive enormous amounts of information from the contours in a scene. As part of this capacity, we are able to use the shapes of image contours (as they are seen by both eyes) to infer the shapes and dispositions in space of the surfaces they lie on, as well as their motion. The interpretation of contours by a binocular observer involves several subproblems (following Witkin [53]):

(a) Locating Contours in the Images. If contours are to be used to infer anything, they must first be found. The human perceiver has little difficulty deciding what is and is not a contour, yet the automatic detection of edges has proved very difficult. Perhaps this fact should not be surprising; the contours that we see in natural images usually correspond to definite physical events, such as shadows, depth discontinuities, color differences, and the like. Our ability to detect these events may say more about their significance for image interpretation than about their ease of detection. Why should we expect events that have simple descriptions in terms of the structure of the scene to have simple descriptions in terms of the image intensity as well? If the physical significance of contours is taken as their primary feature, then at least we know what is being detected, even if we don't know how. But recent research [42] shows that we are reasonably well advanced as far as detection of contours goes. Actually, we can say that we can fairly well detect the contours in an image, even if there are some inaccuracies.

(b) Labeling Contours (i.e., Distinguishing Contours Due to Different Physical Events): If contours correspond to different physical events, then an essential component of their interpretation must be

to decide which contours denote which event, since each kind of contour imparts a different meaning. Recent work has shown that strong structural constraints can be applied to distinguish one kind of contour from another.

(c) Interpreting Contours: Even after contours have been found and labeled, not much is known about the physical structure of the scene. It is clear that contours play an important role in the human perceiver's ability to decide how things are shaped and where they are, apart from the application of specific 'higher-level' knowledge to objects of known shape.

In this section we study this problem of contour interpretation.

3.2 The Passive Approach (Previous Work)

The recovery of three-dimensional shape and surface orientation from a two-dimensional contour is a fundamental process in any visual system. Recently, a number of methods have been proposed for computing shape from contour. For the most part, previous passive techniques have concentrated on trying to identify a few simple, general constraints and assumptions that are consistent with the nature of all possible objects and imaging geometries in order to recover a single 'best' interpretation from among the many that are possible for a given image. For example, Kanade [30] defines shape constraints in terms of image-space regularities such as parallel lines and skew symmetries under orthographic projection. Witkin [53] looks for the most uniform distribution of tangents to a contour over a set of possible inverse projections in object space under orthography. Similarly, Brady and Yuille [14] search for the most compact shape (using the measure of area over perimeter squared) in the object space of inverse projected planar contours. Weiss [54] uses a minimum energy principle for nonplanar contours.

Rather than attempting to maximize some general shape-based evaluation function over the space of possible inverse projective transforms of a given image contour, and adhering to our framework of attempting unique solutions without employing heuristics that might be restrictive, we propose to find a unique solution by using an active approach, since it can be easily proved that one image of a planar contour (under orthography or perspective) admits infinitely many interpretations of the structure of the world plane on which the contour lies, if no other information is known. Finally, the need for a unique solution, which is guaranteed in our approach, arises also from the fact that there exist many real-world counterexamples to the evaluation functions that have been developed to date. For example, Kanade's and Witkin's measures incorrectly estimate surface orientation for regular shapes such as ellipses (which are often interpreted as slanted circles). Brady's compactness measure does not correctly interpret noncompact figures such as a rectangle since he will compute it to be a rotated square (e.g., if we view a rectangular table top, we do not see it as a rotated square surface, but as a rotated rectangle). It is worth noting that the equations used in previous work (the passive approach) are highly nonlinear.

Up to now we have only discussed monocular passive shape from contour. It would seem that detecting shape from contour employing a binocular observer might reduce ambiguity and nonlinearity. It is shown in the sequel that this is not the case, i.e., even if we employ a binocular static observer, the problem is still nonlinear, if we want to avoid solving the correspondence problem between the left and right frames. (We have stated that our approach will be correspondenceless.) Indeed, consider a binocular observer imaging a planar contour C with projections C_l and C_r on the left and right frames respectively. Let a coordinate system $OXYZ$ be fixed with respect to the left camera, with the Z axis pointing along the optical axis. We assume that the image plane I_{m_l} is perpendicular to the Z axis at the point $(0, 0, 1)$ (focal length = 1). Let the nodal point of the right camera be the point $(d, 0, 0)$, and let its image plane I_{m_r} be identical to the previous one. Consider also a plane Π in the world with equation $Z = pX + qY + c$, which contains a contour C, and consider the images (perspective) C_l and C_r of the contour on the left and right image planes respectively.

From now on we will denote the coordinates on the left and right image planes by (x_l, y_l) and (x_r, y_r) respectively. We assume perspective pro-

jection. We can easily prove that if S_l, S_r are the areas enclosed by contours C_l and C_r and (A_L, B_L), (A_R, B_R) the centers of mass of contours C_l and C_r, then

$$\frac{S_L}{S_R} = \frac{1 - A_L p - B_L q}{1 - A_R p - B_R q} \tag{10}$$

where (p,q) is the gradient of the plane Π with equation $Z = pX + qY + c$ with respect to the left frame, and where we have assumed (for simplicity and without loss of generality) that the focal length $f = 1$. For a proof of (10) see appendix A. If we want to recover the shape of the contour in view without any assumptions, what we should do is connect properties of the left and right images—if we don't want to resort to correspondence. Such properties can include area, perimeter, any function of these two, or other functions of the positions of the contours. Equation (10) is linear and is the only linear constraint we have been able to find.

Another constraint can be extracted from the perimeters of the two contours. If we calculate the perimeter of the world contour from each of two projections, then these two results should be equal. From this we can get an additional constraint on p,q.

To do this, we need to develop the first fundamental form of the world plane as a function of the retinal coordinates, in order to be able to compute the length of the world contour (up to a constant factor, of course). If we fix a coordinate system $OXYZ$ with the Z axis as the optical axis and focal length 1 and we consider a plane Π: $Z = pX + qY + c$ in the world with a contour C on it, and we denote by (x,y) the coordinates on the image plane, then a point (X,Y,Z) in the world planar contour C is projected onto the point

$$x = \frac{X}{Z}; \qquad y = \frac{Y}{Z}$$

The inverse imaging function, call it f, is the function that maps the image plane onto the world plane; so, if (x,y) is an image point, the 3D world point on the plane $Z = pX + qY + c$ that has (x,y) as its image is given by

$$f(x,y) = \left(\frac{cx}{1 - px - qy}, \frac{cy}{1 - px - qy}, \frac{c}{1 - px - qy}\right)$$

The first fundamental form of f [Lipschutz 1969] is the quadratic form

$$E\,dx^2 + 2F\,dx\,dy + E\,dy^2$$

with $E = f_x \cdot f_x$, $F = f_x \cdot f_y$, and $G = f_y \cdot f_y$.

