A Hierarchy of Cameras for 3D Photography

Jan Neumann [*,a], Cornelia Fermüller [a] and Yiannis Aloimonos [a]

[a] *Computer Vision Laboratory*
University of Maryland
College Park, MD 20742-3275, USA

Abstract

The view-independent visualization of 3D scenes is most often based on rendering accurate 3-dimensional models or utilizes image-based rendering techniques. To compute the 3D structure of a scene from a moving vision sensor or to use image-based rendering approaches, we need to be able to estimate the motion of the sensor from the recorded image information with high accuracy, a problem that has been well-studied. In this work, we investigate the relationship between camera design and our ability to perform accurate 3D photography, by examining the influence of camera design on the estimation of the motion and structure of a scene from video data. By relating the differential structure of the time varying plenoptic function to different known and new camera designs, we can establish a hierarchy of cameras based upon 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 motion estimation problem can be solved independently of the depth of the scene which leads to fast and robust algorithms for 3D Photography. In between are multiple view cameras with a large field of view which we have built, as well as omni-directional sensors.

Key words: Multi-view Geometry, Spatio-temporal Image Analysis, Camera Design, Structure from Motion, Polydioptric Cameras

* Corresponding author.
Email addresses: jneumann@cfar.umd.edu (Jan Neumann), fer@cfar.umd.edu (Cornelia Fermüller), yiannis@cfar.umd.edu (Yiannis Aloimonos).

1 Introduction

The concept of three-dimensional(3D) photography and imaging was always of great interest to humans. Early attempts to record and recreate images with depth were the stereoscopic drawings of Giovanni Battista della Porta around 1600, and the stereoscopic viewers devised by Wheatstone and Brewster in the 19th century. As described in [34], in the 1860's Francois Villème invented a process known as photo sculpture, which used 24 cameras, to capture the notion of a three-dimensional scene. Later a three-dimensional photography and imaging technique was invented by G. Lippmann in 1908 under the name of integral photography where the object was observed by a large number of small lenses arranged on a photographic sheet resulting in many views of the object from different directions [35]. Today modern electronic display techniques enable the observer to view objects from arbitrary view points and explore virtual worlds freely. These worlds need to be populated with realistic renderings of real life objects to give the observer the feel of truly spatial immersion. This need fuels the demand for accurate ways to recover the 3D shape and motion of real world objects. In general, the approaches to recover the structure of an object are either based on active or passive vision sensors, i.e. sensors that interact with their environment or sensors that just observe without interference. The main examples for the former are laser range scanners [38] and structured light based stereo configurations where a pattern is projected onto the scene and the sensor uses the image of the projection on the structure to recover depth from triangulation [5,12,44]. For a recent overview over different approaches to active range sensing and available commercial systems see [6]. The category of passive approaches consists of stereo algorithms based on visual correspondence where the cameras are separated by a large baseline [39] and structure from motion algorithms on which we will concentrate [17,24]. Since correspondence is a hard problem for widely separated views, we believe that the structure from motion paradigm offers the best approach to 3D photography [37,16], because it interferes the least with the scene being imaged and the recovery of the sensor motion enables us to integrate depth information from far apart views for greater accuracy while taking advantage of the easier correspondence due to dense video. In addition, the estimation of the motion of the camera is a fundamental component of most image-based rendering algorithms.

There are many approaches to structure from motion (e.g. see [17,24] for an overview), but essentially all these approaches disregard the fact that the way how images are acquired already determines to a large degree how difficult it is to solve for the structure and motion of the scene. Since systems have to cope with limited resources, their cameras should be designed to optimize subsequent image processing.

The biological world gives a good example of task specific eye design. It has been estimated that eyes have evolved no fewer than forty times, independently, in diverse parts of the animal kingdom [13]. These eye designs, and therefore the images they capture, are highly adapted to the tasks the animal has to perform. The sophistication of these eyes suggests that we should not just focus our efforts on designing algorithms that optimally process a given visual input, but also optimize the design of the imaging sensor with regard to the task at hand, so the subsequent processing of the visual information is facilitated. This focus on sensor design has already begun, we just mention as an example the influential work on catadioptric cameras [28].

In [32] we presented a framework to relate the design of an imaging sensor to its usefulness for a given task. Such a framework allows us to evaluate and compare different camera designs in a scientific sense by using mathematical considerations.

To design a task specific camera, we need to answer the following two questions:

(1) How is the relevant visual information that we need to extract to solve our task encoded in the visual data that a camera can capture?
(2) What is the camera design and image representation that optimally facilitates the extraction of the relevant information?

To answer the first question, we first have to think about what we mean by visual information. When we think about vision, we usually think of interpreting the images taken by (two) eyes such as our own - that is, perspective images acquired by camera-type eyes based on the pinhole principle. These images enable an easy interpretation of the visual information by a human observer. Therefore, most work on sensor design has focused on designing cameras that would result in pictures with higher fidelity (e.g.[18]). Image fidelity has a strong impact on the accuracy with which we can make quantitative measurements of the world, but the qualitative nature of the image we capture (e.g., single versus multiple view point images) also has a major impact on the accuracy of measurements which cannot be measured by a display-based fidelity measure. Since nowadays most processing of visual information is done by machines, there is no need to confine oneself to the usual perspective images. Instead we propose to study how the relevant information is encoded in the geometry of the time-varying space of light rays which allows us to determine how well we can perform a task given *any* set of light ray measurements.

To answer the second question we have to determine how well a given eye can capture the necessary information. We can interpret this as an approximation problem where we need to assess how well the relevant subset of the space of light rays can be reconstructed based on the samples captured by the eye, our

knowledge of the transfer function of the optical apparatus, and our choice of function space to represent the image. By modeling eyes as spatio-temporal sampling patterns in the space of light rays we can use well developed tools from signal processing and approximation theory to evaluate the suitability of a given eye design for the proposed task and determine the optimal design. The answers to these two questions then allow us to define a fitness function for different camera designs with regard to a given task.

In this work, we will extend our study of the structure of the time-varying plenoptic function captured by a rigidly moving imaging sensor to analyze how the ability of a sensor to estimate its own rigid motion is related to its design and what this effect has on triangulation accuracy, and thus on the quality of the shape models that can be captured.

2 Plenoptic Video Geometry: How is 3D motion information encoded in the space of light rays?

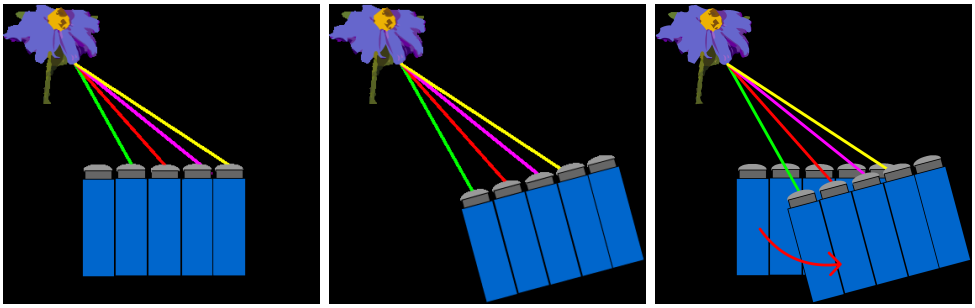


Fig. 1. Illustration of ray incidence and ray identity: (a-b) A multi-perspective system of cameras observes an object in space while undergoing a rigid motion. Each individual camera sees a scene point on the object from a different view point which makes the correspondence of the rays depend on the scene geometry (ray incidence). (c) By computing the inter- and intra-camera correspondences of the set of light rays between the two time instants, we can recover the motion of the camera without having to estimate the scene structure since we correspond light rays and not views of scene points (ray identity).

