Estimating Self-Motion with a Spherical Camera versus Pinhole Camera

Florian Raudies
Center for Computational
Neuroscience and Neural Technology
Center of Excellence for Learning in
Education, Science, and Technology
677 Beacon Street, Boston MA 02215
fraudies@bu.edu

ABSTRACT
Taking inspiration from the eyeball we propose a spherical camera model for the estimation of optic flow and self-motion. Our contribution is the definition of a spherical optic flow model using symmetric spherical coordinates, which allows for the formulation of a linear method estimating self-motion from optic flow. We compare our method with a method that estimates self-motion from optic flow for a pinhole camera. Estimates from both camera models are comparable while varying the zoom, spatial sampling, field of view, or temporal sampling. Selecting 80% of the best optic flow estimates gave better estimates than taking all optic flow for both camera models. For same parameter choices of zoom, sampling, or field of view the spherical camera model gave better estimates of optic flow and self-motion than the pinhole camera model. For the task of estimating self-motion a spherical camera performs no worse than a pinhole camera and provides a wider field of view – an advantage not only to the estimation of self-motion but also for tasks such as visual search, obstacle avoidance, or visual navigation.

Categories and Subject Descriptors
F.2.2 [**Analysis of algorithms and problem complexity**]: Nonnumerical Algorithms and Problems –*geometric problems and computations.*

General Terms
Algorithms, Performance, Experimentation.

Keywords
Spherical camera; optic flow; self-motion estimation; pinhole camera; linear method; eyeball.

1. INTRODUCTION
The semiconductor industry continually increases pixel numbers in consumer products while industrial cameras increase frame rate and pixel numbers alike, e.g. achieving 1,328 × 1,048 pixels at 120 frames per second [19]. However, for estimating self-motion the shape of the sensor and field of view rather than sampling is more important. Self-motion estimation and estimating structure from motion improves for a spherical camera with a large field of view because this larger field of view disambiguates translational from rotational motion [1]. For this reason many methods map images taken from cameras in a rig [2, 12, 14] or from a cardionotopic camera [11] into a unified spherical image space.

We are interested how an ideal spherical camera compares to a pinhole camera. We refer to this camera as ideal because it cannot be manufactured and we generated synthetic images using a ray-tracer [22]. In our comparison we focused on the task of self-motion estimation. To elaborate the comparison between the two cameras we were interested which camera parameters such as zoom, sampling, or field of view to choose for an optimal estimate.

Several methods have been proposed for the estimation of self-motion from mapped or analytical, spherical optic flow. One method applies pattern matching to search for the translational and rotational vector [10]. Another method uses the positive distance constraint to estimate self-motion [23]. Another method simplifies the search space by assuming self-motion while fixating on one point. Projections onto latitudinal direction w.r.t. two different poles allows for two 1D searches along the meridians. One 1D search estimates the translational direction and torsion and the other 1D search estimates the angle between the translational and fixation axis together with the time to collision toward the fixated point [9]. Another method uses catadioptric images. A detection algorithm identifies the catadioptric lines used to estimate the rotational motion. A two-point algorithm estimates the translational motion [3]. Another method assumes only rotational motion and can tolerate small amounts of translation. This method uses the spherical Fourier transform to detect rotational motion by a shift in the spherical spectrum [18]. Another method uses antipodal points to decouple the translation from the rotation. One pair of antipodal points constrains the epipole to lie in a plane. Another pair of points defines another plane. The intersection of these two planes constrains the epipole [15].

Unlike all these previous methods we propose an algebraic method to estimate self-motion from spherical optic flow. Key of our method is to interpret six products of linear and rotational velocities as auxiliary variables to define a quadratic problem, which is solved efficiently.

BICT 2014, December 01-03, Boston, United States

DOI 10.4108/icst.bict.2014.257814

We continue this article with the Methods Section to describe all methods used in our simulations. In the Results Section we interpret the data from four simulations. We finish with the Conclusion Section to answer our introductory questions.

2. METHODS

We describe all the involved parts of our simulation framework starting with a description of the scene, movement trajectory, and the rendering of images. Next, we describe the model assumptions for the estimation of optic flow from frame pairs. We use this estimated optic flow to estimate the camera's self-motion. For this estimation process we briefly recap methods used for the pinhole camera model. Then, we derive our method for spherical optic flow. Last, we define error measures for the estimated optic flow and estimated self-motion.

2.1 Scene, Trajectory, and Rendering

The model camera moves inside a square box that is 2 m × 2 m and 1.5 m high. The camera is 20 cm above the ground and its optical axis is kept parallel to the ground. The camera orbits around the center of the box on a circular track with a radius of r = 75 cm, while keeping the optical axis tangent to the circular track. In the default configuration it takes 1,000 frames to go full circle. This gives a linear speed of 4.71 mm/frame and a rotational yaw speed of 21.6'/frame. The ' indicates arc minutes or 1/60°. The walls, ground, and ceiling are textured. For the rendering we used a ray-tracer that supports a pinhole and a spherical camera model [22]. We rendered 51 images on the circular trajectory, covering only a fraction of the entire circle because of the symmetry of the scene and trajectory. We used a curvilinear path as self-motion because this motion type is common to humans, many ground-dwelling animals, and many robots.

2.2 Estimation of Optic Flow

For the estimation of optic flow we used a published method [27]. In a nutshell this methods uses the gray-value constancy assumption and flow smoothness assumption. Gray-value constancy contributes all change in the gray-value of a pixel between successive frames to the optic flow. Flow smoothness assumes that neighboring flow vectors in the image have similar direction and speed. These two assumptions are used to derive two constraints. Each constraint is embedded into an L1-norm. Two sets of optic flow variables are used, one for each constraint. Corresponding optic flow variables from each set are linked through a square-distance norm. The decoupling of variables per constraint allows for the optimization of the optic flow variables through an alteration of two steps: The solution of the primal and dual problem. As parameters for this estimation method we use the scale factor 0.9 between levels of the image pyramid, one warping step, λ = 50, and 50 iterations to alter between the primal and dual problem. This method uses a frame pair to estimate the optic flow.

As criterion for the selection of the "best" flow estimates we use a metric, which includes the gray-value assumption and smoothness assumption to measure the goodness of the estimated optic flow field [4]. To achieve sparseness in the optic flow we use a percentage of the "best" flow estimates as defined by this metric.