After simple calculations we get

$$E = \frac{c^2}{(1 - px - qy)^4}[(1 - qy)^2 + p^2y^2 + p_2]$$

$$F = \frac{c^2}{(1 - px - qy)^4}[(1 - qy)qx + (1 - px)py + pq]$$

$$G = \frac{c^2}{(1 - px - qy)^4}[q^2x^2 + (1 - px)^2 + q^2]$$

So, if we consider two points (x,y) and $(x+dx, y+dy)$ on the image plane, then the three-dimensional distance dC of the corresponding points on the world plane is given by

$$dC = \sqrt{(E\,dx^2 + 2F\,dx\,dy + G\,dy^2)}$$

Consequently, if we have a contour C on the iamge plane, then the 3D planar contour has length

$$\int_C \sqrt{(E\,dx^2 + 2F\,dx\,dy + G\,dy^2)}$$ on the image plane

Using the above equation we can compute the length of the world contour (up to a constant factor) from both the left and right frames, and equate the results. This will result in an equation (nonlinear) with unknown p,q. This equation may be solved, together with the linear constraint [2], to give the orientation gradient of the contour. Uniqueness is not guaranteed theoretically, and the nonlinearity might create instabilities. Actually it has been shown recently [6] that there can be at most two solutions in the case of some kind of symmetry.

We now proceed to study the same problem, but using an active observer. We distinguish between a monocular and a binocular observer.

3.3 The Active Approach

3.3.1 Monocular Observer. This problem has already been addressed by Aloimonos [2] and

Kanatani [31]. In Kanatani's scheme, which is very elegant, the method developed is an application of the linear-feature theory introduced by Amari [9]. The treatment is for differential motion. In Kanatani's scheme, if we have a closed curve C on the image plane, then features are defined as various line integrals along C of the form

$$I = \int_C F(x,y)\, ds$$

with $ds = \sqrt{dx^2 + dy^2}$, and F any differentiable function. Then, the change dI/dt as the observer moves is connected through linear equations to the gradient (p,q) of the plane on which the contour lies. In other words, if the observer moves with a known motion, then from the two successive images of the contour, the shape (p,q) of the three-dimensional contour is uniquely computed, if certain conditions are satisfied. The solution is given through linear equations. The sensitivity of the method to noise depends on the error introduced from the numerical differentiation and only on this.

3.3.2 Binocular Observer. Here we examine the active perception of shape from contour by a binocular observer. Again, let a coordinate system be fixed with respect to the left camera with the Z axis pointing along the optical axis. We consider that the image plane I_{m_l} is perpendicular to the Z axis at the point (0, 0, 1). Let the nodal point of the right camera be at $(d, 0, 0)$, and let its image plane I_{m_r} be identical to the previous one. Consider a plane P in the world with equation $Z = pX + qY + c$ that contains a contour C, and consider the perspective iamges C_l and C_r of the contour on the left and right image frames respectively. Let S_L and S_R be the areas of the left and right image contours respectively. Then (see appendix A)

$$\frac{S_L}{S_R} = \frac{1 - A_L p - B_L q}{1 - A_R p - B_R q} \tag{11}$$

where (A_L, B_L), (A_R, B_R) are the centers of mass of the left and right contours respectively. We call this constraint the area-ratio constraint. Now, we rotate both eyes by a small angle θ around the X axis (this can also be simulated). The new image coordinates are

$$x' = \frac{r_{11}x + r_{21}y + r_{31}}{r_{13}x + r_{23}y + r_{33}}$$

$$y' = \frac{r_{12}x + r_{22}y + r_{32}}{r_{13}x + r_{23}y + r_{33}}$$

where

$$R = (r_{ij})_{3\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\partial & -\sin\partial \\ 0 & \sin\partial & \cos\partial \end{bmatrix}.$$

Hence

$$x' = \frac{x}{\cos\partial - y\sin\partial} \qquad y' = \frac{\sin\partial + y\cos\partial}{\cos\partial - y\sin\partial}$$

Then the new gradient (p', q') of the world contour is

$$p' = \frac{p}{\cos\partial + q\sin\partial} \tag{12}$$

$$q' = \frac{q\cos\partial - \sin\partial}{\cos\partial + q\sin\partial} \tag{13}$$

But again from the area ratio constraint

$$\frac{S_L^R}{S_R^R} = \frac{1 - A_L^R p' - B_L^R q'}{1 - A_R^R p' - B_R^R q'} \tag{14}$$

where (A_L^R, B_L^R), (A_R^R, B_R^R) are the centers of mass of the left and right contours after the rotation and S_L^R, S_R^R are their areas, respectively.

Using (12), (13), in (14) we get

$$\frac{S_L^R}{S_R^R} = \frac{\cos\partial + q\sin\partial - A_L^R p - B_L^R(q\cos\partial - \sin\partial)}{\cos\partial + \mathrm{q}\sin\partial - A_R^R p - B_R^R(q\cos\partial - \sin\partial)} \tag{15}$$

Equations (11) and (15) constitute a linear system in the unknowns p and q that has a unique solution in general. These equations, however, become degenerate when $p = 0$; this is because when $p = 0$, both equations reduce to

$$\frac{S_L}{S_R} = \frac{1 - B_L q}{1 - B_R q}$$

But the y-coordinates are the same in both images, and so $B_L = B_R$ and consequently $S_L = S_R$. So, if $p = 0$ the areas in both images are equal. Appendix B develops a condition that proves

that this is sufficient too—i.e., $p = 0$ if the areas in both images are equal. Here, we devise a method for computing q in case $p = 0$.

We can easily prove that if both p and q are zero (world plane parallel to image planes), then the length of the contours in both images (perimeters) are equal. It is not sufficient, though, for the lengths of the contours to be equal. This does not imply that $p = q = 0$; there are some degenerate cases, which are discussed in [4]. For the purposes of this section, assuming that we can check for the degenerate cases, we propose the following algorithm for actively perceiving shape from contour in the case of $p = 0$.

Step 1: Rotate the cameras so that discrepancy between the lengths of the left and right contours is minimized.

Step 2: q corresponds to the rotation that minimizes the discrepancy.

Let $(0, q, -1)$ be the surface normal of the world contour. Rotating the camera by θ around the x-axis, we get

$$\begin{pmatrix} p' \\ q' \\ k' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 0 \\ q \\ -1 \end{pmatrix}$$

or

$$\begin{pmatrix} p' \\ q' \\ k' \end{pmatrix} = \begin{pmatrix} 0 \\ q\cos\theta + \sin\theta \\ q\sin\theta - \cos\theta \end{pmatrix}$$

We need $q' = 0$, hence

$$q\cos\theta + \sin u = 0 \quad \text{or} \quad q = -\tan u$$

Thus, if θ is the rotation that minimizes the difference between contour lengths, then $q = -\tan\theta$ gives the corresponding q. A similar analysis can be done for a verging stereo system. The mathematics is much more complicated and it can be found in [8].