The space of light rays is determined by the geometry and motion of the objects in space, their surface reflection properties, and the light sources in the scene. The most general representation for the space of light rays is the plenoptic parameterization. At each location $\boldsymbol{x} \in \mathbb{R}^3$ in free space, the radiance, that is the light intensity or color observed at $\boldsymbol{x}$ from a given direction $\boldsymbol{r} \in \mathbb{S}^2$ at time $t \in \mathbb{R}^+$, is measured by the plenoptic function $\mathcal{L}(\boldsymbol{x}; \boldsymbol{r}; t)$; $\mathcal{L} : \mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{R}_+ \rightarrow \Gamma$. Γ denotes here the spectral energy, and equals $\mathbb{R}$ for monochromatic light, $\mathbb{R}^n$ for arbitrary discrete spectra, or could be a function space for a continuous spectrum. $\mathbb{S}^2$ is the unit sphere of directions in $\mathbb{R}^3$.

We will interpret the images captured by a generalized camera in terms of samples of the plenoptic function. Based on these samples, we can estimate the local geometry of the plenoptic function and then utilize these features in the space of light rays to recover spatio-temporal information about the world. The structure of the world is specified by a directional distance map $\mathcal{Z}(\boldsymbol{x}, \boldsymbol{r}, t) : \mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{R}_+$ which encodes both the shape and motion of the objects in the scene.

We can define a generalized camera in terms of a set of two-dimensional imaging surfaces $\mathcal{C}_i(u, v)$ which are indexed by pixel coordinates (u, v). Associated with each camera surface is a pair of functions that map a pixel to a ray in space. Each ray is defined by a position $(\boldsymbol{x}_i : (u, v) \in \mathbb{R}^2 \rightarrow \mathbb{R}^3)$ and a direction in space $(\boldsymbol{r}_i : (u, v) \in \mathbb{R}^2 \rightarrow \mathbb{S}^2)$. These functions do not need to be continuous, because adjacency in (u, v) does not necessarily imply adjacency in the space of light rays. One can for example think of a camera that consists of a set of mini-lenses on a CCD chip where at the boundaries between adjacent mini-lenses we will have a discontinuity in the observed ray directions and positions.

Each camera surface collects an imaging sequence $I_i(u, v, t)$. We assume that the camera is undergoing a rigid motion in space which is parameterized by a rotation matrix R and a translation vector $\boldsymbol{q}$ thus the world coordinates of a ray entering pixel (u, v) in camera $\mathcal{C}_i$ are given by $\boldsymbol{x}_i(u, v, t) = R(t)\boldsymbol{x}_i(u, v) + \boldsymbol{q}(t)$ and $\boldsymbol{r}_i(u, v, t) = R(t)\boldsymbol{r}_i(u, v)$ as illustrated in Table 1.

Depending on the geometric properties of the camera, there are different types of features that can be computed, and we have two fundamentally different types of constraints.

Camera Type	Ray Incidence	Ray Identity
Single view point	Camera motion $\otimes$ depth (Structure from motion)	Camera rotation (3 d.o.f.)
Multiple view points	Stereo motion Small and large-baseline stereo	Rigid motion (6 d.o.f.) Differential Stereo

Table 1
Quantities that can be computed using the basic light ray constraints for different camera types. $\otimes$ denotes that two quantities cannot be independently estimated without assumptions about scene or motion.

If a scene made up of diffuse reflective surfaces is observed from multiple view points, then the views of the same surface region will look very similar. The intensity function defined on a line pencil through a scene point is expected to have a much lower variance then a line pencil through a point in free air. This photo-consistency constraint [22] allows us to identify corresponding

projections of the same scene point in the different views. This can also be extended to other features such as sets of rays through lines and planes [2,23]. Given corresponding features we can then infer geometrical information such as shape and occlusions about the scene, as well as the position of the cameras. We call these constraints *ray incidence constraints*.

If we move a multi-perspective camera in space, there is a second constraint. If the camera captures overlapping sets of light rays at two time instants, then we can register the two sets of light rays and recover the motion of the camera. We term this constraint the *ray identity constraint*. The general principle is illustrated in Figure 1.

In this paper we will show how these different constraints can be used to compute the motion of a camera and the shape of the scene based on measurements in the space of light rays.

3 Ray incidence constraint

The ray incidence constraint is defined in terms of a scene point $\boldsymbol{P}$ and set of rays $l_i := (\boldsymbol{x}_i, \boldsymbol{r}_i)$. The incidence relation between the scene point $\boldsymbol{P} \in \mathbb{R}^3$ and rays l_i, defined by their origins $\boldsymbol{x}_i \in \mathbb{R}^3$ and directions $\boldsymbol{r}_i \in \mathbb{S}^2$, can be written as

$$[\boldsymbol{r}_i]_\times \boldsymbol{x}_i = [\boldsymbol{r}_i]_\times \boldsymbol{P} \; \forall \; i \tag{1}$$

were $[\boldsymbol{r}]_\times$ denotes the skew-symmetric matrix so that $[\boldsymbol{r}]_\times \boldsymbol{x} = \boldsymbol{r} \times \boldsymbol{x}$. The rays that satisfy this relation form a 3D-line pencil in the space of light rays.

The geometric incidence relations for light rays lead to extensions of the familiar multi-view constraints for light rays. Depending on the design of the imaging sensor there are three types of epipolar constraints that constrain the structure and motion estimation ($\boldsymbol{q}_{j,i}, R_{j,i}$ are the translation and rotation that move l_j into the camera coordinate system of l_i in case they were measured at different times).

(1) Pinhole camera: If we have a conventional single-view point camera, then each scene point is only observed once for each frame and we have the usual single view point epipolar constraint that constrains the camera motion up to scale and has the form

$$\boldsymbol{r}_i^{\mathrm{T}} [\boldsymbol{q}_{i,j}]_\times R_{j,i} \boldsymbol{r}_j = 0.$$

(2) Non-central "Argus Eye" camera: If we observe the world using multiple single viewpoint cameras with non-overlapping field of view, then we can utilize an epipolar constraint on the motion between frames including the

scale of the reconstruction which has the form

$$\boldsymbol{r}_i^{\mathrm{T}}[\boldsymbol{q}_{i,j} + (\boldsymbol{x}_i - R_{j,i}\boldsymbol{x}_j)]_\times R_{j,i}\boldsymbol{r}_j = 0.$$

(3) Polydioptric or stereo camera: If we have a multi-perspective camera and the scene point $\boldsymbol{P}_k$ projects into multiple locations at the same time, then in addition to the previous two types of epipolar constraints, we can also utilize a stereo constraint between two feature locations l_i and l_j that is independent of the motion between frames of the form

$$\boldsymbol{r}_i^{\mathrm{T}}[(\boldsymbol{x}_i - \boldsymbol{x}_j)]_\times \boldsymbol{r}_j = 0.$$

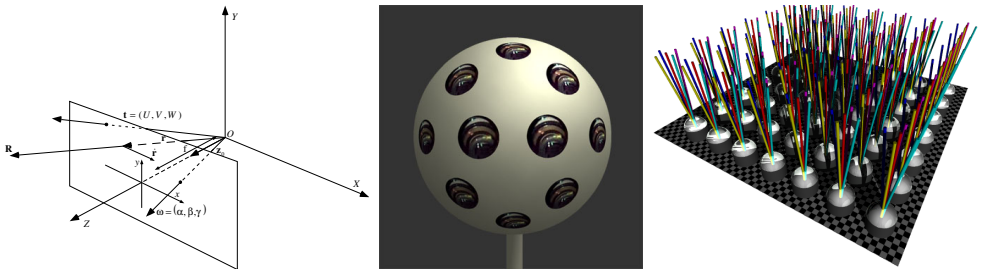


Fig. 2. Examples of the three different camera types: (a) Pinhole camera (b) Spherical Argus eye and (c) Polydioptric lenslet camera.