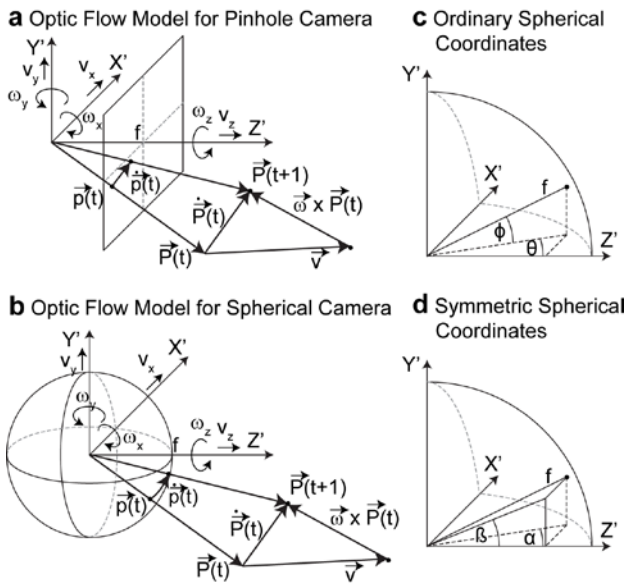

Figure 1. Shows model definitions. In detail it shows the model of optic flow for the (a) pinhole camera model and (b) spherical camera model. (c) Ordinary spherical coordinates are defined through the longitude angle *θ*, the latitude angle *ϕ*, and radius *f*. (d) We define symmetric spherical coordinates through the azimuth angle *α*, the elevation angle *β*, and radius *f*.

2.3 Estimation of Self-Motion

We assume that the self-motion is defined by the instantaneous 3D linear velocity $\vec{v} = (v_x, v_y, v_z)^t$ and the instantaneous 3D rotational velocity $\vec{\omega} = (\omega_x, \omega_y, \omega_z)^t$ (Fig. 1a and 1b). We use the super-script *t* to indicate the vector transpose or matrix transpose. For both velocities the camera coordinate system serves as a reference. These self-motion velocities lead to the velocity $\dot{\vec{P}} = (\dot{X}, \dot{Y}, \dot{Z})^t = \vec{v} + \vec{\omega} \times \vec{P}$ given a sample point $\vec{P} = (X, Y, Z)^t$ as seen by the camera. When this point $\vec{P} = (X, Y, Z)^t$ is projected onto the image, we denote the projected point by $\vec{p} = (x, y, z)^t$ and its corresponding image velocity by $\dot{\vec{p}} = (\dot{x}, \dot{y}, \dot{z})^t$.

2.3.1 Pinhole camera model

For the pinhole camera model we assume the focal length *f*, then a projected point is given by $\vec{p} = f / Z \times \vec{P}$. Taking the partial derivative of this formulation and plugging in the assumption about instantaneous self-motion we get the optic flow model [16]:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} + \frac{1}{f} \begin{pmatrix} x \cdot y & -(f^2 + x^2) & f \cdot y \\ (f^2 + y^2) & -x \cdot y & -f \cdot x \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad . \qquad (1)$$

A simple least-square formulation using this optic flow model from Eq. (1) leads to a nonlinear estimation problem, due to the nonlinear combination of the unknown depth values *Z* with the

linear velocity vector $\vec{v}$. Many approaches have been proposed to solve this estimation problem [6, 21]. For comparison with our proposed method estimating self-motion, which uses a spherical camera model, we use a similar method for a pinhole camera [20].

2.3.2 Spherical camera model

In our first spherical camera model we use ordinary spherical coordinates defined through a longitude angle θ, latitude angle ϕ, and radius f (Fig. 1c):

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = f \cdot \begin{pmatrix} \sin(\theta)\cdot\cos(\phi) \\ \sin(\phi) \\ \cos(\theta)\cdot\cos(\phi) \end{pmatrix} \qquad \begin{pmatrix} \theta \\ \phi \end{pmatrix} = \begin{pmatrix} \arctan(\frac{x}{z}) \\ \arctan(\frac{y}{\sqrt{x^2+z^2}}) \end{pmatrix} \quad (2)$$

We derive the optic flow model for ordinary spherical coordinates by first computing the differentials for the longitude and latitude angle w.r.t. the 3D Cartesian coordinates:

$$\begin{pmatrix} d\theta \\ d\phi \end{pmatrix} = \frac{1}{f\cdot\cos(\phi)} \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ -\sin(\theta)\cdot\sin(\phi)\cdot\cos(\phi) & \cos^2(\phi) & -\sin(\phi)\cdot\cos(\theta)\cdot\cos(\phi) \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \quad (3)$$

Furthermore, the spherical projection can be formulated by dividing through the distance D of a point and multiplying by the focal length f of the spherical camera, which gives

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{f}{D} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad \text{with} \quad D = \sqrt{X^2+Y^2+Z^2}\,. \quad (4)$$

Assuming that the focal length is time-constant and all other variables depend on time we get the temporal derivatives

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \frac{f}{D} \left(D^2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} X^2 & XY & XZ \\ XY & Y^2 & YZ \\ XZ & YZ & Z^2 \end{pmatrix} \right) \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix}. \quad (5)$$

When plugging in the spherical projection function from Eq. (4), the ordinary spherical coordinates from Eq. (2), and the differentials from Eq. (3), we get the optic flow model equation for ordinary spherical coordinates [7]:

$$\vec{\Psi} = \frac{1}{D}\cdot A\vec{v} + B\vec{\omega} \qquad \vec{\Psi} = \begin{pmatrix} \dot{\theta} \\ \dot{\phi} \end{pmatrix} \qquad D = \sqrt{X^2+Y^2+Z^2}$$

$$A = \begin{pmatrix} -\frac{\cos(\theta)}{\cos(\phi)} & 0 & \frac{\sin(\theta)}{\cos(\phi)} \\ \sin(\theta)\cdot\sin(\phi) & -\cos(\phi) & \cos(\theta)\cdot\sin(\phi) \end{pmatrix},$$

$$B = \begin{pmatrix} \frac{\sin(\theta)\cdot\sin(\phi)}{\cos(\phi)} & -1 & \frac{\cos(\theta)\cdot\sin(\phi)}{\cos(\phi)} \\ \cos(\theta) & 0 & -\sin(\theta) \end{pmatrix}. \quad (6)$$

For the method of estimating self-motion it will be necessary that the matrix $B^tA^\perp$ is symmetric whereas $A^\perp$ is defined by swapping the rows in the matrix A and multiplying the 2nd row by (-1). For the matrices A and B the matrix $B^tA^\perp$ is not symmetric. This led us to the use of symmetric spherical coordinates with an azimuth angle α, elevation angle β, and radius f (Fig. 1d):

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{\pm f}{L} \cdot \begin{pmatrix} \tan(\alpha) \\ \tan(\beta) \\ 1 \end{pmatrix} \quad \text{with} \quad L = \sqrt{\tan^2(\alpha)+\tan^2(\beta)+1}\,. \quad (7)$$

We transform ordinary spherical coordinates into symmetric spherical coordinates using

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \theta \\ \arctan\left(\frac{\sin(\phi)}{\cos(\theta)\cos(\phi)}\right) \end{pmatrix} \quad (8)$$

and the differentials

$$\begin{pmatrix} d\alpha \\ d\beta \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{\sin(\theta)\sin(\phi)\cos(\phi)}{\cos^2(\theta)\cos^2(\phi)+\sin^2(\phi)} & \frac{\cos(\theta)}{\cos^2(\theta)\cos^2(\phi)+\sin^2(\phi)} \end{pmatrix} \begin{pmatrix} d\theta \\ d\phi \end{pmatrix}. \quad (9)$$

