CAR-TR-973 IIS-00-8-1365
CS-TR-4318 December 2001

Eyes from Eyes: Analysis of Camera Design Using Plenoptic Video Geometry

Jan Neumann, Cornelia Fermüller and Yiannis Aloimonos

Center for Automation Research
University of Maryland
College Park, MD 20742-3275,USA
{jneumann, fer, yiannis}@cfar.umd.edu

Abstract

We investigate the relationship between camera design and the problem of recovering the motion and structure of a scene from video data. The visual information that could possibly be obtained is described by the plenoptic function. A camera can be viewed as a device that captures a subset of this function, that is, it measures some of the light rays in some part of the space. The information contained in the subset determines how difficult it is to solve subsequent interpretation processes. By examining the differential structure of the time varying plenoptic function we relate different known and new camera models to the spatio-temporal structure of the observed scene. This allows us to define a hierarchy of camera designs, where the order is determined by the stability and complexity of the computations necessary to estimate structure and motion. At the low end of this hierarchy is the standard planar pinhole camera for which the structure from motion problem is non-linear and ill-posed. At the high end is a camera, which we call the *full field of view polydioptric* camera, for which the problem is linear and stable. In between are multiple-view cameras with large fields of view which we have built, as well as catadioptric panoramic sensors and other omni-directional cameras. We develop design suggestions for the polydioptric camera, and based upon this new design we propose a linear algorithm for ego-motion estimation, which in essence combines differential motion estimation with differential stereo.

This research was funded by NSF under Grant # IIS-00-8-1365.

1 Introduction

When we think about vision, we usually think of interpreting the images taken by (two) eyes such as our own - that is, images acquired by planar eyes. These are the so-called camera-type eyes based on the pinhole principle on which commercially available cameras are based. One considers a point in space and the light rays passing through that point. Then the rays are cut with a plane, and a subset of them forms an image. But these are not the only types of eyes that exist; the biological world reveals a large variety of eye designs. Michael Land, a prominent British zoologist and the world's foremost expert on the science of eyes, has provided a landscape of eye evolution [13]. Considering evolution as a mountain, with the lower hills representing the earlier steps in the evolutionary ladder, and the highest peaks representing the later stages of evolution, the situation is pictured in Fig. 1. It has been estimated that eyes have evolved no fewer than forty times, independently, in diverse parts of the animal kingdom. In some cases, eyes use radically different principles; the "eye landscape" of Fig. 1 shows nine basic types. Low in the hierarchy are primitive eyes such as the nautilus' pinhole eye or the marine snail eye. Different types of compound eyes of insects, camera-like eyes of land vertebrates, and fish eyes are all highly evolved.

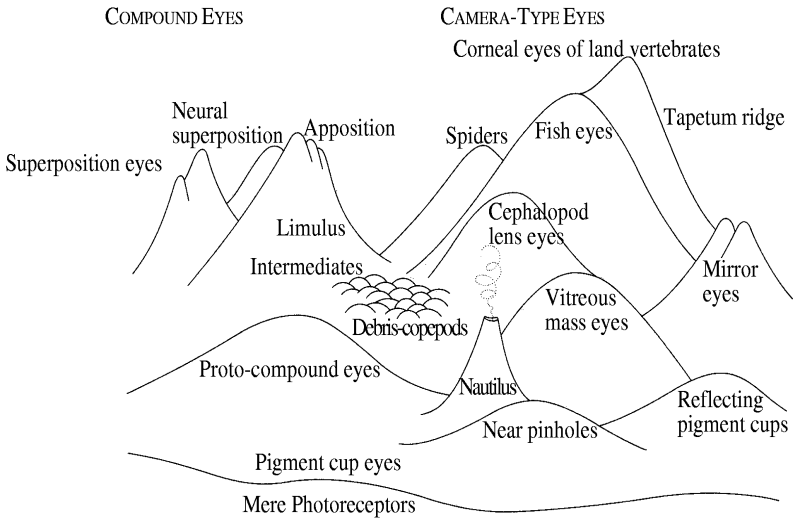


Figure 1: Michael Land's landscape of eye evolution.

An eye or camera is a mechanism that forms images by focusing light onto a light sensitive surface (retina, film, CCD array, etc.). Throughout this paper we will use the term "eye" both for cameras and biological eyes. Different eyes or cameras are obtained by varying three elements: (1) the geometry of the surface, (2) the geometric distribution and optical properties of the photoreceptors, and (3) the way light is collected and projected onto the surface (single or multiple lenses, or tubes as in compound eyes). Vision systems process these images to recognize, navigate, and generally interact with the environment. How advanced this interaction is depends both on the value of the information collected by the eyes and on how difficult it is to create intelligent behavior from such information. Evolutionary considerations tell us that the design of a system's eye is related to the visual tasks the system has to solve. The way images are acquired determines how difficult it is to perform a task, and since systems have limited resources, their eyes should be designed to optimize subsequent image processing as it relates to particular tasks. We would like to gain insight into the relationship of eye design and task

Figure 2: A spherical Argus eye: A compound-like eye composed of conventional video cameras.

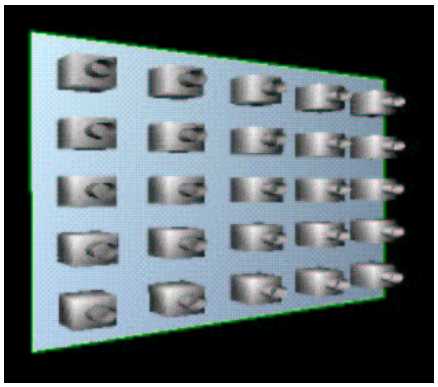


Figure 3: A planar Argus eye: A group of video cameras in a planar configuration This eye is good for motion segmentation.

performance.

Technological advances make it possible to construct integrated imaging devices using electronic and mechanical micro-assembly, micro-optics, and advanced data buses, not only of the kind that exists in nature such as log-polar retinas [8], but also of many other kinds [4]. Thus answers to our question are not only of relevant for explaning biological systems but also have immediate impact on technology.

We wish to evaluate and compare different eye designs in a scientific sense by using mathematical criteria. To do so we must select a (challenging) class of problems. As such a problem we have chosen the recovery of descriptions of space-time models from image sequences—a large and significant part of the vision problem itself. By "space-time models" we mean descriptions of shapes and descriptions of actions. An action is defined as a change of shape over time, which amounts to a 3D motion field. More specifically, we want to determine how we ought to collect images of a (dynamic) scene to best recover the scene's shapes and actions from video sequences. This problem has wide implications for a variety of applications not only in vision and recognition, but also in navigation, virtual reality, tele-immersion, and graphics. At the core of this capability is the celebrated module of structure from motion, and so our question becomes: What eye should we use, for collecting video, so that we can subsequently facilitate the structure from motion problem in the best possible way?

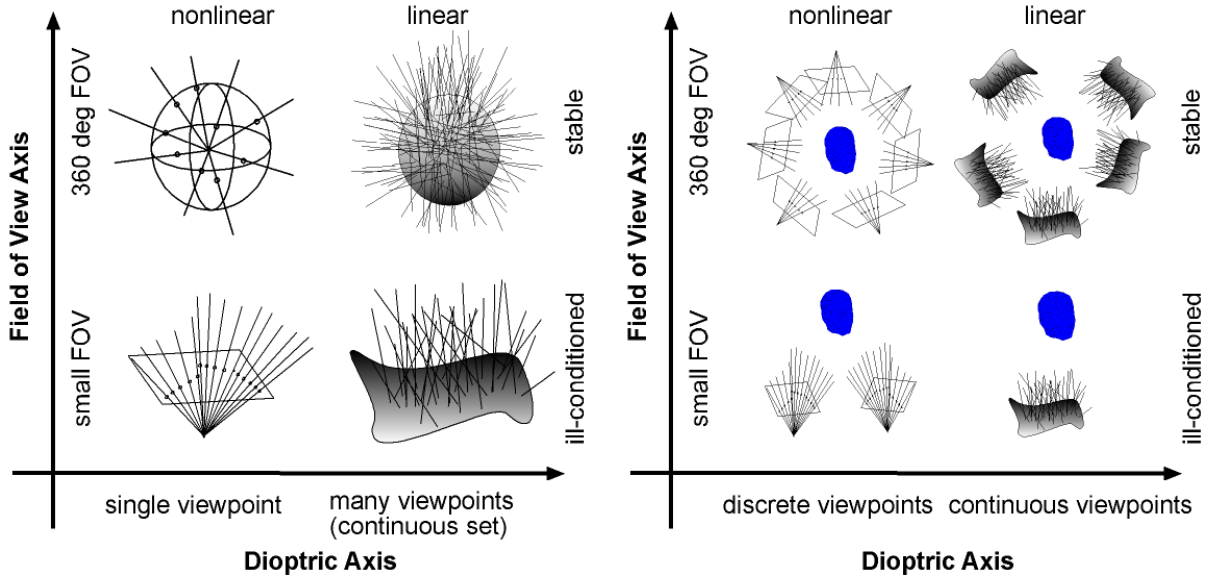
To classify cameras, we will study the most complete visual representation of the scene, namely the plenoptic function as it changes differentially over time. In free space the plenoptic function amounts to the 5D space of time-varying light rays; we denote the changes in this space as *plenoptic ray flow*. Any imaging device captures a subset of the plenoptic function. We would like to know how, by considering different subsets of the light field, the problem of structure from motion becomes easier or harder. The problem of structure from motion amounts to estimating the rigid motion which the camera undergoes on the basis of the rays captured by the camera; then, with knowledge of the camera's location at two (or more) time instants, through triangulation of rays originating from the same scene point, recovery of the scene structure is achieved.