Possible implementations of these three camera types can be seen in Fig. 2

To solve for the motions and 3D positions of points, we need to optimize over the manifold of rotations and translations. The description of this optimization is beyond the scope of this paper. The interested reader is referred to the papers [25] for the two view case and [26] for the multiple view case, as well as the general description of how to optimize functions on spaces with orthogonality constraints [14].

If the camera moves differentially between frames, then the ray intersection constraint leads to the optical flow equations for a rigidly moving camera. If we observe the same scene point $\boldsymbol{P}$ from two different locations $\boldsymbol{x}$ and $\boldsymbol{x}+\Delta\boldsymbol{x}$, then we have for a Lambertian surface

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{r}, t) = \mathcal{L}(\boldsymbol{x}, (\boldsymbol{p} - \boldsymbol{x})/\|\boldsymbol{p} - \boldsymbol{x}\|, t) = \mathcal{L}(\boldsymbol{x} + \Delta\boldsymbol{x}, \frac{\boldsymbol{p} - \boldsymbol{x} - \Delta\boldsymbol{x}}{\|\boldsymbol{p} - \boldsymbol{x} - \Delta\boldsymbol{x}\|}, t) \quad (2)$$

We can write (where $\lambda = \mathcal{D}(\boldsymbol{x}, \boldsymbol{r}, t)$ is the distance between $\boldsymbol{x}$ and $\boldsymbol{P}$).

$$\frac{\boldsymbol{p} - \boldsymbol{x} - \Delta\boldsymbol{x}}{\|\boldsymbol{p} - \boldsymbol{x} - \Delta\boldsymbol{x}\|} = \frac{\lambda\boldsymbol{r} - \Delta\boldsymbol{x}}{\|\lambda\boldsymbol{r} - \Delta\boldsymbol{x}\|} = \boldsymbol{r} - \frac{(I_3 - \boldsymbol{r}\boldsymbol{r}^{\mathrm{T}})\Delta\boldsymbol{x}}{\lambda} + \mathcal{O}(\left\|\frac{\Delta\boldsymbol{x}}{\lambda}\right\|^2)$$

which is the well-known expression for translational motion flow on a spherical imaging surface. Assuming brightness constancy, we have

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{r}, t) = \mathcal{L}(\boldsymbol{x} + \Delta\boldsymbol{x}, \boldsymbol{r} - \mathcal{P}(\boldsymbol{r})\Delta\boldsymbol{x}/\lambda, t) = \mathcal{L}(\boldsymbol{x} + \Delta\boldsymbol{x}, \boldsymbol{r} + \Delta\boldsymbol{r}, t). \quad (3)$$

which relates differential motion and stereo estimation from light ray derivatives to the estimation using image derivatives..

4 Ray Identity Constraint

In a static world, where the albedo of every scene point is not changing over time, the brightness structure of the space of light rays is time-invariant, thus if a camera moves rigidly and captures two overlapping sets of light rays at two different time instants, then a subset of these rays should match exactly and would allow us to recover the rigid motion from the light ray correspondences. Note that this is a true brightness constancy constraint because we compare each light ray to itself. This is in contrast to the usual assumption of brightness constancy, where we have to assume a notion of view invariance since we compare two views of the same scene point. This is illustrated in Fig. 1.

As we had shown in previous work [31,30], this brightness constancy constraint can be utilized to estimate the motion of the camera in a scene-independent way. The set of imaging elements that make up a camera each capture the radiance at a given position $\boldsymbol{x} \in \mathbb{R}^3$ coming from a given direction $\boldsymbol{r} \in \mathbb{S}^2$. If the camera undergoes a rigid motion and we choose the camera coordinate system as our fiducial coordinate system, then we can describe this motion by an opposite rigid coordinate transformation of the ambient space of light rays in the camera coordinate system. This rigid transformation, parameterized by the rotation matrix $R(t)$ and a translation vector $\boldsymbol{q}(t)$, results in the following *exact* equality which is called the *discrete plenoptic motion constraint*

$$\mathcal{L}(R(t)\boldsymbol{x} + \boldsymbol{q}(t); R(t)\boldsymbol{r}; t) = \mathcal{L}(\boldsymbol{x}; \boldsymbol{r}; 0) \tag{4}$$

since the rigid motion maps the time-invariant space of light rays upon itself. Thus, if a sensor is able to capture a continuous, non-degenerate subset of the plenoptic function, then the problem of estimating the rigid motion of this sensor has become an image registration problem that is *independent of the scene*. Therefore the only free parameters are the six degrees of freedom of the rigid motion. This global parameterization leads to a highly constrained estimation problem that can be solved with any multi-dimensional image registration criterion.

If in the neighborhood of the intersection point $\boldsymbol{y} \in \mathbb{R}^3$ of the ray ϕ ($\phi(\lambda) = \boldsymbol{x} + \lambda\boldsymbol{r}$) with the scene surface the albedo is continuously varying and no occlusion boundaries are present, then the plenoptic function $\mathcal{L}$ changes smoothly and we can develop the plenoptic function $\mathcal{L}$ in the neighborhood of $(\boldsymbol{x}; \boldsymbol{r}; t)$ into

a Taylor series (we use $\mathcal{L}_t$ as an abbreviation for $\partial\mathcal{L}/\partial t$):

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}+d\boldsymbol{x};\boldsymbol{r}+d\boldsymbol{r};t+dt) &= \mathcal{L}(\boldsymbol{x};\boldsymbol{r};t) \\ +\mathcal{L}_t dt + \nabla_{\boldsymbol{x}}\mathcal{L}^{\mathrm{T}} d\boldsymbol{x} + \nabla_{\boldsymbol{r}}\mathcal{L}^{\mathrm{T}} d\boldsymbol{r} &+ \mathcal{O}(\|d\boldsymbol{r}, d\boldsymbol{x}, dt\|^2).\end{aligned} \tag{5}$$

where $\nabla_{\boldsymbol{x}}\mathcal{L}$ and $\nabla_{\boldsymbol{r}}\mathcal{L}$ are the partial derivatives of $\mathcal{L}$ with respect to $\boldsymbol{x}$ and $\boldsymbol{r}$. This expression now relates a local change in view ray position and direction to the first-order differential brightness structure of the plenoptic function.

We define the *plenoptic ray flow* $(d\boldsymbol{x}/dt, d\boldsymbol{r}/dt)$ as the difference in position and orientation between the two rays that are captured by the same imaging element at two consecutive time instants. This allows us to use the spatio-temporal brightness derivatives of the light rays captured by an imaging device to constrain the plenoptic ray flow. This generalizes the well-known *Image Brightness Constancy Constraint* to the *Plenoptic Brightness Constancy Constraint*:

$$\frac{d}{dt}\mathcal{L}(\boldsymbol{r};\boldsymbol{x};t) = \mathcal{L}_t + \nabla_{\boldsymbol{r}}\mathcal{L}^{\mathrm{T}}\frac{d\boldsymbol{r}}{dt} + \nabla_{\boldsymbol{x}}\mathcal{L}^{\mathrm{T}}\frac{d\boldsymbol{x}}{dt} = 0. \tag{6}$$

We assume that the imaging sensor can capture images at a rate that allows us to use the instantaneous approximation of the rotation matrix $R \approx I + [\boldsymbol{\omega}]_\times$ where $[\boldsymbol{\omega}]_\times$ is a skew-symmetric matrix parameterized by the axis of the instantaneous rotation $\boldsymbol{\omega}$. Now we can define the plenoptic ray flow for the ray captured by the imaging element located at location $\boldsymbol{x}$ and looking in direction $\boldsymbol{r}$ as

$$\frac{d\boldsymbol{r}}{dt} = \boldsymbol{\omega}\times\boldsymbol{r} \text{ and } \frac{d\boldsymbol{x}}{dt} = \boldsymbol{\omega}\times\boldsymbol{x} + \dot{\boldsymbol{q}} \tag{7}$$

where $\dot{\boldsymbol{q}} = d\boldsymbol{q}/dt$ is the instantaneous translation. As before in the discrete case (Eq.(4)), the plenoptic ray flow is completely specified by the six rigid motion parameters. This regular global structure of the rigid plenoptic ray flow makes the estimation of the differential rigid motion parameters very well-posed.