Using this transform we convert the above optic flow Eq. (6) into the optic flow model for symmetric spherical coordinates:

$$\vec{\Phi} = \frac{1}{D}\cdot E\vec{v} + F\vec{\omega} \qquad \vec{\Phi} = \begin{pmatrix} (\tan^2(\alpha)+1)\cdot\dot{\alpha} \\ (\tan^2(\beta)+1)\cdot\dot{\beta} \end{pmatrix}$$

$$E = \begin{pmatrix} -1 & 0 & \tan(\alpha) \\ 0 & -1 & \tan(\beta) \end{pmatrix}$$

$$F = \begin{pmatrix} \tan(\alpha)\cdot\tan(\beta) & -(1+\tan^2(\alpha)) & \tan(\beta) \\ (1+\tan^2(\beta)) & -\tan(\alpha)\cdot\tan(\beta) & -\tan(\alpha) \end{pmatrix}. \quad (10)$$

2.3.3 Optimization problem for spherical camera

We use this model to estimate the self-motion. A least squares problem formulation for the discretized flow is

$$\min_{\vec{v},\vec{\omega},Z_i} \sum_{i=1}^{n} \left\| \hat{\vec{\Phi}}_i - \frac{1}{D_i} E_i\vec{v} - F_i\vec{\omega} \right\|^2 \quad (11)$$

with $\hat{\vec{\Phi}}_i$ being the estimated optic flow and index i indicating the n samples of flow vectors in the image plane. For instance, for the sampling with 90 × 60 pixels $n = 540$. Optimizing only for the distance D_i gives the bilinear optic flow constraint [5]. Using a weighted version of this bilinear constraint we re-formulate the optimization problem as

$$\min_{\vec{v},\vec{\omega}} \sum_{i=1}^{n} \left(\underbrace{\hat{\vec{\Phi}}_i^t E_i^\perp}_{=\vec{m}}\vec{v} - \vec{\omega}\underbrace{F_i^t E_i^\perp}_{=G_i}\vec{v} \right)^2 \quad E_i = \begin{pmatrix} 0 & -1 & \tan(\beta_i) \\ 1 & 0 & -\tan(\alpha_i) \end{pmatrix}. \quad (12)$$

The weighting introduces a statistical bias, which can be removed through the technique of pre-whitening [17]. This optimization problem is linear in the rotational velocity $\vec{\omega}$ assuming a fixed translational velocity $\vec{v}$ and it is linear in the linear velocity $\vec{v}$ assuming a fixed rotational velocity $\vec{\omega}$. To formulate the quadratic optimization problem

$$\min_{\vec{v},\vec{k}} \sum_{i=1}^{n} (\vec{m}_i^t\vec{v} - \vec{g}_i^t\vec{k})^2 \quad (13)$$

we introduce the vector with six auxiliary variables $\vec{k} = \left(v_x\omega_x, v_y\omega_x, v_z\omega_x, v_y\omega_y, v_z\omega_y, v_z\omega_z\right)$ and the corresponding six linear independent coefficients $g_{i,1} = 1+\tan^2(\beta_i)$, $g_{i,2} = -\tan(\alpha_i)\tan(\beta_i)$, $g_{i,3} = -\tan(\alpha_i)$, $g_{i,4} = 1+\tan^2(\alpha_i)$, $g_{i,5} = -\tan(\beta_i)$, and $g_{i,6} = \tan^2(\alpha_i)+\tan^2(\beta_i)$. This problem is first solved for the auxiliary variables $\vec{k}$ depending on the unknowns $\vec{v}$ and $\vec{\omega}$. The result of this optimization is then plugged into Eq. (13). This gives

$\min_{\vec{v}} \vec{v}^t \cdot (\sum_{i=1}^{n} \vec{l}_i \, \vec{l}_i^{\,t}) \cdot \vec{v}$ with

$$\vec{l}_i^{\,t} = \vec{m}_i^t - \vec{g}_i^t \cdot (\sum_{j=1}^{n} \vec{g}_j \vec{g}_j^t)^{-1} \cdot (\sum_{j=1}^{n} \vec{g}_j \vec{m}_j^t) \quad i = 1...n \cdot \tag{14}$$

This Eq. (14) uses outer products and matrix vector products. The optimization of Eq. (14) leads to a homogenous linear equation system, which has a non-trivial solution in case of optimal data. But measured data is noisy and a robust solution is calculated solving an eigenvalue problem. The eigenvector corresponding to the smallest eigenvalue determines the direction of least variance and, thus, the most likely $\vec{v}$. We denote this solution by $\vec{v}*$ and plug this solution into Eq. (13). Then, we solve for the rotational velocity $\vec{\omega}$. This gives

$$(\vec{\omega}^*)^t = \sum_{j=1}^{n} (\underbrace{\sum_{i=1}^{n} ((\vec{v}^*)^t G_i^t)(G_i \vec{v}^*))^{-1} \cdot ((\vec{v}^*)^t G_i^t)}_{\text{pseudo inverse}} \cdot \underbrace{(\vec{m}_j^t \vec{v}^*)}_{\text{scalar}}$$

$$\text{with } G_i = \begin{pmatrix} g_{i,1} & g_{i,2} & g_{i,3} \\ g_{i,2} & g_{i,4} & g_{i,5} \\ g_{i,3} & g_{i,5} & g_{i,6} \end{pmatrix}. \tag{15}$$

2.4 Error Measures for Optic Flow

We denote the estimated optic flow components by $\dot{x}_i^{est}$ and $\dot{y}_i^{est}$ and the corresponding ground-truth optic flow components by $\dot{x}_i^{gt}$ and $\dot{y}_i^{gt}$ for the pinhole camera model. All definitions are analogous for the spherical camera model. Given these two discretized flow fields we compute several errors characterizing the difference between these flow fields. The first error measure is the absolute speed difference in pixels per frame:

$$s_{err} = \frac{1}{n} \sum_{i=1}^{n} \left| \sqrt{(\dot{x}_i^{gt})^2 + (\dot{y}_i^{gt})^2} - \sqrt{(\dot{x}_i^{est})^2 + (\dot{y}_i^{est})^2} \right| . \tag{16}$$

We compute the mean over all sampled flow vectors indexed by *i*. The second error measure is the angular difference between the ground-truth and estimated optic flow in radians:

$$\varphi_{err} = \frac{1}{n} \sum_{i=1}^{n} \arccos\left(\frac{\dot{x}_i^{gt} \times \dot{x}_i^{est} + \dot{y}_i^{gt} \times \dot{y}_i^{est}}{\sqrt{(\dot{x}_i^{gt})^2 + (\dot{y}_i^{gt})^2} \times \sqrt{(\dot{x}_i^{est})^2 + (\dot{y}_i^{est})^2}} \right) . \tag{17}$$

The third error measure is based on the length of the residual vector between the ground-truth and estimated optic flow. The mean length of this residual vector is

$$r_{err} = \frac{1}{n} \sum_{i=1}^{n} \sqrt{(\dot{x}_i^{gt} - \dot{x}_i^{est})^2 + (\dot{y}_i^{gt} - \dot{y}_i^{est})^2} \cdot \tag{18}$$

The length of the residual vector has the unit pixels per frame. The fourth error measure is called the endpoint error [6] measuring the angular difference of the flow vectors in 3D:

$$\delta_{err} = \frac{1}{n} \sum_{i=1}^{n} \arccos\left(\frac{\dot{x}_i^{gt} \times \dot{x}_i^{est} + \dot{y}_i^{gt} \times \dot{y}_i^{est} + 1}{\sqrt{(\dot{x}_i^{gt})^2 + (\dot{y}_i^{gt})^2 + 1} \times \sqrt{(\dot{x}_i^{est})^2 + (\dot{y}_i^{est})^2 + 1}} \right) . \tag{19}$$

Because of the non-linear interaction between speed and angular error this measure is more difficult to interpret but offers the benefit of being scalar.

