3D Scene Flow from 4D Light Field Gradients

Sizhuo Ma, Brandon M. Smith, and Mohit Gupta

Department of Computer Sciences, University of Wisconsin-Madison, USA
{sizhuoma,bmsmith,mohitg}@cs.wisc.edu

Abstract. This paper presents novel techniques for recovering 3D dense scene flow, based on differential analysis of 4D light fields. The key enabling result is a per-ray linear equation, called the ray flow equation, that relates 3D scene flow to 4D light field gradients. The ray flow equation is invariant to 3D scene structure and applicable to a general class of scenes, but is underconstrained (3 unknowns per equation). Thus, additional constraints must be imposed to recover motion. We develop two families of scene flow algorithms by leveraging the structural similarity between ray flow and optical flow equations: local 'Lucas-Kanade' ray flow and global 'Horn-Schunck' ray flow, inspired by corresponding optical flow methods. We also develop a combined local-global method by utilizing the correspondence structure in the light fields. We demonstrate high precision 3D scene flow recovery for a wide range of scenarios, including rotation and non-rigid motion. We analyze the theoretical and practical performance limits of the proposed techniques via the light field structure tensor, a 3×3 matrix that encodes the local structure of light fields. We envision that the proposed analysis and algorithms will lead to design of future light-field cameras that are optimized for motion sensing, in addition to depth sensing.

1 Introduction

The ability to measure dense 3D scene motion has numerous applications, including robot navigation, human-computer interfaces and augmented reality. Imagine a head-mounted camera tracking the 3D motion of hands for manipulation of objects in a virtual environment, or a social robot trying to determine a person's level of engagement from subtle body movements. These applications require precise measurement of per-pixel 3D scene motion, also known as scene flow [31]. In this paper, we present a novel approach for measuring 3D scene flow with light field sensors [1, 24]. This approach is based on the derivation of a new constraint, the *ray flow equation*, which relates dense 3D motion field of a scene to gradients of the measured light field, as follows:

$$\boxed{L_X V_X + L_Y V_Y + L_Z V_Z + L_t = 0},$$

where V_X, V_Y, V_Z are per-pixel 3D scene flow components, L_X, L_Y, L_Z are spatio-angular gradients of the 4D light field, and L_t is the temporal light field derivative. This simple, linear equation describes the *ray flow*, defined as local changes in the 4D light field, due to small, differential, 3D scene motion. The ray flow equation is independent of the scene depth, and is broadly applicable to a general class of scenes.

The ray flow equation is an under-constrained linear equation with three unknowns (V_X, V_Y, V_Z) per equation. Therefore, it is impossible to recover the full 3D scene flow without imposing further constraints. Our key observation is that, due to the structural similarity between ray flow and the classical optical flow equations [14], the regularization techniques developed over three decades of optical flow research can be easily adapted to constrain ray flow. The analogy between ray flow and optical flow provides a general recipe for designing ray flow based algorithms for recovering 3D dense scene flow directly from measured light field gradients.

We develop two basic families of scene flow recovery algorithms: local *Lucas-Kanade* methods, and global *Horn-Schunck* methods, based on local and global optical flow [14, 20]. We also design a high-performance combined local-global method by utilizing the correspondence structure in the light fields. We adopt best practices and design choices from modern, state-of-the-art optical flow algorithms (*e.g.*, techniques for preserving motion discontinuities, recovering large motions). Using these techniques, we demonstrate 3D flow computation with *sub-millimeter* precision along all three axes, for a wide range of scenarios, including complex non-rigid motion.

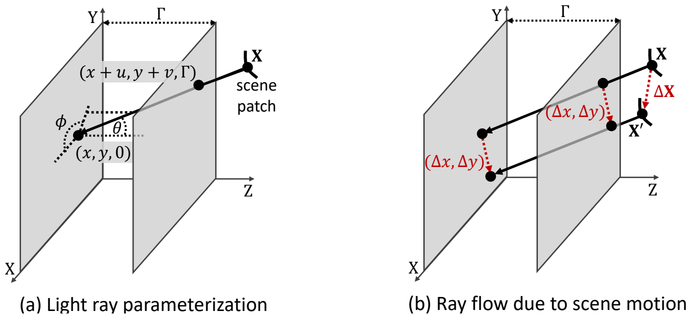
Theoretical and practical performance analysis: What is the space of motions that are recoverable by the proposed techniques? What factors influence their ability to recover 3D motion? To address these fundamental questions, we define the *light field structure tensor*, a 3×3 matrix that encodes local light field structure. We show that the space of recoverable motions is determined by the properties (rank and eigenvalues) of the light field structure tensor, which depends on the scene texture. We also analyze the performance dependence of ray flow techniques on the imaging parameters of the light field camera (*e.g.*, angular resolution, aperture size and field of view [11]). This analysis determines theoretical and practical performance limits of the proposed algorithms, and can also inform design of future light field cameras optimized for motion sensing.

Scope and implications: The main goal of the paper is to establish theoretical foundations of 3D scene flow computation from light field gradients. In doing so, this paper takes the first steps towards positioning light field cameras as effective 3D motion sensors, in addition to their depth estimation capabilities. Although we have implemented several proof-of-concept ray flow methods, it is possible to leverage the vast body of optical flow research and design novel, practical ray flow algorithms in the future. These algorithms, along with novel light field camera designs optimized for motion sensing, can potentially provide high-precision 3D motion sensing capabilities in a wide range of applications, including robotic manipulation, user interfaces, and augmented reality.

2 Related Work

Light field scene flow: State-of-the-art scene flow methods compute the 3D motion by combining optical flow and change of depths (*e.g.*, via stereo [15, 34] or RGB-D cameras [12,29]). Scene flow methods for light fields cameras have also been proposed before [13, 21, 27], where light fields are used for recovering depths. Our goal is different: we use light fields for recovering 3D scene motion directly. Thus, the proposed approaches are not adversely affected by errors in measured depths, resulting in precise motion estimation, especially for subtle motions.

Light field odometry: Light fields have been used for recovering a camera's ego-motion [10, 22], and to compute high-quality 3D scene reconstructions via structure-from-motion techniques [17,35]. These methods are based on a constraint relating cam-



(a) Light ray parameterization

(b) Ray flow due to scene motion

Fig. 1. (a) A light ray is parameterized by 4D coordinates (x, y, u, v), which are determined by the ray's intersection points $(x, y, 0)$ and $(x+u, y+v, \Gamma)$ with planes $Z = 0$ and $Z = \Gamma$, where Γ is a fixed constant. **(b)** Motion (translation) of the scene point that emits or reflects the ray results in a change in the (x, y) coordinates of the ray, but the (u, v) coordinates remain constant.

era motion and light fields. This constraint has the same structural form as the equation derived in this paper, although they are derived in different contexts (camera motion vs. non-rigid scene motion) with different assumptions. These works aim to recover 6-degrees-of-freedom (6DOF) camera motion, which is an over-constrained problem. Our focus is on recovering 3D non-rigid scene motion at every pixel, which is under-constrained due to considerably higher number of degrees of freedom.