Combining Eqs. 6 and 7 leads to the *differential plenoptic motion constraint*

$$\begin{aligned}-\mathcal{L}_t &= \nabla_{\boldsymbol{x}}\mathcal{L}\cdot(\boldsymbol{\omega}\times\boldsymbol{x}+\dot{\boldsymbol{q}}) + \nabla_{\boldsymbol{r}}\mathcal{L}\cdot(\boldsymbol{\omega}\times\boldsymbol{r}) \\ &= \nabla_{\boldsymbol{x}}\mathcal{L}\cdot\dot{\boldsymbol{q}} + (\boldsymbol{x}\times\nabla_{\boldsymbol{x}}\mathcal{L} + \boldsymbol{r}\times\nabla_{\boldsymbol{r}}\mathcal{L})\cdot\boldsymbol{\omega}\end{aligned} \tag{8}$$

which is a linear, scene-independent constraint in the motion parameters and the plenoptic partial derivatives.

We see that we can utilize two different kinds of algorithms. For a single viewpoint camera, we can either make use of an image registration algorithms that finds a parametric mapping between two images (in case of pure rotation or planar scene) or we can use correspondences to solve a global bundle adjustment problem. In general for a single view camera the estimation of motion and structure are coupled and both have to be estimated simultaneously. In

the case of multiple views we have two options. Each measurement that a camera captures corresponds to a bundle of light rays in space, for any scene and motion we can find a rigid motion that maps one set of light rays into another. Thus any changes between light ray images depend only on the motion of the camera, similarly as it is the case for a rotating camera. Or we can utilize the multi-view information to compute approximate 3D information to improve the search for corresponding points and the chance to presegment the scene into different depth layers.

5 Feature computation in the space of light rays

To utilize the constraints described above we need to define the notion of correspondence in mathematical terms. In the case of the ray identity constraint we have to evaluate if two sets of light rays are identical. If the scene is static we can use the difference between the sets of light rays that are aligned according to the current motion estimate as our matching criterion. This criterion is integrated over all rays and we expect that at the correct solution, we will have a minimum. As with all registration algorithms, we need signals that contain enough information for the matching. The amount of information which corresponds to the amount of texture in the perspective images is often measured in terms of the eigenvalues of the structure tensor, that is the outer product of the local intensity gradients integrated over a local neighborhood [40,19]. This criterion can be extended to the space of light rays by examining the structure tensor of the plenoptic function. The plenoptic structure tensor $\nabla\mathcal{L}\nabla\mathcal{L}^{\mathrm{T}}$ can be computed from the intensity gradients of the plenoptic function with respect to view point $\nabla_x\mathcal{L}$ and view direction $\nabla_r\mathcal{L}$. By examining subspaces of the plenoptic structure tensor, we can determine what kind of features can be reliably computed. The structure tensor of the plenoptic function has a simple structure because we have the relationship that $\mathcal{Z}(\boldsymbol{x},\boldsymbol{r},t)\nabla_x\mathcal{L}(\boldsymbol{x},\boldsymbol{r},t) = \nabla_r\mathcal{L}(\boldsymbol{x},\boldsymbol{r},t)$ where $\mathcal{Z}(\boldsymbol{x},\boldsymbol{r},t)$ is again the depth to the scene from location $\boldsymbol{x}$ in direction $\boldsymbol{r}$ at time t.

The amount of texture of the scene can thus be measured by examining the sub-structure tensors $\nabla_x\mathcal{L}\nabla_x\mathcal{L}^{\mathrm{T}}$ and $\nabla_r\mathcal{L}\nabla_r\mathcal{L}^{\mathrm{T}}$ which correspond to the structure tensors of the perspective and orthographic images of the scene. As suggested in the literature before, we can use the inverse of the intensity Hessian as a measure for the variance of the estimated feature positions [40].

If we analyze the portion of the tensor formed by the image derivatives $\nabla_{\mathrm{EPI}}\mathcal{L}$ in the epipolar plane images (EPI) [7], we can analyze how much information we have to compute the depth of the scene. EPI images are two-dimensional subspaces of the plenoptic function where both view point and view direction are varying. We have $\nabla_{\mathrm{EPI}}\mathcal{L} = [\boldsymbol{m}^{\mathrm{T}}\nabla_x\mathcal{L}; \boldsymbol{m}^{\mathrm{T}}\nabla_r\mathcal{L}]$ where $\|\boldsymbol{m}\| = 1$ and $\boldsymbol{m}^{\mathrm{T}}\boldsymbol{r} =$

0. We will have a single non-zero eigenvalue for fronto-parallel planes (lines in the EPI), and two non-zero eigenvalues for depth discontinuities. To estimate the depth and the local shape we need to make differential measurements of the plenoptic function. The accuracy of the depth estimates depend on how accurately we can reconstruct the light field based on the samples captured by a multi-perspective camera. The main advantage of a polydioptric camera is now that geometric properties of the scene such as depth, surface slope, and occlusions can be easily computed by the intensity structure of the polydioptric images. This way we convert the correspondence problem into an interpolation problem [32].

5.1 Effect of the Spacing between View Points on the Motion Estimation

How does the camera spacing affect our ability to make measurements in polydioptric images? Given a camera spacing $\boldsymbol{\Delta}_x$ that is fixed by the design of the polydioptric camera and an estimate of the depth bounds $Z_{\min}$ and $Z_{\max}$, then we can reconstruct a low-pass filtered approximation to the light field that is not affected by aliasing [8]. To exclude the aliased parts of the plenoptic signal we can apply a low-pass filter with cut-off frequency $\boldsymbol{C}_u \leq \frac{2\pi}{f(1/z_{\min}-1/z_{\max})\boldsymbol{\Delta}_x}$ to the (u,v)-subspace of the light field. Since depth is a local property, if possible the low-pass filters should be tuned to the local depth structure in each light field neighborhood. Pyramid-based matching is a well-known and successful tool in the image registration community [42]. As described in [33], if we decompose the (u,v)-subspace of the light field into a pyramidal representation, then based on our knowledge of the depth bounds and camera spacing, we can choose only those levels of the pyramid for the matching that do not contain aliasing components.

In current stereo systems, we have usually much higher resolution in the perspective images, compared to the orthographic images where the view point changes and the view direction is constant. Up to occlusion effects, both images observe the same texture in the scene, thus the local orientation of the gradients in both type of images will be the same, while the ratio between the magnitude of the gradient with respect to view direction and the magnitude of the gradient with respect to view position are proportional to depth of the scene. Thus, we can easily utilize the high resolution orientation information in the perspective images to interpolate and interpret the lower-resolved orthographic images. For the effect of smoothing and interpolation on the motion estimation see Fig. 3.