2.5 Error Measures for Self-Motion

We denote the estimated self-motion by the 3D translational velocity $\vec{v}^{est}$ and the 3D rotational velocity $\vec{\omega}^{est}$ and the corresponding ground-truth values by $\vec{v}^{gt}$ and $\vec{\omega}^{gt}$, respectively. Because of the scaling ambiguity between the translational speed and the absolute depth or distance, we compute an error based on the direction of the translational velocity. For translations we compute the angular difference in radians:

$$v_{err} = \arccos\left(\frac{(v_x^{est} \times v_x^{gt}) + (v_y^{est} \times v_y^{gt}) + (v_z^{est} \times v_z^{gt})}{\sqrt{(v_x^{est})^2 + (v_y^{est})^2 + (v_z^{est})^2} \times \sqrt{(v_x^{gt})^2 + (v_y^{gt})^2 + (v_z^{gt})^2}} \right) . \tag{20}$$

Similarly, we compute the angular difference between the ground-truth rotational axis and the estimated rotational axis. This difference is defined by the angle in radians:

$$\rho_{err} = \arccos\left(\frac{(\omega_x^{est} \times \omega_x^{gt}) + (\omega_y^{est} \times \omega_y^{gt}) + (\omega_z^{est} \times \omega_z^{gt})}{\sqrt{(\omega_x^{est})^2 + (\omega_y^{est})^2 + (\omega_z^{est})^2} \times \sqrt{(\omega_x^{gt})^2 + (\omega_y^{gt})^2 + (\omega_z^{gt})^2}} \right) . \tag{21}$$

As a third error measure we define the absolute speed difference of the rotational ground-truth velocity and the rotational estimated velocity in radians per frame:

$$\zeta_{err} = \left| \sqrt{(\omega_x^{est})^2 + (\omega_y^{est})^2 + (\omega_z^{est})^2} - \sqrt{(\omega_x^{gt})^2 + (\omega_y^{gt})^2 + (\omega_z^{gt})^2} \right| \cdot \tag{22}$$

3. RESULTS

We vary parameters for both the pinhole camera and spherical camera to find optimal values for the estimation of self-motion. Variable parameters are the camera zoom, spatial sampling, field of view, and temporal sampling. We also report mean runtimes for variable spatial samplings. The source code for the methods estimating self-motion is available on the author's webpage at http://cns.bu.edu/~fraudies/SourceCodes/SourceCodes.html.

3.1 A Detailed Example

We illustrate our simulation framework by showing results from all steps involved. With our ray-tracer we rendered images for a pinhole camera model (Fig. 2a and 2b) or a spherical camera model (Fig. 2g and 2h). Given the image coordinates of the camera model, the distance *D* or depth *Z* from the ray-tracer, and the parameters of self-motion we computed analytical optic flow using Eq. 1 and Eq. 6 (Fig. 2c and 2i). Using two successive frames we estimated optic flow for both camera models (Fig. 2d and 2j). The estimated and analytical optic flow fields match closely (compare Fig. 2c with 2d and compare Fig. 2i with 2j). Both pairs of panels refer to the first frame pair in the sequence of 51 frames or 50 estimates for optic flow and self-motion. To give an impression about the error for the 50 estimates for self-motion we plotted ground-truth and estimated angles for longitude and latitude (Fig. 2e and 2k) and we plotted ground-truth and estimated rotational speeds for pitch (x-axis), yaw (y-axis), and roll (z-axis) (Fig. 2f and 2l). Per frame the error varies and for this reason we report the statistics of errors for self-motion estimation in the following simulations.

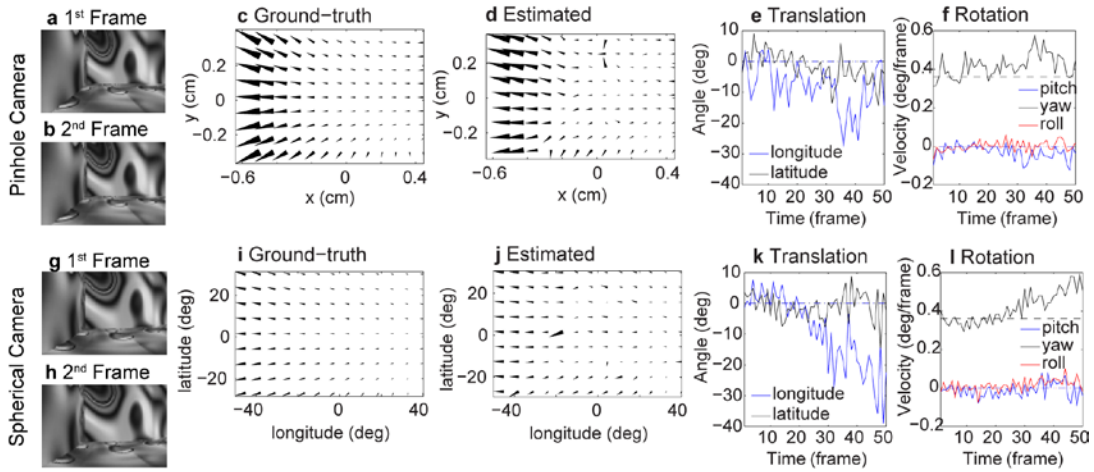

Figure 2. Illustrates the steps of our simulation framework. (a-b) 1st and 2nd frame rendered using the pinhole camera model. (c) Analytical, ground-truth optic flow for the pinhole camera model. (d) Estimated optic flow using the 1st and 2nd frame. (e) Ground-truth (dash-dotted line) and estimated (solid line) longitude and latitude of the translational motion vector. (f) Ground-truth (dash-dotted line) and estimated (solid line) pitch, yaw, and roll rotational velocities. (g-h) 1st and 2nd frame rendered using the spherical camera model. (i) Analytical, ground-truth optic flow calculated using the spherical camera model. (j) Estimated optic flow using the 1st and 2nd frame. (k) Translational error and (l) rotational error with the same encoding as in panels (e) and (f). Notice that flow fields are sampled by factor 7 and scaled by the factor 0.1.