Shape recovery from differential motion: Chandraker *et al.* developed a comprehensive theory for recovering shape and reflectance from differential motion of the light source, object or camera [7–9, 19, 32]. While our approach is also based on a differential analysis of light fields, our goal is different – to recover scene motion itself.

3 The Ray Flow Equation

Consider a scene point P at 3D location $\mathbf{X} = (X, Y, Z)$. Let $L(\mathbf{X}, \theta, \phi)$ be the radiance of P along direction (θ, ϕ), where θ, ϕ are the polar angle and azimuth angle as defined in spherical coordinates. The function $L(\mathbf{X}, \theta, \phi)$ is called the *plenoptic function*: it defines the radiance at all positions, along all possible ray directions. Assuming the radiance does not change along a ray, the 5D function $L(\mathbf{X}, \theta, \phi)$ can be simplified to the 4D *light field* $L(x, y, u, v)$, with each ray parameterized by its intersections with two parallel planes $Z = 0$ and $Z = \Gamma$, where Γ is a fixed constant. This is shown in Figure 1(a). Let the ray intersect the planes at points $(x, y, 0)$ and $(x+u, y+v, \Gamma)$, respectively. Then, the ray is represented by the coordinates (x, y, u, v). Note that (u, v) are *relative coordinates*; they represent the differences in the X and Y coordinates of the two intersection points. This is called the *two-plane parameterization* of the light field [18, 24], and is widely used to represent light fields captured by cameras.

By basic trigonometry, the relationship between the *scene-centric* coordinates (X, Y, Z, θ, ϕ) of a light ray, and its *camera-centric* coordinates (x, y, u, v) is given by:

$$\begin{aligned} x &= X - Z \tan\theta \cos\phi, & u &= \Gamma \tan\theta \cos\phi, \\ y &= Y - Z \tan\theta \sin\phi, & v &= \Gamma \tan\theta \sin\phi. \end{aligned} \tag{1}$$

Effect of scene motion on light fields: Let the 3D locations of a scene point P at time t and $t+\Delta t$ be $\mathbf{X}$ and $\mathbf{X}' = \mathbf{X} + \Delta\mathbf{X}$, where $\Delta\mathbf{X} = (\Delta X, \Delta Y, \Delta Z)$ is the small (differential) 3D motion (shown in Figure 1(b)). Consider a ray reflected (emitted) by P. We assume that the scene patch containing P only translates during motion[1], so that the ray only moves parallel to itself, *i.e.*, (u,v) coordinates of the ray remain constant. Let the coordinates of the ray before and after motion be (x,y,u,v) and $(x+\Delta x, y+\Delta y, u, v)$. Then, assuming that the ray brightness remains constant during motion[2]:

$$L(x,y,u,v,t) = L(x+\Delta x, y+\Delta y, u, v, t+\Delta t)\,. \tag{2}$$

This *ray brightness constancy assumption* is similar to the *scene brightness constancy assumption* made in optical flow. First-order Taylor expansion of Eq. 2 gives:

$$\frac{\partial L}{\partial x}\Delta x + \frac{\partial L}{\partial y}\Delta y + \frac{\partial L}{\partial t}\Delta t = 0\,. \tag{3}$$

We define *ray flow* as the change $(\Delta x, \Delta y)$ in a light ray's coordinates due to scene motion. Eq. 3 relates ray flow and light field gradients $(\frac{\partial L}{\partial x}, \frac{\partial L}{\partial y}, \frac{\partial L}{\partial t})$. From Eq. 1, we can also find a relationship between ray flow and scene motion:

$$\begin{aligned}\Delta x &= \frac{\partial x}{\partial X}\Delta X + \frac{\partial x}{\partial Z}\Delta Z = \Delta X - \frac{u}{\Gamma}\Delta Z\,,\\ \Delta y &= \frac{\partial y}{\partial Y}\Delta Y + \frac{\partial y}{\partial Z}\Delta Z = \Delta Y - \frac{v}{\Gamma}\Delta Z\,.\end{aligned} \tag{4}$$

By substituting Eq. 4 in Eq. 3 and using symbols L_* for light field gradients, we get:

$$\boxed{L_X V_X + L_Y V_Y + L_Z V_Z + L_t = 0}\,, \tag{5}$$

where $L_X = \frac{\partial L}{\partial x}, L_Y = \frac{\partial L}{\partial y}, L_Z = -\frac{u}{\Gamma}\frac{\partial L}{\partial x} - \frac{v}{\Gamma}\frac{\partial L}{\partial y}, L_t = \frac{\partial L}{\partial t}, \mathbf{V} = (V_X, V_Y, V_Z) = (\frac{\Delta X}{\Delta t}, \frac{\Delta Y}{\Delta t}, \frac{\Delta Z}{\Delta t})$. We call this the *ray flow equation*; it relates the 3D scene motion and the measured light field gradients. This simple, yet powerful equation enables recovery of dense scene flow from measured light field gradients, as we describe in Sections 4 to 6. In the rest of this section, we discuss salient properties of the ray flow equation in order to gain intuitions and insights into its implications.

3.1 Ray Flow Due to Different Scene Motions

Ray flows due to different scene motions have interesting qualitative differences. To visualize the difference, we represent a 4D light field sensor as a 2D array of pinhole cameras, each with a 2D image plane. In this representation (u,v) coordinates of the light field $L(x,y,u,v)$ denote the pixel indices within individual images (sub-aperture images). (x,y) coordinates denote the locations of the cameras, as shown in Figure 2.

[1] For a rotating object, in general, the motion of small scene patches can be modeled as translation, albeit with a change in the surface normal. For small rotations (small changes in surface normal), the brightness of a patch can be assumed to be approximately constant [31].

[2] This is true under the assumption that the light sources are distant such that $\mathbf{N}\cdot\mathbf{L}$, the dot-product of surface normal and lighting direction, does not change [31].

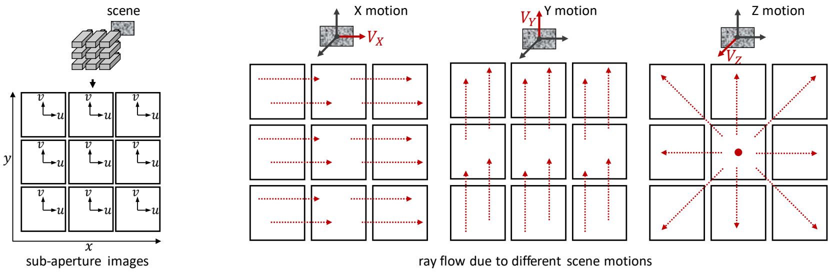


Fig. 2. Ray flow due to different scene motions. (Left) We represent a light field sensor as a 2D array of pinhole cameras, each of which captures a 2D image (sub-aperture images). (u,v) denotes the pixel indices within each sub-aperture image. (x,y) denotes the locations of the cameras. **(Right)** For X/Y scene motion, rays move horizontally/vertically, across sub-aperture images. The amount of change $(\Delta x,\Delta y)$ in the sub-aperture index is independent of the rays' coordinates. For Z-motion, rays shift radially across sub-aperture images. The shift depends on each ray's (u,v) coordinates. Rays at the center of each sub-aperture image $(u=0,v=0)$ do not shift. In all cases, rays retain the same pixel index (u,v), but move to a different sub-aperture image.

