New Eyes for Building Models from Video

Cornelia Fermüller [a], Yiannis Aloimonos [a] and Tomáš Brodský [b]

[a] *Computer Vision Laboratory, Center for Automation Research, Institute for Advanced Computer Studies, and Computer Science Department, University of Maryland, College Park, MD 20742-3275, USA*

[b] *Philips Research, 345 Scarborough Road, Briarcliff Manor, NY 10510, USA*

Abstract

Models of real-world objects and actions for use in graphics, virtual and augmented reality and related fields can only be obtained through the use of visual data and particularly video. This paper examines the question of recovering shape models from video information. Given video of an object or a scene captured by a moving camera, a prerequisite for model building is to recover the three-dimensional (3D) motion of the camera which consists of a rotation and a translation at each instant. It is shown here that a spherical eye (an eye or system of eyes providing panoramic vision) is superior to a camera-type eye (an eye with restricted field of view such as a common video camera) as regards the competence of 3D motion estimation. This result is derived from a geometric/statistical analysis of all the possible computational models that can be used for estimating 3D motion from an image sequence. Regardless of the estimation procedure for a camera-type eye, the parameters of the 3D rigid motion (translation and rotation) contain errors satisfying specific geometric constraints. Thus, translation is always confused with rotation, resulting in inaccurate results. This confusion does not happen for the case of panoramic vision. Insights obtained from this study point to new ways of constructing powerful imaging devices that suit particular tasks in visualization and virtual reality better than conventional cameras, thus leading to a new camera technology. Such new eyes are constructed by putting together multiple existing video cameras in specific ways, thus obtaining eyes from eyes. For a new eye of this kind we describe an implementation for deriving models of scenes from video data, while avoiding the correspondence problem in the video sequence.

1 Introduction

An important problem in virtual reality and the associated areas of augmented reality, telereality, tele-immersion, graphics and visualization is to create models of the environment, i.e., models of space-time. These are descriptions of objects and scenes and descriptions of changes of space over time, that is, events and actions. Availability of such models allows one to insert them in specific settings for the purpose of creating a particular, realistic impression. A lot of progress has been achieved by using synthetic models but real world objects and events are not well generated in this manner. There is a growing sense that this problem will be solved by taking advantage of real images but it is not yet clear how this can be achieved. A field whose main goal is the development of representations of the world from images is known as computational or computer vision. Theoretically, many image cues could be utilized to recover models of the depicted scene. But the most successful cues are based on motion. The reason is that these cues have a geometric character whose basics are more or less understood. Thus, there exist today many approaches attempting to recover models of a scene on the basis of multiple views of that scene, or on the basis of a video depicting the scene. (See [35] for a review.) Advances in technology make it possible to easily digitize video and perform experiments, thus complementing theoretical approaches. Yet, despite the advances in specialized domains—special scenes such as buildings or special motions such as turntables—the problem of recovering shape descriptions of objects as well as descriptions of actions from video is still far from solved. What are some of the fundamental reasons for slow progress? What can be done?

This paper considers this problem as an application of computational geometry (CG). It has not been treated in the traditional CG literature as it is not only a geometric problem but also a statistical one. Measurements in an image are imperfect and one needs to consider how the input data is altered.

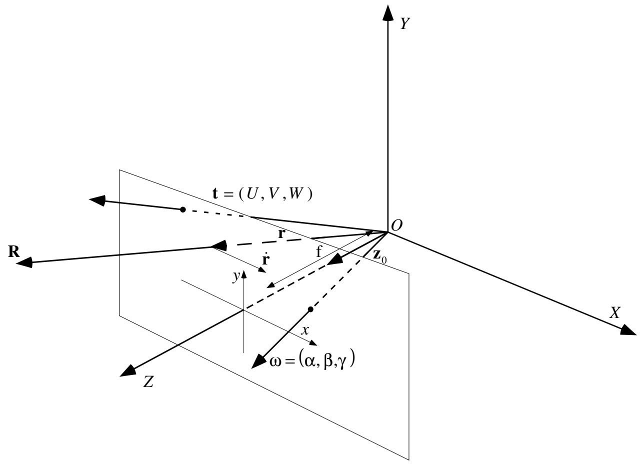
2 Pinpointing the Technical Difficulties

Images, for a standard pinhole camera, are formed by central projection on a plane (Fig. 1a). The focal length is f and the coordinate system $OXYZ$ is attached to the camera, with Z being the optical axis, perpendicular to the image plane.

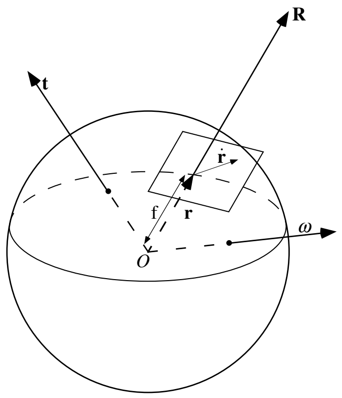
Image points are represented as vectors $\mathbf{r} = [x, y, f]^{\mathrm{T}}$, where x and y are the image coordinates of the point in the coordinate system oxy, with $ox \| OX$, $oy \| OY$ and O the intersection of the axis OZ with the image plane, and f is the focal length in pixels. A scene point $\mathbf{R}$ is projected onto the image point

$$\mathbf{r} = f \frac{\mathbf{R}}{\mathbf{R} \cdot \hat{\mathbf{z}}} \tag{1}$$

where $\hat{\mathbf{z}}$ is the unit vector in the direction of the Z axis.



(a)



(b)

Fig. 1. Image formation on the plane (a) and on the sphere (b). The system moves with a rigid motion with translational velocity $\mathbf{t}$ and rotational velocity $\boldsymbol{\omega}$. Scene points $\mathbf{R}$ project onto image points $\mathbf{r}$ and the 3D velocity $\dot{\mathbf{R}}$ of a scene point is observed in the image as image velocity $\dot{\mathbf{r}}$.

In the case of video, the camera is moved to different locations while acquiring new images. Thus, video acquired by a moving camera amounts to a collection of images of a scene, i.e., projections onto a plane, acquired from different viewpoints. Figuring out a model for the scene and the movement in the scene becomes a problem of relating the different projections (images) to each other.

In general, when a scene is viewed from two positions, there are two concepts of interest:

(a) The 3D transformation relating the two viewpoints. This is a rigid motion transformation, consisting of a translation and a rotation (six degrees of freedom). When the viewpoints are close together, this transformation is modeled by the 3D motion of the eye (or camera).
(b) The 2D transformation relating the pixels in the two images, i.e., a transformation that given a point in the first image maps it onto its corresponding one in the second image (that is, these two points are the projections of the same scene point). When the viewpoints are close together, this transformation amounts to a vector field denoting the velocity of each pixel, called an image motion field.

Perfect knowledge of both transformations described above leads to perfect knowledge of models of space and action. Regarding models of space, this is easy to understand. Knowing exactly how the two viewpoints and the images are related provides the exact position of each scene point in space. Regarding models of action, knowing the exact velocity of each image point, by projecting back onto the scene, for which a model is available by the previous step, we can find the *3D motion vector* for each scene point. This sequence of evolving

3D motion fields constitutes a general model of action (since action is the extension of shape into time).

Thus, a key to the basic problem of building models of space-time is the recovery of the two transformations described before and any difficulty in building such models can be traced to the difficulty of estimating these two transformations. Conceptually, it doesn't matter if the viewpoints are close to each other or far apart. For consistency, since we will be working with video data where the successive viewpoints are differentially related, we consider them close by and thus the 3D transformation becomes a camera's *3D motion* and the 2D transformation becomes an *image motion field*. These two transformations are further discussed in the rest of this section, after some nomenclature is introduced.

2.1 Nomenclature

Consider a camera with the geometric model of Fig. 1a moving in a static environment with instantaneous translation $\mathbf{t} = (U, V, W)$ and instantaneous rotation $\boldsymbol{\omega} = (\alpha, \beta, \gamma)$ (measured in the coordinate system $OXYZ$). Then a scene point $\mathbf{R}$ moves with velocity (relative to the camera)

$$\dot{\mathbf{R}} = -\mathbf{t} - \boldsymbol{\omega} \times \mathbf{R} \tag{2}$$

The image motion field is then [16]

$$\dot{\mathbf{r}} = -\frac{1}{(\mathbf{R} \cdot \hat{\mathbf{z}})}(\hat{\mathbf{z}} \times (\mathbf{t} \times \mathbf{r})) + \frac{1}{f}\hat{\mathbf{z}} \times (\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})) = \frac{1}{Z}\mathbf{u}_{\mathrm{tr}}(\mathbf{t}) + \mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega}) \tag{3}$$

where Z is used to denote the scene depth $(\mathbf{R} \cdot \hat{\mathbf{z}})$, and $\mathbf{u}_{\mathrm{tr}}, \mathbf{u}_{\mathrm{rot}}$ the direction of the translational flow and the rotational flow respectively. Due to the scaling ambiguity, only the direction of translation (focus of expansion—FOE, or focus of contraction—FOC, depending on whether the observer approaches or moves away from the scene), also known as the epipole, and the three rotational parameters can be estimated from monocular image sequences [5].