A theoretical model for a camera that captures the plenoptic function in some part of the space is a surface S that has at every point a pinhole camera. With such a camera we observe every point in the scene in view from many different viewpoints (theoretically, from every point on S) and thus we capture many rays emanating from that point. This principle was already employed in a photographic technique called "Integral Photography" that was described by G.Lippmann and H.E. Ives at the beginning of the 20th century [24] and a possible implementation was presented in [2]. A theoretical parameterization for

these general cameras has been introduced recently in [18]. We call this camera a *polydioptric* camera[1].

Standard single-pinhole cameras capture only one ray from each point in space, and from this ray at different times the structure and motion must be estimated. This makes estimation of the viewing geometry a non-linear problem. The additional information in the polydioptric camera (multiple rays from the same scene point) makes estimation of the viewing geometry linear. There is another factor that affects the estimation, namely the surface S on which light is captured. It has long been known that there is an ambiguity in the estimation of the motion parameters for small field of view cameras, but only recently has it been noticed that this ambiguity disappears for a full field of view camera.

Thus there are two principles relating camera design to performance in structure from motion – the field of view and the linearity of the estimation. These principles are summarized in Fig. 4.



(a) Static World - Moving Cameras

(b) Moving Object - Static Cameras

Figure 4: Hierarchy of Cameras. We classify the different camera models according to the field of view (FOV) and the number and proximity of the different viewpoints that are captured (Dioptric Axis). This in turn determines if structure from motion estimation is a well-posed or an ill-posed problem, and if the motion parameters are related linearly or non-linearly to the image measurements. The camera models are clockwise from the lower left: small FOV pinhole camera, spherical pinhole camera, spherical polydioptric camera, and small FOV polydioptric camera.

For a planar camera, which by construction has a limited field of view, the problem is nonlinear and ill-posed. If, however, the field of view approaches $360°$, that is, the pencil of light rays is cut by a sphere, then the problem becomes well-posed and stable, although still nonlinear. It is currently technologically impossible to implement a high-resolution spherical camera. Catadioptric sensors have been used to capture a full field of view [10, 16], but they do not provide the resolution necessary for model building. One can, however, approximate such a camera by using several conventional cameras, capable of synchronized recording and arranged on a surface so that they capture a number of light ray pencils simultaneously. We use the name "Argus Eye" for such a device [5]. When the cameras are arranged on a sphere (or any other surface that enables sampling of the full field of view), then we

[1]A "plenoptic camera" had been proposed in [2], but since no physical device can capture the true time-varying plenoptic function, we prefer the term polydioptric to emphasize the difference between the continuous concept and the discrete implementation.

obtain a spherical Argus eye for which the structure from motion problem is well-posed (see Fig. 2 for a spherical Argus eye).

A polydioptric camera can be obtained if we arrange ordinary cameras very close to each other (Fig. 5). This camera has an additional property arising from the proximity of the individual cameras: it can form a very large number of orthographic images, in addition to the perspective ones. Indeed, consider a direction $\mathbf{r}$ in space and then consider in each individual camera the captured ray parallel to $\mathbf{r}$. All these rays together, one from each camera, form an image with rays that are parallel. Furthermore, for different directions $\mathbf{r}$ a different orthographic image can be formed. For example, Fig. 6 shows that we can select one appropriate pixel in each camera to form an orthographic image that looks to one side (blue rays) or another (red rays). Fig. 7 shows all the captured rays, thus illustrating that each individual camera collects conventional pinhole images.

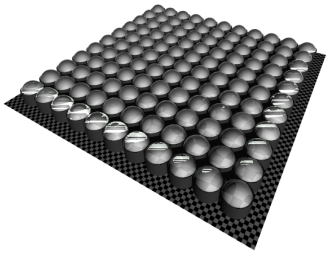


Figure 5: Design of a Polydioptric Camera

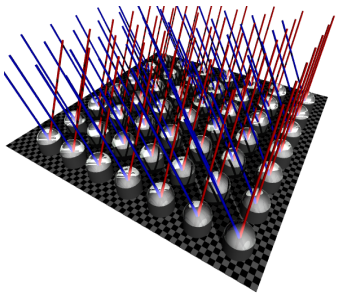


Figure 6: capturing Parallel Rays

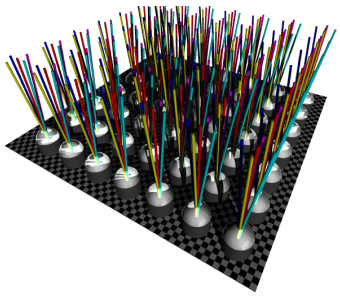


Figure 7: capturing Pencil of Rays

Thus, a polydioptric camera has the unique property that it captures, simultaneously, a large number of perspective and affine images (projections). We will demonstrate that it also makes the structure from motion problem linear. A polydioptric spherical camera is therefore the ultimate camera since it combines the stability of full field of view motion estimation with linearity of the problem, as well as the ability to reconstruct scene models with minimal reconstruction errors.

The general outline of this paper is as follows. We will define the framework of plenoptic video geometry (Section 2) and use it to analyze the effect of the field of view of a camera on motion estimation (Section 3). Then we relate the information in the polydioptric camera to the information in ordinary pinhole cameras and differential stereo setups, and also examine the relation between the depth of the scene and the plenoptic derivatives and their relative scales (Section 4). Finally, we propose a feedback algorithm to accurately compute the structure and motion using all the plenoptic derivatives, and we conclude with suggestions about how to implement and construct polydioptric and Argus eyes.

2 Plenoptic Video Geometry: The Differential Structure of the Space of Light Rays

2.1 Photometric Properties of the Scene

Let the scene surrounding the image sensor be modeled by the signed distance function $f(\mathbf{x};t) : \mathbb{R}^3 \times \mathbb{R}_+ \rightarrow \mathbb{R}$. The insides of the objects in the scene are defined by $f(\mathbf{x};t) \leq 0$, therefore, the iso-surface $f(\mathbf{x};t) = 0$ is a representation of the surfaces in the scene. The normal to the surfaces are given by $\mathbf{n}(\mathbf{x};t) = \nabla f(\mathbf{x},t)$. At each location $\mathbf{x}$ in free space $(f(\mathbf{x};t) > 0)$, the radiance, that is the light intensity or color observed at $\mathbf{x}$ from a given direction $\mathbf{r}$ at time t, can be measured by the plenoptic function $L(\mathbf{r};\mathbf{x};t)$; $L : \mathbb{S}^2 \times \mathbb{R}^3 \times \mathbb{R}_+ \rightarrow \mathbb{R}^d$, where $d = 1$ for intensity, $d = 3$ for color images, and $\mathbb{S}^2$ is the unit sphere of directions in $\mathbb{R}^3$ [1].

Assuming that at the intersection $\mathbf{y}$ of the ray ϕ ($\phi(\lambda) = \mathbf{x} + \lambda\mathbf{r}$) with the scene surface ($f(\mathbf{y}) = 0$) the albedo $\rho(\mathbf{y}; t)$ is continuously varying and no occlusion boundaries are present ($\mathbf{r} \cdot \mathbf{n}(\mathbf{y}, t) \neq 0$), then we can develop the plenoptic function $L(\mathbf{r}; \mathbf{x}; t)$ into a Taylor series

$$L(\mathbf{r} + \Delta\mathbf{r}; \mathbf{x} + \Delta\mathbf{x}; t + \Delta t) = L(\mathbf{r}; \mathbf{x}; t) + L_t \Delta t + \nabla_r L \cdot \Delta\mathbf{r} + \nabla_x L \cdot \Delta\mathbf{x} + \mathcal{O}(\|\Delta\mathbf{r}, \Delta\mathbf{x}, \Delta t\|^2) \quad (1)$$

where $L_t = \partial L/\partial t, \nabla_r = (\partial/\partial r_1, \partial/\partial r_2, \partial/\partial r_3)^{\mathrm{T}}, \nabla_x = (\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3)^{\mathrm{T}}$.