For X/Y scene motion, a light ray *shifts horizontally/vertically* across sub-aperture images. The amount of shift $(\Delta x,\Delta y)$ is *independent* of the ray's original coordinates, as evident from Eq. 4. For Z-motion, the ray *shifts radially* across sub-aperture images. The amount of shift *depends* on the ray's (u,v) coordinates (c.f. Eq. 4). For example, rays at the center of each sub-aperture image $(u=0,v=0)$ do not shift. In all cases, rays retain the *same pixel index* (u,v) after the motion, but in a *different sub-aperture image* (x,y), since scene motion results in rays translating parallel to themselves.

3.2 Invariance of Ray Flow to Scene Depth

An important observation is that the ray flow equation does not involve the depth or 3D position of the scene point. In conventional motion estimation techniques, depth and motion estimation are coupled together, and thus need to be performed simultaneously [2]. In contrast, the ray flow equation decouples depth and motion estimation. This has important practical implications: 3D scene motion can then be directly recovered from the light field gradients, without explicitly recovering scene depths, thereby avoiding the errors due to the intermediate depth estimation step.

Notice that although motion estimation via ray flow does not need depth estimation, the accuracy of the estimated motion depends on scene depth. For distant scenes, the captured light field is convolved with a 4D low-pass point spread function, which makes gradient computation unreliable. As a result, scene motion cannot be estimated reliably.

3.3 Similarities between Ray Flow and Optical Flow

For every ray in the captured light field, we have one ray flow equation with three unknowns to solve, which gives us an under-constrained system. Therefore additional assumptions need to be made to further constrain the problem. This is similar to the well-known *aperture problem* in 2D optical flow, where the optical flow equation $I_x u_x + I_y u_y + I_t = 0$ is also under-constrained (1 equation, 2 unknowns (u_x,u_y)). There are some

Optical Flow	Ray Flow
Linear equation: $I_x u_x + I_y u_y + I_t = 0$	Linear equation: $L_X V_X + L_Y V_Y + L_Z V_Z + V_t = 0$
Coefficients: Image gradients (I_x, I_y, I_t)	Coefficients: Light field gradients (L_X, L_Y, L_Z, L_t)
2 unknowns per pixel: Pixel motion (u_x, u_y)	3 unknowns per pixel: Scene motion (V_X, V_Y, V_Z)
Motion (u_x, u_y) computed in 2D image space (pixels)	Motion (V_X, V_Y, V_Z) computed in 3D scene space
Gradients (I_x, I_y) defined on 2D image grid	Gradients (L_X, L_Y, L_Z) defined on 4D light-field grid
u_x and u_y flow computations are symmetric	X/Y and Z motion computations are asymmetric
Size of structure tensor: 2×2	Size of structure tensor: 3×3
Possible ranks of structure tensor: $[0, 1, 2]$	Possible ranks of structure tensor: $[0, 2, 3]$

Table 1. Comparisons between optical flow and ray flow.

interesting differences between ray flow and optical flow (see Table 1), but the key similarity is that both ray flow and optical flow are *under-constrained linear equations.*

Fortunately, optical flow is one of the most researched problems in computer vision. Broadly, there are two families of differential optical flow techniques, based on the additional constraints imposed for regularizing the problem. The first is local methods (*e.g.*, Lucas-Kanade [20]), which assume that the optical flow is constant within small image neighborhoods. Second is global methods (*e.g.*, Horn-Schunck [14]), which assume that the optical flow varies smoothly across the image. By exploiting the structural similarity between the optical flow and ray flow equations, we develop two families of ray flow techniques accordingly: local ray flow (Section 4) and global ray flow (Section 5).

4 Local 'Lucas-Kanade' Ray Flow

In this section, we develop the local ray flow based scene flow recovery methods, inspired by Lucas-Kanade optical flow [20]. This class of ray flow methods assume that the motion vector $\mathbf{V}$ is constant in local 4D light field windows. Consider a ray with coordinates $\mathbf{x}_c = (x, y, u, v)$. We stack all the equations of form Eq. 5 from rays in a local neighborhood of $\mathbf{x}_c$, $\mathbf{x}_i \in \mathcal{N}(\mathbf{x}_c)$ into a linear system $\mathbf{AV} = \mathbf{b}$, where:

$$\mathbf{A} = \begin{bmatrix} L_X(\mathbf{x}_1) & L_Y(\mathbf{x}_1) & L_Z(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ L_X(\mathbf{x}_n) & L_Y(\mathbf{x}_n) & L_Z(\mathbf{x}_n) \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -L_t(\mathbf{x}_1) \\ \vdots \\ -L_t(\mathbf{x}_n) \end{bmatrix}. \tag{6}$$

Then, the motion vector $\mathbf{V}$ can be estimated by the normal equation:

$$\mathbf{V} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}. \tag{7}$$

4.1 What is the Space of Recoverable Motions?

In the previous section, we discussed that it is impossible to recover the complete 3D motion vector from a single ray flow equation. A natural question to ask is: what is the space of recoverable motions with the additional local constancy constraint? Intuitively it depends on the local structure of the light field. For example, if the local window corresponds to a textureless scene, then no motion is recoverable. One way to address this question is by understanding the properties of the 3×3 symmetric matrix $\mathbf{S} = \mathbf{A}^T \mathbf{A}$.

$$\mathbf{S} = \begin{bmatrix} \sum_{\mathbf{i}=1}^{\mathbf{n}} L_{Xi}^2 & \sum_{\mathbf{i}=1}^{\mathbf{n}} L_{Xi} L_{Yi} & \sum_{\mathbf{i}=1}^{\mathbf{n}} L_{Xi} L_{Zi} \\ \sum_{\mathbf{i}=1}^{\mathbf{n}} L_{Yi} L_{Xi} & \sum_{\mathbf{i}=1}^{\mathbf{n}} L_{Yi}^2 & \sum_{\mathbf{i}=1}^{\mathbf{n}} L_{Yi} L_{Zi} \\ \sum_{\mathbf{i}=1}^{\mathbf{n}} L_{Zi} L_{Xi} & \sum_{\mathbf{i}=1}^{\mathbf{n}} L_{Zi} L_{Yi} & \sum_{\mathbf{i}=1}^{\mathbf{n}} L_{Zi}^2 \end{bmatrix}, \tag{8}$$