Eq. (3) demonstrates model construction. If the image motion vector $\dot{\mathbf{r}}$ is known at point $\mathbf{r}$, then knowledge of $\mathbf{t}$ (up to scale) and $\boldsymbol{\omega}$ provides Z (up to scale), i.e., the depth at point $\mathbf{r}$ in the camera's coordinate system. Knowledge of Z (or, equivalently, $\mathbf{R}$) for all image points $\mathbf{r}$ provides a model for the scene in view, for the current viewpoint of the camera. Knowledge of $\mathbf{t}, \boldsymbol{\omega}$ and $\mathbf{R}$ provides then, from eq. (2), knowledge of $\dot{\mathbf{R}}$ (up to scale), that is, the 3D motion vector. A sequence of 3D motion vector fields is a model of action, as it shows how different parts of space move. Of course, there exist many issues to be addressed before models can be built, but recovery of the camera's 3D motion and the image motion field are the essential prerequisites for acquiring scene depth, which is the cornerstone of the model building process. It is thus important to understand any inherent geometric limitations in accomplishing these two processes.

3 Inherent Limitations

3.1 Image motion fields

If $\mathbf{r}$ is an image point (x, y, f), the motion vector $\dot{\mathbf{r}}$ lies on the image plane and its third coordinate is zero. Let us express by (u, v) the first two components of $\dot{\mathbf{r}}$.

If $I(x, y, t)$ represents the intensity of the time varying image, the motion field (u, v) at an image point satisfies the following constraint [15]:

$$I_x u + I_y v + I_t = 0 \qquad \text{or} \qquad (I_x, I_y) \cdot (u, v) = -I_t$$

where I_x, I_y are the spatial derivatives and I_t the temporal derivative of the image. The vector (u, v) shows how the image point moves on the basis of image measurement. This approximation of the motion field is called optic flow or just flow. The above equation shows that the projection of flow vector (u, v) on the image gradient (I_x, I_y) (i.e., perpendicular to the local edge) is known. This quantity is called normal flow and denoted as u_n. In our nomenclature,

$$u_n = \dot{\mathbf{r}} \cdot \mathbf{n}. \tag{4}$$

where $\mathbf{n}$ is a unit vector at an image point denoting the orientation of the gradient at that point. The normal flow is a robust measurement from a moving image and can be computed locally and in parallel. To compute then the values of the flow, one would need to utilize the normal flow values along with additional constraints.

Estimation of optic flow is a problem for which thousands of references can be found in the literature. (See [35] for a review.) All approaches are basically relying on the following concept: Consider a small image patch for which normal flow values $u_{n_1}, u_{n_2}, \ldots, u_{n_p}$ (the motion components perpendicular to local edges) have been computed. Then, one assumes a model for the flow in that patch and uses the normal flow values to fit this model. For example, if the flow is assumed constant in the neighborhood, one searches for the value (u, v) that when projected onto the local gradients best fits the values $u_{n_1}, u_{n_2}, \ldots, u_{n_p}$, where usually a least square minimization is employed but other minimizations have also been used. Alternately, one may assume more sophisticated models, such as affine, quadratic, high level polynomial and the like. Some popular approaches are based on smoothness, i.e., no parametric model is assumed but instead the assumption is made that the flow field in the patch is a smooth function and the solution is obtained through a regularization procedure [1]. The problem with all these approaches is that they work adequately when the image patch considered is the image of a smooth scene patch, but they lead to wrong estimates when the scene patch contains depth discontinuities, because the values of the flow on the two sides of the discontinuity are quite different and thus the assumed models are not valid. There exist more sophisticated optimization approaches that attempt to estimate the flow while at the same time locating discontinuities, but these techniques are very slow, require a large number of resources, and their success is data dependent. They do not work successfully all the time and we cannot know a priori whether they will be successful.

If we knew where the discontinuities are, estimating flow would be easy, but to know where the discontinuities are we need first to find 3D motion and use it to find depth—but to do that we need to know the values of the flow! The whole problem is clearly a chicken/egg problem.

There exists an additional reason causing incorrect flow estimates that only recently was understood [14], and is related to the image texture. It has to do with the statistical difficulty of integrating local, 1D motion signals into 2D image velocity measurements. Any procedure for estimating image motion has to start with normal flow measurements, that is, the image motion component perpendicular to local edges. It has been shown [14] that when these local measurements are combined in a neighborhood to produce image motion, an estimate of flow is obtained which is biased. The estimated value depends on the distribution of image gradients, the actual flow and the error in the normal flow. This is strikingly observed in the Ouchi illusion (Fig. 2). The pattern in Fig. 2 has the surprising property that small motions can cause illusory relative motion between the inset and background regions. [1] The reason for this illusion is that for the particular spatial gradient distributions of the Ouchi pattern, the bias in the estimation of flow is highly pronounced, giving rise to a large difference in the velocity estimates in the two regions. Situations like this occur too often in real imagery (neighboring textures of different orientation).

Thus, there are two basic problems with the estimation of correspondence,

[1] The effect can be attained with small retinal motions or a slight jiggling of the paper and is robust over large changes in the patterns, frequencies and boundary shapes.

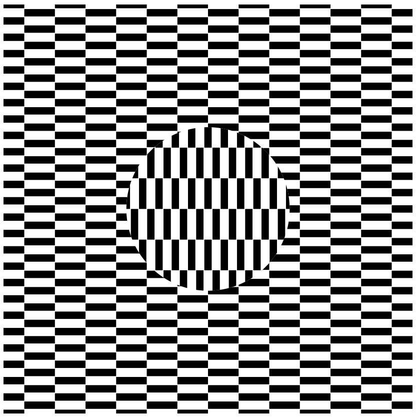


Fig. 2. A pattern similar to one by Ouchi.

i.e., the motion field. One is geometric, related to scene discontinuities, and the other is statistical, related to how the image texture looks.

3.2 3D motion

Let's assume that, despite the problems mentioned, a motion field can be estimated to some degree of accuracy, and thus optic flow is available. There exists a veritable cornucopia of techniques for finding 3D motion from optic flow. Almost all techniques are based on the so-called epipolar constraint, which shows how the motion of image points is related to 3D rigid motion and the scene. The epipolar constraint can be easily understood in the discrete case. Consider two cameras at two positions, with their coordinate systems related by a rigid transformation, and a scene point. The scene point, together with the camera centers define the so called epipolar plane which intersects the image planes in the epipolar lines. The epipolar constraint then states that a point in one image has to be matched with a point lying on the corresponding epipolar line in the other image. The distance of the matched point from the epipolar line is called the epipolar error. Minimization of epipolar errors is the basis of most 3D motion estimation algorithms. For the differential case of video, the epipolar constraint is obtained from the image motion equations as $(\mathbf{t}\times\mathbf{r})\cdot(\dot{\mathbf{r}}+\boldsymbol{\omega}\times\mathbf{r}) = 0$ [8]. One is interested in the estimates of translation $\hat{\mathbf{t}}$ and rotation $\hat{\boldsymbol{\omega}}$ which best satisfy the epipolar constraint at every point $\mathbf{r}$ according to some criteria of deviation. Usually the Euclidean norm is considered leading to the minimization of function.

$$M_{ep} = \iint\limits_{\text{image}} \left[\left(\hat{\mathbf{t}} \times \mathbf{r}\right) \cdot \left(\dot{\mathbf{r}} + \hat{\boldsymbol{\omega}} \times \mathbf{r}\right)\right]^2 d\mathbf{r} \tag{5}$$

Experience has shown that estimating 3D motion by minimizing the above functional, or variations of it, is a very difficult problem. This is the reason for the large amount of literature on this issue. One main reason for this difficulty has to do with the apparent confusion between translation and rotation in the motion field. This is easy to understand at an intuitive level. If we look straight ahead at a shallow scene, whether we rotate around our vertical axis or translate parallel to the scene, the motion field at the center of the image is very similar in the two cases. Thus, for example, translation along the x axis is confused with rotation around the y axis. The basic understanding of this confusion has attracted few investigators over the years. (See [7,8] for a review.) In this paper it is shown that the confusion exists no matter what estimator is used, proving that there is an inherent limitation to the estimation of 3D motion from data of only a limited field of view. To be more precise, a statistical analysis of all the possible computational models that can be used to derive 3D motion is given. Next, this analysis is carried out for the classic epipolar minimization.

