A Linear Approach to Absolute Pose Estimation for Light Fields

Anonymous 3DV submission

Paper ID 143

Abstract

This paper presents the first absolute pose estimation approach tailored to Light Field cameras. It builds on the observation that the ratio between the disparity arising in different sub-aperture images and their corresponding baseline is constant. Hence, we augment the 2D pixel coordinates with the corresponding normalised disparity to obtain the Light Field feature. This new representation allows for linear estimation of the absolute pose of a Light Field camera using the well-known Direct Linear Transformation algorithm. We evaluate the resulting absolute pose estimates with extensive simulations and experiments involving real Light Field datasets, demonstrating the competitive performance of our linear approach. Furthermore, we integrate our approach in a state-of-the-art Light Field Structure from Motion pipeline and demonstrate accurate multi-view 3D reconstruction.

1. Introduction

A well-known computer vision problem with many applications in Structure from Motion (SfM) is that of estimating the absolute pose of a camera [39, 40, 10]. In this work, we focus on the estimation of absolute pose for non-central projection cameras with overlapping fields-of-view, specifically Light Field (LF) cameras. We propose the first domain-specific approach that is linear, non-iterative and provides a unique solution for an arbitrary number of corresponding input points.

Multi-camera arrays have long been used in LF or plenoptic imaging [26]. LF cameras avoid the angular integration of rays impinging on each pixel, and therefore capture a 4D slice of the plenoptic function [1]. An alternative to multi-camera arrays are monocular microlens-based LF cameras, which have a microlens array placed between the main lens and a conventional image sensor. The introduction of this microlens array allows the imaged scene to be captured from multiple viewpoints, termed sub-aperture images [33], trading off reduced spatial resolution on the sensor with increased angular resolution. It is also worth
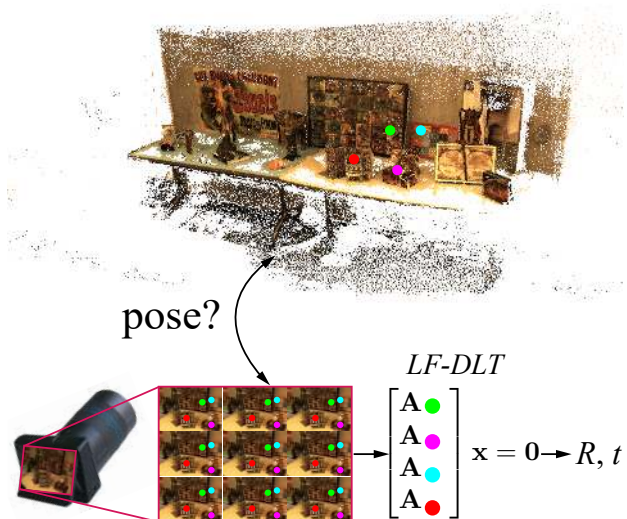

Figure 1. Our method linearly estimates the camera pose from a set of at least four 3D points projecting on a LF image.

noting that an increasing number of recent smartphones feature LF cameras (either arranged in a grid or in the form of dual-pixel cameras [9]). Various applications have been explored for post-processing LF images [51], such as depth estimation [16], deblurring [42], view synthesis [18], among which SfM has attracted considerable interest [17, 52, 36]. SfM with LF cameras (LF-SfM for short) is also the topic where our work finds its primary application.

LF cameras can be considered as a special case of a generalised camera model [11]. Contrary to central (i.e pinhole) cameras, in which all rays captured by the camera are constrained to pass through a single point known as the camera optical center, each pixel in a generalised camera samples an arbitrary 3D ray. After extensive research on absolute pose estimation for central cameras, efforts are now also concerned with absolute pose estimation for generalised cameras, *e.g.* [46, 49, 30, 31, 4]. Although the generalised imaging model is an elegant formulation, we argue (and we will discuss in more detail in Sec. 3.3) that LF cameras have implicit constraints which, if directly exploited, can yield very efficient and accurate algorithms.