Disregarding the higher-order terms, we have a linear function which relates a local change in view ray position and direction to the brightness structure of the plenoptic function. Since a transparent medium such as air does not change the color of the light, we have a constant radiance along the view direction $\mathbf{r}$:

$$L(\mathbf{r}; \mathbf{x}; t) = L(\mathbf{r}; \mathbf{x} + \lambda\mathbf{r}; t)\ \forall\lambda : f(\mathbf{x} + \lambda\mathbf{r}) > 0 \text{ which implies} \quad (2)$$

$$\nabla_x L \cdot \mathbf{r} = \nabla_r L \cdot \mathbf{r} = 0\ \forall\mathbf{x} \in \mathbb{R}^3, f(\mathbf{x}; t) > 0. \quad (3)$$

Therefore, the plenoptic function in free space reduces to five dimensions – the time-varying space of directed lines for which many representations have been presented (for an overview see Camahort and Fussel [7]).

Since we assume a static world, the albedo of a world point does not change over time ($d\rho/dt = 0$), and the normals and illumination on the object surface are constant as well ($d/dt[\mathbf{n} \cdot \mathbf{s}] = 0$). It follows that the total time derivative of a single light ray vanishes:

$$\frac{d}{dt} L(\mathbf{r}; \mathbf{x}; t) = \frac{d}{dt}(\rho(\mathbf{x}; t)[\mathbf{n}(\mathbf{x}; t) \cdot \mathbf{s}(\mathbf{x}; t)]) = 0. \quad (4)$$

In other words, we assume that the intensity of a ray remains constant over consecutive time instants. This allows us to use the spatio-temporal brightness derivatives of the light rays captured by an imaging surface to constrain the *plenoptic ray flow*, that is the change in position and orientation between rays at consecutive time instants, by generalizing the well-known *Image Brightness Constancy Constraint* to the *Plenoptic Brightness Constancy Constraint*:

$$\frac{d}{dt} L(\mathbf{r}; \mathbf{x}; t) = L_t + \nabla_r L \cdot \frac{d\mathbf{r}}{dt} + \nabla_x L \cdot \frac{d\mathbf{x}}{dt} = 0. \quad (5)$$

2.2 Plenoptic Motion Equations

In this section we will relate the motion of an imaging sensor to the plenoptic brightness constancy constraint (Eq. 5). Assuming that the imaging sensor undergoes a rigid motion with instantaneous translation $\mathbf{t}$ and rotation $\boldsymbol{\omega}$ around the origin of the fiducial coordinate system, we can define the plenoptic ray flow for the ray captured by the imaging element located at location $\mathbf{x}$ and looking in direction $\mathbf{r}$ as

$$\frac{d\mathbf{r}}{dt} = \boldsymbol{\omega} \times \mathbf{r} \text{ and } \frac{d\mathbf{x}}{dt} = \boldsymbol{\omega} \times \mathbf{x} + \mathbf{t} \quad (6)$$

Combining Eqs. 5 and 6 leads to the *plenoptic motion constraint*

$$-L_t = \nabla_{\mathbf{x}} L \cdot (\boldsymbol{\omega} \times \mathbf{x} + \mathbf{t}) + \nabla_{\mathbf{r}} L \cdot (\boldsymbol{\omega} \times \mathbf{r}) = \nabla_{\mathbf{x}} L \cdot \mathbf{t} + (\mathbf{x} \times \nabla_{\mathbf{x}} L + \mathbf{r} \times \nabla_{\mathbf{r}} L) \cdot \boldsymbol{\omega} \quad (7)$$

which is a linear constraint in the motion parameters and relates them to all the differential image information that a sensor can capture. This equation will be our main tool for comparing the different camera models. To our knowledge, this is the first time that the temporal properties of the plenoptic function

have been related to the structure from motion problem. In previous work, the plenoptic function has mostly been studied in the context of image-based rendering in computer graphics under the names light field [21] and lumigraph [17], and only the 4D subspace of the static plenoptic function corresponding to the light rays in free space was examined. The advantages of multiple centers of projection with regard to the stereo estimation problem had been studied in [26] using tools similar those in [25].

It is important to realize that the derivatives $\nabla_{\mathbf{r}} L$ and $\nabla_{\mathbf{x}} L$ can be obtained from the image information captured by a polydioptric camera. Recall that a polydioptric camera can be envisioned as a surface where every point corresponds to a pinhole camera (see Fig. 5). $\nabla_{\mathbf{r}} L$, the plenoptic derivative with respect to direction, is the derivative with respect to the image coordinates that one finds in a traditional pinhole camera. One keeps the position and time constant and changes direction (Fig. 7). The second plenoptic derivative, $\nabla_{\mathbf{x}} L$, is obtained by keeping the direction of the ray constant and changing the position along the surface (Fig. 6). Thus, one captures the change of intensity between parallel rays. This is similar to computing the derivatives in an affine or orthographic camera. In section 1 we mentioned that a polydioptric camera captures perspective and affine images. $\nabla_{\mathbf{r}} L$ is found from the perspective images and $\nabla_{\mathbf{x}} L$ from the affine images. The ability to compute all the plenoptic derivatives depends on the ability to capture light at multiple viewpoints coming from multiple directions. This corresponds to the ability to incorporate stereo information into motion estimation, since multiple rays observe the same part of the world. For single-viewpoint cameras this is inherently impossible, and thus it necessitates nonlinear estimation over both structure and motion to compensate for this lack of multi-view (or equivalently depth) information. This will amplify the systematic errors in the estimated motion, as we describe in the next section.

3 Ambiguities due to the Field of View

For standard cameras with only one imaging surface and one pinhole there is only information about the change of intensity with direction ($\nabla_{\mathbf{r}} L$).

If the imaging surface is a sphere of unit radius centered at the origin of the fiducial coordinate system ($\mathbf{x} = \mathbf{0}$), we consider a parameterization of the imaging surface by the directional coordinates $\mathbf{r}$ where $\mathbf{R}$ is the scene point projected on the imaging surface at $\mathbf{r}$ and thus $\mathbf{r} = \mathbf{R}/|\mathbf{R}|$.

Assuming the intensity at corresponding image points to be the same, we obtain the *image brightness constraint equation*

$$L_t + \nabla_{\mathbf{r}} L \cdot \frac{d\mathbf{r}}{dt} = 0 \tag{8}$$

which we relate to the motion parameters as follows:

$$-L_t = \nabla_{\mathbf{r}} L \cdot \frac{d\mathbf{r}}{dt} = \nabla_{\mathbf{r}} L \cdot \left(\frac{1}{|\mathbf{R}|} \mathbf{t} + (\boldsymbol{\omega} \times \mathbf{r}) \right). \tag{9}$$

For planar cameras we parameterize the imaging surface by $\mathbf{r} = (x, y, f)$ with f the focal length ($\mathbf{r} = f\mathbf{R}/(\hat{\mathbf{z}} \cdot \mathbf{R})$) and obtain, by substituting for the image motion,

$$-L_t = \nabla_{\mathbf{r}} L \cdot \frac{d\mathbf{r}}{dt} = \nabla_{\mathbf{r}} L \cdot \left(\frac{1}{(\mathbf{R} \cdot \hat{\mathbf{z}})} (\hat{\mathbf{z}} \times (\mathbf{t} \times \mathbf{r})) + \frac{1}{f} \hat{\mathbf{z}} \times (\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})) \right) \tag{10}$$

where $\hat{\mathbf{z}}$ is a unit vector in the direction of the Z axis, and $\mathbf{R}$ is the scene point.

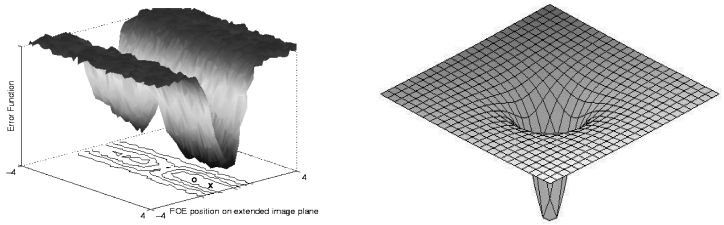
The most common approach to motion estimation on the basis of this input proceeds in two computational steps. First, on the basis of the image derivatives $(L_t, \nabla_{\mathbf{r}} L)$ one estimates an approximation to the motion field $d\mathbf{r}/dt$, the so-called optical flow field. To do so, one has to make assumptions about the flow which in essence amount to assumptions about the scene in view; usually the flow field is modeled

as varying smoothly. In a second step one solves for the parameters of the rigid motion, that is, the direction of the translational velocity $\mathbf{t}/|\mathbf{t}|$ and the rotational velocity $\boldsymbol{\omega}$. This is accomplished by minimizing deviation from the *epipolar constraint*. This constraint for both the plane and the sphere takes the form $(d\mathbf{r}/dt - \boldsymbol{\omega} \times \mathbf{r}) \cdot (\mathbf{t} \times \mathbf{r}) = 0$. As can be seen, this equation is non-linear in the motion parameters. Other approaches, often called direct approaches, relate the image derivatives directly to the 3D motion parameters. Without making any assumptions about the scene in view, the only constraint that can be used is the *depth positivity constraint*, that is, the depth has to have positive value. Algorithms that implement this constraint search (in appropriate subspaces) for the 3D motion parameters which yield the smallest number of negative depth values. Another way of relating image derivatives directly to the 3D motion is through the *depth variability constraint*. In this case one assumes the scene to be patch-wise smooth. A possible way to implement this constraint is through a search in the space of the directions of translation. For each translation candidate the rotation and inverse depth can be solved linearly.