3.2 Simulation I: Zoom

In this simulation we varied the zoom using values of 30° × 20°, 60° × 40°, 90° × 60°, 120° × 80°, and 150° × 100° while keeping the spatial sampling constant at 90 × 60 pixels. With increased zoom out the angular error for the estimation of optic flow increases from ≈ 4° to ≈ 16° (Fig. 3a). The endpoint error plateaus around 90° × 60° with a value of ≈ 4.5°. This plateauing happens because the endpoint error depends on the speed difference between ground-truth and estimates besides depending on the angular difference between ground-truth and estimate [6]. To separate angular and speed error we report the speed error and residual error. For the pinhole camera the speed error and residual error follows a U-shaped curve with its minimum for a zoom of 60° × 40° (Fig. 3b – the red and green boxes). For the spherical camera speed and residual error decrease with an increasing zoom out, being lowest for the zoom 150° × 100° (Fig. 3b – the magenta and blue boxes). Self-motion estimates become worse when zooming out (Fig. 3c). For the zoom 30° × 20° angular errors for the translational axis are ≈ 3° (Fig. 3c – red and blue boxes) and for the rotational axis these errors are ≈ 5° (Fig. 3c – the green and magenta boxes). For the rotational speed difference, the difference in length of the rotational axes, the error is ≈ 0.02°/frame for the zoom 30° × 20° (Fig. 3d). For the large zooms of 120° × 80° and 150° × 100° the angular error of estimates of self-motion are smaller for the spherical camera model than those for the pinhole camera model (Fig. 3c) and for the speed error this is the case only for 120° × 80° (Fig. 3d). Overall, large zoom-outs produce large errors in the estimation of optic flow and self-motion and estimates using the spherical camera are equally good or better than those for the pinhole camera.

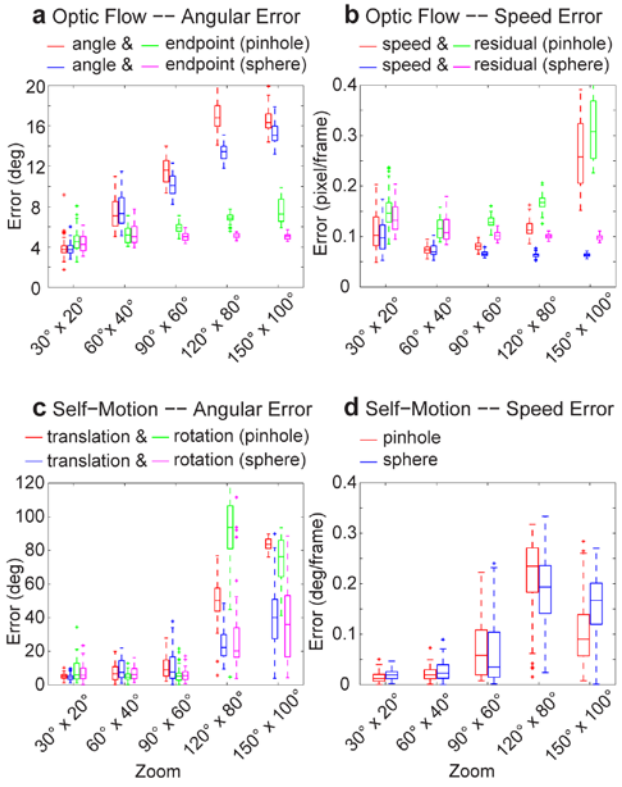

Figure 3. Shows an increase in error for large zoom out. (a) Angular error φ_{err} and endpoint error δ_{err} of optic flow estimates for variable zoom. (b) Speed error s_{err} and residual error r_{err} of optic flow estimates. (c) Angular error for the estimated translation axis v_{err} and rotation axis ρ_{err} of the self-motion. (d) Speed error for the estimated rotational self-motion ζ_{err}. In all boxplots whiskers extend the 25% or 75% percentile by 1.5, crosses denote outliers above or below the whiskers, and the solid, horizontal line within the box denotes the median or 50% percentile.

3.3 Simulation II: Spatial Sampling

In this simulation we varied the spatial sampling with values of 30 × 20 pixels, 60 × 40 pixels, 90 × 60 pixels, 120 × 80 pixels, 150 × 100 pixels, 180 × 120 pixels, and 240 × 160 pixels while keeping the zoom fixed at 90° × 60°. The angular error and endpoint error for the estimation of optic flow decreases with increased spatial

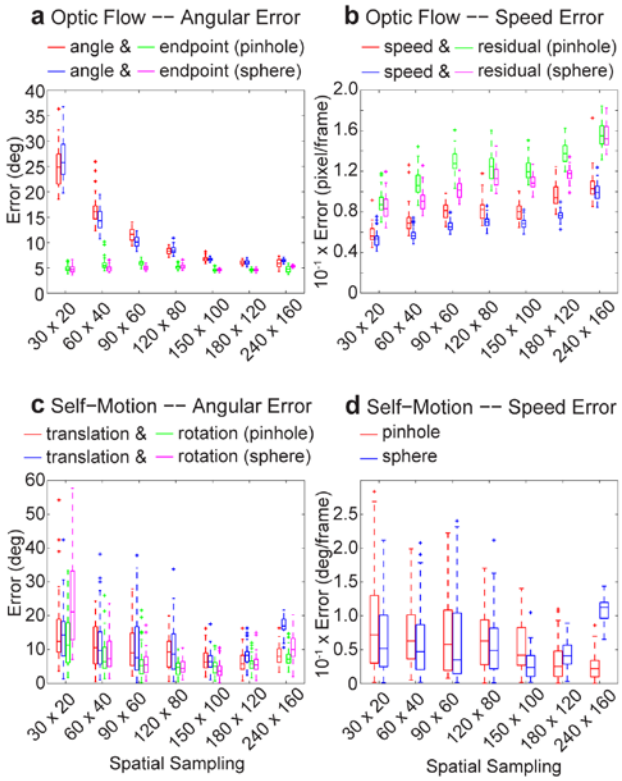

Figure 4. Shows a higher spatial sampling is not always better for self-motion estimation. (a) Angular and endpoint error of optic flow for variable spatial sampling. (b) Speed and residual error. (c) Angular error for translation and rotation. (d) Speed error for rotation. The caption of Fig. 3 describes the parameters of boxplots.