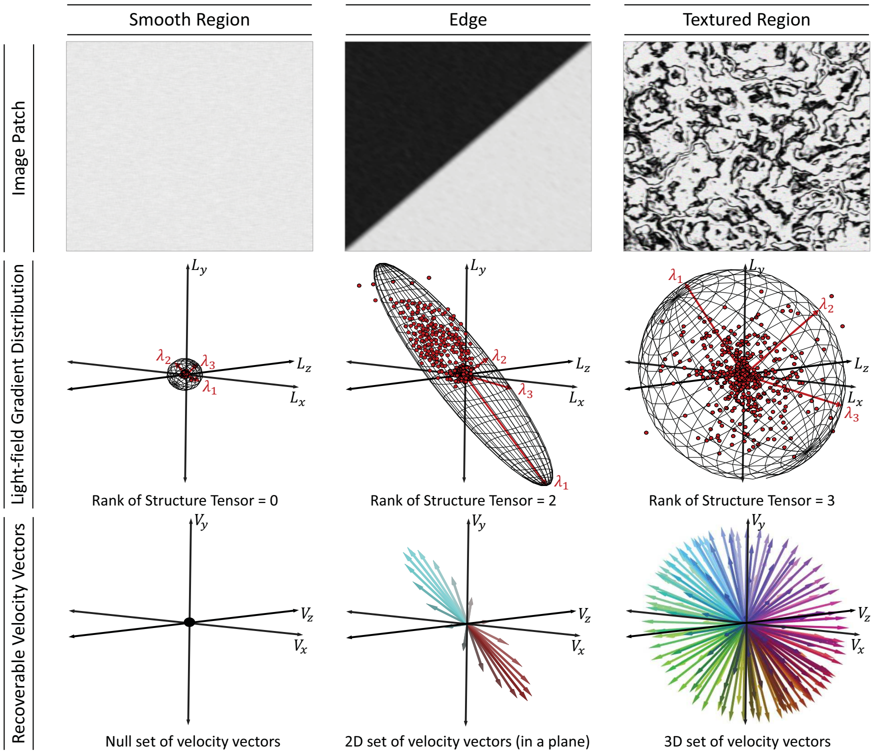

Smooth Region
Edge
Textured Region
Image Patch
Light-field Gradient Distribution
Recoverable Velocity Vectors
L_y
L_z
L_x
λ_1
λ_2
λ_3
Rank of Structure Tensor = 0
Rank of Structure Tensor = 2
Rank of Structure Tensor = 3
V_y
V_z
V_x
Null set of velocity vectors
2D set of velocity vectors (in a plane)
3D set of velocity vectors

Fig. 3. Relationship between scene texture, rank of the light field structure tensor, and the space of recoverable motions. (Top) Scene patches. **(Middle)** Distribution of light field gradients; each dot represents the gradient (L_X, L_Y, L_Z) computed at one location in a light field window. The covariance of the gradients is represented by ellipsoids whose principal axes are proportional to the three eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the structure tensor. **(Bottom)** Set of recoverable motion vectors. **(Left)** For a light field window corresponding to a smooth patch, the gradients (L_X, L_Y, L_Z) are approximately zero, and concentrated around the origin in the gradient space. The rank of the structure tensor is 0, implying that no motion vector can be recovered reliably. **(Center)** For a patch with a single edge, non-zero gradients are distributed approximately along a plane in the gradient space, resulting in a rank 2 structure tensor (1-D null space). As a result, a 2D family of motions (orthogonal to the edge) can be recovered. **(Right)** For a patch with 2D texture, non-zero gradients are distributed nearly isotropically in the gradient space. Therefore, structure tensor has rank = 3. Thus, the entire space of 3D motions are recoverable.

where L_{*i} is short for $L_*(\mathbf{x}_i)$. We define $\mathbf{S}$ as the *light field structure tensor*; it encodes the local structure of the light field.[3] To estimate motion using Eq. 7, $\mathbf{S}$ must be invertible. Thus, the performance of the local method can be understood in terms of $rank(\mathbf{S})$.

Result (Rank of structure tensor) Structure tensor $\mathbf{S}$ has three possible ranks: 0, 2, and 3 for a local 4D light field window. These correspond to scene patches with no texture (smooth regions), an edge, and 2D texture, respectively.

[3] Structure tensors have been researched and defined differently in the light field community (*e.g.*, [23]). Here it is defined by the gradients w.r.t. the 3D motion and is thus a 3×3 matrix.

Intuition: In the following, we provide an intuition for the above result by considering three cases. A detailed proof is given in the supplementary technical report.

Case 1: Smooth region. In this case, $L_X = L_Y = L_Z = 0$ for all the locations in the light field window. Therefore, all the entries of the structure tensor (given in Eq. 8) are zero, resulting in it being a rank 0 matrix. All three eigenvalues $\lambda_1, \lambda_2, \lambda_3 = 0$, as shown in the left column of Figure 3. As a result, it has a 3-D null space, and no motion vector can be recovered reliably for this window.

Case 2: Single step edge. Without loss of generality, suppose the light field window corresponds to a fronto-parallel scene patch with a vertical edge, *i.e.*, $L_Y = 0$. The middle row of the structure tensor is all zeros, resulting in a rank 2 matrix, with a 1-D null space (only one eigenvalue $\lambda_3 = 0$). As a result, a 2D family of motions (motion orthogonal to the edge) can be recovered, as illustrated in the second column of Figure 3.

Case 3: 2D texture. All three derivatives are non-zero and independent. The structure tensor is full rank (rank $= 3$) and the entire space of 3D motions are recoverable.

Comparisons with structure tensor for optical flow: The structure tensor for 2D optical flow is a 2×2 matrix and can have all possible ranks from 0 to 2 [26]. *For light fields, the structure tensor cannot have rank* 1. This is because even a 4D window with a single step edge results in a rank 2 structure tensor.[4] For more conceptual comparisons between optical flow and ray flow, please refer to Table 1.

Dependence on camera parameters. Besides scene texture and light field structure, the imaging parameters of the light field camera also influences the performance of ray flow methods. Using the ray flow equation requires computing angular light field gradients (L_X and L_Y), whose accuracy depends on the angular resolution of the light field camera. Most off-the-shelf light field cameras have a relatively low angular resolution (*e.g.*, 15×15 for Lytro Illum), resulting in aliasing [22]. To mitigate aliasing, we apply Gaussian pre-filtering before computing the gradients. Another important parameter is the aperture size, which limits the range of recoverable motion. This is because ray flow changes the (x, y) coordinates of the ray. When the motion is too large, most of the rays will escape the aperture and the motion cannot be recovered (see Fig. 2). See the supplementary report for a detailed discussion on the effects of various camera parameters.