3DV
#143

3DV
#143

3DV 2020 Submission #143. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

To the best of our knowledge, there exists no other absolute pose estimation algorithm customised for LF cameras. Thus, the contributions of this work are the following:

- A new representation for Light Field points and derivation of the LF projection matrix for a 3D point.
- A linear solution for LF camera absolute pose estimation based on the Direct Linear Transformation.
- Extensive evaluation of our method with simulations and datasets acquired with commercial LF cameras.

The rest of the paper is organized as follows. We discuss related prior work in Section 2. The *normalised disparity* concept and the proposed LF feature representation are presented in Section 3. Based on the introduced feature representation, we derive in Section 4 the proposed absolute pose estimation algorithm, which we call *LF-DLT*. We evaluate the performance of *LF-DLT* and compare it with state-of-the-art methods in Section 5. A LF-SfM pipeline using *LF-DLT* is presented and assessed in Section 6, and the paper is concluded in Section 7.

2. Related Work

Throughout this paper, the term Light Field (LF) will refer to a configuration where the camera centers are arranged on a rectangular grid (cf. Fig. 2). This can stem from either a camera array [50] (a configuration also found on some recent smartphones) or a set of sub-aperture images after the calibration of a micro-lens based LF camera [7, 3, 35]. In both cases, we consider an LF frame to consist of a set of different central views, *i.e.* the sub-aperture images. The sub-aperture image whose optical center coincides with the origin of the LF frame is referred to as the *central* sub-aperture image [36, 52].

2.1. Absolute Pose for Light Fields

Central Cameras: When all rays converge to a single optical center, the pose of a central camera can be determined from n 2D-3D correspondences, solving what is known as the Perspective-n-Point (*PnP*) problem. PnP is a widely studied topic in computer vision, *e.g.* [8, 54, 32]. The minimal case, namely *P3P*, requires 3 correspondences for which efficient and accurate solvers exist, since the resulting polynomials can be solved non-iteratively by radicals.

Non-central (generalised) cameras: In the non-central case, the problem is known as generalised or non-perspective PnP (respectively *gPnP* or *NPnP*), in which correspondences between 3D rays and 3D points are assumed. For the minimal case involving 3 ray-point correspondences, Chen *et al.* [5] derived one of the earliest algorithms. They compute the pose by solving an eighth degree polynomial and proposed to randomly select triplets to obtain an initial solution, which is further optimized using *ICP* [2]. Additional solvers for the *NP3P* problem were designed by Nistér and Stewénius [34] and Lee *et.*

al [23], which reduce to the solution of octic polynomials. If 4 correspondences are available, the scale of the translation between cameras can be recovered in addition to pose. Thus, Ventura *et al.* [49] derive a minimal solution for generalised pose and scale using Gröbner basis techniques whereas Kukelova *et. al* [22] present a solution based on an efficient algorithm for finding all intersections of 3 quadrics.

Several algorithms for the overdetermined case of arbitrary $n > 3$ have also been proposed. For example, Schweighofer and Pinz developed an iterative globally optimal $O(n)$ solution, where the distance of a 3D point and its projection to the line of sight is minimised using Semi-Definite Positive Programming (SDP) [41]. An extension of *EPnP* [25] to the case of generalised cameras is proposed in [20]. A unified approach to both central and non-central cameras, namely *UPnP*, is developed in [21]. However, despite its flexibility, *UPnP* still needs to disambiguate between 16 solutions and its implementation is quite involved. Sweeney *et al.* [45] formulate the pose-and-scale problem for generalised cameras as a minimisation of a least squares cost function, which can be solved as a system of third degree polynomials derived from $n \geq 4$ correspondences.

Assuming that the gravity vector direction is given, Sweeney *et al.* recover the depths of the points by solving a quadratic equation and then align the reconstructed point clouds [44]. Similarly, using lines as features (*i.e. gPnL*) and given the vertical direction, Horanyi *et al.* derive a minimal solver using 3 line correspondences [14]. A more general solution to *gPnL* (*i.e.* without any prior knowledge) is proposed by Miraldo *et al.* where the pose of the generalised camera is obtained by minimising the Klein quadric between 3D lines and their corresponding projection lines [30]. More recently, minimal solvers for combinations of features (*e.g.* 1 point-2 lines or 2 lines-1 point) were proposed in [31].