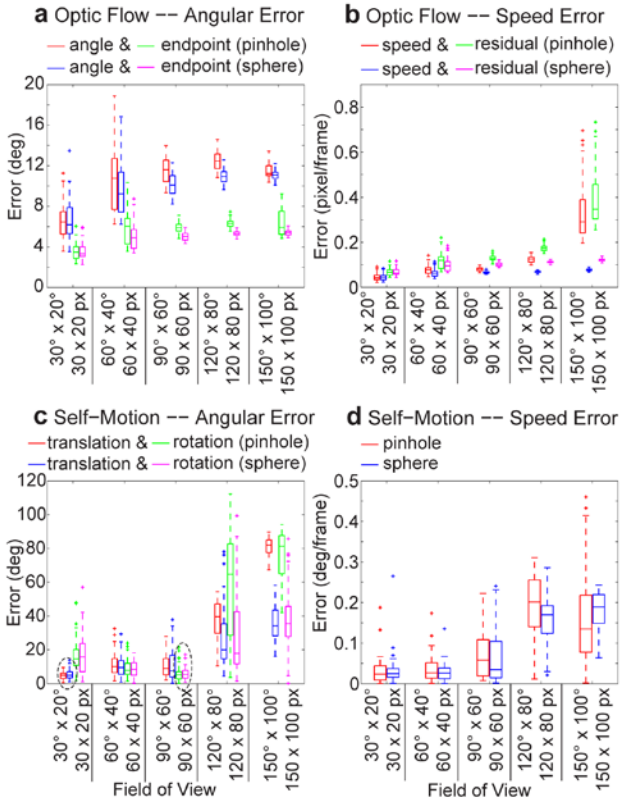

Figure 5. For a large field of view with large zoom out errors in the estimated optic flow and errors in the estimated self-motion are large. (a) Angular and endpoint error of optic flow for variable field of views. (b) Speed and residual error. (c) Angular error for translation and rotation. (d) Speed error for rotation. The caption of Fig. 3 describes the parameters of boxplots.

sampling (Fig. 4a). In contrast the speed and residual error for the estimation of optic flow increases with increased spatial sampling (Fig. 4b). Speed errors of optic flow estimates are lower for the spherical camera model compared to those for the pinhole camera model (Fig. 4b – green vs. magenta and red vs. blue). The decrease in angular error but increase in speed error leads to a U-shaped curve in the error for the estimation of self-motion. The angular errors for the translation and rotation axes are lowest for the spatial sampling 150 × 100 pixels (Fig. 4c). The speed error for the rotational self-motion is lowest for the spatial sampling 240 × 160 pixels for the pinhole camera (Fig. 4d – red boxes) and is lowest for the spatial sampling 150 × 100 pixels for the spherical camera (Fig. 4d – blue boxes). A small error for low spatial samplings is opposite to the trend of increasing the spatial sampling in modern sensors to values as high as 5,184 × 3,456 pixels [8]. Our results show that for self-motion estimation a low spatial sampling rate of ≈ 5/3 pixels per degree of visual field angle gives the best estimation results. Another advantage of a low spatial resolution is the time saving in computing optic flow.

3.4 Simulation III: Field of View

While the previous two simulations I and II have varied either the zoom or sampling rate in this simulation III we vary both simultaneously keeping a constant sampling rate of one pixel per degree. We used zooms of 30° × 20°, 60° × 40°, 90° × 60°, 120° × 80°, and 150° × 100° together with the spatial sampling rates of 30 × 20 pixels, 60 × 40 pixels, 90 × 60 pixels, 120 × 80 pixels, and 150 × 100 pixels. For the estimation of optic flow the angular error increases from ≈ 6° to ≈ 10° for the angular difference between estimated and ground-truth flow vectors (Fig. 5a – red and blue boxes) and from ≈ 3.5° to ≈ 5° for the endpoint error (Fig. 5a – green and magenta boxes). The speed and residual error increases with an increased field of view (Fig. 5b). For self-motion estimation the angular error for the axis of translational is lowest for a field of view of 30° × 20° (Fig. 5c – red and blue boxes). The angular error for the axis of rotation is lowest for a field of view of 90° × 60° (Fig. 5c – green and magenta boxes). A field of view beyond 90° × 60° produces large image and flow distortions for the pinhole camera model and subsequently leads to large estimation errors in self-motion (Fig. 5c and 5d). Errors in the rotation speed increase with an increased field of view (Fig. 5d). In sum, a larger field of view increases the accuracy in estimating the orientation of the rotational axis while a smaller field of view increases the accuracy in estimating the orientation of the translational axis.

3.5 Simulation IV: Temporal Sampling

We explored the temporal sampling rate by varying the samples in the circular track with values m of 100, 200, 500, 1000, 2000, 5000, and 10,000, which gives the translational speed $s = 2\pi \times r / m$ and the rotational speed $\omega = 2\pi / m$. The angular error of the estimated optic flow vectors increases with an

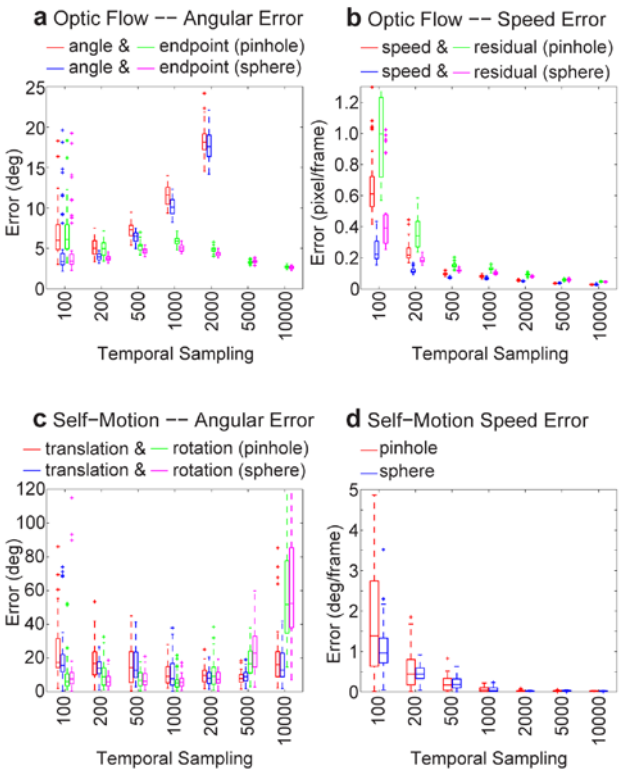

Figure 6. Shows that the temporal sampling rate has to be adapted to the spatial resolution and field of view. (a) Angular and endpoint error of optic flow for variable temporal sampling rates – errors for 5,000 and 10,000 samples are above the interval. (b) Speed and residual error. (c) Angular error for translation and rotation. (d) Speed error for rotation. The caption of Fig. 3 describes the parameters of boxplots.

increasing number of samples (Fig. 6a – red and blue boxes), while the endpoint error has an inverted U-shape with its maximum at 1,000 samples (Fig. 6 – green and magenta boxes). The speed error and residual error of optic flow estimates decreases with an increasing number of temporal samples (Fig. 6b). This increase in speed error is partly due to the decrease in the length of flow vectors. There is a trade-off between angular error and speed error for optic flow vectors and the temporal sampling rate. Low sample rates have small angular errors and high sample rates have small speed errors. But what is the optimal temporal sampling rate for the estimation of self-motion? In our example, this optimum appears for 2,000 samples where the errors for the translation and rotation axes are ≈ 10° (Fig. 6c). The error for the rotational speed decreases with an increased number of samples (Fig. 6d). In sum, for an increased temporal sampling rate the absolute speed error for optic flow and self-motion decreases at the cost of an increase in accuracy in angular direction. For self-motion estimation lowest errors for the angle of the translational and rotational axis appear for 2,000 samples. This corresponds to a translational speed of 2.36 mm/frame and a rotational speed of 10.8'/frame.