3.4 Summary and Discussion

We have presented a theory for the computation of shape from contour by an active observer. The constraints involved demonstrate the superiority of an active observer vs. a passive one, with respect to computation. Uniqueness and linearity vs. ill-posedness and nonlinearity make the AV paradigm with respect to the shape-from-contour problem very appealing and worth studying. The advantage of binocular shape from contour over monocular shape from contour lies in the fact that the monocular case has been demonstrated to be unstable. The problem with the active binocular shape-from-contour theory is that the contours in the left and right images have to be corresponded. We have not solved this problem but it is certainly easier to correspond macrofeatures (contours) rather than microfeatures. Finally, the problem of shape from nonplanar contour, i.e., understanding the structure of the contour without correspondences in an active way, is treated in Aloimonos and Tsakiris [8]. We chose here the case of a planar contour, because this case has been addressed extensively by past research.

4 Shape from Texture

The problem of shape from texture has received a lot of attention in the past few years and some excellent research on the topic has been published [2,18,21,31,46,52]. The problem in the passive case is defined as 'finding the orientation of a textured surface from a static monocular view of it.' This problem is ill-posed in the sense that there exist infinitely many solutions. To restrict the space of solutions, assumptions have to be made about the texture. Assumptions such as directional isotropy and uniform density have been employed in previous research. Density has been defined as density of texels or density of the sum of the lengths of the contours (zero-crossings) in the image.

It is very clear that even though some of the assumptions used in the literature for the recovery of shape from texture are general enough, they are not powerful enough to capture a very large subset of natural images. As a result, the developed algorithms fail when they are applied to many real surfaces. Furthermore, there is no way to check in advance whether or not a par-

ticular assumption is valid for the surface that is imaged. This problem alone is enough to demonstrate the restricted applicability of the existing shape-from-texture algorithms (or of the ones yet to come). We will show that if the observer is active then the shape-from-texture problem, or the problem of shape detection from surface intensity and markings, becomes easy, in the sense that no restrictive assumptions are necessary and the solution is obtained from linear equations.

The next section introduces the mathematical prerequisites. For simplicity, and without loss of generality, we will assume that the surface in view is planar (as in previous shape-from-texture research). If the surface in view is nonplanar, then the problem can be addressed either by applying our theory locally in the image, i.e., by assuming that the surface in view is locally planar, or if a parametric model for the surface is assumed, then the same basic principles reported here may be used to recover the parameters of the surface (with the difference that the resulting equations might not be linear).

4.1 Prerequisites

The treatment and symbolism here follow those in [29]. Suppose that the camera is looking at a planar surface. Assume further that the camera is moving. For our analysis we assume that the surface is moving. This is equivalent to the motion of the camera, and it is done here for simplification of the formulas. Call the planar surface in the world W and the image plane R. Suppose that point $X = (X, Y, Z) \in W$ is projected onto point $\mathbf{x} = (x, y) \in R$. Let the motion of the surface consist of a translation $\mathbf{T} = (t_1, t_2, t_3)$ and a rotation $\boldsymbol{\Omega} = (\omega_1, \omega_2, \omega_3)$, or $\mathbf{V}(\mathbf{X}) = \mathbf{T} + \boldsymbol{\Omega} \times \mathbf{X}$, where $\mathbf{V}(\mathbf{X})$ is the velocity of a point $\mathbf{X} \in W$. Then this velocity can be written as

$$\mathbf{V}(\mathbf{X}) = \sum_{k=1}^{6} r_k \mathbf{V}_k(\mathbf{X})$$

where

$$r_1 = t_1, \quad \mathbf{V}_1(\mathbf{X}) = (1 \quad 0 \quad 0)^T$$
$$r_2 = t_2, \quad \mathbf{V}_2(\mathbf{X}) = (0 \quad 1 \quad 0)^T$$
$$r_3 = t_3, \quad \mathbf{V}_3(\mathbf{X}) = (0 \quad 0 \quad 1)^T$$
$$r_4 = \omega_1, \quad \mathbf{V}_4(\mathbf{X}) = (0 \quad -Z \quad Y)^T$$
$$r_5 = \omega_2, \quad \mathbf{V}_5(\mathbf{X}) = (Z \quad 0 \quad -X)^T$$
$$r_6 = \omega_3, \quad \mathbf{V}_6(\mathbf{X}) = (-Y \quad X \quad 0)^T$$

Then, it can be easily proved that the optic flow (image velocity) at a point $\mathbf{x} = (x, y)$ is $\dot{\mathbf{x}} = \Sigma_{k=1}^{6} r_k \mathbf{u}_k(\mathbf{x})$, where $\mathbf{u}_k$ is a function depending on the shape.

We prove the above equation avoiding the details of the perspective projection. Let the projection from the world to the image plane be $\mathbf{P}$, with $\mathbf{P}(\mathbf{X}) = \mathbf{x} = (x, y) = [\mathbf{P}_1(\mathbf{X}), \mathbf{P}_2(\mathbf{X})]$. If the shape of the surface W is given (a function h), a mapping: P_h^{-1}: $R \rightarrow$ (object surface) is defined such that $\mathbf{P}[P_h^{-1}(\mathbf{x})] = \mathbf{x}$.

The optic flow $\dot{\mathbf{x}}$ at the point $\mathbf{x} = (x, y)$ is then given by

$$\dot{\mathbf{x}} = \frac{d}{dt}\mathbf{P}(\mathbf{X}) = \frac{\partial \mathbf{P}}{\partial \mathbf{X}}\mathbf{V}$$

where both $\partial \mathbf{P}/\partial \mathbf{X}$ and $\mathbf{V}$ are functions of retinal coordinates.

So, since $\mathbf{V} = \Sigma_{k=1}^{6} r_k \mathbf{V}_k$, we have

$$\dot{\mathbf{x}} = \sum_{k=1}^{6} r_k \mathbf{u}_k(\mathbf{x}) \tag{16}$$

with

$$\mathbf{u}_k(\mathbf{x}) = \frac{\partial \mathbf{P}}{\partial \mathbf{X}}[P_h^{-1}(\mathbf{x})]\mathbf{V}_k[P_h^{-1}(\mathbf{x})]$$

Equation (16) will be used very frequently in the sequel.

4.2 Linear Features

Here we introduce the concept of a *linear feature vector* that has proved to be a powerful device for handling several problems in cybernetics [9]. Let the image intensity function be denoted by $s(x, y)$. A *linear feature* (LF) is a linear function f over the image, i.e.,

$$f = \iint s(x, y) m(x, y)\, dx\, dy$$

where m is called a measuring function, s is the image brightness, and the integration is taken

over the area of interest. A *linear feature vector* **f** (LFV) is a vector of linear features, i.e.,

$$\mathbf{F} = [f_1 \quad f_2 \quad \cdots \quad f_n]^T$$

with

$$f_i = \iint s m_i \, dx \, dy$$

where m_i is a measuring function, for $i = 1, \ldots, n$. $\{m_i\}$ could be any set; one good example is $\{m_i\} = \{m_{pq}\} = \{e^{i(px+qy)}\}$, in which case a linear feature corresponds to a Fourier component of the image.