Owing to their dependence on the solution of high order polynomials which cannot be solved non-iteratively by radicals, *gP3P* solvers are significantly slower than their *P3P* counterparts. Our linear estimation trades accuracy for efficiency, which is often essential in practical applications.

2.2. SfM for LF Cameras

Recently, Nousias *et. al* [36] developed *uLF-SfM*, the first large-scale SfM pipeline tailored to LF images, in which the scene is incrementally reconstructed. *uLF-SfM*'s reconstruction accuracy is competitive to that of mature conventional pipelines like *COLMAP* [40] using only the central sub-aperture images, but is obtained at a fraction of the computational cost. In Sec. 6 we will report a modification of *uLF-SfM* to include the proposed method, hence we provide next a brief description of the former's operation.

uLF-SfM starts by selecting the two views with the highest number of matches, avoiding degenerate configurations

(*e.g.* pure rotation or zoom-in motion). Their relative motion is computed with the 17-pt algorithm [27]. Despite the fact that outlier removal in central cameras is straightforward, it is more involved in LF cameras primarily due to the fact that the Generalised Epipolar Constraint (GEC) [38] should be satisfied by all pairs of corresponding rays between LF frames. As a consequence, an outlying ray in a LF frame may result in a set of outliers after RANSAC. *uLF-SfM* meticulously filters outliers by examining the number of incidences of each feature, so that correct features are not discarded. The scene is reconstructed incrementally by registering LF frames with *gP3P* [20] and performing robust multi-view triangulation among the sub-aperture images of two-view-LF frames, taking into consideration the idiosyncrasies of small baselines. Drift is bounded by periodically performing bundle adjustment.

3. The Light Field Projection

In this section, we present our main observation that the normalised disparity is constant for a grid of cameras (Sec. 3.1), and proceed in developing the proposed LF features in Section 3.2. We discuss how these features can be extracted in LF cameras in Section 3.3 and develop the LF projection matrix in Section 3.4.

3.1. The Normalised Disparity

Referring to the simplified 2D case depicted in Figure 2, consider the case where a point $\mathbf{O} = \begin{bmatrix} X & Z \end{bmatrix}^\mathsf{T}$ is observed by camera C_i, $i \in \{1, 2, 3\}$. Let o_i denote $\mathbf{O}$'s corresponding 1D pixel projection, $d_{ji} = o_j - o_i$ and b_{ij} be the baseline between the i-th and j-th camera. From Fig. 2, using the rule of similar triangles it follows that

$$\frac{d_{ij}}{b_{ij}} = \frac{f}{Z}, \qquad (1)$$

where f is the focal length of the cameras. It follows from eq. (1) that for any two cameras that are stereo-rectified, *i.e.* aligned either horizontally or vertically, the ratio between the disparity and their baseline is constant and inversely proportional to the depth of the 3D point they observe. This fact is well-known in the stereo literature, see *e.g.* [37]. Intuitively, one can think that scaling the baseline between cameras (*i.e.* moving the cameras further away or bringing them closer) results in scaling the disparity so that the ratio remains constant.

In general, for a set of N collinear cameras with parallel optical axes and a 3D point $\mathbf{O}_k$, we can rewrite eq. (1) as

$$\forall i, j \in \{1, 2, \ldots, N\}, \; \frac{d_{ij}}{b_{ij}} = \frac{f}{Z_k}. \qquad (2)$$

In total, there are $\binom{N}{2}$ scaled disparity estimates. From hereon, we will refer to the ratio between the disparity and the baseline as the normalised disparity, denoted by ρ.
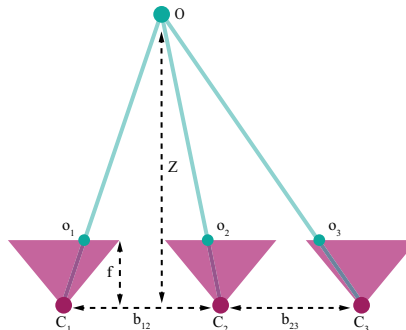

Figure 2. A 3D point projecting on three sub-aperture images arranged on a 1D grid.