Any approach to 3D motion estimation using as input optic flow would minimize function (5). Thus, we perform a topographic analysis of the five-dimensional surface described by this function (two dimensions for $\mathbf{t}/|\mathbf{t}|$ and three for $\boldsymbol{\omega}$). We want to know how the valleys of (5) are structured and what the properties of the minima are at the locations that will be found by different estimators. Specifically, we are interested in the relationship between the 3D motion errors in the minima of (5). Expressing $\dot{\mathbf{r}}$ in terms of the real motion, function (5) can be expressed in terms of the actual and estimated motion parameters $\mathbf{t}$, $\boldsymbol{\omega}$, $\hat{\mathbf{t}}$ and $\hat{\boldsymbol{\omega}}$ (or, equivalently, the actual motion parameters $\mathbf{t}, \boldsymbol{\omega}$ and the errors $\mathbf{t}_\epsilon = \mathbf{t} - \hat{\mathbf{t}}$, $\boldsymbol{\omega}_\epsilon = \boldsymbol{\omega} - \hat{\boldsymbol{\omega}}$) and the depth Z of the viewed scene. To conduct any analysis, a model for the scene is needed. We are interested in the statistically expected values of the motion estimates resulting from all possible scenes. Thus, as our probabilistic model we assume that the depth values of the scene are uniformly distributed between two arbitrary values $Z_{\min}$ and $Z_{\max}$ ($0 < Z_{\min} < Z_{\max}$).

Thus, we obtain the function

$$E_{ep} = \int\limits_{Z=Z_{\min}}^{Z=Z_{\max}} M_{ep} dZ \tag{6}$$

measuring deviation from the epipolar constraint. Since for the scene in view we employ a probabilistic model, the results are of a statistical nature, that is, the geometric constraints between $\mathbf{t}_\epsilon, \boldsymbol{\omega}_\epsilon$ at the minima of (6) that we shall uncover should be interpreted as being likely to occur. Our approach expresses function (6) in terms of $\mathbf{t}$, $\boldsymbol{\omega}$, $\mathbf{t}_\epsilon$ and $\boldsymbol{\omega}_\epsilon$ and finds the conditions that $\mathbf{t}_\epsilon$ and $\boldsymbol{\omega}_\epsilon$ satisfy at the local minima which represent solutions of the different estimation algorithms. Procedures for estimating 3D motion can be classified into those estimating either the translation or rotation as a first step and the remaining component (that is, the rotation or translation) as a second step, and those estimating all components simultaneously. Procedures of the former kind result when systems utilize inertial sensors which provide them with estimates of one of the components of the motion, or when two-step motion estimation algorithms are used.

Thus, three cases need to be studied: the case were no prior information about 3D motion is available and the cases where an estimate of translation or rotation is available with some error. Imagine that somehow the rotation has been estimated, with an error $\boldsymbol{\omega}_\epsilon$. Then our function becomes two-dimensional in the variables $\mathbf{t}_\epsilon$ and represents the space of translational error parameters corresponding to a fixed rotational error. Similarly, given a translational error $\mathbf{t}_\epsilon$, the functions become three-dimensional in the variables $\boldsymbol{\omega}_\epsilon$ and represent the space of rotational errors corresponding to a fixed translational error. To study the general case, one needs to consider the lowest valleys of the functions in 2D subspaces which pass through 0. In the image processing literature, such local minima are often referred to as ravine lines or courses.

The following convention is employed throughout the paper. We use letters with hat signs to represent estimated quantities, unmarked letters to represent the actual quantities and the subscript "ϵ" to denote errors, where the error quantity is defined as the actual quantity minus the estimated one. For example, $\mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega})$ represents actual rotational flow, $\mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}})$ estimated rotational flow, $\mathbf{t}_\epsilon$ the translational error vector, $x_{0_\epsilon} = x_0 - \hat{x}_0$, $\alpha_\epsilon = \alpha - \hat{\alpha}$, etc.

Let $\mathbf{t} = (x_0, y_0, 1)$ and $\boldsymbol{\omega} = (\alpha, \beta, \gamma)$. Since the field of view is small, the quadratic terms in the image coordinates are very small relative to the linear and constant terms, and are therefore ignored. All the computations are carried out with the symbolic algebraic computation software Maple, and, for abbreviation, intermediate results are not given. The case of noise-free flow is studied, in which case the analysis becomes a study of the inherent geometric confusion between rotation and translation.

Considering a circular aperture of radius e, setting the focal length $f = 1$, $W = 1$ and $\hat{W} = 1$, the function in (6) becomes

$$E_{ep} = \int\limits_{Z=Z_{\min}}^{Z_{\max}} \int\limits_{r=0}^{e} \int\limits_{\phi=0}^{2\pi} \Bigg\{ \Bigg(\left(\frac{x - x_0}{Z} - \beta_\epsilon + \gamma_\epsilon y + x \right) (y - \hat{y}_0) - \left(\frac{y - y_0}{Z} + \alpha_\epsilon - \gamma_\epsilon x + y \right) (x - \hat{x}_0) \Bigg)^2 r \Bigg\} dr\, d\phi\, dZ$$

where (r, ϕ) are polar coordinates $(x = r\cos\phi, y = r\sin\phi)$. Performing the integration, one obtains

$$\begin{aligned} E_{ep} = \pi e^2 \Bigg(&(Z_{\max} - Z_{\min}) \left(\frac{1}{3}\gamma_\epsilon^2 e^4 + \frac{1}{4}\Big(\gamma_\epsilon^2 \left(\hat{x}_0^2 + \hat{y}_0^2\right) + 6\gamma_\epsilon \left(\hat{x}_0\alpha_\epsilon + \hat{y}_0\beta_\epsilon\right) \right. \\ &\left. + \alpha_\epsilon^2 + \beta_\epsilon^2\Big) e^2 (\hat{x}_0\alpha_\epsilon + \hat{y}_0\beta_\epsilon)^2 \right) + (\ln(Z_{\max}) - \ln(Z_{\min})) \left(\frac{1}{2}(3\gamma_\epsilon(x_{0_\epsilon} y_0 \right. \\ &\left. - y_{0_\epsilon} x_0) + x_{0_\epsilon}\beta_\epsilon - y_{0_\epsilon}\alpha_\epsilon)e^2 + 2\left(x_{0_\epsilon} y_0 - y_{0_\epsilon} x_0\right)\left(\hat{x}_0\alpha_\epsilon + \hat{y}_0\beta_\epsilon\right) \right) \\ &+ \left(\frac{1}{Z_{\min}} - \frac{1}{Z_{\max}} \right) \left(\frac{1}{4}\left(y_{0_\epsilon}^2 + x_{0_\epsilon}^2\right) e^2 + \left(x_{0_\epsilon} y_0 - y_{0_\epsilon} x_0\right)^2 \right) \Bigg) \end{aligned} \tag{7}$$

(a) Assume that the translation has been estimated with a certain error $\mathbf{t}_\epsilon = (x_{0_\epsilon}, y_{0_\epsilon}, 0)$. Then the relationship among the errors in 3D motion at the minima of (7) is obtained from the first-order conditions $\frac{\partial E}{\partial \alpha_\epsilon} = \frac{\partial E}{\partial \beta_\epsilon} = \frac{\partial E}{\partial \gamma_\epsilon} = 0$, which yield

$$\alpha_\epsilon = \frac{y_{0_\epsilon}\left(\ln(Z_{\max}) - \ln(Z_{\min})\right)}{Z_{\max} - Z_{\min}} \qquad \beta_\epsilon = \frac{-x_{0_\epsilon}\left(\ln(Z_{\max}) - \ln(Z_{\min})\right)}{Z_{\max} - Z_{\min}} \qquad \gamma_\epsilon = 0 \tag{8}$$

It follows that $\alpha_\epsilon/\beta_\epsilon = -x_{0_\epsilon}/y_{0_\epsilon}, \gamma_\epsilon = 0$. The first of these constraints is called the orthogonality constraint $((\alpha_\epsilon, \beta_\epsilon) \perp (x_{0_\epsilon}, y_{0_\epsilon}))$.

(b) Assuming that rotation has been estimated with an error $(\alpha_\epsilon, \beta_\epsilon, \gamma_\epsilon)$, the relationship among the errors is obtained from $\frac{\partial E}{\partial x_{0_\epsilon}} = \frac{\partial E}{\partial y_{0_\epsilon}} = 0$. In this case, the relationship is very elaborate and the translational error depends on all the other parameters—that is, the rotational error, the actual translation, the image size and the depth interval.