4.3 The Constraint

The object's motion, induces optical flow that satisfies the following equation:

$$s_x u + s_y v + s_t = 0$$

where (u, v) is the optic flow at a point (x, y) and s_x, s_y, s_t are the spatiotemporal derivatives of the image intensity function at the point (x, y). This equation can be written as

$$\frac{\partial s}{\partial t} = -\dot{\mathbf{x}} \cdot \nabla s$$

The time derivative of an LFV will be

$$\dot{\mathbf{f}} = [\dot{f}_1 \quad \dot{f}_2 \quad \cdots \quad \dot{f}_n]$$

where

$$\dot{f}_i = \iint \frac{\partial s}{\partial t} m_i \, dx \, dy = -\iint m_i (\dot{\mathbf{x}} \cdot \nabla s) \, dx \, dy$$

The optic flow field (from equation (16) can be written in the form

$$\dot{\mathbf{x}} = \sum_{k=1}^{6} r_k \mathbf{u}_k = \sum_{k=1}^{6} r_k \begin{bmatrix} u_k(\mathbf{x}) \\ u_k(\mathbf{x}) \end{bmatrix}$$

From this,

$$\dot{f}_i = -\iint m_i (\dot{\mathbf{x}} \cdot \nabla s) \, dx \, dy, \quad \text{or}$$

$$\dot{f}_i = -\sum_{k=1}^{6} r_k \iint m_i (u_k s_x + v_k s_y) \, dx \, dy, \quad \text{or}$$

$$\dot{f}_i = \sum_{k=1}^{6} r_k h_{ik}$$

with

$$h_{ik} = -\iint m_i (u_k s_x + v_k s_y) \, dx \, dy$$

So, we have found that

$$\dot{\mathbf{f}} = H \cdot r \tag{17}$$

where $H = (h_{ik})$ and $r = (r_1 r_2 \cdots r_6)^T$, the motion parameters.

Matrix H contains the parameters of the plane in a linear form. So, equation (17) relates linear features with shape and motion parameters. Furthermore, it is linear in the shape of the planar surface in view. So, a simple linear least-squares method or a Hough transform technique is sufficient for the recovery of the gradient of the plane in view. Depth can be computed too, if desired. Here we must emphasize the fact that no local correspondence has been used. The only computed quantity was the time derivative of a linear-feature vector, that involves the whole image. This approach was introduced by Amari and emphasized by Kanatani [31].

Finally, we want to stress here the fact that in this algorithm, the spatial derivatives of the intensity function don't need to be computed. This is due to the linear-feature vector approach. (Integration by parts avoids differentiation of the intensity function. Instead, the derivative of the analytic measuring function m_i has to be computed. So, we prefer to avoid differentiating the image intensity, which is discrete, because numerical differentiation is an ill-posed problem.) More importantly, the same approach can be followed if the image is a dot pattern (or a line pattern—zero-crossings), i.e., it is discontinuous. The reason for this is again the fact that the spatial derivatives of the intensity function do not have to be estimated. Only the temporal derivative of the image needs be estimated. This approach has been initiated in [28].

4.4 Summary and Discussion

We have presented a method for the recovery of the shape of a planar surface by an active observer. Our method does not rely on any assumptions about the texture and it does not require the

image to be spatially differentiable. The approach is based on the fact that the observer is moving with a known motion. If the observer is moving with an unknown motion, then again the problem is solvable, and it has been addressed by Aloimonos [2], Negahdaripour [40], and Amari and Maruyama [10]. We will report elsewhere our research on this case.

5 Structure from Motion

5.1 Introduction

The problem of structure from motion has received considerable attention lately [35,49,50]. The problem is to recover the three-dimensional motion and structure of a moving object from a sequence of its images. Even though computation of structure and 3D motion are equivalent when the retinal motion is given, the two problems have received different names, the former "Structure from Motion" and the latter "Passive Navigation." We will refer to them interchangeably.

Basically there have been two approaches toward solving this problem. The first assumes 'small' motion. In this case, if the three-dimensional intensity function (two spatial and one temporal argument) is locally well behaved and its spatiotemporal gradients are defined, then the image velocity field (or optic flow) may be computed [23,26]. Algorithms developed using this approach use the velocity field to compute 3D motion and structure. The second approach assumes that the motion is large, and measurement of image motion entails solving the correspondence problem. Imaged feature points due to the same three-dimensional artifact (e.g., texture element or edge junction) in two successive dynamic frames are assumed to be identified correctly. Algorithms using this approach compute 3D transformation parameters from the above-mentioned displacements field [38,44,49]. There is a third approach, which computes 3D motion directly from brightness patterns, but the general case (unrestricted motion) has not yet been solved [5,40].

The 'small' (continuous) and 'large' (discrete) motion cases are slightly different in terms of the constraints that relate the 3D to the 2D motion, although the results are essentially the same [19]. So, we concentrate here only on the continuous case, where the input to the 'structure from motion' perceptual process is the optic-flow field. What will be developed in the sequel has meaning if and only if the optic flow (image velocity) is the projection of the three-dimensional motion. We state this explicitly, because the velocity of the brightness patterns in the image is computed (according to the existing literature) using some assumptions, which might violate the fact that image velocity is the projection of 3D motion. In the case where optic flow is used as input, there is some excellent research [16,42]. The basic problems of this passive approach are:

1. The constraint that relates 3D to 2D motion is nonlinear.
2. The dimensionality of the space of unknows is high (five, if one camera is used).

Sometimes, closed-form solutions may be found [35], but higher-order derivatives of the optic flow are involved. Thus, given that optic flow will be noisy (there is no algorithm to date that can compute optic flow in natural scenes with high accuracy) and also given that numerical differentiation is an ill-posed problem [42], the efficacy of these approaches is questionable. The possible employment of active tracking to facilitate navigation has been suggested by visual psychologists [17]. An active binocular observer greatly simplifies the constraints used in the analysis of motion and permits simple closed-form solutions of the resulting parameter equations.

It will be shown (for details see [13]) that when the observer is able to track a prominent feature point in the imaged scene, ,the task of navigation is facilitated since it is easier to compute egomotion parameters, compared to the nontracking case. The emphasis in this section is on the mathematics governing the imaging equations that are obtained while the system is tracking.

5.2 Constraints on Visual Motion for the Active Observer

In the monocular imaging situation, we have a sensor moving relative to a static scene. The

reference coordinate frame (X, Y, Z) is fixed to the sensor and the viewing direction is along the positive Z-axis. There is another coordinate frame fixed at a point S on the body. The point S has the velocity $T_S = (U_S, V_S, W_S)$. At the time of observation the reference and the body frame axes are parallel to each other. The rotational velocity of the body is given by the vector $\Omega = (\alpha, \beta, \gamma)$. The 3D velocity of a point $P = (X, Y, Z)$ on the body is given by the equation

$$V = T_S + [R](P - S) \tag{18}$$

where $X_S = (X_S, Y_S, Z_S)$ denote the position of the body origin S, V denotes the 3D velocity of P, and R is a matrix representation for Ω. Image formation is modeled by perspective projection. The projection of a point $P = (X, Y, Z)$ is denoted by $p = (x, y) = (fX/Z, fY/Z)$. The constant f is the focal length of the imaging system. It is the distance separating the nodal point of the camera (or eye) and the image plane, moving along the optical axis (i.e., Z axis). In subsequent steps the constant f is assumed to be unity. The velocity (u,v) of an image point (x,y) in the 2D image space is called optical flow. The relations between the 2D and 3D velocities are obtained as

$$\begin{aligned} u = \dot{x} &= \frac{U_S - xW_S}{Z} - \alpha\left[xy - x\frac{Y_S}{Z}\right] \\ &\quad + \beta\left[1 - \frac{Z_S}{Z} + x^2 - x\frac{X_S}{Z}\right] \\ &\quad - \gamma\left[y - \frac{Y_S}{Z}\right] \\ v = \dot{y} &= \frac{V_S - yW_S}{Z} - \alpha\left[1 - \frac{Z_S}{Z} + y^2 - y\frac{Y_S}{Z}\right] \\ &\quad + \beta\left[xy - y\frac{X_S}{Z}\right] + \gamma\left[x - \frac{X_S}{Z}\right] \end{aligned} \tag{19}$$