3.2. Light Field Features

LF systems are composed of either multiple pinhole cameras arranged in a grid or from micro-lens based LF cameras (whose sub-aperture images after calibration are equivalent to a set of pinhole cameras arranged on a grid). As discussed in Sec. 3.1, for a $N \times N$ grid and a single 3D point we can obtain $2N\binom{N}{2}$ estimates of the normalised disparity ρ (stemming from the vertical and horizontal configurations). We define the homogeneous *LF feature* as:

$$\mathbf{l} = \begin{bmatrix} x_{cent} \\ y_{cent} \\ \rho \\ 1 \end{bmatrix}, \qquad (3)$$

where (x_{cent}, y_{cent}) are the pixel coordinates in the central sub-aperture image. Note that given $\mathbf{l}$, the projections in other sub-aperture images can be obtained simply by multiplying its 3rd element, *i.e.* ρ, by the baseline between the central camera (or sub-aperture image) and the camera (or sub-aperture image) we wish to project to.

3.3. Light Field Features in LF-SfM

The prevalent approach for feature extraction in LFs is to repeatedly apply a 2D detector across sub-aperture images and match the corresponding descriptors, which gives rise to sets of features. Each of these features describes the projections of a 3D point to the sub-aperture images. These sets of features are converted to 3D rays, provided that the LF system is internally calibrated, to be used for either relative or absolute pose estimation. However, we argue that due to the very short baseline of LF cameras, these rays do not provide valuable information [24].

Consider, for example, a LF camera of focal length f and calibration matrix $\mathbf{K}$. Let two matched features be $\mathbf{s}_1 = \begin{bmatrix} u & v & 1 \end{bmatrix}^T = \begin{bmatrix} 192 & 276 & 1 \end{bmatrix}^{T\,1}$ and $\mathbf{s}_2 = \begin{bmatrix} u+d & v & 1 \end{bmatrix}^T$, where d is the disparity between the feature locations. Converting pixels to rays (*i.e.* homogeneous

---

[1] Sub-aperture images in a Lytro Illum camera are 383×552 pixels.

3DV
#143

3DV
#143

3DV 2020 Submission #143. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

coordinates) as $\mathbf{K}^{-1}\mathbf{s}_1$ and $\mathbf{K}^{-1}\mathbf{s}_2$, we observe that the difference in the x-coordinate of the rays is d/f. This means that for disparities less than 1 pixel and a focal length of 800 pixels, the angle between the resulting rays will be $0.07°$, which numerically can as well be considered as the same ray corrupted with noise. As a consequence, the sampled rays from a LF camera are not sufficiently distinct from each other and may lead to significant errors (especially when the LF camera is modeled as a generalised camera).

Departing from the naive and exhaustive feature extraction approach, Dansereau *et al.* [6] recently developed *LiFF*, a SIFT-like feature detection algorithm in which they extend the idea of the standard SIFT detector to the 4D LF. LiFF provides the pixel location, scale, orientation and *slope* of each interest point. For a calibrated LF camera, slope can be mapped to depth [6]. This is very similar to our feature representation of normalised disparity since ρ is inversely proportional to depth.

Our approach can use both types of features, *i.e.* either SIFT features in sub-aperture images or LiFF features. Specifically, one can extract SIFT features in all the sub-aperture images, estimate different values of ρ (resulting, for example, from all pairwise combinations of collinear cameras), and use the median of these values as a robust estimate of ρ. An alternative would be to extract LiFF features and simply invert the depth value obtained from them.

In our experiments, we compute multiple estimates of ρ using the features linked to the same 3D point within a LF frame (a.k.a. intra-frame), as provided by [36]. A LF feature is then obtained by robustly estimating ρ from the median of all corresponding values.