4.2 Enhanced Local Methods

Our analysis so far assumes small (differential) scene motion. If the inter-frame scene motion is large, then the simple linear ray flow equation is not valid. Another way to relate the scene motion and the resulting change in the captured light field is to define a warp function on the light field, which describes the change in coordinates $\mathbf{x} = (x, y, u, v)$ of a light ray due to scene motion $\mathbf{V}$ (Eq. 1):

$$\mathbf{w}(\mathbf{x}, \mathbf{V}) = (x + V_X - \frac{u}{\Gamma} V_Z, y + V_Y - \frac{v}{\Gamma} V_Z, u, v)\,. \tag{9}$$

[4] Although the structure tensor theoretically has rank 2, the ratio $\frac{\lambda_1}{\lambda_2}$ of the largest and second largest eigenvalues can be large. This is because the eigenvalue corresponding to Z motion depends on the range of (u, v) coordinates, which is limited by the size of the light field window. Therefore, a sufficiently large window size is required for motion recovery.

Then, the local method can be formulated as a local light field registration problem:

$$\min_{\mathbf{V}} \sum_{\mathbf{x_i} \in \mathcal{N}(\mathbf{x_c})} (L_0(\mathbf{x_i}) - L_1(\mathbf{w}(\mathbf{x_i}, \mathbf{V})))^2 . \tag{10}$$

The method described by Eq. 7 is *the same* as locally linearizing Eq. 10. Using this formulation, we develop an enhanced local method where the motion vector $\mathbf{V}$ is solved over a light field pyramid for dealing with large (non-differential) scene motions.

5 Global 'Horn-Schunck' Ray Flow

The local constancy assumption made by the local ray-flow methods is too restrictive when dealing with non-rigid motion. In this section, we propose a family of global ray flow methods that are inspired by global 'Horn-Schunck' optical flow [14]. The basic, less limiting assumption is that the 3D flow field varies smoothly over the scene. Therefore, we regularize the flow computation by introducing a smoothness term that penalizes large variations of $\mathbf{V}$ and minimize a global functional:

$$E(\mathbf{V}) = E_D(\mathbf{V}) + E_S(\mathbf{V}), \qquad \text{where} \tag{11}$$

$$E_D(\mathbf{V}) = \int_\Omega (L_X V_X + L_Y V_Y + L_Z V_Z + L_t)^2 \, dx\,dy\,du\,dv \, ,$$

$$E_S(\mathbf{V}) = \int_\Omega \left(\lambda |\nabla V_X|^2 + \lambda |\nabla V_Y|^2 + \lambda_Z |\nabla V_Z|^2\right) dx\,dy\,du\,dv \, .$$

Note that Ω is the 4D light field domain, and ∇p is the 4D gradient of a scalar field p: $\nabla p = (\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial u}, \frac{\partial p}{\partial v})$. Since the computation of X/Y flow and Z flow are asymmetric, we use different weights for the X/Y and Z smoothness terms. In practice we use $\lambda = 8$ and $\lambda_Z = 1$. $E(\mathbf{V})$ is a convex functional, and its minimum can be found by the Euler-Lagrange equations. See the supplementary technical report for details.

Enhanced global methods. The quadratic penalty functions used in the basic global ray flow method (Eq. 11) penalizes flow discontinuities, leading to over-smoothing around motion boundaries. In the optical flow community [3, 5, 25], it has been shown that robust penalty functions perform significantly better around motion discontinuities. Based on this, we developed an enhanced global method that uses the generalized Charbonnier function $\rho(x) = (x^2 + \varepsilon^2)^a$ with $a = 0.45$ as suggested in [28].

6 Combined Local-Global Ray Flow

The ray flow methods considered so far treat the motion of each light ray separately. However, a light field camera captures multiple rays from the same scene point, all of which share the same motion. Can we exploit this constraint to further improve the performance of ray flow based motion recovery methods? Consider a ray with coordinates (x, y, u, v), coming from a scene point $S = (X, Y, Z)$. The coordinates of all the rays coming from S form a 2D plane $\mathscr{P}(u, v)$ [10, 17, 27] in the 4D light-field:

$$\mathscr{P}(u,v) = \{(x_i, y_i, u_i, v_i) \mid u_i = u - \alpha(x_i - x), v_i = v - \alpha(y_i - y)\}, \tag{12}$$

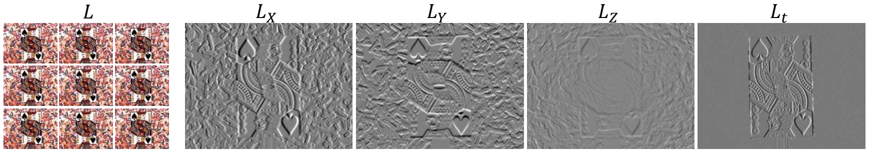


Fig. 4. Measured light field gradients. Light field for an example scene (a card moving in the X-Z plane in front of a static background) is shown as a 3×3 subset of sub-aperture images (left). Light field gradients are only shown for the central sub-aperture. **Zoom in for details.**

where the parameter $\alpha = \frac{\Gamma}{Z}$ is the disparity between sub-aperture images, and is a function of the depth Z of S. All these rays share the same flow vector $\mathbf{V} = (V_X, V_Y, V_Z)$. Therefore, we can estimate $\mathbf{V}$ by minimizing the following function:

$$\min_{\mathbf{V}} \sum_{\mathbf{x}_i \in \mathscr{P}(u,v)} (L_{Xi}V_X + L_{Yi}V_Y + L_{Zi}V_Z + L_{ti})^2. \tag{13}$$

Given the parameter α (which can be determined using light-field based depth estimation [33]), this function can be minimized similarly as the local method (Section 4), which assumes constancy of ray motion in a local 4D ray neighborhood $\mathscr{N}(u,v)$. While the local constancy assumption is only approximate, the constancy of motion over the 2D plane described in Eq. 12 is an *exact constraint*, resulting in better performance. Moreover, in order to further regularize the problem, we can leverage the global smoothness of motion assumption used in global methods in Section 5. Based on these observations, we propose a *combined local-global* (CLG) ray flow method [6], whose data term is given by minimizing the local term (Eq. 13) for each ray *in the central view Ω_c*:

$$E_D(\mathbf{V}) = \int_{\Omega_c} \sum_{\mathbf{x}_i \in \mathscr{P}(u,v)} (L_{Xi}V_X + L_{Yi}V_Y + L_{Zi}V_Z + L_{ti})^2 du\, dv\,. \tag{14}$$