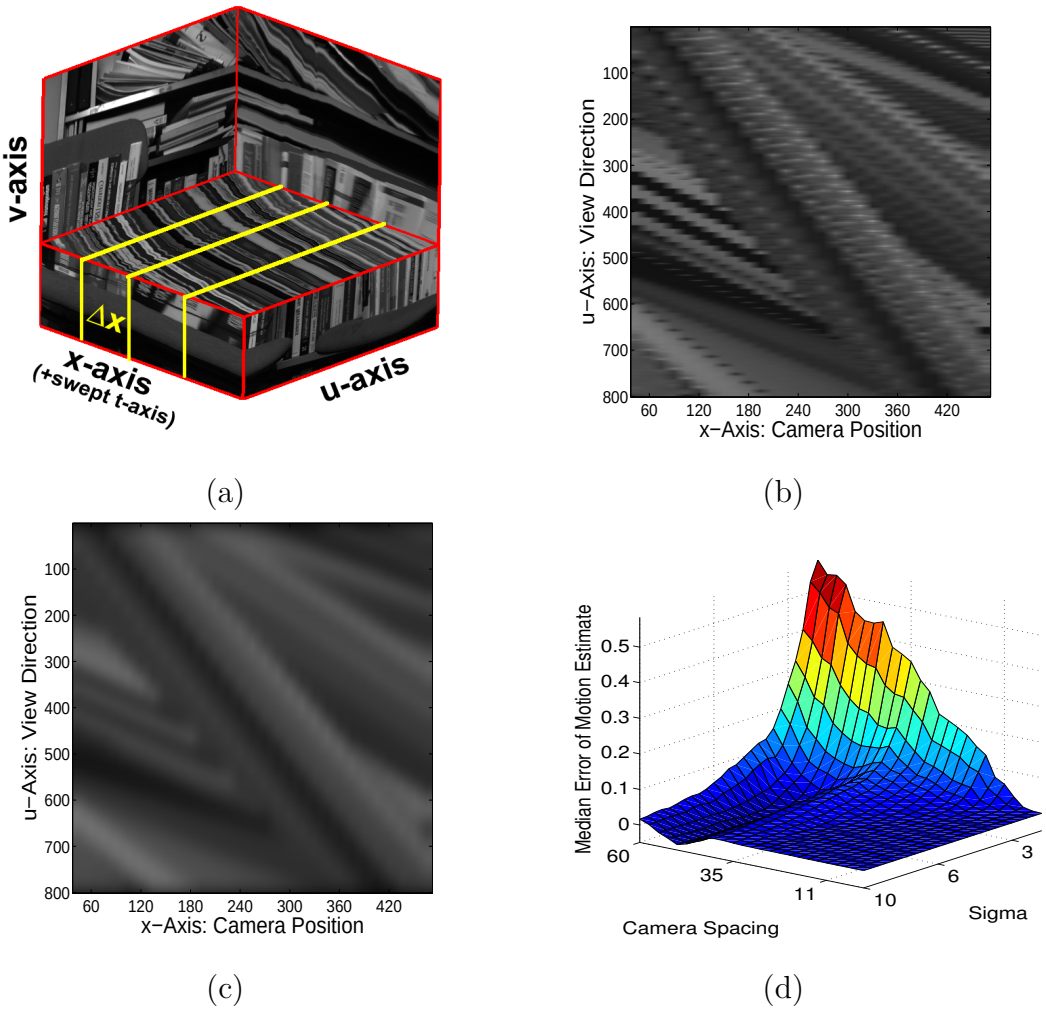


Fig. 3. (a) One of the epipolar volumes used in the experiments, the slices E_k are marked in bright yellow. (b) Aliasing effects in (x,u)-space (29 cameras spaced apart such that the average disparity is around 15 pixels). (c) Smoothing along the u-dimension reduces aliasing in (x,u)-space and makes differential motion estimation possible. (d) Relationship between smoothing along u-dimension (we vary the standard deviation of the Gaussian filter used in the smoothing) and the camera spacing (both variables are in pixel units). We see that the motion can be recovered accurately even for larger camera spacings if we filter the images along the u-dimension.

6 Hierarchy of Cameras for 3D Photography

Another important criteria for the sensitivity of the motion estimation problem is the size of the field of view (FOV) of the camera system. The basic understanding of these difficulties has attracted few investigators over the years [10,11,20,21,27,36]. These difficulties are based on the geometry of the problem and they exist in the cases of small and large baselines between the views, that is for the case of continuous motion as well as for the case of

discrete displacements of the cameras.

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 [15]. Catadioptric sensors could provide the field of view but they have poor resolution, making it difficult to recover shape models. Thus, in our lab we built the Argus eye [3,1], a construction consisting of six cameras pointing outwards. When this structure is moved arbitrarily in space, then 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, the motion estimation for this camera is more elaborate than for a spherical one, but because we know the geometric configuration between the cameras (from calibration) we can obtain all three translational parameters.

For every direction of translation we find the corresponding best rotation which minimizes deviation from a brightness-based constraint. Fig. 4 shows (on the sphere of possible translations) the residuals of the epipolar error color coded for each individual camera computed for a real image sequence that we captured with a multi-perspective camera setup, the "Argus Eye". Noting that the red areas are all the points within a small percentage of the minimum, we can see the valleys which clearly demonstrates the ambiguity theoretically shown in the proofs in the literature. In contrast, we see in Fig. 5

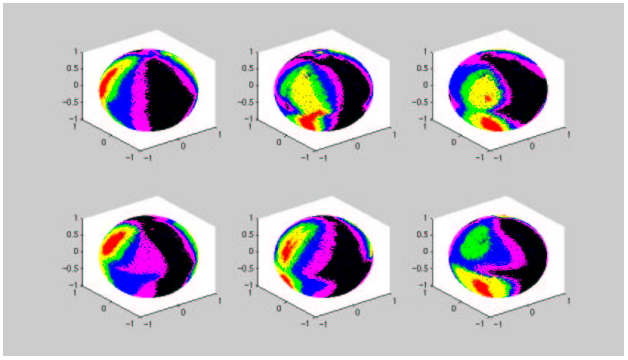


Fig. 4. Deviation from the epipolar constraints for motion estimation from individual cameras (from [1])

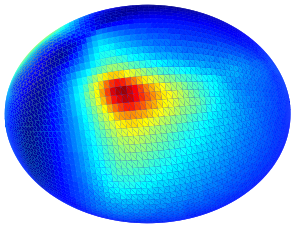
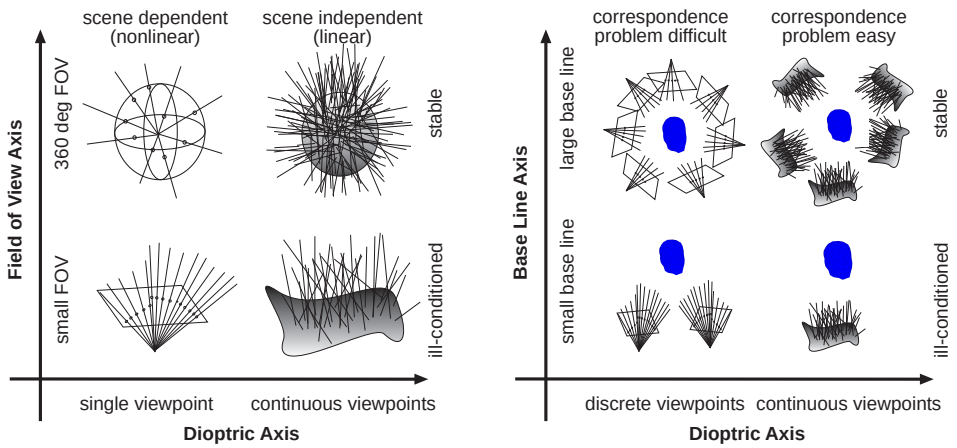


Fig. 5. Combination of error residuals for a six camera Argus eye (from [1])

a well-defined minimum (in red) when estimating 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.

Combining the two criteria, the field of view and the subset of the space of light rays that a sensor captures, we can rank different eye design in a hierarchy as shown in Fig. 6 which expresses a qualitative measure of how hard the task of motion estimation is to solve for a given sensor design [33].