3.4. Light Field Projection Matrix

Let $\mathbf{X}_w = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^{\mathsf{T}}$ be the homogeneous coordinates of a 3D point in the world coordinate frame. Using eq. (3), the homogeneous projection $\mathbf{l}$ of $\mathbf{X}_w$ in a light field can be obtained as:

$$\mathbf{l} = \frac{1}{Z_c} \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 0 & f \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}_w, \qquad (4)$$

where $\mathbf{R}$, $\mathbf{t}$ are respectively the rotation matrix and translation vector specifying the displacement from the world coordinate frame to the LF camera coordinate frame, and (c_x, c_y) is the principal point of the central sub-aperture image. For simplicity, we assume that the intrinsic parameters are the same for all the sub-aperture images[2]. Finally, Z_c is the z-coordinate of the 3D point expressed in the LF camera coordinate frame (*i.e.* after applying the rotation and translation in eq. (4)).

---

[2]If the intrinsic parameters differ among sub-aperture images, they can be made identical by suitable image coordinate transformations.

We denote the *LF Projection Matrix* as:

$$\mathbf{L} = \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 0 & f \\ 0 & 0 & 1 & 0 \end{bmatrix}. \qquad (5)$$

For later use, we also denote with $\mathbf{T}$ the camera pose matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \qquad (6)$$

4. Absolute Pose Estimation

In this section, we estimate absolute pose based on the projection equation defined in Section 3. Specifically, in Section 4.1 we show how we can obtain the projection matrix up to scale. In Section 4.2, we examine the number of constraints provided by a single LF feature. Finally, in Section 4.3 we discuss how to extract translation and rotation from the recovered projection matrix.

4.1. Light Field DLT

Close inspection of the projection equation (4) reveals that there exists an equivalence relation between LF points and their corresponding 3D points. We can rewrite eq. (4) as:

$$Z_c \mathbf{l} = \mathbf{L}\,\mathbf{T}\,\mathbf{X}_w, \qquad (7)$$

or

$$\mathbf{l} \simeq \mathbf{L}\,\mathbf{T}\,\mathbf{X}_w = \mathbf{P}\,\mathbf{X}_w, \qquad (8)$$

with $\simeq$ denoting equality up to a scale factor and $\mathbf{P} \equiv \mathbf{L}\,\mathbf{T}$.

Note that the absolute dual quadric matrix $\mathbf{\Omega} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$ projects to the dual image of the absolute quadric (DIAC) [47, 29]:

$$\omega = \mathbf{P}\,\mathbf{\Omega}\,\mathbf{P}^{\mathsf{T}} = \mathbf{K}\,\mathbf{K}^{\mathsf{T}}, \qquad (9)$$

where $\mathbf{K}$ is the intrinsic parameters matrix:

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (10)$$

Essentially, the upper 3×3 block of $\mathbf{K}$ becomes the upper block of matrix $\mathbf{L}$. We will use eq. (9) to convert constraints on the matrix $\mathbf{K}$ to constraints on candidate projections $\mathbf{P}$.

In order to estimate the pose of the LF camera, we start by rewriting eq. (8) as:

$$\lambda \mathbf{l} = \mathbf{P}\,\mathbf{X}_w, \qquad (11)$$

where λ is an unknown scalar that needs to be eliminated. In the classic Direct Linear Transformation (DLT) algorithm [43], elimination of λ is achieved by taking the cross

3DV
#143

3DV
#143

3DV 2020 Submission #143. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

product of the left and right-hand sides [12]. However, since $\mathbf{l} \in \mathbb{R}^4$, this approach is not feasible here as the cross-product operation is not defined. Yet, DLT can still be applied to eq. (11) using the fact that for any skew-symmetric matrix $\mathbf{S}$ and vector $\mathbf{x}$, the quadratic form associated with $\mathbf{S}$ vanishes[3], i.e. $\mathbf{x}^\mathsf{T} \mathbf{S} \mathbf{x} = 0$. Thus, the scalar λ can be eliminated by forming the inner product of the right-hand side of eq. (11) with $\mathbf{l}^\mathsf{T} \mathbf{S}$. Specifically, given a skew-symmetric matrix $\mathbf{S}$, we obtain:

$$\mathbf{l}^\mathsf{T} \mathbf{S} \mathbf{P} \mathbf{X}_w = 0. \qquad (12)$$

Using the Kronecker product, eq. (12) can be rewritten as:

$$\left(\mathbf{X}_w^T \otimes (\mathbf{l}^\mathsf{T} \mathbf{S})\right) \mathrm{vec}(\mathbf{P}) = 0, \qquad (13)$$

where $\mathrm{vec}(\mathbf{P}) \in \mathbb{R}^{16}$ is the vector formed by stacking the columns of $\mathbf{P}$.