Solving accurately for 3D motion parameters using conventional small field of view cameras turned out to be a very difficult problem. The main reason for this has to do with the apparent confusion between translation and rotation in the image displacements. This is easy to understand on 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 fields or the correspondences at the center of the image are very similar in the two cases. Thus, for example, translation along the x axis is confused with rotation around the y axis. This really is a geometric problem and it exists for both small and large baselines between the views, that is, for the case of continuous motion as well as in the case of discrete displacements of the cameras. The basic understanding of these difficulties has attracted only a few investigators over the years [3, 11, 12, 20, 22].

Having in mind the design of an optimal sensor, we are interested in how the stability of the estimation of motion changes with the field of view. In particular, we have compared the planar small field of view camera with a spherical camera [14]. Since motion estimation amounts to solving some minimization function, we analyzed the minimization functions corresponding to the different constraints described above. To be more precise, we performed a geometric statistical analysis; we compared the expected values of the different functions parameterized by the motion parameters. The topographic structure of the surfaces defined by these functions defines the behavior of the motion estimation.

We found that 3D motion estimation is much better behaved for cameras with a full field of view – spherical cameras – than for small field of view planar cameras.



(a) (b)

Figure 8: Schematic illustration of error function in the space of the direction of translation. (a) A valley for a planar surface with a limited field of view. (b) A clearly defined minimum for a spherical field of view.

Intuitively speaking, for imaging surfaces with small fields of view the minima of the error functions lie in a valley. This is a cause for inherent instability because, in a real situation, any point of that valley or flat area could serve as the minimum, thus introducing errors in the computation (see Fig. 8a). For imaging surfaces with a large field of view, on the other hand, the functions have a well defined-

minimum, as shown in Fig. 8b, and thus there is no ambiguity in the solution.

To give some geometric intuition, we write the motion constraint on the sphere (Eq. 9) as

$$-L_t = \nabla_r L \cdot \frac{\mathbf{t}}{|\mathbf{R}|} + (\mathbf{r} \times \nabla_r L) \cdot \boldsymbol{\omega} \tag{11}$$

Since $\nabla_r L$ is perpendicular to $(\mathbf{r} \times \nabla_r L)$, for a small field of view ($\mathbf{r}$ varies very little) and little variation in depth, a translational error $\mathbf{t}_\epsilon$ can be compensated by a rotational error $\boldsymbol{\omega}_\epsilon$ without violating the constraint in Eq. 11 as long as the errors have the following relationship:

$$\frac{1}{|\mathbf{R}|}\mathbf{r} \times \mathbf{t}_\epsilon = -\mathbf{r} \times (\mathbf{r} \times \boldsymbol{\omega}_\epsilon). \tag{12}$$

That is, the projections of the translational and rotational errors on the tangent plane to the sphere at $\mathbf{r}$ need to be perpendicular. We call this the orthogonality constraint on the plane. If we now increase the field of view, the constraint on the errors in Eq. 12 cannot be satisfied for all $\mathbf{r}$, thus the confusion disappears.

There is another ambiguity. Looking at the first term in Eq.11, that is $\nabla_r L \cdot \mathbf{t}/|\mathbf{R}|$, we see that the component of $\mathbf{t}$ parallel to $\mathbf{r}$ does not factor into the equation (since $\nabla_r L \cdot \mathbf{r} = 0$) and therefore cannot be recovered from the projection onto the gradients for a small field of view. We call this the line constraint on the plane, because the projections of the actual $\mathbf{t}$ (FOE) and the estimated $\tilde{\mathbf{t}} = \mathbf{t} + \lambda\mathbf{r}, \lambda \in \mathbb{R}$ onto the image plane lie on a line through the image center. Again an increase in the field of view will eliminate this ambiguity, since then measurements at other image locations enable us to estimate the component of $\mathbf{t}$ parallel to $\mathbf{r}$.

In particular, the ambiguity for planar surfaces is as follows: Denote the five unknown motion parameters as (x_0, y_0) (direction of translation) and (α, β, γ) (rotation). Then, if the camera has a limited field of view, *no matter how 3D motion is estimated from the motion field*, the expected solution will contain errors $(x_{0_\epsilon}, y_{0_\epsilon})$, $(\alpha_\epsilon, \beta_\epsilon, \gamma_\epsilon)$ that satisfy two constraints:

(a) The orthogonality constraint: $\dfrac{x_{0_\epsilon}}{y_{0_\epsilon}} = -\dfrac{\beta_\epsilon}{\alpha_\epsilon}$

(b) The line constraint: $\dfrac{x_0}{y_0} = \dfrac{x_{0_\epsilon}}{y_{0_\epsilon}}$

In addition, $\gamma_\epsilon = 0$.

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 because the depth $|\mathbf{R}|$ that is computed by substituting the estimated motion parameters into Eq. 9 will be distorted (see Section 4.2).

The proofs described here are of a theoretical nature. Nevertheless, we found experimentally that there were valleys in the function to be minimized for any indoor or outdoor sequence we worked on. Often we found the valley to be rather wide, but in many cases it was close in position to the predicted one.

If we increase the field of view of a sensor to $360°$, proofs in the literature show that we should be able to accurately recover 3D motion and subsequently shape [14]. Catadioptric sensors can provide the field of view, but the low and non-uniform resolution makes the signal processing difficult that is necessary to recover shape models. Thus, we built the Argus eye [5], a construction consisting of six cameras pointing outward (as in Fig. 2). Clearly, only parts of the sphere are imaged. When this structure is moved arbitrarily in space, data from all six cameras can be used to very accurately recover 3D motion, which can then be used in the individual videos to recover shape.

Since the six cameras do not have the same center of projection, motion estimation for this camera is more elaborate than for a spherical camera with a single center of projection, but because we know the geometric configuration of the cameras (from calibration) we can obtain all three translational parameters. The algorithm we implemented, in a nutshell, works as follows: For each individual camera we estimate the rigid motion in the coordinate system of the camera. In particular, we use the epipolar constraint. For every direction of translation we find the corresponding best rotation which minimizes deviation from this constraint. Fig. 9 shows (on the sphere of possible translations) the residuals of the epipolar error color coded for each individual camera. Noting that the red areas are all the points within a small percentage of the minimum, we can see the valleys, which clearly demonstrate the ambiguity theoretically shown in the proofs. Next, we consider for each camera the valleys with the best

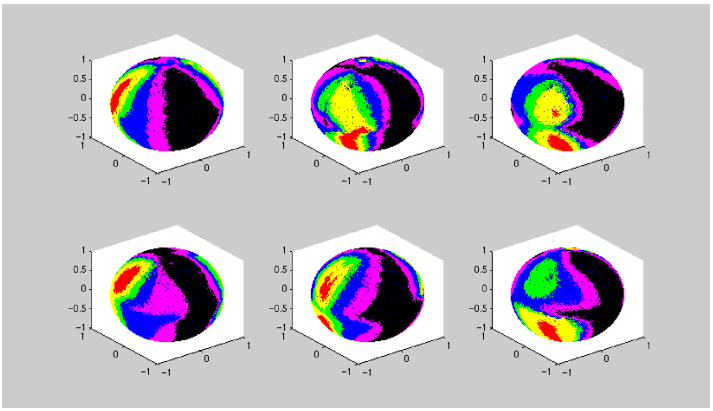


Figure 9: Deviation from the epipolar constraints for motion estimation from individual cameras

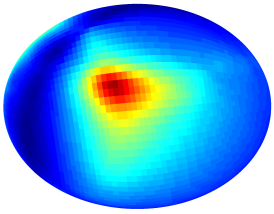


Figure 10: Combination of error residuals for a six camera Argus eye

candidate translations (The translational components differ, because the nodal points of the cameras are not aligned). To each translation corresponds a best rotation, and the intersection of the rotations corresponding to all the valleys provides the rotation of the system. Then, subtracting the rotational component, we find for each camera candidates for the translation. Their intersection provides the translation of the system.

In contrast, we see in Fig. 10 a well-defined minimum (in red) when we estimate the motion globally over all the Argus cameras, indicating that the direction of the translation obtained is not ambiguous when using information from a full field of view.