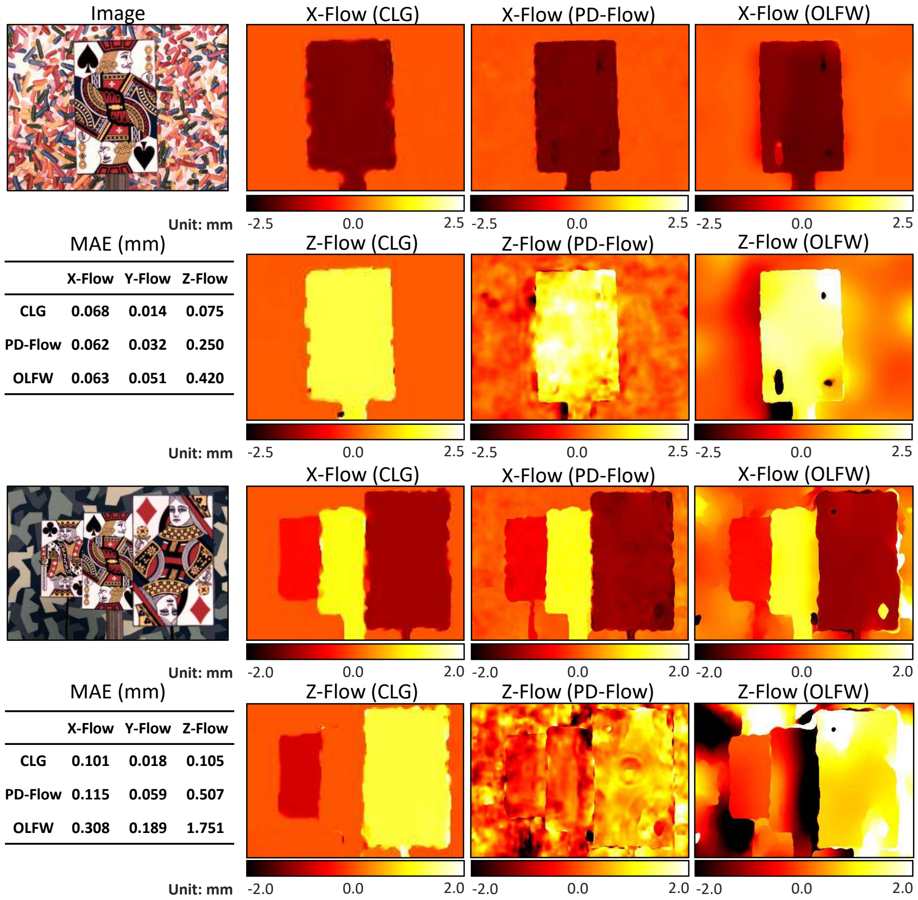
This local data term is combined with a global smoothness term defined on Ω_c.

$$E_S(\mathbf{V}) = \int_{\Omega_c} \left(\lambda |\nabla V_X|^2 + \lambda |\nabla V_Y|^2 + \lambda_Z |\nabla V_Z|^2\right) du\, dv\,. \tag{15}$$

This formulation estimates motion only for the 2D central view Ω_c while utilizing the information from the whole light field, thereby simultaneously achieving computational efficiency and high accuracy. Furthermore, by adopting the enhancements of local and global methods, the CLG method outperforms individual local and global methods. Therefore, in the rest of the paper, we show results only for the CLG method. Also notice that the CLG ray flow method uses the estimated depths only *implicitly* as an additional constraint for regularization. Therefore, unlike previous methods [13, 21, 27], estimating depths accurately is not critical for recovering motion. Please see the supplementary technical report for implementation details of the CLG method, a comparison between the local, global and CLG methods and simulation results demonstrating the effect of depth accuracy on the CLG method.

7 Experimental Results

For our experiments, we use a Lytro Illum camera, calibrated using a geometric calibration toolbox [4]. We extract the central 9×9 subaperture images, each of which



MAE (mm)

	X-Flow	Y-Flow	Z-Flow
CLG	0.068	0.014	0.075
PD-Flow	0.062	0.032	0.250
OLFW	0.063	0.051	0.420

MAE (mm)

	X-Flow	Y-Flow	Z-Flow
CLG	0.101	0.018	0.105
PD-Flow	0.115	0.059	0.507
OLFW	0.308	0.189	1.751

Fig. 5. Controlled experiments on a translation stage. (Top) A single card moving diagonally. **(Bottom)** Three cards moving diagonally forward, laterally, and diagonally backward, respectively. Mean absolute error (MAE) for the three motion components are shown in the tables. While all methods recover the lateral motion relatively accurately, the proposed CLG ray-flow approach estimates the Z-motion more accurately than previous approaches. This is because previous approaches rely on, and are thus prone to errors in, depth estimation. In contrast, our approach estimates the motion directly from light-field gradients, thereby achieving high accuracy.

has a spatial resolution of 552 × 383. Figure 4 shows an example light field and the computed gradients. We compare our combined local-global method (CLG) with the RGB-D scene flow method (PD-Flow) of Jaimez *et al*. [16] and light field scene flow method (called OLFW in this paper) of Srinivasan *et al*. [27]. For a fair comparison, we use the same modality (light fields) for depth estimation in PD-Flow (depth estimated from light field is the depth channel input), using the same algorithm as in OLFW [30]. Please refer to the supplementary video for a better visualization of the scene motion.

Controlled experiments on a translation stage. Figure 5 shows scene flow recovery

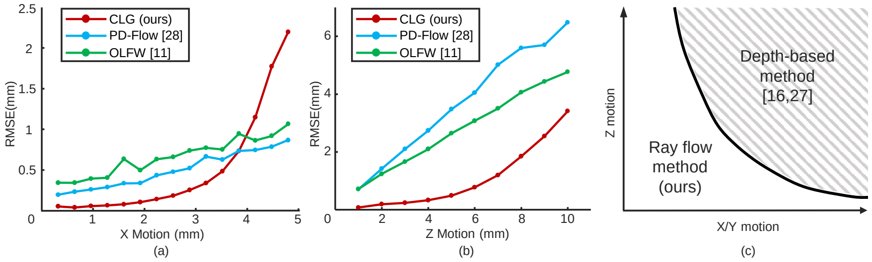


Fig. 6. Effect of the amount and kind of motion. We use a single textured plane as the scene to exclude the effect of other factors (motion boundaries, occlusions). **(a)** For X-motion, the error of our method increases rapidly when the motion is larger than 3.5mm, while PD-Flow and OLFW degrade gracefully. **(b)** For Z-motion, our method outperforms previous methods since it does not rely on accurate depth estimates. **(c)** This plot qualitatively shows the method best suited for estimating different amount and kind of motion. While previous approaches can reliably measure large motions, the proposed method is better suited for small, especially axial, motions.