(c) In the general case, we need to study the subspaces in which E changes least at its absolute minimum; that is, we are interested in the direction of the smallest second derivative at 0, the point where the motion errors are zero. To find this direction, we compute the Hessian at 0, that is the matrix of the second derivatives of E with respect to the five motion error parameters, and compute the eigenvector corresponding to the smallest eigenvalue. The scaled components of this vector amount to

$$x_{0_\epsilon} = x_0 \quad y_{0_\epsilon} = y_0 \quad \beta_\epsilon = -\alpha_\epsilon \tfrac{x_0}{y_0} \quad \gamma_\epsilon = 0$$
$$\alpha_\epsilon = 2y_0 Z_{\min} Z_{\max} \left(\ln\left(Z_{\max}\right) - \ln\left(Z_{\min}\right)\right) \Big/ \Big(\left(Z_{\max} - Z_{\min}\right)\left(Z_{\max} Z_{\min} - 1\right) + \Big(\left(Z_{\max} - Z_{\min}\right)^2 \left(Z_{\max} Z_{\min} - 1\right)^2 + 4 Z_{\max}^2 Z_{\min}^2 (\ln\left(Z_{\max}\right) - \ln\left(Z_{\min}\right))^2 \Big)^{1/2} \Big)$$

As can be seen, for points defined by this direction, the translational and rotational errors are characterized by the "orthogonality constraint" $\alpha_\epsilon/\beta_\epsilon = -x_{0_\epsilon}/y_{0_\epsilon}$ and by the constraint $x_0/y_0 = \hat{x}_0/\hat{y}_0$, which is called the "line constraint." It basically means that (x_0, y_0)—the direction of the real translation, and $(\hat{x}_0, \hat{y}_0)$—the direction of the estimated translation, lie on a line passing from the origin.

The result states that the solution contains errors satisfying the orthogonality constraint and the line constraint and thus are mingled and create a confusion between rotation and translation that cannot be cleared up. The errors may be small or large, but their expected value will always satisfy the above conditions. Although the 3D-motion estimation approaches described above may provide answers that could be sufficient for various navigation tasks, they cannot be used for deriving object models. This result demonstrates that recovering 3D motion from a video stream is an ill-posed problem.

4 New Eyes for Virtual Reality

Why is it that biological systems that need to fly and thus require good estimates of 3D motion (insects, birds) have panoramic vision implemented either as a compound eye or by placing camera-type eyes on opposite sides of the head? This is a fascinating question that has remained open since the time of the pioneer investigator, Sigmund Exner, at the beginning of this century. The obvious answer is, of course, that flying systems should perceive the whole space around them—thus panoramic vision emerged. There is, however, a deeper mathematical reason and it has to with the ability of a system to estimate 3D motion when it analyzes panoramic images, as shown in this section. Put simply, a spherical eye (360 degree field of view) is superior to a planar eye (restricted field) with regard to 3D motion estimation. Recall from Section 3.2 that, given a sequence of images, 3D motion is estimated by minimizing function E that represents deviation from the epipolar constraint. It was shown that in the case of images captured by a planar eye (e.g., a common video camera), this function has a special topography which is such that the errors in the motion are mingled, causing confusion between rotation and translation and thus producing a wrong result. If, however, the field of view goes to 360 degrees, the topography of the surface drastically changes with the minimum clearly standing out in most cases. It is no wonder then that flying organisms possess panoramic vision!

The analysis that leads to this result is almost identical to the analysis performed for planar eyes. Panoramic vision is modeled by projecting onto a sphere, with the sphere's center as the center of projection (Fig. 1b). In this case, the image $\mathbf{r}$ of any point $\mathbf{R}$ is $\mathbf{r} = \frac{\mathbf{R}f}{|\mathbf{R}|}$, with R being the norm of $\mathbf{R}$ (the range), and the image motion is

$$\dot{\mathbf{r}} = \frac{1}{|\mathbf{R}|f}\left((\mathbf{t}\cdot\mathbf{r})\,\mathbf{r} - \mathbf{t}\right) - \boldsymbol{\omega}\times\mathbf{r} = \frac{1}{R}\mathbf{u}_{\mathrm{tr}}(\mathbf{t}) + \mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega}). \tag{9}$$

The function M_{ep} representing deviation from the epipolar constraint on the sphere has the exact same form as in the plane for our nomenclature. We integrate over the range R within an interval bounded by $R_{\min}$ and $R_{\max}$ and obtain

$$E_{ep} = \int\limits_{R_{\min}}^{R_{\max}} \iint\limits_{\text{sphere}} \left\{\left(\frac{\mathbf{r}\times(\mathbf{r}\times\mathbf{t})}{R} - (\boldsymbol{\omega}_\epsilon\times\mathbf{r})\right)\cdot\left(\hat{\mathbf{t}}\times\mathbf{r}\right)\right\}^2 dA\,dR$$

where A refers to a surface element. Due to the sphere's symmetry, for each point $\mathbf{r}$ on the sphere, there exists a point with coordinates $-\mathbf{r}$. Since $\mathbf{u}_{\mathrm{tr}}(\mathbf{r}) = \mathbf{u}_{\mathrm{tr}}(-\mathbf{r})$ and $\mathbf{u}_{\mathrm{rot}}(\mathbf{r}) = -\mathbf{u}_{\mathrm{rot}}(-\mathbf{r})$, when the integrand is expanded the product terms integrated over the sphere vanish. Thus

$$E_{ep} = \int\limits_{R_{\min}}^{R_{\max}} \iint\limits_{\text{sphere}} \left\{\frac{\left(\left(\mathbf{t}\times\hat{\mathbf{t}}\right)\cdot\mathbf{r}\right)^2}{R^2} + \left((\boldsymbol{\omega}_\epsilon\times\mathbf{r})\cdot\left(\hat{\mathbf{t}}\times\mathbf{r}\right)\right)^2\right\} dA\,dR$$

(a) Assuming that translation $\hat{\mathbf{t}}$ has been estimated, the $\boldsymbol{\omega}_\epsilon$ that minimizes E_{ep} is $\boldsymbol{\omega}_\epsilon = 0$, since the resulting function is non-negative quadratic in $\boldsymbol{\omega}_\epsilon$ (minimum at zero). The difference between sphere and plane is already clear. In the spherical case, as shown here, if an error in the translation is made we do not need to compensate for it by making an error in the rotation ($\boldsymbol{\omega}_\epsilon = 0$), while in the planar case we need to compensate to ensure that the orthogonality constraint is satisfied!

(b) Assuming that rotation has been estimated with an error $\boldsymbol{\omega}_\epsilon$, what is the translation $\hat{\mathbf{t}}$ that minimizes E_{ep}? Since R is assumed to be uniformly distributed, integrating over R does not alter the form of the error in the optimization. Thus, E_{ep} consists of the sum of two terms:

$$K = K_1 \iint\limits_{\text{sphere}} \left(\left(\mathbf{t} \times \hat{\mathbf{t}}\right) \cdot \mathbf{r}\right)^2 dA \text{ and } L = L_1 \iint\limits_{\text{sphere}} \left((\boldsymbol{\omega}_\epsilon \times \mathbf{r}) \cdot \left(\hat{\mathbf{t}} \times \mathbf{r}\right)\right)^2 dA,$$

where K_1, L_1 are multiplicative factors depending only on $R_{\min}$ and $R_{\max}$. For angles between $\mathbf{t}, \hat{\mathbf{t}}$ and $\hat{\mathbf{t}}, \boldsymbol{\omega}_\epsilon$ in the range of 0 to $\pi/2$, K and L are monotonic functions. K attains its minimum when $\mathbf{t} = \hat{\mathbf{t}}$ and L when $\hat{\mathbf{t}} \perp \boldsymbol{\omega}_\epsilon$. Fix the distance between $\mathbf{t}$ and $\hat{\mathbf{t}}$ leading to a certain value K, and change the position of $\hat{\mathbf{t}}$. L takes its minimum when $(\mathbf{t} \times \hat{\mathbf{t}}) \cdot \boldsymbol{\omega}_\epsilon = 0$, as follows from the cosine theorem. Thus E_{ep} achieves its minimum when $\hat{\mathbf{t}}$ lies on the great circle passing through $\mathbf{t}$ and $\boldsymbol{\omega}_\epsilon$, with the exact position depending on $|\boldsymbol{\omega}_\epsilon|$ and the scene in view.

(c) For the general case where no information about rotation or translation is available, we study the subspaces where E_{ep} changes the least at its absolute minimum, i.e., we are again interested in the direction of the smallest second derivative at 0. For points defined by this direction we calculate, using Maple, $\mathbf{t} = \hat{\mathbf{t}}$ and $\boldsymbol{\omega}_\epsilon \perp \mathbf{t}$.

4.1 What if correspondence is not available?

The preceding sections investigated the differences between camera-type eyes (restricted field of view) and spherical eyes (full field of view) with regard to 3D motion estimation, when an estimate of correspondence or flow was available. One may wonder how this comparative analysis becomes when correspondence is not available, but all we have at our disposal is the normal flow. This case is harder to analyze and we provide here results from proofs that appeared recently [11,12].