Let the moving observer track a single feature point so that it appears stationary on the retina at position (0, 0). The tracking motion consists of rotations about the axes that are orthogonal to the line of sight or the optical axis of the lens. It is a rotation $(\omega_x, \omega_y, 0)$, which is superimposed upon the actual parameters of motion.

Let $S = (0, 0, Z_0)$ be the spatial coordinates of the point being tracked. Assume that the observer can track an environmental point and hold it steady on the optical axis (Z-axis). Therefore the optical flow field will have a singularity at the origin of the retinal frame, where the flow value is zero.

The observer is moving with translation (U, V, W), measured with respect to the origin, and rotation (α, β, γ). Then the resultant parameters for the tracking observer are related by

$$\begin{aligned} U' &= \frac{U}{Z_0} = -(\beta + \omega_y) = -B \\ V' &= \frac{V}{Z_0} = (\alpha + \omega_x) = A \\ W' &= W \end{aligned} \tag{20}$$

When the optical axes of the two cameras converge onto a point in the environment, the tracking velocities, which are assumed to be observable, can be used to compute egomotion parameters. We will also assume a rectilinearly moving observer, with negligible instantaneous acceleration rates. The analysis generally deals with the left coordinate frame, with respect to which the various quantities will be written as in the monocular case. When it is needed to reference the quantities with respect to the right frame these will be written primed (e.g., x'). The tracking motion involves three independent rotation velocities $(\omega, \omega_y, \omega_y')$. The rotation ω is about the baseline of the imaging system. Hence the tracking motion of the left frame is given by $(\omega_x = -\omega \sin\theta, \omega_y, \omega_z = \omega\cos\theta)$. If Z_0 is the depth of the tracked point in the left frame then

$$\frac{2d}{\sin(\theta + \theta')} = \frac{Z_0}{\sin(\theta')} = \frac{Z_0'}{\sin(\theta)}$$

Thus we can write

$$Z_0(t) = 2d\,\frac{\sin(\theta')}{\sin(\theta + \theta')} = F[\theta(t), \theta'(t)]$$

also

$$\dot{F} = 2d\left[\frac{\dot{\theta}'\cos\theta'\sin(\theta + \theta') - (\dot{\theta} + \dot{\theta}')\sin\theta'\cos(\theta + \theta')}{\sin^2(\theta + \theta')}\right]$$

which simplifies to

$$\dot{F} = 2d\left[\frac{\dot{\theta}'\sin\theta - \dot{\theta}\sin\theta'\cos(\theta + \theta')}{\sin^2(\theta + \theta')}\right]$$

Differentiating the above relation with respect to time once more, we have

$$\ddot{F} = 2d\left[\frac{\ddot{\theta}' \cos\theta'}{\sin(\theta+\theta')} - \frac{(\dot{\theta}+\dot{\theta}')\sin\theta'\cos(\theta+\theta')}{\sin^2(\theta+\theta')}\right] + 2d\left[\frac{(\dot{\theta}+\dot{\theta}')^2\sin\theta' - (\dot{\theta})^2\sin\theta'}{\sin(\theta+\theta')}\right] - 4d\left[\frac{(\dot{\theta}+\dot{\theta}')\dot{\theta}'\cos\theta'\cos(\theta+\theta')}{\sin^2(\theta+\theta')} - \frac{(\dot{\theta}+\dot{\theta}')^2\sin\theta'\cos(\theta+\theta')}{\sin^3(\theta+\theta')}\right]$$

and simplifying the above leads to

$$\ddot{F} = 2d\left[\frac{\ddot{\theta}'\sin\theta - \ddot{\theta}\sin\theta'\cos(\theta+\theta')}{\sin^2(\theta+\theta')}\right] + 2d\left[\frac{\dot{\theta}(\dot{\theta}+2\dot{\theta}')\sin\theta'}{\sin(\theta+\theta')}\right] - 2(\dot{\theta}+\dot{\theta}')\cos(\theta+\theta')\dot{F}$$

Let the motion of the observer be described by the translational velocity $T = (U, V, W)$ and rotational velocity $\Omega = (\alpha, \beta, \gamma)$. These parameters are defined with respect to the point L in the body, which also happens to be the origin of the left coordinate frame. The tracking motion of the system consists of three independent rotations with respect to the observer. The tracking angular motion of the left frame with respect to the observer is given by ω_t and that of the right frame is ω_t'. Note that we will express all the motion parameters measured in a frame with respect to basis vectors defined in that frame. Therefore

$$\omega_t = (-\omega\sin\theta, \dot{\theta}, \omega\cos\theta)$$

$$\omega_t' = (-\omega\sin\theta', \dot{\theta}', -\omega\cos\theta')$$

If the rigid motion parameters with respect to the right coordinate frame are given by the translational velocity T' and the rotational velocity Ω', then we have

$$T' = R_\lambda \cdot (T + \Omega \times \rho)$$

$$\Omega' = R_\lambda \cdot \Omega$$

where $\times$ denotes the vector product and $\cdot$ denotes matrix multiplication. In addition, the rotation matrix R_λ expresses the transformation due to the rotation by $\lambda = \pi - (\theta + \theta')$ between the left and right frames. Now from equation (20) we have

$$\begin{aligned} \dot{F}(t) &= W \\ U + (\beta + \omega_y)F(t) &= 0 \\ V - (\alpha + \omega_x)F(t) &= 0 \end{aligned} \tag{21}$$

Observe that the above equations involve five unknown motion parameters. If we now differentiate these equations we have

$$\begin{aligned} \ddot{F}(t) &= \dot{W} \\ \dot{U} + \dot{\beta}F(t) + \beta\dot{F}(t) + \dot{\omega}_y F(t) + \omega_y\dot{F}(t) &= 0 \\ \dot{V} - \dot{\alpha}F(t) - \alpha\dot{F}(t) - \dot{\omega}_x F(t) - \omega_x\dot{F}(t) &= 0 \end{aligned} \tag{22}$$