3.1 Noise Sensitivity due to Field of View

This section contains some observations describing how the field of view influences the sensitivity of the estimation for the polydioptric camera. A rigorous analysis comparing the sensitivity in polydioptric and conventional cameras is still needed and is the subject of current work. If we stack the motion constraints (Eq. 7) for all the measurements, we can form the system of linear equations

$$A\mathbf{b} = \mathbf{c} \text{ with} \tag{13}$$

$$A_i = [\nabla_x L_i, (\mathbf{x}_i \times \nabla_{\mathbf{x}} L_i + \mathbf{r}_i \times \nabla_{\mathbf{r}} L_i)] \tag{14}$$

$$\mathbf{b} = [\mathbf{t}; \boldsymbol{\omega}] \tag{15}$$

$$c_i = -(L_t)_i \tag{16}$$

where i denotes the index of the measurement. The least-squares solution to this overdetermined system is given by

$$\mathbf{b} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}\mathbf{c} = A^{+}\mathbf{c} \tag{17}$$

where A^{+} is the pseudo-inverse of A.

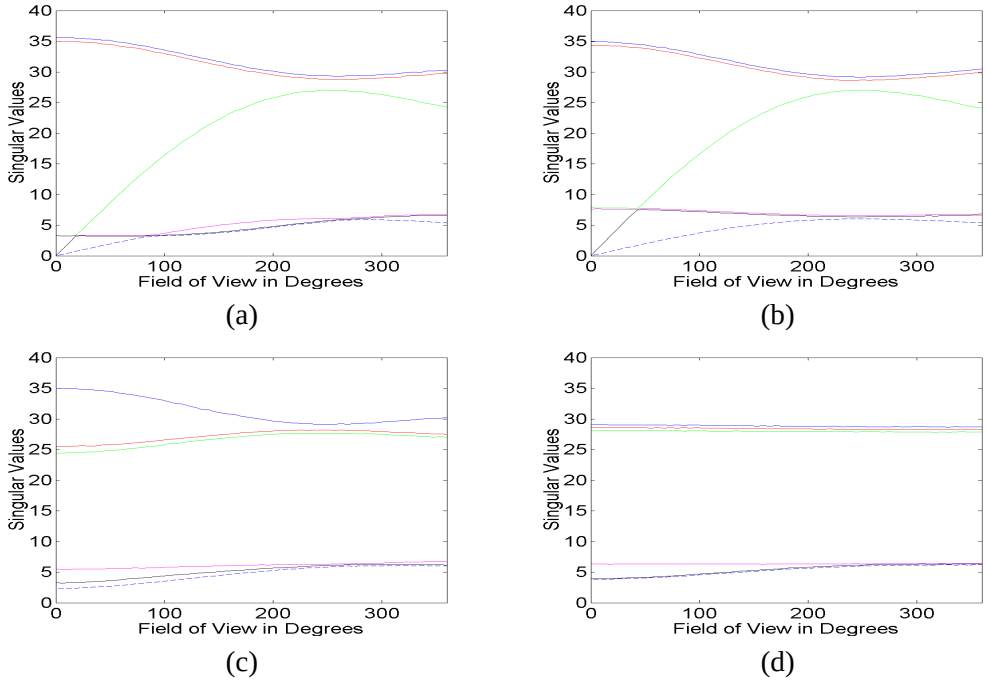


Figure 11: Singular values of matrix A for different camera setups in dependence on the field of view of the individual cameras:
(a) single camera, (b) two cameras facing forward and back, (c) two cameras facing forward and sideward, and (d) three cameras facing foward, sideways, and upward

Since our measurements are not perfect, we have to use an errors-in-the-variables model [27] to examine the sensitivity of the motion estimation problem based on the plenoptic motion constraint. Given errors in our derivative measurements, we must replace the exact quantities in the linear system by their estimated quantities $\tilde{\mathbf{c}} = \mathbf{c} + \mathbf{c}_\epsilon$ and $\tilde{A} = A + A_\epsilon$. Solving this linear system with errors defined in the least squares sense, will result in the following expected estimates for the motion parameters $\mathbf{b}$:

$$\tilde{\mathbf{b}} = \mathbf{b} - A^{+}A_\epsilon\mathbf{b} + (A^{\mathrm{T}}A)^{-1}A_\epsilon^{\mathrm{T}}\mathbf{c}_\epsilon = \mathbf{b} + (A^{\mathrm{T}}A)^{-1}(A_\epsilon^{\mathrm{T}}\mathbf{c}_\epsilon - A^{\mathrm{T}}A_\epsilon\mathbf{b}). \tag{18}$$

We can see that the amplification of the noise depends on the matrix $(A^{\mathrm{T}}A)^{-1}$. If A is well-conditioned, i.e. the ratio between its largest and smallest singular values is close to one, then the solution $\mathbf{b}$ will not be sensitive to small perturbations A_ϵ and c_ϵ, but if A is badly conditioned, i.e. A is close to singular and therefore some eigenvalues of $(A^{\mathrm{T}}A)^{-1}$ are very large, then small errors in the measurements can cause large errors in the motion estimate. Therefore, the effect of the field of view on the sensitivity of the motion estimation can be analyzed by examining how the singular values of the matrix A depend on the field of view. We did some synthetic experiments, where we simulated four different spatial arrangements of polydioptric cameras (see Fig. 11) similar to possible polydioptric Argus eye configurations. For each individual camera we defined the imaging surface to be the set of rays that made an angle of less than α degrees with the optical axis, where α was varied

to simulate different fields of view. The sum of the measurements for the whole system of cameras was kept constant to make the results for the different setups comparable. The same experiments also tell us about the influence of the field of view in the case of single-viewpoint cameras with known depth. For a forward-looking camera (Fig. 11 a) and forward and backward looking cameras (Fig. 11 b), two of the singular values vanish for a small field of view, implying that the estimation of the motion parameters is ill-posed. If we increase the field of view the linear system becomes better and better conditioned. When we arrange the cameras so that they face in perpendicular directions (Figs. 11 c, d), we see that the conditioning of the linear system is nearly independent of the field of view of the individual cameras, thus suggesting that motion estimation using an Argus eye configuration of conventional planar cameras is as robust as when using an ideal spherical eye.

4 Depth Information from Plenoptic Derivatives

Although the estimation of structure and motion for a single-viewpoint spherical camera, or its implementation as an Argus eye, is stable and robust, it is still non-linear, and the algorithms which give the most accurate results are search techniques, and thus rather elaborate. Therefore, we will now focus on the advantages of having access to both directional ($\nabla_r L$) as well as positional plenoptic derivatives ($\nabla_x L$) by analyzing the relationship between the plenoptic intensity derivatives. It will be shown how the depth information is encoded in the relations between the plenoptic derivatives $\nabla_x L$, $\nabla_r L$, L_t, and the camera motion parameters $\mathbf{t}$ and $\boldsymbol{\omega}$. The relationship between $\nabla_x L$ and $\nabla_r L$ had been previously utilized in differential stereo and epipolar plane image analysis [6], while $\nabla_r L$, L_t, $\mathbf{t}$, and $\boldsymbol{\omega}$ are used in differential motion estimation algorithms. Our work for the first time integrates differential motion information and differential stereo in a plenoptic framework using *all* the plenoptic derivatives.

4.1 Relationship between Plenoptic Derivatives

Comparing the plenoptic motion constraint (Eq. 7) for a single-viewpoint camera moving through the plenoptic space to the single-viewpoint motion constraint on the sphere (Eq. 9), we see that these constraints are nearly identical. Choosing the sensor viewpoint to be the origin of the coordinate system ($\mathbf{x} = \mathbf{0}$), the only difference is that in one case the translational component is given by $\nabla_x L \cdot \mathbf{t}$ and in the other case by $\nabla_r L \cdot \mathbf{t}/|\mathbf{R}|$. We can interpret the classical single-viewpoint motion constraint as being obtained from the plenoptic motion constraint by just substituting for the positional derivatives the directional derivatives and additional depth information. That is,

$$-L_t = \nabla_x L \cdot \mathbf{t} + \nabla_r L \cdot (\boldsymbol{\omega} \times \mathbf{r}) = 1/|\mathbf{R}| \; \nabla_r L \cdot \mathbf{t} + \nabla_r L \cdot (\boldsymbol{\omega} \times \mathbf{r}). \tag{19}$$

This relationship between the derivatives can easily be shown by the law of similar triangles. Since $f(\mathbf{x} + |\mathbf{R}|\mathbf{r}) = 0$ and the surface is assumed to have Lambertian reflectance, we can apply a simple triangulation argument (see Fig. 12) to get the following identity (because $\nabla_r L \cdot \mathbf{r} = 0$ we choose $\Delta\mathbf{x} = \Delta\mathbf{r}$ such that $\mathbf{r}\Delta\mathbf{x} = \mathbf{r} \cdot \Delta\mathbf{r} = 0$):

$$L(\mathbf{r} + \Delta\mathbf{r}; \mathbf{x}; t) = L(\mathbf{r}; \mathbf{x} + |\mathbf{R}|\Delta\mathbf{r}; t) \text{ or} \tag{20}$$

$$L(\mathbf{r}; \mathbf{x} + \Delta\mathbf{x}; t) = L(\mathbf{r} + \Delta\mathbf{x}/|\mathbf{R}|; \mathbf{x}; t) \tag{21}$$