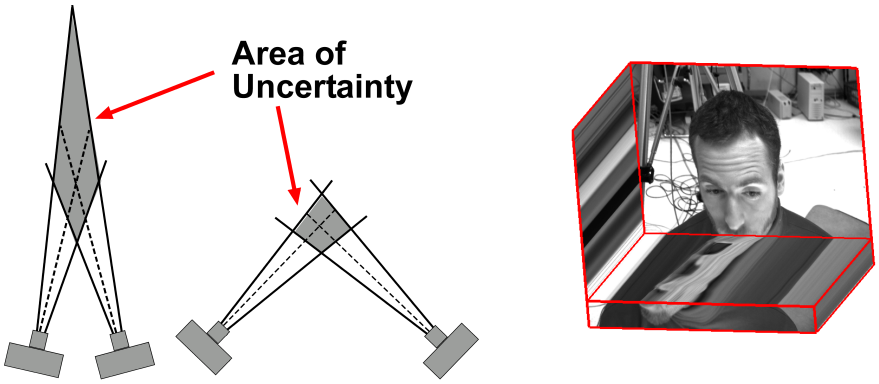


(a) (b)

Fig. 6. (a) Hierarchy of Cameras for 3D Motion Estimation and (b) 3D Shape estimation. The different camera models are classified according to the field of view (FOV) and the number and proximity of the different viewpoints that are captured (Dioptric Axis). The camera models are clockwise from the lower left: small FOV pinhole camera, spherical pinhole camera, spherical polydioptric camera, and small FOV polydioptric camera.

One can see in the figure that the conventional pinhole camera is at the bottom of the hierarchy because the small field of view makes the motion estimation ill-posed and it is necessary to estimate depth and motion simultaneously. Although the estimation of the 3D motion for a single-viewpoint spherical camera is stable and robust, it is still scene-dependent, and the algorithms which give the most accurate results are search techniques, and thus rather elaborate. One can conclude that a spherical polydioptric camera is the camera of choice to solve the 3D motion estimation problem since it combines the stability of full field of view motion estimation with the linearity and scene independence of the polydioptric motion estimation.

For 3D photography we need to reconstruct the scene structure from multiple views. How well this can be done depends mainly on our abilities to compute correspondences between the views and then how accurately we can triangulate the correct position of the scene points. If we have a polydioptric camera we can compute local shape estimates from the multiple small-baseline stereo systems which allows one to use shape invariants in addition to intensity invariants to find correspondences between different views, while a single view point camera has to rely completely on intensity information. The accuracy of the triangulation depends on the base line between the cameras, the larger the baseline the more robust the estimation as can be seen in Fig. 7. Based on these two criteria, we can also define a hierarchy of cameras for the 3D shape estimation problem.



(a) (b)

Fig. 7. (a) Triangulation-Correspondence Tradeoff in dependence on baseline between cameras. For a small baseline system, we can solve the correspondence problem easily, but have a high uncertainty in the depth structure. For a large baseline system this uncertainty is much reduced, but the correspondence problem is harder. (b) Motion and Shape can be reconstructed directly from feature traces in spatio-temporal and epipolar image volumes.

7 Sensitivity of motion and depth estimation using perturbation analysis

To assess the performance of different camera designs we have to make sure that the algorithms we use to estimate the motion and shape are comparable. In this work we will restrict our analysis to the case of instantaneous motion of the camera. We will compare a number of standard algorithms for egomotion estimation as described in [43] to solving a linear system based on the plenoptic motion flow equation Eq. (7). This linear system relates the plenoptic motion flow to the rigid motion parameters, that is the translation $\boldsymbol{q}$ and axis of rotation $\boldsymbol{\omega}$. Since multi-view information can be used to infer depth information, we will project the translational flow onto a spherical retina and account for the multi-view information by scaling the flow by an approximate inverse depth value. Then the flow for the ray $l_i = (\boldsymbol{x}_i, \boldsymbol{r}_i, t)$ is given by (we drop the argument and write $Z_i = \mathcal{Z}(\boldsymbol{x}_i, \boldsymbol{r}_i, t)$):

$$-\frac{1}{Z_i}[\boldsymbol{r}_i]_x^2 \boldsymbol{q} - (\frac{1}{Z_i}[\boldsymbol{r}_i]_x^2 [\boldsymbol{x}_i]_x + [\boldsymbol{r_i}]_x)\boldsymbol{\omega} = \dot{\boldsymbol{r}}_i \tag{9}$$

Given a multi-perspective camera system we can utilize the multi-view information to generate an approximate local depth map. The accuracy of these 3D measurements depends on the noise in our image measurements, the accuracy of the calibration of the camera, and the baseline between the views. In this section we will apply the ideas of stochastic perturbation theory [41] to analyze the influence of depth errors on the accuracy of instantaneous mo-

tion estimation. We will denote the inverse depth at a given point $\boldsymbol{P}_i$ by $D_i = 1/Z_i$. Since each measurement is scaled by the depth individually, we can combine the individual depth measurements to form the diagonal matrices $D = \text{diag}(D_1, \ldots, D_n)$ and $Z = \text{diag}(Z_1, \ldots, Z_n)$. If we know the inverse depths in D and the spherical flow $\dot{\boldsymbol{r}}$ in the images, then we can form a linear system of the form ($\boldsymbol{m} = [\boldsymbol{q}; \boldsymbol{\omega}]$):

$$A\boldsymbol{m} = \boldsymbol{b} \text{ that is } DA_z[\boldsymbol{q}; \boldsymbol{\omega}] + A_\omega \boldsymbol{\omega} = [\dot{\boldsymbol{r}}_1; \dot{\boldsymbol{r}}_2; \dot{\boldsymbol{r}}_3; ...] \tag{10}$$

where $\boldsymbol{b}$ contains the flow between frames. $A_z = [A_q, A_c]$ is formed by stacking the 3×3 matrices $A_{qi} := [\boldsymbol{r}_i]_x^2$ and $A_{ci} := [\boldsymbol{r}_i]_x^2 [\boldsymbol{x}_i]_x$ that contain the terms in Eq. (9) that are scaled by the depth, and (A_ω) is constructed by stacking the terms $[\boldsymbol{r_i}]_x$ that do not get scaled by the depth.

There are two sources for error in this system: the error in the inverse depth D_ϵ and the error in the computed optical flows $\dot{\boldsymbol{r}}_i$ which we stack to form the vector $\boldsymbol{b}$. Stochastic perturbation theory [41] allows us to analyze the effect of these errors on the motion estimate of the camera. We write the estimated inverse depth as $\tilde{D} = D + D_\epsilon$ and estimated spherical flow as $\tilde{\boldsymbol{b}} = \boldsymbol{b} + \boldsymbol{b}_\epsilon$. Then we can write the linear system including the error terms as:

$$(A + E)\boldsymbol{m} = (D + D_\epsilon)A_z\boldsymbol{m} + A_\omega\boldsymbol{\omega} = \boldsymbol{b} + \boldsymbol{b}_\epsilon = \tilde{\boldsymbol{b}} \tag{11}$$

We can characterize the error matrix in a probabilistic sense by writing it as a stochastic matrix [41] of the form

$$E := S_c H S_r = \hat{D}_\epsilon H A_z \tag{12}$$

where H is a diagonal stochastic matrix where the entries have zero mean and unit variance. The entries in H are multiplied from the left by the the inverse depth errors in the diagonal matrix $\hat{D}_\epsilon$ i to the correct size and multiplied from the right by A_z, the entries of the system matrix A that get scaled by the depth, to account for the correlations between the depth values and the motion parameters. depth values that are introduced while computing depth.