Although here we consider a rectilinearly moving observer, the translational velocity (U,V,W) undergoes change due to the rotation of the frame in which the observations are made. Thus we obtain

$$\begin{aligned} \dot{U} &= (\beta + \omega_y)W - (\gamma + \omega_z)V \\ \dot{V} &= (\alpha + \omega_x)W + (\gamma + \omega_z)U \\ \dot{W} &= (\alpha + \omega_x)V - (\beta + \omega_y)U \end{aligned} \tag{23}$$

Similarly the rotational velocity (α, β, γ), undergoes change due to the tracking motion, as follows:

$$\begin{aligned} \dot{\alpha} &= \omega_y\gamma - \omega_z\beta \\ \dot{\beta} &= -\omega_x\gamma + \omega_z\alpha \\ \dot{\gamma} &= 0 \end{aligned}$$

Introducing the parameters $A = \alpha + \omega_x$, $B = \beta + \omega_y$, and $C = \gamma + \omega_z$, substituting for $\dot{U}$, $\dot{V}$, $\dot{W}$, $\dot{\alpha}$, and $\dot{\beta}$ from the above relations, and replacing U and V from (21), we have, from the last two equations in (22)

$$C = \frac{2B\dot{F}(t) + A\omega_z F(t) + \dot{\omega}_y F(t)}{(A + \omega_x)F(t)}$$

and

$$C = \frac{2A\dot{F}(t) - B\omega_z F(t) + \dot{\omega}_x F(t)}{-(B + \omega_y)F(t)}$$

Finally, eliminating C from the above pair of

equations and using the remaining equation of (22), we obtain the pair of independent equations

$$A^2 + B^2 = \frac{\ddot{F}(t)}{F(t)} = \phi_1 \tag{24a}$$

$$\phi_2 A + \phi_3 B + \phi_4 = 0 \tag{24b}$$

where

$$\phi_2 = 2\omega_x \dot{F}(t)F(t) + \dot{\omega}_x F^2(t) + \omega_y \omega_z F^2(t)$$

$$\phi_3 = 2\omega_y \dot{F}(t)F(t) + \dot{\omega}_y F^2(t) - \omega_x \omega_z F^2(t)$$

$$\phi_4 = (\omega_x \dot{\omega}_x + \omega_y \dot{\omega}_y)F^2(t) + 2\dot{F}(t)\ddot{F}(t)$$

From (24) we obtain two sets of solutions for the motion parameters. Eliminating the parameter B we have

$$aA^2 + bA + c = 0$$

where $a = \phi_2^2 + \phi_3^2$, $b = 2\phi_2\phi_4$, and $c = \phi_4^2 - \phi_1\phi_3^2$.

To summarize, the solution method consists of obtaining the solutions to the pair of equations (24a) and (24b). Since closed-form solutions are obtained at every time instant and assuming the computation errors to be uniformly random, we perform smoothing on the time series of the computed parameters to eliminate a large portion of the error.

The important aspects of this method of computation of the motion parameters are as follows:

(a) The solution is in closed-form, requiring no iteration or search.
(b) The constraints are derived from the observed tracking velocities and rotations. We do not need the optical flow measurements.
(c) Here the observables are, $(\theta, \theta', \dot{\theta}, \dot{\theta}', \ddot{\theta}, \ddot{\theta}')$. These can be measured quite accurately by analog measurement apparatus. This possibility forms a strong motivation for the tracking approach.
(d) The optical flow field in our motion perception scheme is used only to disambiguate between the possible interpretations computed by the tracking module. This is always possible since under extended periods of observation the optical flow field generated is compatible with one and only one intepretation [13].

Simulation experiments based on the above analysis show the approach to be robust against noise.

5.3 Summary and Conclusions

We have shown that a binocular observer that tracks a point on a moving object can recover 3D motion using a closed-form solution. It is worth noting that optic flow has not been used in the proposed schema. However, the obtained solution may be used as an initial estimator in a cooperative optic flow and structure computation algorithm. Optic flow and structure computation are strongly interdependent. An independent computational mechanism for motion parameter estimation can guide the motion correspondence process and enable the computation of structure. Namely, in our method, motion parameters are computed without using optical flow, and then these parameters along with the intensity profile enable us to uniquely compute optic flow.

6 Conclusions and Future Research

We have proposed a new paradigm for visual perception called active vision. Our methodology was demonstrated by showing its applicability in the abovementioned important areas of low- and intermediate-level vision. It was shown that the controlled alteration of viewing parameters uniquely computes shape and motion.

The basis for the approach lies in being able to work in a rich stimulus domain with a partially known parametrization. This knowledge is due to the fact that the viewing transformation is known. As the viewing parameters are continuously varied, the observed visual stimuli undergo local transformations that are measurable and provide powerful constraints for the computation of the unknown scene parameters. We are, of course, interested in the rates of these stimulus changes, which have traditionally been thought to be difficult to measure. However, it should be pointed out that in the present scheme we do not work with a small set of discrete observations, but with trajectories in the stimulus space, termed flow lines. These trajectories are

smooth, since the viewing transformations we use are themselves smooth, and therefore can be computed accurately enough for our purposes. Thus we do not need to rely on the smoothness of properties of the observed scene, such as illumination and depth. The real power of the method is due to the avoidance of complications usually associated with multiview approaches to visual perception. For instance, the problem of correspondence of microfeatures is not involved in the current approach.

The treatment in this paper has been largely theoretical, because we want to create a sound framework for what we believe is a promising methodology for computer vision. We plan to verify the analysis with experiments on synthetic and natural images. Also, we plan to continue our theoretical analysis for active visual computations before we build a system (possibly with special hardware) that will carry out active visual computations. The initial developments of the theory of active vision have been outlined. There are many important issues that need to be explored in this context. For instance, we have not looked at the question of whether there is any viewing parameter trajectory that is preferable to other trajectories in tackling the computational task. This could be termed exploratory visual computation, where the knowledge about scene constraints available at any stage of the visual process determines the way in which the control parameters will be altered. Also, a theoretical error analysis of the proposed algorithms has to be done, taking into account discretization effects and noise in images. Finally, the problem of learning the viewing parameter trajectory that is the best (in computational terms) with respect to a particular problem, using a neural (connectionist) network [20], has to be addressed. Examination of such issues forms an important future research goal, and our formulation for this task will follow the one presented by Aloimonos and Shulman [7].

Appendix A

Theorem. Let a coordinate system $OXYZ$ be fixed with respect to the left camera, with the Z-axis pointing along the optical axis. We assume that the image plane Im_1 is perpendicular to the Z axis at the point $(0, 0, 1)$ and that O is the nodal point of the left camera. Let the nodal point of the right camera be the point (R, L, O) and let its image plane be identical to the previous one, i.e., $\mathrm{Im}_1 = \mathrm{Im}_2$. Consider a polygon P on the world plane $Z = pX + qY + c$, defined by the points (X_i, Y_i, Z_i), $i = 1, \ldots, n$, and having area S_W. Let S_1, S_2 be the areas of the paraperspective projections of P on the left and right cameras respectively and S_1', S_2' the areas of the perspective projections of the polygon P on the left and right cameras respectively. Then

$$\frac{S_1}{S_2} = \frac{S_1'}{S_2'}$$

Proof. The proof is given in several parts.