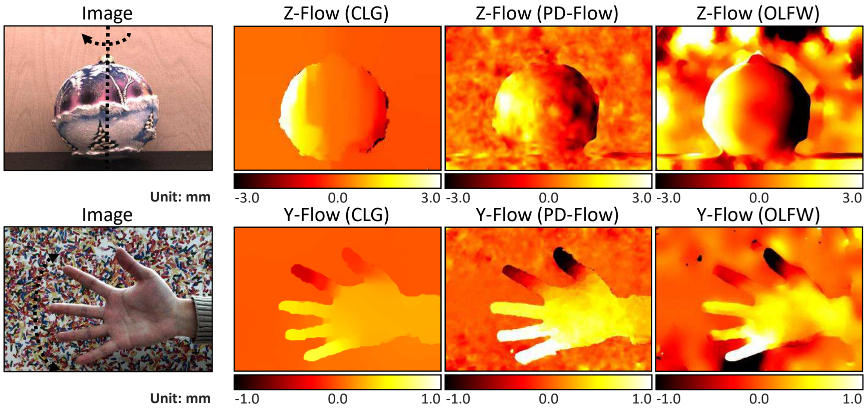


Fig. 7. Recovering non-planar and non-rigid motion. (Top) A rotating spherical ornament. All methods can estimate the gradually changing Z-motion, but only our method recovers the background correctly. **(Bottom)** An expanding hand. The expansion is demonstrated by the different Y-motion of the fingers.

results for a scene that is intentionally chosen to have simple geometry and sufficient texture to compare the baseline performance of the methods. The moving objects (playing cards) are mounted on controllable translation stages such that they can move in the X-Z plane with measured ground truth motion. Mean absolute error (MAE) for the three dimensions (ground truth Y-motion is zero) are computed and shown in the table. All three methods perform well for recovering the X-motion. However, PD-Flow and OLFW cannot recover the Z-motion reliably because errors in depth estimation are large compared to the millimeter-scale Z-motion. The proposed ray flow methods estimates the Z-motion directly, thereby achieving higher accuracy.

Dependency of the performance on the amount and kind of motion. We mount a textured plastic sheet on the translation stage and move it either laterally (X-motion) or

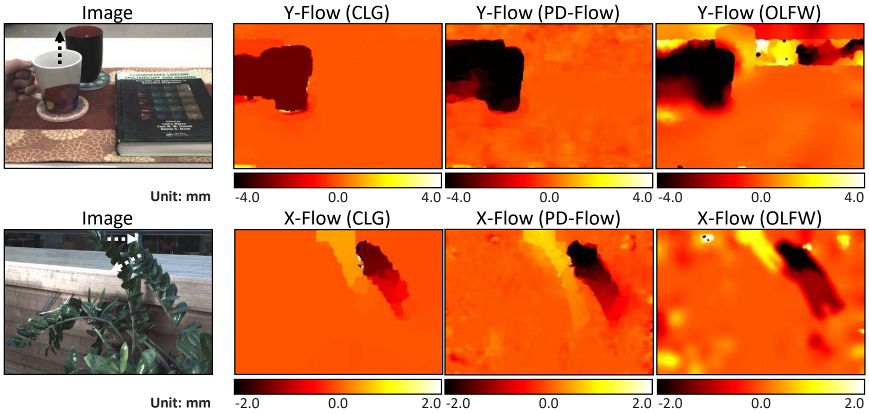


Fig. 8. Recovering motion in natural environments with occlusions. (Top) The mug on the left is picked up by a hand. Our method estimates the motion boundaries accurately. **(Bottom)** The top two vertical branches of the plant quiver in the wind. Our method can correctly compute the motion of the two complex-shaped branches.

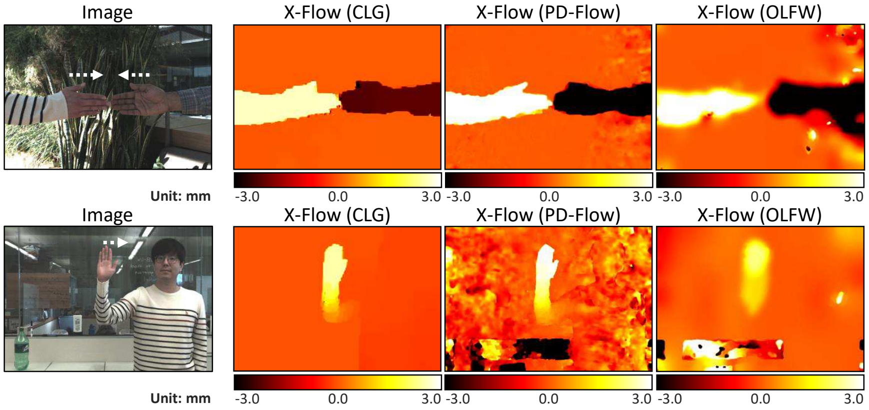


Fig. 9. Recovering human actions. (Top) Handshaking. All the three methods compute the joining movements of the hands correctly, while our method preserves the hand boundary best. **(Bottom)** Waving hand. Our method correctly estimates the motion in spite of the reflections and textureless regions in the background, which is challenging for depth estimation algorithms.

axially (Z-motion). Figures 6(a)(b) plot the RMSE of the estimated motion, against the amount of motion. The proposed method achieves higher precision for small motion. However, its accuracy decreases as the amount of motion increases. This is because of the limit imposed by the aperture size, as discussed in Section 4.1. On the other hand, previous depth-based methods [27] can recover motion over a large range, albeit with lower precision. This complementary set of capabilities of our method and previous methods are shown qualitatively in Figure 6(c). Although for the rest of the paper we focus on showing our methods' capability in recovering small motion (*e.g.*, for appli-

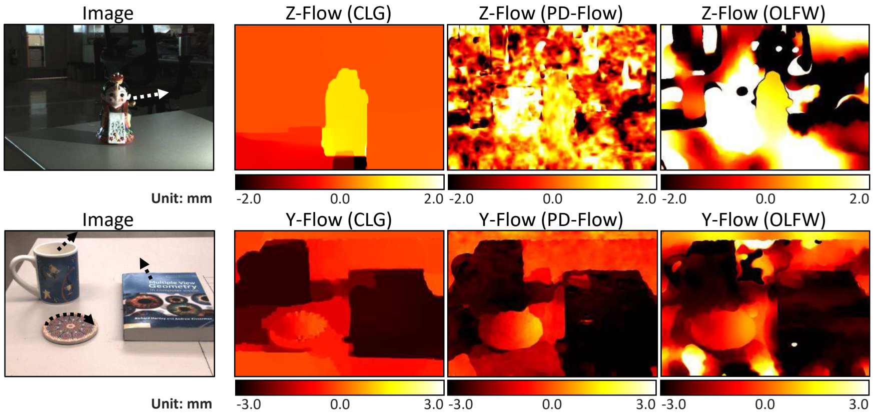


Fig. 10. Recovering motion under challenging lighting conditions. (Top) A figurine moves under weak, directional lighting. Our method still preserves the overall shape of the object, although its reflection on the table is also regarded as moving. **(Bottom) Failure case:** a few objects move independently. Due to shadows and lack of texture in the background, boundaries of the objects are not distinguishable in the recovered motion field of all the three methods.

cations in finger gesture and facial expression recognition), previous approaches [27] may perform better for measuring large scale motion, such as gait recognition.