If we write $C = A^{\mathrm{T}}A$, then the the first order change in the solution is given by [41]:

$$\hat{\boldsymbol{m}} = \boldsymbol{m} - A^+ E\boldsymbol{m} + C^{-1}E^{\mathrm{T}}\boldsymbol{b}_\epsilon \tag{13}$$

The stochastic norm $\|A\|_s$ of a matrix A is defined as the expected Frobenius norm of the matrix that is $\|A\|_s = E[\|A\|_F] = E[\sqrt{\text{trace}(A^{\mathrm{T}}A)}]$. We can express the difference between the true and estimated motion parameters in terms of stochastic norms as:

$$\|\hat{\boldsymbol{m}} - \boldsymbol{m}\|_S = \sqrt{\|A^+ S_c\|_F^2 \|S_r\boldsymbol{m}\|^2 + \|S_c\boldsymbol{b}_\epsilon\|^2 \|S_r C^{-1}\|_F^2} \tag{14}$$

or for each component separately

$$\|\hat{\boldsymbol{m}}_i - \boldsymbol{m}_i\|_S = \sqrt{\|S_c A_i^+\|^2 \|S_r \boldsymbol{m}\|^2 + \|S_c \boldsymbol{b}_\epsilon\|^2 \|S_r C_i^{-1}\|^2} \tag{15}$$

where A_i^+ and C_i^{-1} are the i-th rows of the matrices A^+ and C^{-1}. Assuming that the errors in the stereo correspondences and the errors in the optical flow are identically independently distributed with variances σ_D and σ_b, then we can simplify the expressions for the error in the motion parameters. In this case we have that $S_c = \hat{D}_\epsilon = \sigma_D I$ and $S_r \boldsymbol{m} = A_z \boldsymbol{m}$ is the magnitude of the depth dependent flow for a scene of unit depth. The term $\|S_c \boldsymbol{b}_\epsilon\|^2 = \|\sigma_D I \boldsymbol{b}_\epsilon\|^2$ reduces to $\sigma_D^2 \sigma_b^2$. Then we can rewrite the expected error in the motion estimate as

$$\|\hat{\boldsymbol{m}} - \boldsymbol{m}\|_S = \sqrt{\sigma_D^2 \|C^{-1}(DA_z + A_\omega)^{\mathrm{T}}\|_F^2 \|A_z \boldsymbol{m}\|^2 + \sigma_D^2 \sigma_b^2 \|A_z C^{-1}\|_F^2} \tag{16}$$

It is to note that the matrix C^{-1} determines the sensitivity of the motion estimation to errors in the depth and in the flow. How much C^{-1} inflates the errors depends on the eigenvalue distribution of the matrix C which in turn is determined by the field of view of the camera. The larger the field of view, the smaller will be the condition number of C, and thus the error in the motion estimates will be less affected by errors in the depth and flow estimates.

8 Experimental Results

To assess the performance of different camera models with regard to motion estimation, we compare a number of standard algorithms for ego-motion estimation as described in [43] against a multi-camera stereo system. We used Jepson and Heeger's linear subspace algorithm and Kanatani's normalized minimization of the epipolar constraint. We assume similar error distributions for the optical flow and disparity distributions, both varying from 0 to 0.04 radians (0-2 degrees) in angular error. This corresponds to about a pixel error of 0 to 4.5 pixels in a 256x256 image. We ran each each algorithm 100 times on different randomly generated flow vectors and point clouds, and measured the angular deviation from the true translation and rotation. The results in Figure 8 demonstrate that for a similar distribution of errors in the disparities and the optical flow, solving a linear system with approximate depth knowledge outperforms the algorithms that algebraically eliminate the depth from the equation noticeably.

The effect of the camera on depth estimation, can similarly analyzed. To estimate the depth from a motion estimate, we can invert Eq. (10) pointwise to get:

$$Z = \frac{A_q \boldsymbol{q} + A_c \boldsymbol{\omega}}{\boldsymbol{b} - A_\omega \boldsymbol{\omega}} \tag{17}$$

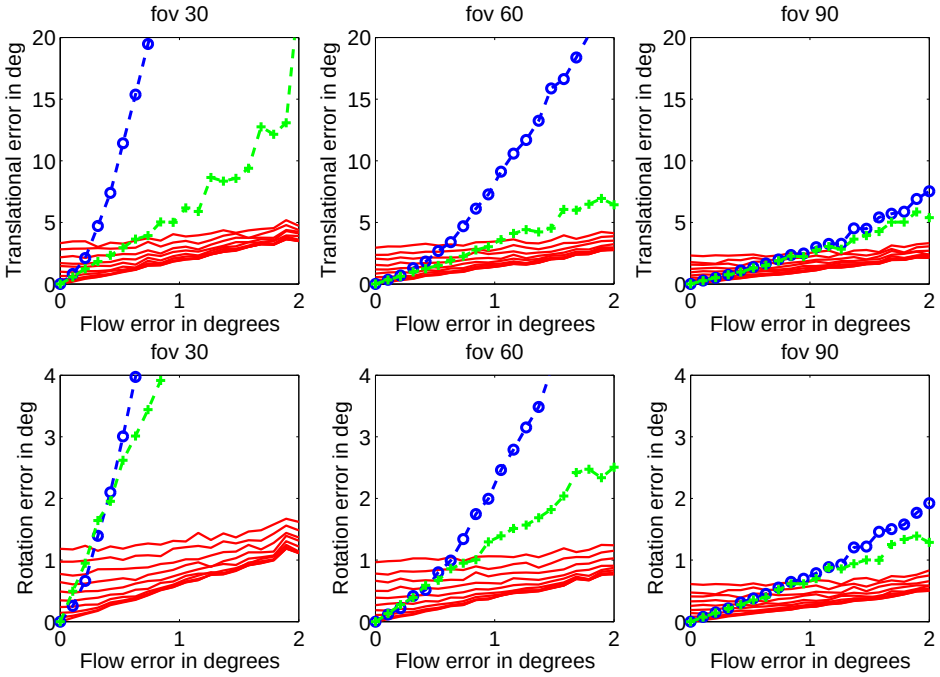
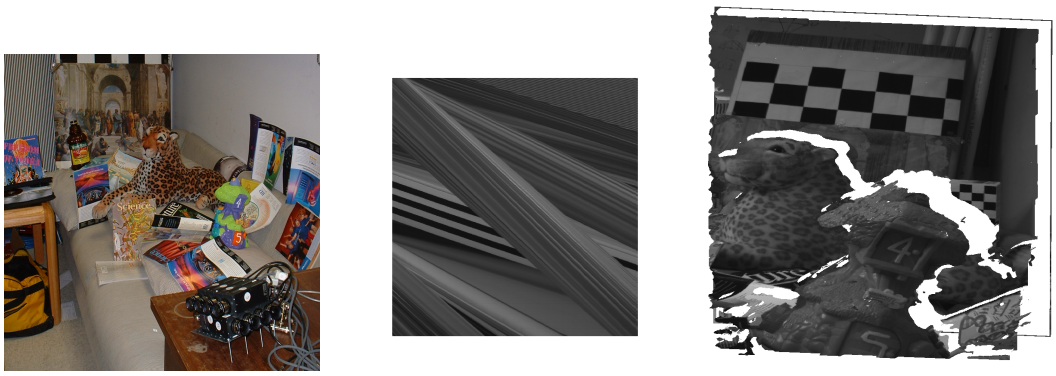


Fig. 8. Comparison of motion estimation using single and multi-perspective cameras. The errors in the correspondences varied between 0 and 0.04 radians (0-2 degrees), and we computed the mean squared error in the translation and rotation direction for 3 different camera field of views (30,60, and 90 degrees). The blue line (–o–) and green line (–+–) are Jepson-Heeger's subspace algorithm and Kanatani's normalized minimization of the instantaneous differential constraint as implemented by [43]. The red lines denote the performance of motion estimation using Eq. (10) where the errors in the disparities are normally distributed with a standard deviation that varies between 0 and 5 degrees.

For the case of stereo, we have $\boldsymbol{\omega} = 0$, and the equation reduces to $Z = \frac{\boldsymbol{r}_\perp^{\mathrm{T}}[\boldsymbol{r}]_\times^2 \boldsymbol{q}}{\boldsymbol{r}_\perp^{\mathrm{T}} \boldsymbol{b}}$. It has been observed before that if we have errors in the motion estimation, then the reconstructed shape will be distorted [4]. This necessitates the use of Kalman filters or sophisticated fusion algorithms. By using polydioptric cameras, we can solve the correspondence problem easily due to the small baseline, and at the same time, since we know the calibrated imaging geometry, we can estimate the local depth models with greater accuracy. This improves the motion estimation and allows for easier correspondence over larger baselines. Finally, stochastic fusion algorithms such as described in [9] can be used to integrate the local depth estimates.