If normal flow is given, the vector equations (3) and (9) cannot be used directly. The only constraint is scalar equation (4), along with the inequality $Z > 0$ which states that since the surface in view is in front of the eye its depth must be positive. Substituting (3) or (9) into (4) and solving for the estimated depth $\hat{Z}$ or range $\hat{R}$, we obtain for a given estimate $\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}$ at each point $\mathbf{r}$:

$$\hat{Z}(\text{or } \hat{R}) = \frac{\mathbf{u}_{\text{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}}{(\dot{\mathbf{r}} - \mathbf{u}_{\text{rot}}(\hat{\boldsymbol{\omega}})) \cdot \mathbf{n}}. \tag{10}$$

If the numerator and denominator of (10) have opposite signs, negative depth is computed. Thus, to utilize the positivity constraint one must search for the motion $\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}$ that produces a minimum number of negative depth estimates. Formally, if $\mathbf{r}$ is an image point, define the indicator function

$$I_{nd}(\mathbf{r}) = \begin{cases} 1 \text{ for } \left(\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}\right)(\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}})) < 0 \\ 0 \text{ for } \left(\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}\right)(\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}})) > 0 \end{cases}.$$

Then estimation of 3D motion from normal flow amounts to minimizing [9,10,16] the function

$$M_{nd} = \iint\limits_{\text{image}} I_{nd}(\mathbf{r}) d\mathbf{r}. \tag{11}$$

As before, we assume a probabilistic model for the scene with the depth values uniformly distributed between two arbitrary values $Z_{\min}$ (or $R_{\min}$) and $Z_{\max}$ (or $R_{\max}$). We further assume that the directions in which flow measurements are made are uniformly distributed in every direction for every depth. Parameterizing $\mathbf{n}$ by ψ, the angle between $\mathbf{n}$ and the x axis, we thus obtain the following function:

$$E_{nd} = \int\limits_{\psi=0}^{\pi} \int\limits_{Z=Z_{\min}}^{Z_{\max}} M_{nd} dZ \, d\psi, \tag{12}$$

measuring the amount of negative depth.

To study the negative depth values described by function (12) a more geometric interpretation is needed. Substituting into (10) the value of $\dot{\mathbf{r}}$ from (3) or (9) gives

$$\hat{Z}(\text{or } \hat{R}) = \frac{\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}}{\left(\frac{\mathbf{u}_{\mathrm{tr}}(\mathbf{t})}{Z(\text{or } R)} - \mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega}_\epsilon)\right) \cdot \mathbf{n}}$$

This equation shows that for every $\mathbf{n}$ and $\mathbf{r}$ a range of values for Z (or R) is obtained which result in negative estimates of $\hat{Z}$ (or $\hat{R}$). Thus for each direction $\mathbf{n}$, considering all image points $\mathbf{r}$, we obtain a volume in space corresponding to negative depth estimates. The sum of all these volumes for all directions is termed the "negative depth" volume, and calculating 3D motion in this case amounts to minimizing this volume. Minimization of this volume provides conditions for the errors in the motion parameters.

Applying this analysis to the plane and the sphere provides the following results.

Plane:

(a) Assume that rotation has been estimated with an error $(\alpha_\epsilon, \beta_\epsilon, \gamma_\epsilon)$. Then the error $(x_{0_\epsilon}, y_{0_\epsilon})$ that minimizes the negative depth volume satisfies the orthogonality constraint $x_{0_\epsilon}/y_{0_\epsilon} = -\beta_\epsilon/\alpha_\epsilon$.

(b) In the absence of any prior information about the 3D motion, the solution obtained by minimizing the negative depth volume has errors that satisfy the orthogonality constraint $x_{0_\epsilon}/y_{0_\epsilon} = -\beta_\epsilon/\alpha_\epsilon$, the line constraint $x_0/y_0 = \hat{x}_0/\hat{y}_0$ and $\gamma_\epsilon = 0$

Sphere:

(a) Assume that rotation has been estimated with an error $\boldsymbol{\omega}_\epsilon$. Then the translational error $\mathbf{t}_\epsilon$ minimizing the negative depth volume is zero.

(b) In the absence of any prior information about the 3D motion, the solution obtained by minimizing the negative depth volume has errors $\mathbf{t}_\epsilon, \boldsymbol{\omega}_\epsilon$ satisfying $\mathbf{t}_\epsilon = 0$ and $\boldsymbol{\omega}_\epsilon \perp \mathbf{t}$.

The following table summarizes the results for both cases.

4.2 Eyes from eyes

The preceding results demonstrate the advantages of spherical eyes for the process of 3D motion estimation. Table 1 lists the eight out of ten cases which lead to clearly defined error configurations. It shows that 3D motion can be estimated more accurately with spherical eyes. Depending on the estimation procedure used—and systems might use different procedures for different tasks—either the translation or the rotation can be estimated very accurately. For planar eyes, this is not the case, as for all possible procedures there exists confusion between the translation and rotation. The error configurations also allow systems with inertial sensors to use more efficient estimation procedures. If a system utilizes a gyrosensor which provides an approximate estimate of its rotation, it can employ a simple algorithm based on the negative depth constraint for only translational motion fields to derive its translation and obtain a very accurate estimate. Such algorithms are much easier to implement than algorithms designed for completely unknown rigid motions, as they amount to searches in 2D as opposed to 5D spaces [9]. Similarly, there exist computational advantages for systems with translational inertial sensors in estimating the remaining unknown rotation.

Since it turns out that spherical eyes such as the ones of insects, or, in general, panoramic vision provides much better capability for 3D motion estimation, and since our problem of building accurate space and action descriptions depends on accurate 3D motion computation, it makes sense to reconsider what the eye for our problem should be. There are a few ways to create panoramic vision cameras, and the recent literature is rich in alternative approaches, but

Table 1: Summary of results

	Spherical Eye	Camera-type Eye
Epipolar minimization, given optic flow	(a) Given a translational error $\mathbf{t}_\epsilon$, the rotational error $\boldsymbol{\omega}_\epsilon = 0$ (b) Without any prior information, $\mathbf{t}_\epsilon = 0$ and $\boldsymbol{\omega}_\epsilon \perp \mathbf{t}$	(a) For a fixed translational error $(x_{0_\epsilon}, y_{0_\epsilon})$, the rotational error $(\alpha_\epsilon, \beta_\epsilon, \gamma_\epsilon)$ is of the form $\gamma_\epsilon = 0$, $\alpha_\epsilon/\beta_\epsilon = -x_{0_\epsilon}/y_{0_\epsilon}$ (b) Without any a priori information about the motion, the errors satisfy $\gamma_\epsilon = 0$, $\alpha_\epsilon/\beta_\epsilon = -x_{0_\epsilon}/y_{0_\epsilon}$, $x_0/y_0 = x_{0_\epsilon}/y_{0_\epsilon}$
Minimization of negative depth volume, given normal flow	(a) Given a rotational error $\boldsymbol{\omega}_\epsilon$, the translational error $\mathbf{t}_\epsilon = 0$ (b) Without any prior information, $\mathbf{t}_\epsilon = 0$ and $\boldsymbol{\omega}_\epsilon \perp \mathbf{t}$	(a) Given a rotational error, the translational error is of the form $-x_{0_\epsilon}/y_{0_\epsilon} = \alpha_\epsilon/\beta_\epsilon$ (b) Without any error information, the errors satisfy $\gamma_\epsilon = 0$, $\alpha_\epsilon/\beta_\epsilon = -x_{0_\epsilon}/y_{0_\epsilon}$, $x_0/y_0 = x_{0_\epsilon}/y_{0_\epsilon}$

there is a way to take advantage of both the panoramic vision of flying systems and the high resolution vision of primates. An eye like the one in Fig. 3, assembled from a few video cameras arranged on the surface of a sphere, [2] can easily estimate 3D motion since, while it is moving, it is sampling a spherical motion field!

An eye like the one in Fig. 3 not only has panoramic properties, eliminating the rotation/translation confusion, but it has the unexpected benefit of making it easy to estimate image motion with high accuracy. Any two cameras with overlapping fields of view also provide high-resolution stereo vision, and this collection of stereo systems makes it possible to locate a large number of depth discontinuities. It is well known that, given scene discontinuities, image motion can be estimated very accurately. As a consequence, the eye in Fig. 3 is very

[2] Like a compound eye with video cameras replacing ommatidia

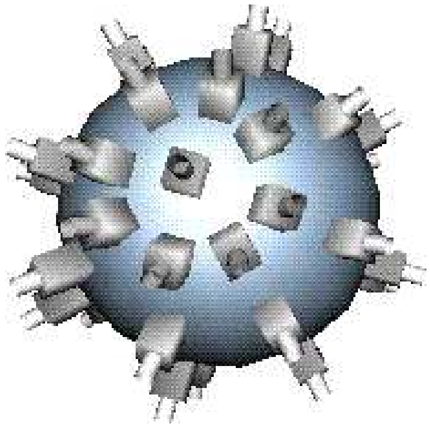


Fig. 3. A compound-like eye composed of conventional video cameras.

well suited to developing accurate models of the world.

5 Algorithms

There is a very large number of ways in which one can utilize multiple videos like the ones captured by the cameras of the sensor in Fig. 3 for recovering 3D structure and motion. The obvious ones include: (a) treat the flow fields close to the center of each camera as parts of a spherical motion field and apply algorithms such as those in [11]; (b) perform epipolar minimization in each video while enforcing the constraints relating the motions of different cameras comprising the sensor. The results of Table 1 can serve as a guide for choosing particular algorithmic procedures, e.g., should rotation or translation be estimated first, or should all parameters be estimated simultaneously, depending on whether epipolar or negative depth minimization is used, depending on whether inertial sensors are available, etc.