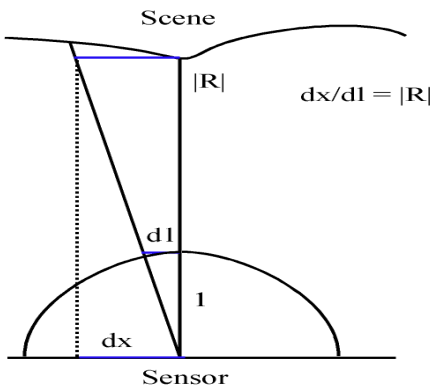


Figure 12: Relationship between directional and positional derivatives

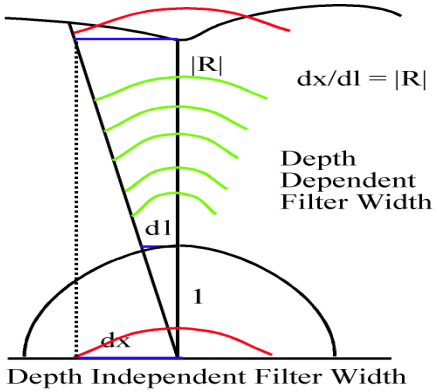


Figure 13: Depth dependence of derivative scale (matching scale in red)

If we now compute the directional derivative of the plenoptic function along direction $\Delta\mathbf{x}$, we get

$$\begin{aligned}\nabla_r L|_{\Delta\mathbf{x}} &= \lim_{\|\Delta\mathbf{x}\|\to 0} \frac{L(\mathbf{r}+\Delta\mathbf{x};\mathbf{x};t) - L(\mathbf{r};\mathbf{x};t)}{\|\Delta\mathbf{x}\|} \\ &= \lim_{\|\Delta\mathbf{x}\|\to 0} \frac{L(\mathbf{r};\mathbf{x}+|\mathbf{R}|\Delta\mathbf{x};t) - L(\mathbf{r};\mathbf{x};t)}{\|\Delta\mathbf{x}\|} \\ &= \lim_{\|\Delta\mathbf{x}\|\to 0} \frac{L(\mathbf{r};\mathbf{x}+\Delta\mathbf{x};t) - L(\mathbf{r};\mathbf{x};t)}{\frac{\|\Delta\mathbf{x}\|}{|\mathbf{R}|}} = |\mathbf{R}|\nabla_x L|_{\Delta\mathbf{x}}.\end{aligned} \tag{22}$$

and see that depth is encoded as the ratio between the positional and directional derivatives.

In differential stereo we have two (or more) cameras separated by a small baseline $\mathbf{b}$ (translating camera with known motion), and we want to recover depth by relating the image difference between the two cameras $L_2 - L_1$ to the spatial derivative $\nabla_r L$ in one of the images (again, using derivatives on the sphere we have $\nabla_r L \cdot \mathbf{r} = 0$). The familiar formulation of the differential stereo equation on the left hand side can now be restated using the fundamental relationship between the positional and directional plenoptic derivatives on the right hand side since for a small baseline $L_2 - L_1 \approx \nabla_x L \cdot \mathbf{b}$:

$$L_2 - L_1 = \frac{\nabla_r L \cdot \mathbf{b}}{|\mathbf{R}|} \rightarrow \frac{\nabla_r L \cdot \mathbf{b}}{L_2 - L_1} = |\mathbf{R}| \approx \frac{\nabla_r L \cdot \mathbf{b}}{\nabla_x L \cdot \mathbf{b}}. \tag{23}$$

Thus, we can interpret plenoptic motion estimation as the integration of differential motion estimation with differential stereo.

4.2 Structure from Plenoptic Derivatives and Motion Information

There are three ways to compute depth with a polydioptric sensor based on the measured plenoptic derivatives. Using the analysis of Section 4 and the plenoptic motion constraint Eq. 7, we can express the depth using each pairwise relation between the temporal, positional, and directional derivatives as follows:

$$|\mathbf{R}| = \frac{\nabla_r L}{\nabla_x L} = -\frac{\nabla_r L \cdot \mathbf{t}}{L_t + \nabla_r L \cdot (\boldsymbol{\omega} \times \mathbf{r})} = -\frac{L_t + \nabla_x L \cdot \mathbf{t}}{\nabla_x L \cdot (\boldsymbol{\omega} \times \mathbf{r})}. \tag{24}$$

For the latter two, we need to have an accurate estimate of the parameters of the rigid motion; otherwise the depth will be distorted as follows ($|\tilde{\mathbf{R}}|$ is the estimated depth, while the errors in the

parameters are denoted by the subscript ϵ):

$$|\tilde{\mathbf{R}}| = -\frac{\nabla_r L \cdot (\mathbf{t} + \mathbf{t}_\epsilon)}{L_t + \nabla_r L \cdot ((\boldsymbol{\omega} + \boldsymbol{\omega}_\epsilon) \times \mathbf{r})} = |\mathbf{R}| \left(-\frac{\nabla_r L \cdot (\mathbf{t} + \mathbf{t}_\epsilon)}{\nabla_r L \cdot (\mathbf{t} + |\mathbf{R}|(\boldsymbol{\omega}_\epsilon \times \mathbf{r}))} \right) = |\mathbf{R}| \cdot D_l. \quad (25)$$

$$= -\frac{L_t + \nabla_x L \cdot (\mathbf{t} + \mathbf{t}_\epsilon)}{\nabla_x L \cdot ((\boldsymbol{\omega} + \boldsymbol{\omega}_\epsilon) \times \mathbf{r})} = |\mathbf{R}| \left(-\frac{\nabla_x L \cdot (\mathbf{t}_\epsilon / |\mathbf{R}| + (\boldsymbol{\omega} \times \mathbf{r}))}{\nabla_x L \cdot ((\boldsymbol{\omega} + \boldsymbol{\omega}_\epsilon) \times \mathbf{r})} \right) = |\mathbf{R}| \cdot D_x. \quad (26)$$

The distortion of the depth and its dependence on the errors in the estimated motion parameters has been studied before[15], and it was shown that the distortion function D_l (Eq. 25) belongs to the family of Cremona transformations. The distortion function D_x has not been studied yet we believe that a distortion framework involving all the relations between the plenoptic derivatives is worth further study and will help us improve the accuracy of the scene structure that we recover.

4.3 Scale Dependence of the Plenoptic Derivatives

The accuracy of the linearization of the time-varying plenoptic function (Eq. 1) depends on the compatibility of the plenoptic derivatives. This means that the computation of all the plenoptic derivatives needs to be based upon similar subsets of the scene radiance. Unfortunately, an imaging sensor cannot measure the plenoptic derivatives, that is the derivatives of the scene radiance, directly. Instead we have to estimate the differential image properties by applying finite filters to image regions. Notice that if we combine information from neighboring measurements in directional space at a fixed position, we integrate radiance information over a region of the scene surface whose area scales with the distance from the sensor to the scene. In contrast, if we combine information over neighboring measurements in positional space for a fixed direction, we integrate information over a region of the scene surface whose area is independent of the depth (illustrated in Fig.13).

Unless the brightness structure of the scene has enough similarity across scales (e.g., if the local scene radiance changes linearly on the scene surface), so that the derivative computation is invariant to our choice of filter size, we have to make sure when we compute the plenoptic derivatives with respect to time, direction, and position that the domains of integration of our derivative filters relative to the scene are as similar as possible. This means that if we choose the plenoptic derivative filters such that the widths and orientations of the filters used to compute the positional and directional derivatives are related to the projection of the temporally integrated camera motion as follows ($\Delta\mathbf{x}$ and $\Delta\mathbf{r}$ denote the orientation and magnitude of the displacement that we use to compute the derivatives):

$$\Delta t \cdot (\mathbf{r} \times (\mathbf{r} \times (\mathbf{t} + |\mathbf{R}|(\boldsymbol{\omega} \times \mathbf{r})))) = \Delta\mathbf{r}|\mathbf{R}| = \Delta\mathbf{x}, \quad (27)$$

then Eq. 1 is a valid description of the local scene radiance and thus we will be able to compute accurate sensor motion and scene depth based upon this formulation.

One way to adjust the filter sizes would be to compute the temporal, directional and positional derivatives at many scales and use Eq. 24 as a constraint to find the best relative shift in scale space between the three derivatives.