We used the multi-perspective camera concepts described in this paper to recover shape from real world sequences. We built a polydioptric camera consisting of two linear arrays of cameras looking at perpendicular directions. This camera configuration was moved in a planar motion in front of an office scene (Fig. 9a). Using the simple linear algorithm described in [33], we estimated the motion based on the plenoptic derivatives and were able to com-



(a) (b) (c)

Fig. 9. (a) Example scene for motion estimation. The camera can be seen on the table. It was moved in a planar motion on the table. (b) The epipolar image that was recovered after compensating for varying translation and rotation of the camera. The straight lines indicate that the motion has been accurately recovered. (c) Recovered depth model.

(a) (b)

Fig. 10. (a) Camera Design for 3D Photography (each camera in the figure is a multi-perspective polydioptric camera formed by 3 conventional cameras in a small baseline trinocular stereo setup) (b) Renderings from animated computer graphics model.

pute accurately rectified epipolar image volumes (Fig. 9b). Finally, we used the recovered motion to segment the scene according to depth and recovered the position of the main objects in the scene (Fig. 9c).

For another example, to demonstrate the use of polydioptric cameras for 3D shape estimation we arranged polydioptric cameras around a human head and created an animated computer graphics model (Renderings of the 3D model from novel non-camera views can be seen in Fig.10b). Each of the polydioptric cameras consisted of a triple of conventional cameras in a trinocular small baseline stereo configuration. The 3D shape and motion of the head model were computed by integrating the multi-perspective image sequences captured by each polydioptric camera in a global optimization step that was adapted from [29].

9 Conclusion

According to ancient Greek mythology Argus, the hundred-eyed guardian of Hera, the goddess of Olympus, alone defeated a whole army of Cyclops, one-eyed giants. Inspired by the mythological power of many eyes we proposed in this paper a mathematical framework for the design of cameras used for 3D photography. Based on this framework we developed hierarchies of cameras for 3D motion and 3D shape estimation. We analyzed the structure of the space of light rays, and found that large field of view polydioptric cameras, these are generalized cameras which capture a multi-perspective subset of the plenoptic function, are best suited to robustly recover accurate estimates of motion and shape based on local image measurements.

Cameras nowadays become smaller and more affordable by the day, thus soon it will be in anyones reach to use polydioptric assemblies of cameras for the tasks they try to solve. Since these assemblies can be reconfigured easily, the design of novel polydioptric eyes will be possible with little effort. The main challenges that remain for the implementation of polydioptric cameras are the current size of the individual imaging elements that make up the polydioptric sensor. Since the surface of the camera is restricted to be two-dimensional and we are sampling a four-dimensional function, the spacing between the imaging elements will always be discontinuous along some dimensions. Thus it is paramount to develop small lens-sensor systems (MEMS, nano-technology) that address the optical imaging as well as the sensing problem. The smaller the imaging elements become, the more noise our measurements will contain due to the quantum nature of light. This necessitates intelligent sensor that adaptively combine information from neighboring measurements. We believe that the analysis of camera designs based on the structure of the space of light rays has great potential especially with advent of optical nano-technology around the corner which will offer new opportunities to design cameras that sample the space of light rays in ways unimaginable to us today.

Acknowledgment

The support through the National Science Foundation Award 0086075 is gratefully acknowledged.

References

[1] P. Baker, R. Pless, C. Fermuller, and Y. Aloimonos. A spherical eye from multiple cameras (makes better models of the world). In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[2] Patrick Baker and Yiannis Aloimonos. Structure from motion of parallel lines. In *Proc. Europ. Conf. Computer Vision*, volume 4, pages 229–240, 2004.

[3] Patrick Baker, Robert Pless, Cornelia Fermuller, and Yiannis Aloimonos. Camera networks for building shape models from video. In *Workshop on 3D Structure from Multiple Images of Large-scale Environments (SMILE 2000)*, 2000.

[4] Gregory Baratoff and Yiannis Aloimonos. Changes in surface convexity and topology caused by distortions of stereoscopic visual space. In *Proc. European Conference on Computer Vision*, volume 2, pages 226–240, 1998.

[5] P.J. Besl. Active optical range imaging sensors. *Machine Vision Appl.*, 1(2):127–152, 1988.

[6] F. Blais. A review of 20 years of ranges sensor development. In *Videometrics VII Proceedings of SPIE-IST Electronic Imaging*, volume 5013, pages 62–76, 2003.

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

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

[9] Amit Roy Chowdhury and Rama Chellappa. Stochastic approximation and rate-distortion analysis for robust structure and motion estimation. *International Journal of Computer Vision*, 55(1):27–53, October 2003.

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

[11] K. Daniilidis and M. Spetsakis. Understanding noise sensitivity in structure from motion. In *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, chapter 4, pages 61–88. Lawrence Erlbaum Associates, Hillsdale, NJ, 1997.

[12] James Davis, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *2003 Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, pages 359–366, June 2003.

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

[14] A. Edelman, T. Arian, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 1998.

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

[16] P. Fua. Regularized bundle-adjustment to model heads from image sequences without calibration data. *International Journal of Computer Vision*, 38:153–171, 2000.

[17] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[18] F. Huck, C. Fales, and Z. Rahman. *Visual Communication*. Kluwer, Boston, 1997.

[19] B. Jähne, J. Haussecker, and P. Geissler, editors. *Handbook on Computer Vision and Applications*. Academic Press, Boston, 1999.

[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] Jana Kosecka, Yi Ma, and Shankar S. Sastry. Optimization criteria, sensitivity and robustness of motion and structure estimation. In *Vision Algorithms Workshop, ICCV*, 1999.

[22] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38:199–218, 2000.

[23] Y. Ma, K. Huang, R. Vidal, J. Kosecka, and S. Sastry. Rank condition on the multiple view matrix. *International Journal of Computer Vision*, (2), 2004.

[24] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003.

[25] Yi Ma, Jana Kosecka, and Shankar S. Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 44(3):219–249, 2001.

[26] Yi Ma, Rene Vidal, Shawn Hsu, and Shankar S. Sastry. Optimal motion from multiple views by normalized epipolar constraints. *Communications in Information and Systems*, 1(1), 2001.

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

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

[29] J. Neumann and Y. Aloimonos. Spatio-temporal stereo using multi-resolution subdivision surfaces. *International Journal of Computer Vision*, 47(1/2/3):181–193, 2002.

[30] J. Neumann and C. Fermüller. Plenoptic video geometry. *Visual Computer*, 19(6):395–404, 2003.

[31] J. Neumann, C. Fermüller, and Y. Aloimonos. Eyes from eyes: New cameras for structure from motion. In *IEEE Workshop on Omnidirectional Vision 2002*, pages 19–26, 2002.

[32] J. Neumann, C. Fermüller, and Y. Aloimonos. Eye design in the plenoptic space of light rays. In *Proc. International Conference on Computer Vision*, volume 2, pages 1160–1167, 2003.

[33] J. Neumann, C. Fermüller, and Y. Aloimonos. Polydioptric camera design and 3d motion estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 294–301, 2003.

[34] B. Newhall. Photosculpture. *Image*, 7(5):100–105, 1958.

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

[36] J. Oliensis. The error surface for structure from motion. Neci tr, NEC, 2001.

[37] Marc Pollefeys, Reinhard Koch, and Luc J. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *ICCV*, pages 90–95, 1998.

[38] M Rioux. Laser range finder based on synchronized scanners. *Applied Optics*, 23(21):3837–3844, 1984.

[39] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):17–42, 2002.

[40] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 593 – 600, 1994.

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

[42] P. Thévenaz, U.E. Ruttimann, and M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing*, 7(1):27–41, January 1998.

[43] T. Tian, C. Tomasi, and D. Heeger. Comparison of approaches to egomotion computation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 315–320, June 1996.

[44] L. Zhang, B. Curless, and S. M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 367–374, 2003.