Although good flow values can be obtained since many discontinuities are provided by the multitude of stereo systems, the image texture may provide bias in the flow, as discussed in Section 3.1. Thus, it is desirable to avoid flow values early on in the computation.

We describe here an approach which is not based on correspondence or optic flow [4,6,9,13], and for tractability reasons we develop it for a single video. Extension to multiple videos is straightforward.

Consider an estimate of the 3D rigid motion $\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}$. At an image point $\mathbf{r}$ where the normal flow direction is $\mathbf{n}$, the depth $\hat{Z}$ can be estimated from (3) as:

$$\frac{1}{\hat{Z}} = \frac{\dot{\mathbf{r}} \cdot \mathbf{n} - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) \cdot \mathbf{n}}{\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}} \tag{13}$$

where $\mathbf{u}_{\rm rot}(\hat{\boldsymbol{\omega}}), \mathbf{u}_{\rm tr}(\hat{\mathbf{t}})$ are the estimated rotational and the direction of the translational flow, respectively. Substituting the value of $\dot{\mathbf{r}}$ from (3) into (13), one obtains:

$$\frac{1}{\hat{Z}} = \frac{1}{Z}\frac{\mathbf{u}_{\rm tr}(\mathbf{t})\cdot\mathbf{n} - Z\mathbf{u}_{\rm rot}(\delta\boldsymbol{\omega})\cdot\mathbf{n}}{\mathbf{u}_{\rm tr}(\hat{\mathbf{t}})\cdot\mathbf{n}}$$

where $\mathbf{u}_{\rm rot}(\delta\boldsymbol{\omega})$ is the rotational flow due to the rotational error $\delta\boldsymbol{\omega} = (\hat{\boldsymbol{\omega}} - \boldsymbol{\omega})$. The above equation can be further expressed as:

$$\hat{Z} = Z \cdot D \quad \text{with} \tag{14}$$

$$D = \frac{\mathbf{u}_{\rm tr}(\hat{\mathbf{t}})\cdot\mathbf{n}}{(\mathbf{u}_{\rm tr}(\mathbf{t}) - Z\mathbf{u}_{\rm rot}(\delta\boldsymbol{\omega}))\cdot\mathbf{n}} \tag{15}$$

where D hereafter is termed the distortion factor. Eq. (14) shows how wrong depth estimates are produced due to inaccurate 3D motion values. The distortion factor for any direction $\mathbf{n}$ corresponds to the ratio of the projections of the two vectors $\mathbf{u}_{\rm tr}(\hat{\mathbf{t}})$ and $\mathbf{u}_{\rm tr}(\mathbf{t}) - Z\mathbf{u}_{\rm rot}(\delta\boldsymbol{\omega})$ on $\mathbf{n}$. The larger the angle between these two vectors is, the more the distortion will be spread out over the different directions. Thus, considering a patch of a smooth surface in space and assuming that normal flow measurements are taken along many directions, a rugged (i.e., unsmooth) surface will be computed on the basis of wrong 3D motion estimates.

This observation constitutes the main idea behind the approach. It amounts to searching for the 3D motion that will minimize some measure of depth variability: Divide the image into small patches. Consider further a search procedure that searches for the 3D motion which when used with the normal flow data of any particular patch provides depth values in the patch that vary the least. The 3D motion that minimizes depth variation of all the image patches constitutes the solution. This is easy to understand if the scene in view is smooth (containing no depth discontinuities) and static (no independent motion). If the scene in view contains depth discontinuities and is static, then there will be patches, namely the ones containing discontinuities, for which the correct 3D motion produces large variability of depth; but these patches can still be subdivided into two smaller patches, one on each side of the discontinuity, for which small variability of depth can be achieved.

The reader may wonder about the relationship of the introduced algorithm to the epipolar constraint and the positive depth constraint used in the previous analysis. This is explained in the sequel. In this algorithm we utilize normal flow, but we do not minimize negative depth directly. Instead, we minimize a function related to the distortion of depth, but this function is closely related statistically to negative depth. To make this clear, in our algorithm we estimate the variation of depth (or functions of the depth) within single image patches, and we estimate the 3D motion by minimizing the variation of depth within all image patches. The estimated depth (or range) at a point as given by (15) corresponds to the ratio of the projections of the two vectors $\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})$ and $\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}})$ on $\mathbf{n}$. The larger the angle between $\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})$ and $\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}})$ the more normal flow directions will give rise to negative depth. But also, the larger the angle, the more the variation of $\hat{R}$ will be spread out over the different directions $\mathbf{n}$. Thus at an image patch corresponding to a smooth scene patch which has flow measurements along many directions the amount of depth variation is directly related to the amount of negative depth values. By this argument, a measure of negative depth also represents a measure of the depth variability.

This new approach, minimizing depth variation within patches also has a close relationship to epipolar minimization, but in general is more powerful. This is explained here. Consider a small patch P that contains a set of measurements $\dot{\mathbf{r}}_i$ in directions $\mathbf{n}_i$ (normal flows). Given candidate motion parameters $\hat{\mathbf{t}}$ and $\hat{\boldsymbol{\omega}}$ we can estimate depth up to an overall scale ambiguity. One possible measure of depth variation is the variance S_0 of the depth values, or rather the sum of squared differences of the depth values from a mean $1/\bar{Z}$. For the purpose of relating S_0 to the epipolar constraint, a similar way of writing measure S_0 without explicitly computing depth is:

$$S_0(\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}, P) = \sum_i W_i(\dot{\mathbf{r}}_i \cdot \mathbf{n}_i - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) \cdot \mathbf{n}_i - \frac{1}{\hat{Z}} \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i) \tag{16}$$

where $1/\hat{Z}$ is the depth estimate minimizing the measure, not necessarily the mean $1/\bar{Z}$. If we set $W_i = 1/(\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)^2$, (16) expresses the variance of depth values.

Assuming that the depth is constant in the small patch (fronto-parallel plane) [3], the best inverse depth $1/\hat{Z}$ minimizing S is

$$\frac{1}{\hat{Z}} = \frac{\sum_i W_i(\dot{\mathbf{r}}_i \cdot \mathbf{n}_i - \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) \cdot \mathbf{n}_i - \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)}{\sum_i W_i(\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) \cdot \mathbf{n}_i)^2} \tag{18}$$

[3] If more precision is required, one can model the scene patch by a general plane and use a linear approximation $1/\hat{Z} = \mathbf{z} \cdot \mathbf{r}$ (note that the third component of $\mathbf{r}$ is a constant f, so $\mathbf{z} \cdot \mathbf{r}$ is a general linear function in the image coordinates). Then we have

$$\partial S_0 / \partial \mathbf{z} = 0 \tag{17}$$

providing three linear equations in the elements of $\mathbf{z}$.

Substituting (18) (or the solution of (17)) into (16), we obtain $S(\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}, P)$, a function of $\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}$ whose minimum provides the desired 3D motion.

Consider the function S_0 in a small image region P. The vectors $\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})$ and $\mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}})$ are polynomial functions of image position $\mathbf{r}$ and can usually be approximated by constants within the region. We use a local coordinate system where $\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})$ is parallel to $[1,0,0]^{\mathrm{T}}$. Without loss of generality we can write (in that coordinate system)

$$\begin{aligned} \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}}) &= [1,0,0]^{\mathrm{T}}, \mathbf{u}_{\mathrm{rot}}(\hat{\boldsymbol{\omega}}) &&= [u_{\mathrm{rx}}, u_{\mathrm{ry}}, 0]^{\mathrm{T}} \\ \mathbf{n}_i &= [\cos\psi_i, \sin\psi_i, 0]^{\mathrm{T}}, u_{\mathrm{n}_i} &&= \dot{\mathbf{r}}_i \cdot \mathbf{n}_i \end{aligned} \tag{19}$$

Fitting the best constant optical flow $(u_{\mathrm{x}}, u_{\mathrm{y}})$ to the measurements in P amounts to minimizing

$$\sum_i W_i \left(u_{\mathrm{n}_i} - (u_{\mathrm{x}}, u_{\mathrm{y}}) \cdot \mathbf{n}_i\right)^2 \tag{20}$$

To simplify the notation, we define several sums

$$\begin{aligned} A &= \textstyle\sum W_i \cos^2\psi_i & D &= \textstyle\sum W_i\, u_{\mathrm{n}_i} \cos\psi_i \\ B &= \textstyle\sum W_i \cos\psi_i \sin\psi_i & E &= \textstyle\sum W_i\, u_{\mathrm{n}_i} \sin\psi_i \\ C &= \textstyle\sum W_i \sin^2\psi_i & F &= \textstyle\sum W_i\, u_{\mathrm{n}_i}^2 \end{aligned} \tag{21}$$