Let (A_1, B_1) and (A_2, B_2) be the centers of mass of the projections of the contour P on the left and right image planes respectively (it has to be noted that (A_1, B_1) and (A_2, B_2) are the centers of mass of the actual left and right images as opposed to the projections of the center of mass of P onto the left and right image planes). Then we have

$$\frac{S_2}{S_1} = \frac{1 - A_2 p - B_2 q}{1 - A_1 p - B_1 q} \tag{A.1}$$

We will prove the above equation to be exact under perspective projection.

Now

$$A_1 = \frac{1}{n}\sum\left(\frac{X_i}{Z_i}\right), \qquad B_1 = \frac{1}{n}\sum\left(\frac{Y_i}{Z_i}\right)$$

and

$$A_2 = \frac{1}{n}\sum\left(\frac{X_i - R}{Z_i}\right), \qquad B_2 = \frac{1}{n}\sum\left(\frac{Y_i - L}{Z_i}\right)$$

Substituting in (A.1) we get, after some tedious manipulations,

$$\frac{S_2}{S_1} = 1 + \frac{pR + qL}{c} \tag{A.2}$$

On the other hand, we can easily prove that

$$\frac{S_2}{S_1} = 1 + R\,\frac{\sum\left(\frac{Y_i - Y_{i+1}}{Z_i Z_{i+1}}\right)}{M}$$

$$+ L \frac{\sum \left(\frac{X_i - X_{i+1}}{Z_i Z_{i+1}} \right)}{M} \tag{A.3}$$

with

$$M = \sum \left(\frac{X_i Y_{i+1} - X_{i+1} Y_i}{Z_i Z_{i+1}} \right) \tag{A.4}$$

We can also easily prove that

$$\frac{\sum \left(\frac{Y_i - Y_{i+1}}{Z_i Z_{i+1}} \right)}{M} = \frac{p}{c} \tag{A.5}$$

and

$$\frac{\sum \left(\frac{X_i - X_{i+1}}{Z_i Z_{i+1}} \right)}{M} = \frac{q}{c} \tag{A.6}$$

From equations (A.2), (A.3), (A.4), (A.5), and (A.6) the proof of the theorem is immediate. The previous theorem can be easily shown to hold true even if we do not consider the contours as a collection of points.

Appendix B

Theorem. With the nomenclature of the previous theorem (appendix A) and assuming only horizontal displacement of the cameras, if S_L, S_R denote the areas of the left and right images, and the world plane from which the image contour is obtained is $Z + pX + qY + c$, then

$$\frac{S_L}{S_R} = \frac{1}{1 + \frac{dx}{c} p}$$

(dx = displacement between camera 1 and camera 2).

Proof.

$$S_L = \frac{1}{2} \left(\sum x_i^L y_{i+1}^L - \sum x_{i+1}^L y_i^L \right) \qquad \left[\begin{matrix} \text{Area of a} \\ \text{polygon} \end{matrix} \right]$$

$$S_R = \frac{1}{2} \left(\sum x_i^R y_{i+1}^R - \sum x_{i+1}^R y_i^R \right)$$

where (x^L, y^L), (x^R, y^R) are the coordinates in the left and right image planes, respectively. Now

$$y_{is} = y_i^R \tag{i}$$

and

$$x_i^R = \frac{f(x_i^L - dx)}{Z_i} = x_i^L - \frac{f\,dx}{Z_i} \tag{ii}$$

So

$$\frac{S_L}{S_R} = \frac{\frac{1}{2} \left(\sum x_i^L y_{i+1}^L - \sum x_{i+1}^L y_i^L \right)}{\frac{1}{2} \left(\sum x_i^R y_{i+1}^R - \sum x_{i+1}^R y_i^R \right)}$$

Then, using (i) and (ii),

$$\frac{S_L}{S_R} = \frac{1}{\frac{\sum \left(x_i^L - \frac{f\,dx}{Z_i} \right) y_{i+1}^L - \sum \left(x_{i+1}^L - \frac{f\,dx}{Z_{i+1}} \right) y_i^L}{\sum x_i^L y_{i+1}^L - \sum x_{i+1}^L y_i^L}}$$

$$= \frac{1}{1 + f\,dx \; \frac{\sum \frac{y_{i+1}^L}{Z_i} - \sum \frac{y_i^L}{Z_{i+1}}}{\sum x_i^L y_{i+1}^L - \sum x_{i+1}^L y_i^L}}$$

$$= \frac{1}{1 + \frac{dx}{c} p \; \frac{fc}{p} \; \frac{\sum \frac{y_i^L}{Z_{i+1}} - \sum \frac{y_{i+1}^L}{Z_i}}{\sum x_i^L y_{i+1}^L - \sum x_{i+1}^L y_i^L}}$$

Thus we need to show

$$\frac{fc}{p} \frac{\left(\sum \frac{Y_i^L}{Z_{i+1}} - \sum \frac{y_{i+1}}{Z_i} \right)}{\left(\sum x_i^L y_{i+1}^L - \sum x_{i+1}^L y_i^L \right)} = 1$$

that is,

$$\sum y_i^L \frac{fc}{Z_{i+1}} - \sum y_{i+1}^L \frac{fc}{Z_i} = \sum (p x_i^L) y_{i+1}^L$$

$$- \sum (p x_{i+1}^L) y_i^L$$

But

$$Z_i = p x_i^L + q y_i^L + c$$

$$\Rightarrow \frac{c}{Z_i} = 1 - p \frac{x_i^L}{Z_i} - 1 \frac{y_i^L}{Z_i}$$

$$\Rightarrow \frac{fc}{Z_i} = f - p\frac{fx_i^L}{Z_i} - q\frac{fy_i^L}{Z}$$
$$= f - px_i^L - qy_i^L$$

Hence

$$\sum y_i^L \frac{fc}{Z_{i+1}} - \sum y_{i+1}\frac{fc}{Z_i}$$
$$= \sum y_i^L(f - px_{i+1}^L - qy_{i+1}^L)$$
$$- \sum y_{i+1}^L(f - px_i^L - qy_i^L)$$
$$= f\left(\sum y_i^L - \sum y_{i+1}^L\right)$$
$$+ \left(\sum (px_i^L)y_{i+1}^L - \sum (px_{i+1}^L)y_i^L\right)$$
$$+ q\left(\sum y_i^L y_{i+1}^L - \sum y_{i+1}^L y_i^L\right) \qquad \text{(since, } y_{n+1} = y_1$$
$$= \sum (px_i^L)y_{i+1}^L - \sum (px_{i+1}^L)y_i^L \qquad \text{by notation)}$$

Thus we have proved the thorem.