Qualitative comparisons. Figure 7–10 shows qualitative comparisons of the three methods for complex, non-rigid motion and in challenging natural environments. For each experiment we only show one component of the recovered 3D flow. Please see the supplementary report for the full 3D flow visualization and more experiments. In all the examples, our method is able to estimate the complex, gradually changing motion fields and preserve the motion boundaries better than the other methods, especially for experiments involving small Z-motion, and where depth estimation is unreliable (*e.g.*, scenes with occlusions or reflections in the background). In Fig. 10 (bottom) all three methods have difficulty in preserving the object boundaries due to shadows, which is a inherent drawback of the brightness constancy assumption.

8 Limitations

Recoverable range of motion: As discussed in Sections 4.1 and 7, the maximum recoverable amount of motion for ray flow methods is limited by the aperture size. A future research direction is to develop hybrid methods that combine the ray flow method and depth-based methods [16, 27] according to the amount and nature of scene motion.

Running time: Currently our methods are implemented via unoptimized MATLAB code, which takes approximately 10 minutes to compute scene flow between two frames. Further work includes reducing the computational complexity of the algorithm and implementing the algorithm efficiently (*e.g.*, on a GPU), for real-time applications.

Acknowledgement. The authors would like to thank ONR grant number N00014-16-1-2995 and DARPA REVEAL program for funding this research.

References

1. Adelson, E.H., Wang, J.Y.A.: Single Lens Stereo with a Plenoptic Camera. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **14**(2), 99–106 (1992)
2. Alexander, E., Guo, Q., Koppal, S., Gortler, S., Zickler, T.: Focal flow: Measuring distance and velocity with defocus and differential motion. In: European Conference on Computer Vision (ECCV). pp. 667–682. Springer, Heidelberg (2016)
3. Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer Vision and Image Understanding **63**(1), 75–104 (1996)
4. Bok, Y., Jeon, H.G., Kweon, I.S.: Geometric calibration of micro-lens-based light field cameras using line features. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **39**(2), 287–300 (2017)
5. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. European Conference on Computer Vision (ECCV) pp. 25–36 (2004)
6. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. International journal of computer vision (IJCV) **61**(3), 211–231 (2005)
7. Chandraker, M.: On shape and material recovery from motion. In: European Conference on Computer Vision (ECCV). pp. 202–217. Springer, Heidelberg (2014)
8. Chandraker, M.: What camera motion reveals about shape with unknown brdf. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2171–2178. IEEE, Washington (2014)
9. Chandraker, M.: The information available to a moving observer on shape with unknown, isotropic brdfs. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **38**(7), 1283–1297 (2016)
10. Dansereau, D.G., Mahon, I., Pizarro, O., Williams, S.B.: Plenoptic flow: Closed-form visual odometry for light field cameras. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4455–4462. IEEE, Washington (2011)
11. Dansereau, D.G., Schuster, G., Ford, J., Wetzstein, G.: A wide-field-of-view monocentric light field camera. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Washington (2017)
12. Gottfried, J.M., Fehr, J., Garbe, C.S.: Computing range flow from multi-modal kinect data. In: International Symposium on Visual Computing. pp. 758–767. Springer, Heidelberg (2011)
13. Heber, S., Pock, T.: Scene flow estimation from light fields via the preconditioned primal-dual algorithm. In: Jiang, X., Hornegger, J., Koch, R. (eds.) Pattern Recognition. pp. 3–14. Springer International, Cham (2014)
14. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3), 185–203 (1981)
15. Hung, C.H., Xu, L., Jia, J.: Consistent binocular depth and scene flow with chained temporal profiles. International journal of computer vision (IJCV) **102**(1-3), 271–292 (2013)
16. Jaimez, M., Souiai, M., Gonzalez-Jimenez, J., Cremers, D.: A primal-dual framework for real-time dense rgb-d scene flow. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 98–104. IEEE, Washington (2015)
17. Johannsen, O., Sulc, A., Goldluecke, B.: On linear structure from motion for light field cameras. In: IEEE International Conference on Computer Vision (ICCV). pp. 720–728. IEEE, Washington (2015)
18. Levoy, M., Hanrahan, P.: Light field rendering. In: SIGGRAPH Conference on Computer Graphics and Interactive Techniques. pp. 31–42. ACM, New York (1996)

19. Li, Z., Xu, Z., Ramamoorthi, R., Chandraker, M.: Robust energy minimization for brdf-invariant shape from light fields. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 1. IEEE, Washington (2017)
20. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence. pp. 674–679. Morgan Kaufmann, San Francisco (1981)
21. Navarro, J., Garamendi, J.: Variational scene flow and occlusion detection from a light field sequence. In: International Conference on Systems, Signals and Image Processing (IWSSIP). pp. 1–4. IEEE, Washington (2016)
22. Neumann, J., Fermuller, C., Aloimonos, Y.: Polydioptric camera design and 3d motion estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. II–294. IEEE, Washington (2003)
23. Neumann, J., Fermüller, C., Aloimonos, Y.: A hierarchy of cameras for 3d photography. Computer Vision and Image Understanding **96**(3), 274–293 (2004)
24. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. Computer Science Technical Report CSTR **2**(11), 1–11 (2005)
25. Odobez, J.M., Bouthemy, P.: Robust multiresolution estimation of parametric motion models. Journal of Visual Communication and Image Representation **6**(4), 348–365 (1995)
26. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 593–600. IEEE, Washington (1994)
27. Srinivasan, P.P., Tao, M.W., Ng, R., Ramamoorthi, R.: Oriented light-field windows for scene flow. In: IEEE International Conference on Computer Vision (ICCV). pp. 3496–3504. IEEE, Washington (2015)
28. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2432–2439. IEEE, Washington (2010)
29. Sun, D., Sudderth, E.B., Pfister, H.: Layered rgbd scene flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 548–556. IEEE, Washington (2015)
30. Tao, M.W., Hadap, S., Malik, J., Ramamoorthi, R.: Depth from combining defocus and correspondence using light-field cameras. In: IEEE International Conference on Computer Vision (ICCV). pp. 673–680. IEEE, Washington (2013)
31. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. In: IEEE International Conference on Computer Vision (ICCV). vol. 2, pp. 722–729. IEEE, Washington (1999)
32. Wang, T.C., Chandraker, M., Efros, A.A., Ramamoorthi, R.: Svbrdf-invariant shape and reflectance estimation from light-field cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5451–5459. IEEE, Washington (2016)
33. Wanner, S., Goldluecke, B.: Variational light field analysis for disparity estimation and super-resolution. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **36**(3), 606–619 (2014)
34. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: European Conference on Computer Vision (ECCV). pp. 739–751. Springer, Heidelberg (2008)
35. Zhang, Y., Li, Z., Yang, W., Yu, P., Lin, H., Yu, J.: The light field 3d scanner. In: IEEE International Conference on Computational Photography (ICCP). pp. 1–9. IEEE, Washington (2017)