This suggests the following plenoptic structure from motion algorithm. Using the proposed plenoptic motion framework, one can envision a feedback loop algorithm, where we use all the plenoptic derivatives to compute an estimate of the camera motion using Eq. 7. Since we are solving a linear system, the computation of the motion parameters is fast and we do not have any convergence issues as in the nonlinear methods necessary for single-viewpoint cameras. Then we can use the recovered motion together with the plenoptic derivatives to compute a scene depth estimate. If the three estimates in Eq. 24 do not match, we adapt the integration domains of the temporal, directional and positional derivative filters until we compute consistent depth and motion estimates. This is repeated for each

frame of the input video, while simultaneously we use the computed motion trajectory to integrate and refine the instantaneous depth maps in a large-baseline stereo optimization.

Another issue that needs to be considered is how densely the plenoptic space needs to be sampled by a polydioptric camera to be able to compute accurate structure and motion information. A similar problem has been studied in computer graphics under the name plenoptic sampling [9] with the aim of improving view interpolation in image-based rendering. The sampling density necessary to avoid aliasing depends of course on the brightness profile and depth structure of the scene, thus the choice of the correct smoothing filter depends very much on prior knowledge. The question how to design this optimal smoothing filter for motion estimation is beyond the scope of this paper.

4.4 Experiments

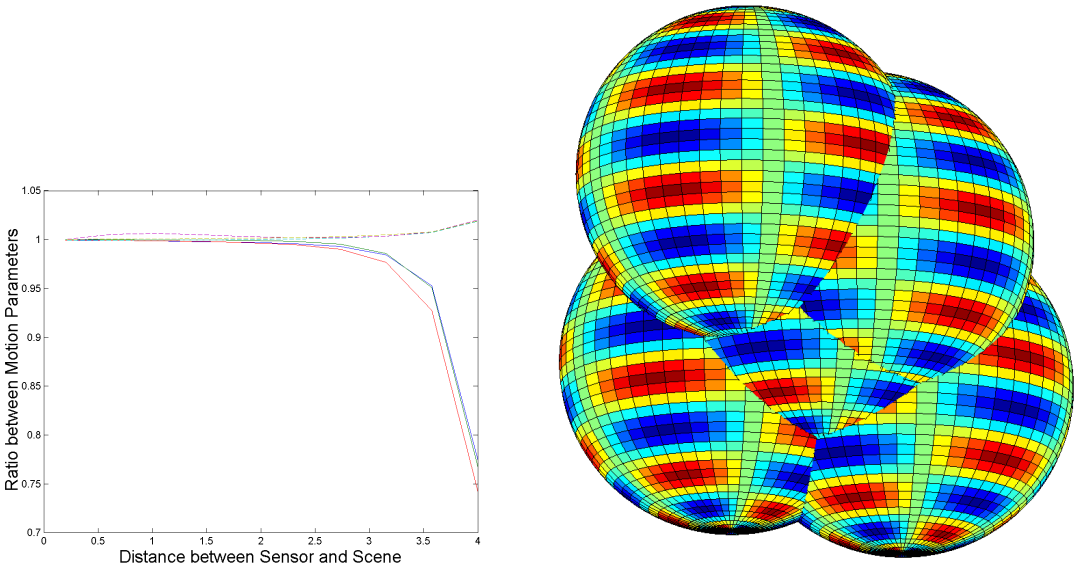


Figure 14: Left: Accuracy of Plenoptic Motion Estimation. The plot shows the relation between the ratio of the true and estimated motion parameters (vertical axis) and the distance between the sensor surface and the scene (horizontal axis). The solid lines denote the ratios of the rotation parameters, and the dashed lines the ratios of the translation parameters.
Right: Subset of an Example Scene.

To examine the performance of an algorithm using the plenoptic motion constraint, we did experiments with synthetic data. We distributed spheres, textured with a smoothly varying pattern, randomly in the scene so that they filled the horizon of the camera. We then computed the plenoptic derivatives through raytracing, stacked the linear equations (Eq. 7) to form a linear system, and solved for the motion parameters. Even using derivatives only at one scale, we find that the motion is recovered very accurately. As long as the relative scales of the derivatives were not too different (as in a distant scene) the error in the motion parameters varied between 1% and 3% as seen in Fig.14.

Since at depth discontinuities the plenoptic brightness constraint is not satisfied, we used the constraint that $\nabla_x L$ and $\nabla_r L$ are parallel by definition to prune the measurements, this improved the estimation for scenes with larger depth variation.

5 The Physical Implementation of Argus and Polydioptric Eyes

We are currently working on developing a highly integrated tennis-ball-sized Argus eye with embedded DSP power, integrated spherical image frame memory and high speed interface to a PC. A possible appearance of such an Argus eye is shown in Fig. 15. A simplified block diagram is shown in Fig. 16. A number of CCD or CMOS image sensor chipsets are interfaced to their own DSP chips. Normally, DSP chips have fast ports for communicating with other DSP chips, forming a parallel processor. Also, modern DSPs provide a host port through which a host computer can address and control the DSP as well as access its memory space. We envision using P1394 serial bus (Firewire). Through six wires this serial bus can recreate a complete parallel bus (e.g., PCI bus) at a remote location away from the host PC. In our case this remote location is inside the Argus eye. In effect, Firewire brings the PCI bus inside the Argus eye, allowing complete addressability, programmability and control from a single PC. Ultimately, the Argus eye will integrate our motion estimation algorithms within on-board DSP chips.

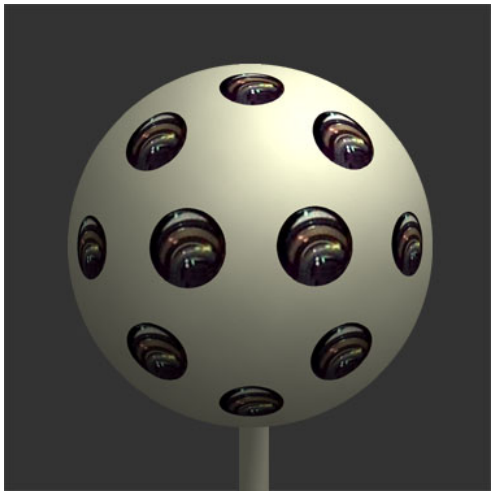


Figure 15: A highly integrated tennis-ball-sized Argus eye.

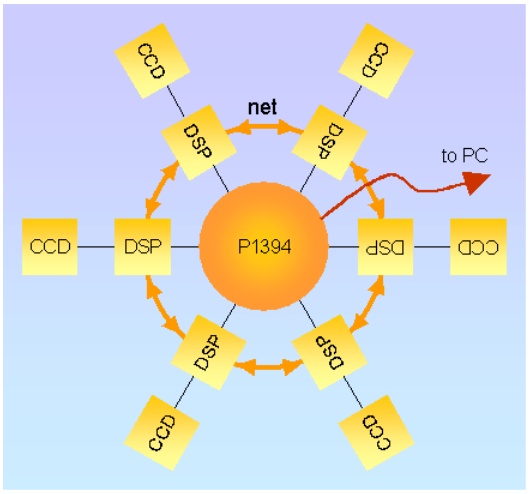


Figure 16: Block diagram of an Argus eye.

Of course one could think of numerous alternative implementations of Argus and polydioptric eyes. Ideas involving fish lenses and catadioptric mirrors [23] to project wide-angle panoramas onto a single sensor are first to come to mind. In principle, such cameras can be used for experimentation. Nonetheless, any optical approach that warps the panoramic optical field to project it onto one or two planar image sensors will suffer reduced spatial resolution, because too wide of an angle is squeezed onto a limited number of pixels. Given the low price of common resolution image sensors[2] as well as inexpensive plastic optics, there is no reason not to build Argus eye as suggested above – by pointing many cameras to look all around. In fact, the Argus eye is not only a panoramic spherical camera; it is a compound eye with many overlapping fields of view. A simple Argus eye could also be built by combining mirrors with conventional cameras, using the mirrors to split the field of view in such a way that we capture many directions in one image (e.g., one forward and one sideways).

By using special pixel readout from an array of tightly spaced cameras we can obtain a polydioptric camera. Perhaps the biggest challenge in making a polydioptric camera is to make sure that neighboring cameras are at a distance that allows estimation of the "orthographic" derivatives of rays, i.e., the change in ray intensity when the ray is moved parallel to itself. For scenes that are not too close to the cameras

[2] A 1/4" quality color CCD chipset with about 640×480 pixels can be purchased for about $100. It is estimated that in 10 years such cameras will cost only a few dollars.

it is not necessary to have the individual cameras very tightly packed; therefore, miniature cameras may be sufficient. The idea of lenticular sheets has appeared in the literature [2, 24] for 3D imaging. These sheets employ cylindrical lenses and are not very appealing because of the blurring they create. There are, however, similar micro-image formation ideas that would fully support the mathematical advantages of polydioptric eyes suggested in the previous section. One such idea is shown in Fig. 17. A micro-lens array is mounted on the surface of an image sensor directly, emulating an insect compound eye. Fig. 18 shows the imaging geometry. As an alternative to micro-lens arrays one could also consider coherent optical fiber bundles as image guides.