3.6 Simulation V: Sparseness

In this simulation we took the best performing cases from the four prior simulations. For "zoom" the selected best case has 30° × 20° zoom sampled by 90 × 60 pixels. For the "spatial sampling" the selected best case has 90° × 60° zoom sampled by 150 × 100 pixels. For the "field of view" the selected best case has 60° × 40° zoom sampled by 60 × 40 pixels. In these three best cases the temporal sampling was 1,000. For the "temporal sampling" the selected best case has the parameters 2,000 samples, 90° × 60° zoom, and 90 × 60 pixels spatial sampling. For these four cases we compared the estimate of self-motion using all optic flow vectors versus an estimate of self-motion that takes the 80% best estimates of optic flow vectors. For cases "zoom" and "temporal sampling" there is no clear reduction in translational error when using 80% of all flow vectors (Fig. 7a – 1st and 3rd group) while for the cases "spatial sampling" and "temporal sampling" (Fig. 7a – 2nd and 4th group) there is a clear reduction in translational error when using 80% of all flow vectors compared to using 100% of flow vectors. The angular error for the rotational axis has no clear reduction when using 80% of all flow vectors compared to using 100% of all flow vectors (Fig. 7b). But the error for the rotational speed is reduced again for the two cases "spatial sampling" and "temporal sampling" when using 80% of flow vectors compared to a 100% of flow vectors (Fig. 7c). In sum, sparseness of the optic flow generated by selecting the 80% of best flow estimates helps to improve the estimation error for self-motion in some cases; and never yielded worse results compared to using all available flow vectors.

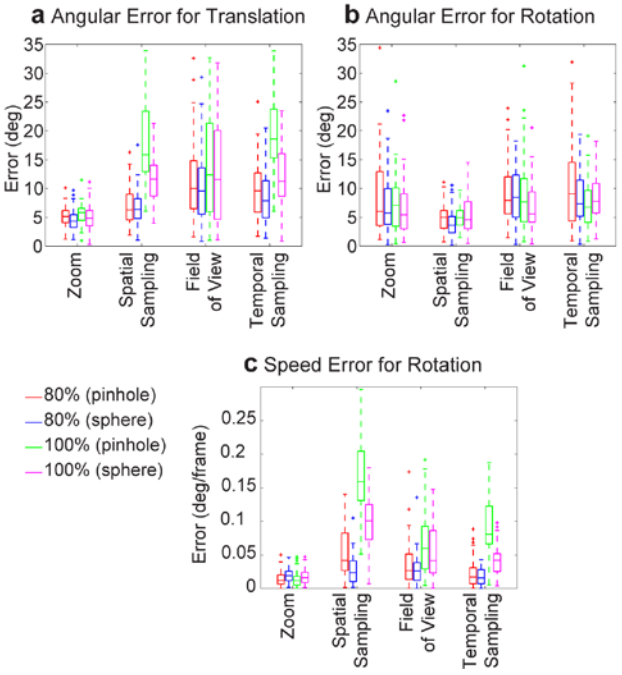

Figure 7. Shows that taking the 80% best optic flow estimates improves self-motion estimates. (a) Angular error between the estimated axis of translation and ground-truth axis of translation. (b) Angular error for the rotational axes. (c) Rotational speed error. The caption of Fig. 3 describes the parameters of boxplots.

3.7 Simulation VI: Runtimes

We measure runtimes per frame for variable spatial samplings using the pinhole camera model and spherical camera model. Implementations of the ray-tracer, optic flow estimation, and self-motion estimation are run in Matlab 7.12 (R2011a) on a Windows 7 Professional with an Intel Core i5 CPU M 520 @ 2.4 GHz and 4 GBy of RAM. We measure runtime using the Matlab function `cputime`. We compute the mean runtime for 50 repeated measurements corresponding to the 50 sample points of

simulation II. Because of the short runtime for self-motion estimation we use 1,000 repetitions for each sample point.

Table 1 lists runtimes for the pinhole camera model. Most of the runtime is spent with ray-tracing and estimation of optic flow. These two computational tasks take about the same time. The estimation of self-motion is about 1,000× faster. For a realistic robotic platform the task of ray-tracing is replaced by some initial image processing, e.g. filtering to suppress noise, which can be done much faster than the ray-tracing.

Table 1. Runtimes for calculations using a pinhole camera.

Sampling	Ray-tracer (sec)	Optic Flow (sec)	Self-Motion (msec)
30 × 20 pixels	0.153	0.174	0.330
60 × 40 pixels	0.736	0.739	1.017
90 × 60 pixels	1.474	1.608	1.892
120 × 80 pixels	2.521	2.561	3.290
150 × 100 pixels	3.905	4.182	5.192
180 × 120 pixels	5.593	6.561	7.859
240 × 160 pixels	10.227	14.575	15.961

Table 2 lists the runtimes for the spherical camera model. The runtime for the ray-trace and optic flow estimation is very similar to that of the pinhole camera. The runtime for the self-motion estimation is larger than that for the pinhole camera (compare Table 1 – last column – with Table 2 – last column). This difference in runtime is explained by the evaluation of trigonometric functions not necessary for the pinhole camera. In addition, the ray-tracer uses ordinary spherical coordinates, which have to be converted into symmetric spherical coordinates. This conversion is included in the runtime for the estimating self-motion. To avoid these "overheads" in runtime the ray-tracer could use symmetric spherical coordinates and the trigonometric functions for image locations can be pre-computed. In general, we see that the estimation of self-motion from optic flow is not the bottleneck in the processing pipeline; rather it is the computation of optic flow that takes most of the runtime.

Table 2. Runtimes for calculations using a spherical camera.

Sampling	Ray-tracer (sec)	Optic Flow (sec)	Self-Motion (msec)
30 × 20 pixels	0.262	0.179	1.103
60 × 40 pixels	0.753	0.748	3.276
90 × 60 pixels	1.523	1.662	6.712
120 × 80 pixels	2.588	2.671	12.273
150 × 100 pixels	4.081	4.422	19.980
180 × 120 pixels	5.703	6.689	29.675
240 × 160 pixels	10.290	14.785	54.611

4. CONCLUSION

We proposed an algebraic method for the estimation of self-motion from spherical optic flow. This method is based on the bilinear optic flow constraint [5] and interprets products of linear and rotational velocity components as auxiliary variables, similar to [20, 26]. An efficient, algebraic optimization of the bilinear constraint equation with six such variables was only possible when using symmetric spherical coordinates.

Our simulation results suggest avoiding large zoom-outs when estimating self-motion. A fairly low spatial resolution of ≈ 5/3 pixels per degree of visual field suffices. A field of view of 90° × 60° proved best for the estimation of self-motion given our box environment. A temporal sampling rate that led to translational speeds of ≈ 2 mm/frame and rotational yaw speed of 10'/frame were best. The translational speed depends on the distance or depth of the sample points, which was small with being smaller than 1 m in our setup. When assuming a translational speed of 1 m/sec in a 2 m × 2 m enclosure, then we would require a sample rate of about 500 frames per second to match the 2 mm/frame translational speed. This temporal sample rate is higher than commercial cameras offer, but a typical scene would be larger than the 2 m × 2 m leading to slower image speeds due to the larger distance values (compare with Eq. (1) and Eq. (10)). Selecting 80% of the best optic flow estimates improved the estimates of self-motion estimation for optimal values of spatial sampling and field of view.