The vector $(u_{\mathrm{x}}, u_{\mathrm{y}})$ minimizing (20) is obtained by differentiating (20) and solving a linear system to obtain

$$\begin{pmatrix} u_{\mathrm{x}} \\ u_{\mathrm{y}} \end{pmatrix} = \frac{1}{AC - B^2} \begin{pmatrix} C & -B \\ -B & A \end{pmatrix} \begin{pmatrix} D \\ E \end{pmatrix} \tag{22}$$

and the minimum error is

$$E_{\mathrm{F}} = F - \frac{1}{AC - B^2}(E^2 A + D^2 C - 2DEB) \tag{23}$$

Using the notation (19) we have

$$S_0 = \sum_i W_i \left(u_{\mathrm{n}_i} - u_{\mathrm{rx}} \cos\psi_i - u_{\mathrm{ry}} \sin\psi_i - \frac{1}{\hat{Z}} \cos\psi_i\right)^2 \tag{24}$$

It can be verified that u_{rx} only shifts the best $1/\hat{Z}$, but it does not influence the final measure. Thus we can set u_{rx} to zero without loss of generality and expand S_0 to $S_0 = F + u_{\mathrm{ry}}^2 C + (1/\hat{Z})^2 A - 2u_{\mathrm{ry}} E - 2(1/\hat{Z})D + 2(1/\hat{Z})u_{\mathrm{ry}} B$. Let us denote $u_{\mathrm{ry}} = u_{\mathrm{y}} + \delta u_{\mathrm{ry}}$. Minimization of S_0 yields $\frac{1}{\hat{Z}} = \frac{D - u_{\mathrm{ry}} B}{A}$. Our measure S of depth variability is obtained by substituting the above equation into S_0. Using (23) it can be written as

$$S = \frac{AC - B^2}{A}\ \delta u_{\mathrm{ry}}^2 + E_{\mathrm{F}} \tag{25}$$

The epipolar constraint can be written as $(\hat{\mathbf{z}} \times \mathbf{u}_{\mathrm{tr}}(\mathbf{t})) \cdot (\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega})) = 0$. In practice, epipolar minimization amounts to optimizing the sum of $E = \frac{[(\hat{\mathbf{z}} \times \mathbf{u}_{\mathrm{tr}}(\mathbf{t})) \cdot (\dot{\mathbf{r}} - \mathbf{u}_{\mathrm{rot}}(\boldsymbol{\omega}))]^2}{||\mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})||^2}$, with the normalization term included for reducing bias. Assuming that $\dot{\mathbf{r}}$ has been obtained from minimization of (20), we can write it as (u_x, u_y) and substituting into E we obtain $E = (u_{\mathrm{y}} - u_{\mathrm{ry}})^2 = \delta u_{\mathrm{ry}}{}^2$. In other words, $\delta u_{\mathrm{ry}}{}^2$ represents the distance of the estimated flow from the epipolar line, i.e., the quantity optimized in epipolar minimization.

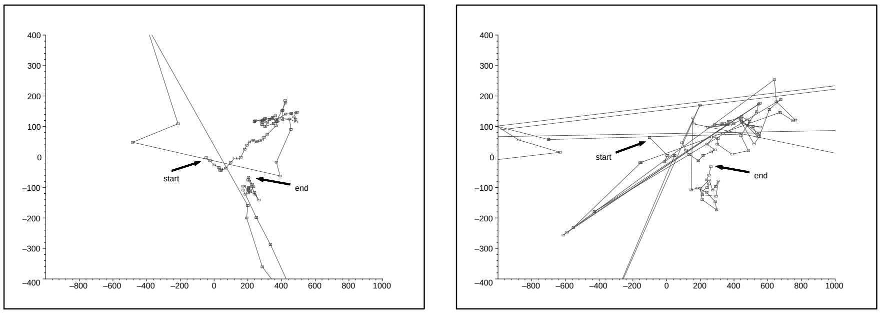
The first component of our proposed measure in (25) is related to the epipolar constraint and it depends on the 3D motion estimate, as well as the gradient distribution in the patch. The second component in (25), E_{F}, represents how well the scene patch is approximated by a plane and it is independent of the 3D motion estimate. In classic approaches, after optic flow is computed, the term E_{F} is not considered any further and the estimation of 3D motion parameters is based only on the distance from the epipolar line. Here we keep this term to utilize it for segmentation.

Several experiments demonstrate both the promise of the new constraint and the superiority of multi-view structure from motion. Movie 1 [19] shows a sequence captured by a hand-held camera in the lab. [4] For one frame of the sequence, see Fig. 4. Movie 2 [20] shows the recovered epipole from our technique (green dot) and epipolar minimization, and movie 3 [21] shows the recovered instantaneous depth using our technique. For an evolution of the recovered epipole positions, see Fig. 5. Fig. 6 shows recovered depth maps for different parts of the sequence. Fig. 7 shows two views of the recovered scene from fifteen image frames. Epipolar minimization results in poor reconstruction, even providing negative depth in about 25% of the frames. Movies 4 [22], 5 [23], 6 [24] and 7 [25] show original sequences and the corresponding recovered shapes (with texture mapping) demonstrating the power of the depth variability technique. Movie 8 [26] shows the well-known Yosemite sequence, movie 9 [27] the reconstructed shape in the form of a mesh and movie 10 [28] with painted texture. Reconstruction from multiple videos gives very good

[4] A hand-held camera guarantees that the motion changes at every time instant, making the problem the hardest possible.

results. Movie 11 [29] shows an original sequence depicting an object (Pooh game). Movie 12 [30] shows a fly-through of the model constructed by finding 3D motion from that video and movie 13 [31] shows the model constructed by finding 3D motion from multiple videos. Clearly, the results are greatly improved in this case. Of course, the models constructed amount to parts of the scene visible in the video stream. To create a volumetric, object-centered model, one would need video imaging over the whole object. The problem of linking the camera-centered representations into an object-centered representation is one of our future research goals.

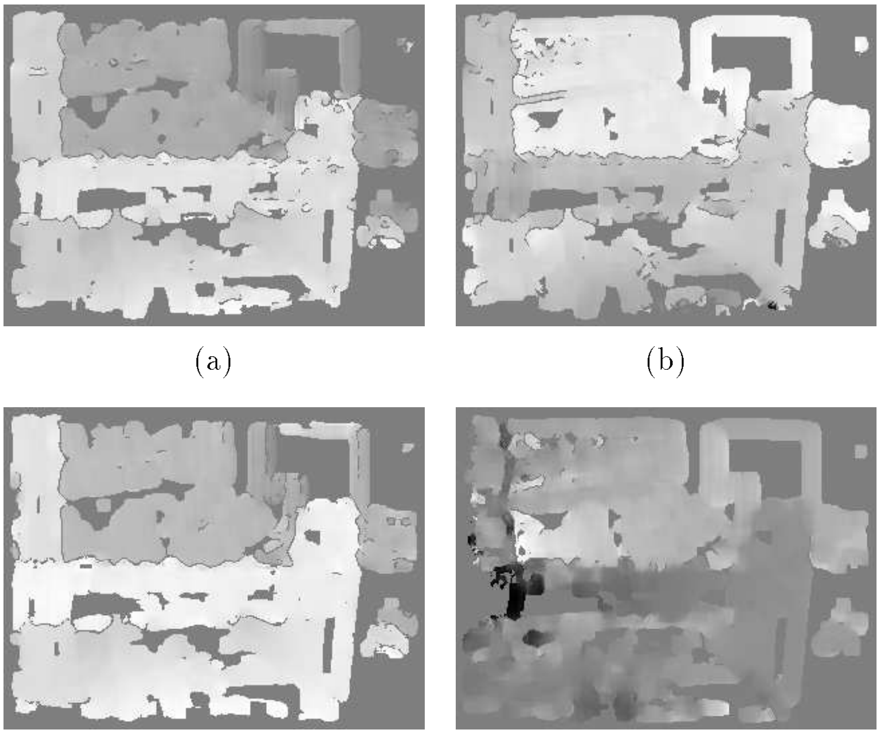
Fig. 4.



(a) (b)

Fig. 5. The evolution of recovered epipole positions over 90 frames. (a) The proposed algorithm. (b) Epipolar minimization. Both graphs show an identical part of the image plane (x between -1000 and 1000, y between -400 and 400). The image size was 320×240 pixels.

To summarize, a full visual field provides 3D motion very accurately, and thus very good models of the world. Existing sensors for capturing panoramic images (such as [18]) are not adequate for this problem due to low resolution. One would need a high-resolution spherical field of view. As this is currently technologically impossible, we resort to sampling the whole visual field with high resolution. See, for example, the sensor in Fig. 8, built in our laboratory, consisting of four cameras looking in different directions. If all cameras shared a common nodal point, then the cameras would sample parts of a sphere. When this is not true, a calibration is required. Knowledge of the rigid transformations relating the difference camera coordinate systems, allows 3D motion



(a) (b) (c) (d)

Fig. 6. Comparison of recovered inverse depth for the lab sequence using the epipole positions as estimated with the proposed algorithm and the epipolar minimization. (a), (b) Frame 134. (a) Depth variation, epipole: $(377, -125)$, (b) Epipolar minimization, epipole: $(-612, 256)$. (c), (d) Frame 142. (a) Depth variation, epipole: $(483, -123)$, (b) Epipolar minimization, epipole: $(-153, 18)$.