In this example, micro-lenses are focused at infinity. Advances in MEMS and micro-machining has resulted in the wide availability of micro-optics and lens arrays. They can be as small as tens of microns in diameter and can be arranged in a rectangular or hexagonal grid. The image sensor detects a plurality of optical images. These images are very small, perhaps 16–32 pixels on a side. They may not be useful for extensive imaging; however, they directly support computation that unleashes the power of polydioptric eyes. Fig. 19 depicts how these small images might look; they appear more like textures than like detailed images of the scene. However, these textures are sufficient for obtaining the rotational and translational derivatives of the light rays as 3D motion of the camera (or scene) occurs (i.e., $\nabla_r L$ and $\nabla_x L$). Fig. 19 shows that the derivatives of ray rotation (blue to red ray change in Fig. 18) would be computed by estimating the texture motion in individual sub-retinas over time. On the other hand, the derivatives of ray translation would be computed by estimating the time-evolving disparity of texture motions across pairs of sub-retinas. This is only one idea for building a miniature polydioptric eye. With a conventional image sensor we could have as many as 60×60 micro-lenses over a single image sensor. A plurality of such polydioptric eyes could be arranged to yield new camera topologies for vision computation.

Some form of acceptance optics would be needed to resize the field of view to meaningful sizes. These are commonly used as image tapers to resize the field of view in scientific cameras such as digital

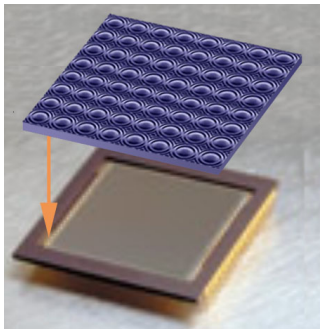


Figure 17: Forming a kind of polydioptric eye.

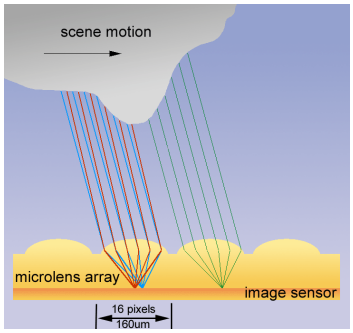


Figure 18: Plenoptic projection geometry for micro-lenses.

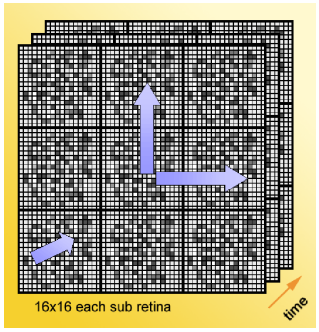


Figure 19: Plenoptic image processing.

X-ray cameras. Also, image guides are manufactured for use in medical instruments for laparoscopy. A plurality of image guides would be used. The acceptance faces of the image guides would be tightly arranged in an imaging surface; the exit faces could then be spaced apart and coupled to image sensors. A bundle of thousands of such fibers, appropriately calibrated, may constitute the ultimate design. We are currently experimenting with small versions and investigating the possibility of different optical materials. Using image guides is a more expensive proposition, but it is an attractive alternative, as this approach builds a superb compound eye with each "ommatidium" having a complete perspective image of the scene. Similar methods could be used to build small polydioptric domes: the acceptance faces

would surround the dome volume, and at some distance behind them, the image sensors can be placed.

Finally, we should emphasize that not all images collected by an Argus eye need to be kept in memory. For example, a polydioptric eye with a few thousand optical fibers can create the same number of perspective images plus several thousand more affine images for different directions. As we only need some calculations of raw derivatives in order to derive at a scene model, we can be selective regarding the images we wish to keep. In fact, our dedicated DSP engines can sift through their individual image data and preserve only higher-level entities, such as derivatives, needed for later computation. Perhaps one spherical image would also be computed and stored locally, but not all the raw images from the eye.

6 Conclusion

According to ancient Greek mythology Argus, the hundred-eyed guardian of Hera, the goddess of Olympus, alone defeated a whole army of Cyclopes, one-eyed giants. The mythological power of many eyes became real in this paper, which proposed a mathematical analysis of new cameras. This paper also, introduced for the first time, the relation between the local differential structure of the time-varying plenoptic function and the rigid motion of an imaging sensor. This relationship was used to formulate design principles for new cameras. Using the two principles relating camera design to the performance of structure from motion algorithms, the field of view, and the linearity of the estimation, we defined a hierarchy of camera designs. Although the mathematics of visual computing have been considerably advanced, the cameras we use in computer vision applications have basically been using the same principles for the past century: They capture a pencil of light rays with a limited field of view. In this paper, based upon the two design principles that we have formulated, we have introduced two new families of cameras, Argus eyes and polydioptric cameras. Argus eyes result from placing many conventional pinhole cameras, capable of synchronized recording, on a surface. The shape of the surface is essential. Spherical Argus eyes are very powerful, as they can estimate 3D motion from video in an unambiguous manner, and subsequently develop correct scene models. Polydioptric cameras are generalizations of Argus eyes with the individual cameras very close to each other. Polydioptric cameras capture all the rays falling on a surface and allow estimation of the change in any light ray under any rigid movement. This provides polydioptric cameras with the capability of solving for scene models in a linear manner, as described in the last section, opening new avenues for a variety of applications. For example, polydioptric domes open new avenues for 3D video development. We have analyzed the properties of the plenoptic derivatives and proposed a feedback algorithm to optimize the recovered motion of the imaging sensor and subsequently the structure of the scene based on this analysis.

Currently, we are developing different physical implementations of Argus and polydioptric eyes as described in Section 5, and we will evaluate the proposed plenoptic structure from motion algorithm on a benchmark set of image sequences captured by these new cameras.

Acknowledgement

The support of the National Science Foundation is gratefully acknowledged. Figures 15-19 are courtesy of Vladimir Brajovic, CMU.

References

[1] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J. A. Movshon, editors, *Computational Models of Visual Processing*, pages 3–20. MIT Press, Cambridge, MA, 1991.

[2] E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Trans. PAMI*, 14:99–106, 1992.

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

[4] J. Arnspang, H. Nielsen, M. Christensen, and K. Henriksen. Using mirror cameras for estimating depth. In *Computer Analysis of Images and Patterns, 6th International Conference*, pages 711–716, 1995.

[5] P. Baker, C. Fermüller, and Y. Aloimonos. A spherical eye from multiple cameras (or how to make better models). In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 576–583, 2001.

[6] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *Int. Journal of Computer Vision*, 1:7–55, 1987.

[7] E. Camahort and D. Fussell. A geometric study of light field representations. Technical Report TR99-35, Department of Computer Sciences, The University of Texas at Austin, 1999.

[8] C. Capurro, F. Panerai, and G. Sandini. Vergence and tracking fusing log-polar images. In *Proc. International Conference on Pattern Recognition*, 1996.

[9] J. Chai, X. Tong, and H. Shum. Plenoptic sampling. In *Proc. of ACM SIGGRAPH*, pages 307–318, July 2000.

[10] P. Chang and M. Hebert. Omni-directional structure from motion. In *Proc. IEEE Workshop on Omnidirectional Vision*, pages 127–133, Hilton Head Island, SC, June 2000. IEEE Computer Society.

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

[12] 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*. Lawrence Erlbaum Associates, Mahwah, NJ, 1997.

[13] R. Dawkins. *Climbing Mount Improbable*. Norton, New York, 1996.

[14] C. Fermüller and Y. Aloimonos. Observability of 3D motion. *International Journal of Computer Vision*, 37:43–63, 2000.

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

[16] J. Gluckman and S. K. Nayar. Egomotion and omnidirectional sensors. In *Proc. International Conference on Computer Vision*, 1998.

[17] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *Proc. of ACM SIGGRAPH*, 1996.

[18] Michael D. Grossberg and Shree K. Nayar. A general imaging model and a method for finding its parameters. In *Proc. International Conference on Computer Vision*, July 2001.

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

[20] A. D. Jepson and D. J. Heeger. Subspace methods for recovering rigid motion II: Theory. Technical Report RBCV-TR-90-36, University of Toronto, 1990.

[21] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. of ACM SIGGRAPH*, 1996.

[22] S. J. Maybank. Algorithm for analysing optical flow based on the least-squares method. *Image and Vision Computing*, 4:38–42, 1986.

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

[24] T. Okoshi. *Three-dimensional Imaging Techniques*. Academic Press, 1976.

[25] H. Sahabi and A. Basu. Analysis of error in depth perception with vergence and spatially varying sensing. *Computer Vision and Image Understanding*, 63(3):447–461, 1996.

[26] H.Y. Shum, A. Kalai, and S. M. Seitz. Omnivergent stereo. In *Proc. International Conference on Computer Vision*, 1999.

[27] G.W. Stewart. Stochastic perturbation theory. *SIAM Review*, 32:576–610, 1990.