References

1. G. Adiv, "Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field," PROC. IEEE CONF. ON COMPUTER VISION AND PATTERN RECOGNITION, pp. 70-77, 1985.
2. J. Aloimonos, "Computing intrinsic images," Ph.D. thesis, Univ. of Rochester, 1986.
3. J. Aloimonos and A. Bandyopadhyay, "Correspondence is not necessary for motion perception," submitted.
4. J. Aloimonos and A. Basu, "Shape from contour," submitted, 1986.
5. J. Aloimonos and C.M. Brown, "Direct processing of curvilinear sensor motion from a sequence of perspective images," PROC. IEEE WORKSHOP ON COMPUTER VISION, Annapolis, MD, 1984.
6. J. Aloimonos and J.-Y. Herve, "Correspondenceless detection of depth and motion for a planar surface," to appear, 1987.
7. J. Aloimonos and D. Shulman, "Learning shape computations," to appear, 1987.
8. J. Aloimonos and D. Tsakiris, "Shape from nonplanar contour," in preparation, 1987.
9. S. Amari, "Feature spaces which admit and detect invariant signal transformations," PROC. ICPR, Tokyo, 1978.
10. S. Amari and S. Maruyama, "Computation of structure from motion," personal communication, 1986.
11. R. Bajcsy, "Active perception vs. passive perception," PROC. IEEE WORKSHOP ON COMPUTER VISION, Ann Arbor, MI, 1986.
12. A. Bandyopadhyay, personal communication, 1987.
13. A. Bandyopadhyay, "A computational study of rigid motion perception," Ph.D. Thesis, Dept. of Computer Science, Univ. of Rochester, 1986.
14. J. Brady and A. Yuille, "An extremum principle for shape from contour," IEEE TRANS. PAMI-6, pp. 288-301, 1984.
15. C.M. Brown, Personal communication, 1986.
16. A. Bruss and B.K.P. Horn, "Passive navigation," COMPUTER VISION, GRAPHICS AND IMAGE PROCESSING, vol. 21, 1983.
17. J.E. Cutting, "Motion parallax and visual flow: how to determine direction of locomotion," Dept. of Psychology, Cornell Univ., 1982.
18. L. Davis, L. Janos, and S. Dunn, "Efficient recovery of shape from texture," Tech. Report 1133, Computer Vision Laboratory, Univ. of Maryland, 1982.
19. J.Q. Fang and T.S. Huang, "Solving three dimensional small rotation motion equations: uniqueness, algorithms, and numerical results," COMPUTER VISION, GRAPHICS AND IMAGE PROCESSING, vol. 26, pp. 183-206, 1984.
20. J. Feldman, "Four frames suffice: A provisional model of vision and space," BEHAVIORAL AND BRAIN SCIENCES, June, 1985.
21. J.J. Gibson, THE PERCEPTION OF THE VISUAL WORLD. Houghton Mifflin: Boston, 1951.
22. J. Hadamard, LECTURES ON THE CAUCHY PROBLEM IN LINEAR PARTIAL DIFFERENTIAL EQUATIONS. New Haven: Yale University Press.
23. E.C. Hildreth, "Computations underlying the measurement of visual motion," ARTIFICIAL INTELLIGENCE, vol. 23, pp. 309-354, 1984.
24. B.K.P. Horn, ROBOT VISION, McGraw-Hill: New York, 1986.
25. B.K.P. Horn, "Understanding image intensities," ARTIFICIAL INTELLIGENCE, vol. 8, 1977.
26. B.K.P. Horn and B. Schunck, "Determining optical flow," ARTIFICAL INTELLIGENCE, vol. 17, pp. 185-204, 1981.
27. K. Ikeuchi and B.K.P. Horn, "Numerical shape from shading and occluding boundaries," ARTIFICIAL INTELLIGENCE, vol. 17, 1981.
28. E. Ito and J. Aloimonos, "Determining transformation parameters from images: theory," in PROC. IEEE CONF. ON ROBOTICS AND AUTOMATION, 1987a.
29. E. Ito and J. Aloimonos, "Shape from nonplanar contour," to appear, 1987b.
30. T. Kanade, "Determining the shape of an object from a single view," ARTIFICIAL INTELLIGENCE, 17, 1981.
31. K. Kanatani, "Group theoretical methods in image understanding," Tech. Report 1692, Computer Vision Laboratory, Univ. of Maryland, 1986.
32. S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi, "Op-

timization by simulated annealing," RC 9355 (#41093), IBM T.J. Watson Research Center, Yorktown Heights, NY.
33. D. Lee and T. Pavlidis, personal communication, 1986.
34. H.C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," NATURE, vol. 293, pp. 133–135, 1981.
35. H.C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," PROC. ROY. SOC. (LONDON), vol. B208, pp. 385–397, 1980.
36. D. Marr, VISION. W.H. Freeman: San Francisco, 1982.
37. V.A. Morozov, REGULARIZATION METHODS FOR SOLVING ILL-POSED PROBLEMS. Springer-Verlag: New York, 1984.
38. H.H. Nagel and B. Neumann, "On 3-D reconstruction from two perspective view," PROC. SEVENTH INT. JOINT CONF. ARTIF. INTEL., Vancouver, pp. 661–663, 1981.
39. V. Nalwa, "Detecting edges in images," personal communication, 1985.
40. S. Negahdaripour and B.K.P. Horn, "Determining 3-D motion of planar objects from image brightness patterns," PROC. NINTH INT. JOINT CONF. ARTIF. INTEL., Los Angeles, pp. 898–901, 1985.
41. T. Poggio and C. Koch, "Ill-posed problems in early vision: from computational theory to analog networks," PROC. ROY. SOC. (LONDON), B, 1985.
42. T. Poggio and the staff, "MIT progress in understanding images," PROC. DARPA IMAGE UNDERSTANDING WORKSHOP, Miami, 1985.
45. K. Prazdny, "Determining the instantaneous direction of motion from optical flow generated by curvilinearly moving observer," COMPUTER VISION, GRAPHICS AND IMAGE PROCESSING, vol. 17, pp. 94–97, 1981.
45. J.W. Roach and J.K. Aggarwal, "Determining the movement of objects from a sequence of images," IEEE TRANS. PAMI–2, 1980.
46. D. Shulman and J. Aloimonos, "A linear theory of discontinuous regularization," submitted, 1987.
47. K. Stevens, "The information content in texture gradients," BIOLOGICAL CYBERNETICS, vol. 42, pp. 95–105, 1982.
48. D. Terzopoulos, "Regularization of inverse problems involving discontinuities," IEEE TRANS. PAMI–8, pp. 413–425, 1986.
49. A.N. Tichonov and V.Y. Arsenin, SOLUTIONS OF ILL-POSED PROBLEMS. Winston: Washington, 1977.
50. 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, pp. 13–27, 1984.
51. S. Ullman, "The interpretation of structure from motion," PROC. ROY. SOC. (LONDON), vol. B203, pp. 405–426, 1979.
52. A. Waxman and S. Ullman, "Surface structure and 3-D motion from image flow: a kinematic analysis," CAR-TR-24, Center for Automation Research, Univ. of Maryland, October 1983.
53. A. Witkin, "Recovering surface orientation and shape from texture," ARTIFICIAL INTELLIGENCE, vol. 17, 1981.
54. L. Weiss, "3-D shape reconstruction on a varying mesh," PROC. IMAGE UNDERSTANDING WORKSHOP, 1987.