In our simulations, the spherical camera model had better optic flow and self-motion estimates compared to the pinhole camera model for high temporal sampling (more than 2,000 samples) or large field of views (larger or equal than 120° × 80°), which is similar to the visual field of the human eye [13]. In all remaining cases there was no clear enough difference in estimates between the two camera models.

The best optic flow estimate did not always lead to the best self-motion estimate (e.g. compare Fig. 3a with Fig. 3c). Thus, we conclude that besides the four error measures for optic flow other indicators such as the spatial distribution of errors is important for a small error in the estimation of self-motion.

We suggest that the spherical camera will become part of panoramic cameras and sensor devices of the future. For a spherical camera peripheral distortions present in a pinhole camera model are absent. Furthermore, the human eyeball has proven capable as powerful frontend for cognitive tasks and high resolving power, e.g. hyper acuity [24] attributed to a non-uniform and overlapping arrangement of receptors on the retina.

Future work will focus on the use of space-variant sampling schemes for the estimation of self-motion. Such schemes put more emphasize on the peripheral vision for the estimation of self-motion and have been suggested to avoid independently moving objects that appear in the central part of the visual field [25].

5. ACKNOWLEDGMENTS

This work was supported in part by CELEST, a National Science Foundation Science of Learning Center under grant NSF SMA-0835976, by the grant NSF BCS-1147440, and the Office of Naval Research by the grant ONR MURI N00014-10-1-0936.

6. REFERENCES

[1] Adiv, G. (1985) Inherent ambiguities in recovering 3D motion and structure from a noisy flow field. In Proceedings of the Conference on Computer Vision and Pattern Recognition, 70—77.

[2] Baker, P., Pless, R., Fermüller, C., and Aloimonos, Y. (2000). New eyes for shape and motion estimation. In Proceedings of the Conference on Biologically Motivated Computer Vision, 118—128.

[3] Bazin, J.C., Demonceaux, C., Vasseur, P., and Kweon, I.S. (2010). Motion estimation by decoupling rotation and translation in catadioptric vision. Computer Vision and Image Understanding 114, 254—273.

[4] Bruhn, A. (2001). Regularization in motion estimation. Diploma Thesis in the Department of Mathematics and Computer Science at the University of Mannheim, Germany.

[5] Bruss, A. and Horn, B.K.P. (1983). Passive navigation. Computer Vision, Graphics, and Image Processing 21, 3—20.

[6] Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. International Journal of Computer Vision 12(1), 43—77.

[7] Calow, D., Krüger, N., and Wörgötter, F. (2004). Statistics of optic flow for self-motion through natural scenes. In Ilg, U.J., Blüthoff, H.H., Mallot, H.A. (eds.) Proceedings of Dynamic Perception, IOS Press, Akademische Verlagsgesellschaft Aka GmBH, Berlin, 133—138.

[8] Canon, 2014. EOS Rebel T3i DSLR Camera data sheet. http://www.usa.canon.com/cusa/support/consumer/eos_slr_camera_systems/eos_digital_slr_cameras/eos_rebel_t3i_18_55mm_is_ii_lens_kit#Specifications accessed 08/04/2014.

[9] Daniilidis, K. (1997). Fixation simplifies 3D motion estimation. Computer Vision and Image Understanding 68(2), 158—169.

[10] Fermüller, C. and Aloimonos, Y. (1995) Direct perception of three-dimensional motion from patterns of visual motion. Science 270, 1973—1976.

[11] Gluckman, J. and Nayar, S.K. (1998) Ego-motion and omnidirectional cameras. In Proceedings of the Conference on Computer Vision, 999—1005.

[12] Hui, T.-W. and Chung, R. (2013). Determining motion directly from normal flows upon the use of a spherical eye platform. In Proceedings of the Conference on Computer Vision an Pattern Recognition 2267—2274.

[13] Human Eye. Encyclopaedia Britannica, http://www.britannica.com/EBchecked/topic/1688997/human-eye accessed 10/09/2014.

[14] Kim, J.-S., Hwangbo, M., and Kanade, T. (2010). Spherical approximation for multiple cameras in motion estimation: its applicability and advantages. Computer Vision and Image Understanding 114, 1068—1083.

[15] Lim, J. and Barnes, N. (2007) Estimation of the epipole using optical flow at antipodal points. In Proceedings of the International Conference on Computer Vision, 1—6.

[16] H.C. Longuet-Higgins and K. Prazdny (1980). The interpretation of a moving retinal image. Proceedings of the Royal Society of London, Series B, Biology Sciences 208, 385—397.

[17] MacLean, W. (1999). Removal of translation bias when using subspace methods. International Conference on Computer Vision, pages 753—758.

[18] Makadia, A. and Daniilidis, K. (2003) Direct 3D-rotation estimation from spherical images via a generalized shift theorem. In Proceedings of Computer Vision and Pattern Recognition, 217—224.

[19] PointGrey (2014). Product catalogue for common industrial standard CCD cameras from PointGrey. http://ww2.ptgrey.com/PTGREY_Complete_Catalog.pdf accessed 08/04/2014.

[20] Raudies, F. and Neumann, H. (2009). An efficient linear method for the estimation of ego-motion from optical flow. In J. Denzler, G. Notni, and H. Süße (Eds.): DAGM 2009, LNCS 5748, 11—20.

[21] Raudies, F. and Neumann, H. (2012) A Review and Evaluation of Methods Estimating Ego-Motion. Computer Vision and Image Understanding 116, 606—633.

[22] Raudies, F. (2014). Ray-tracer implemented in Matlab. https://code.google.com/p/matlab-ray-tracer/ accessed 08/04/2014.

[23] Sinclair, D., Blake, A. and Murray, D. (1994). Robust estimation of egomotion from normal flow. International Journal of Computer Vision 13, 57—69.

[24] Westheimer, G. and McKee, S. P. (1977). Spatial configurations for visual hyperacuity. Vision Research 17, 941—947.

[25] Wagner, R.E. (2004). Stereo active vision and peripheral optic flow: computer vision applications of the wide-field human visual representation. PhD Thesis in the Graduate School for Arts and Sciences at Boston University, USA.

[26] Zhuang, X., Huang, T., Ahuja, N., and Haralick, R. (1988). A simplified linear optic flow algorithm. Computer Vision, Graphics and Image Processing 42, 334—344.

[27] Zach, C. Pock, T., and Bischof, H. (2007). A duality based approach for realtime TV-L1 optical flow. In Proceedings of the 29th DAGM conference on Pattern recognition, Fred A. Hamprecht, Christoph Schnörr, and Bernd Jähne (Eds.). Springer-Verlag, Berlin, Heidelberg, 214—223.