Fig. 7. Two views of a 3D reconstruction of the recovered depth combined from fifteen image frames.

and structure estimation through the use of all videos. Issues of optimality regarding such sensor design remain open.

Fig. 8. The Argus eye resting on a cabinet (courtesy of R. Pless and P. Baker).

6 Conclusions and Future Research

Taking inspiration from nature, especially from the compound eyes of insects that look like the eye in Fig. 3 by replacing the video cameras with many poor quality cameras (ommatidia), we studied computational advantages of panoramic and multiple view vision. At the same time, we introduced a new constraint for structure from motion that is more powerful than epipolar minimization. (See also [3].) The past few years have seen some work in computer vision conducted in a multi-view environment (for example, work on image based rendering [33], multi-view structure from motion [32], multi-view human motion capture systems and multi-view virtualized reality [17]). As this promising recent work is driven by applications of all sorts, there isn't yet a clear understanding of how to put together multiple cameras to solve problems. Our ideas showed how we can make multi-view eyes with provable properties, and we introduced a new (compound-like) eye for developing shape descriptions.

The geometric results presented point to new ways of building cameras for a variety of applications. This alternative camera technology calls for building "eyes from eyes," that is, new sensors out of existing ones. One such very useful sensor results by modifying the sensor in Fig. 3 so that the cameras point inwards as opposed to outwards (Fig. 9). Imaging a moving rigid object at the center of the sphere creates image motion fields at the center of each camera which are the same as the ones that would be created if the whole spherical dome were moving with the opposite rigid motion! Thus, utilizing information from all the cameras, the 3D rigid motion of the object inside the sphere can be accurately estimated, and at the same time accurate shape models can be obtained from the motion field of each camera. The negative spherical eye or variants of it also allow for accurate recovery of models of action, such as human movement, because putting together motion and shape, sequences of 3D motion fields representing the motion inside the dome can be estimated. Such action models have many applications in telereality, graphics and recognition. [5]

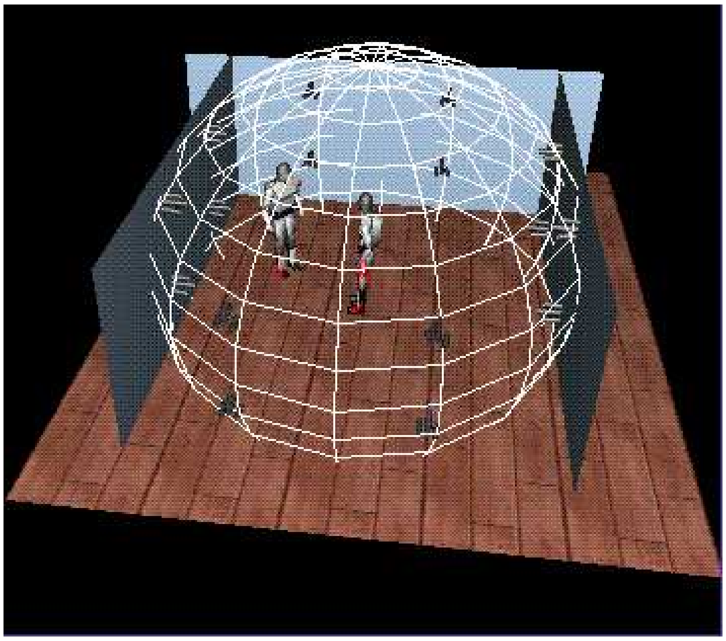


Fig. 9. A negative spherical eye.

The sphericity of the sensors in Figs. 3 and 9 is not absolutely essential. In actual fact, the cameras could be arranged on any surface as long as they can image the object of interest from a large collection of surrounding viewpoints. By placing the cameras in canonical positions and making sure that the relative orientation between neighboring cameras is exactly the same, we obtain

[5] Action descriptions cannot of course be recovered using the introduced theory on rigid motion since they basically amount to nonrigid motion. Current efforts along this line of work are in [34].

an additional constraint on the whole system. This constraint constitutes an invariant which could further enhance the performance of the overall system.

An implementation of these techniques using a multi-camera environment with many cameras raises fruitful prospects for very important technology, such as 3D video. The Institute of Advanced Computer Studies at the University of Maryland obtained a gift from the Keck Foundation to establish the Keck Laboratory for the study of visual movement. The Laboratory consists of a large number of cameras (currently sixty-four)[6] and a network of PCs[7] with the capability of simultaneous recording and synchronization among all sensors. We are currently implementing these ideas in the Keck Laboratory, by combining our results with volume carving techniques. For a detailed description, see [2].

The above described configurations are examples of alternative sensors, and they also demonstrate that multiple-view vision has great potential. Different arrangements best suited for other problems can be imagined. This was perhaps foreseen in ancient Greek mythology, which has Argus, the hundred-eyed guardian of Hera, the goddess of Olympus, defeating a whole army of Cyclopes, one-eyed giants!

Acknowledgements

Special thanks to Sara Larson for her editorial and graphics assistance.

References

[1] J. Aloimonos and D. Shulman. *Integration of Visual Modules: An Extension of the Marr Paradigm.* Academic Press, Boston, 1993.

[2] http://www.cfar.umd.edu/users/yiannis/research.html.

[3] T. Brodský, C. Fermüller, and Y. Aloimonos. Structure from motion: Beyond the epipolar constraint. Technical Report CAR-TR-911, Center for Automation Research, University of Maryland, 1998.

[4] T. Brodský, C. Fermüller, and Y. Aloimonos. Shape from video. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 146–151, June 1999.

[6] The cameras are Kodak ES-310 and can provide images at a rate of eighty-five frames per second.

[7] There are sixteen dual processor Pentium 450s connected by a high-speed network (Myrinet). Each PC has 1 Gbyte of memory and capture boards.

[5] A. Bruss and B. K. P. Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21:3–20, 1983.

[6] L. Cheong, C. Fermüller, and Y. Aloimonos. Effects of errors in the viewing geometry on shape estimation. *Computer Vision and Image Understanding*, 71:356–372, 1998.

[7] K. Daniilidis. *On the Error Sensitivity in the Recovery of Object Descriptions*. PhD thesis, Department of Informatics, University of Karlsruhe, Germany, 1992. In German.

[8] K. Daniilidis and M. E. Spetsakis. Understanding noise sensitivity in structure from motion. In Y. Aloimonos, editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, Advances in Computer Vision, chapter 4. Lawrence Erlbaum Associates, Mahwah, NJ, 1997.

[9] C. Fermüller and Y. Aloimonos. Direct perception of three-dimensional motion from patterns of visual motion. *Science*, 270:1973–1976, 1995.

[10] C. Fermüller and Y. Aloimonos. Qualitative egomotion. *International Journal of Computer Vision*, 15:7–29, 1995.

[11] C. Fermüller and Y. Aloimonos. Ambiguity in structure from motion: Sphere versus plane. *International Journal of Computer Vision*, 28:137–154, 1998.

[12] C. Fermüller and Y. Aloimonos. Geometry of eye design: Biology and technology. Technical Report CAR-TR-901, Center for Automation Research, University of Maryland, 1998.

[13] C. Fermüller, L. Cheong, and Y. Aloimonos. Visual space distortion. *Biological Cybernetics*, 77:323–337, 1997.

[14] C. Fermüller, R. Pless, and Y. Aloimonos. The Ouchi illusion as an artifact of biased flow estimation. *Vision Research*, 1999. In press.

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

[16] B. K. P. Horn and E. J. Weldon, Jr. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76, 1988.

[17] P. Narayanan, P. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proc. International Conference on Computer Vision*, pages 3–10, Bombay, January 1998.

[18] S. Nayar. Catadioptric omnidirectional camera. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, Puerto Rico, 1997.

[19] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-1.mpg.

[20] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-2.mpg.

[21] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-3.mpg.

[22] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-4.mpg.

[23] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-5.mpg.

[24] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-6.mpg.

[25] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-7.mpg.

[26] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-8.mpg.

[27] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-9.mpg.

[28] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-10.mpg.

[29] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-11.mpg.

[30] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-12.mpg.

[31] http://www.cfar.umd.edu/users/larson/NewEyesMovies/NEM-13.mpg.

[32] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. International Conference on Computer Vision*, pages 90–95, 1998.

[33] S. M. Seitz and K. N. Kutulakos. Plenoptic image editing. In *Proc. International Conference on Computer Vision*, pages 17–24, 1998.

[34] B. V. Stuart. *3D Video*. PhD thesis, Department of Computer Science, University of Maryland, 2000. To appear.

[35] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach.* Kluwer Academic Publishers, 1996.