Each choice of a matrix $\mathbf{S}$ provides one constraint on the elements of $\mathbf{P}$. The space of skew-symmetric matrices in $\mathbb{R}^{4 \times 4}$ is six-dimensional[4]. Considering that the general form of a 4×4 skew-symmetric matrix is

$$\mathbf{S} = \begin{bmatrix} 0 & s_1 & s_2 & s_3 \\ -s_1 & 0 & s_4 & s_5 \\ -s_2 & -s_4 & 0 & s_6 \\ -s_3 & -s_5 & -s_6 & 0 \end{bmatrix}, \qquad (14)$$

we can express $\mathbf{S}$ as a linear combination of six basis matrices, i.e.

$$\mathbf{S} = \sum_{i=1}^{6} s_i \mathbf{B}_i. \qquad (15)$$

Each $\mathbf{B}_i$ in eq. (15) is a skew-symmetric matrix, for example

$$\mathbf{B}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (16)$$

Therefore, each $\mathbf{B}_i$ yields one constraint on the elements of $\mathbf{P}$. Note that $\mathbf{P}$ has 3 zero elements in the third row, thus there are 13 unknowns in total.

Given n correspondences, we can estimate $\mathbf{P}$ from the singular value decomposition (SVD) of matrix $\mathbf{A}$ made up of n six-row matrices $\mathbf{A}_i$, i.e.

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{A}_n \end{bmatrix}, \quad \mathbf{A}_i = \begin{bmatrix} \mathbf{X}_{w,i}^T \otimes (\mathbf{l}_i^T \mathbf{B}_1) \\ \mathbf{X}_{w,i}^T \otimes (\mathbf{l}_i^T \mathbf{B}_2) \\ \mathbf{X}_{w,i}^T \otimes (\mathbf{l}_i^T \mathbf{B}_3) \\ \mathbf{X}_{w,i}^T \otimes (\mathbf{l}_i^T \mathbf{B}_4) \\ \mathbf{X}_{w,i}^T \otimes (\mathbf{l}_i^T \mathbf{B}_5) \\ \mathbf{X}_{w,i}^T \otimes (\mathbf{l}_i^T \mathbf{B}_6) \end{bmatrix}. \qquad (17)$$

---

[3] $\mathbf{x}^\mathsf{T} \mathbf{S} \mathbf{x} = \mathbf{x}^\mathsf{T} \mathbf{S}^\mathsf{T} \mathbf{x} \in \mathbb{R}$ and, since $\mathbf{S}^\mathsf{T} = -\mathbf{S}$, $\mathbf{x}^\mathsf{T} \mathbf{S} \mathbf{x} = 0$.

[4] In general, the space of skew-symmetric $n \times n$ matrices over a field has dimension $n(n-1)/2$.

4.2. A Note on Matrix $\mathbf{A}_i$

Consider a sub-block $\mathbf{A}_i$ of $\mathbf{A}$, which originates from a LF point correspondence. We state the following:

Theorem 4.1. *The rank of $\mathbf{A}_i$ is 3.*

Proof. We can rewrite $\mathbf{A}_i$ from (17) as:

$$\mathbf{A}_i = \mathbf{X}_{w,i}^T \otimes \begin{bmatrix} \mathbf{l}_i^T \mathbf{B}_1 \\ \mathbf{l}_i^T \mathbf{B}_2 \\ \mathbf{l}_i^T \mathbf{B}_3 \\ \mathbf{l}_i^T \mathbf{B}_4 \\ \mathbf{l}_i^T \mathbf{B}_5 \\ \mathbf{l}_i^T \mathbf{B}_6 \end{bmatrix} = \mathbf{X}_{w,i}^T \otimes \mathbf{B}_{l_i}. \qquad (18)$$

From the Kronecker product properties, we have that $\mathrm{rank}(\mathbf{X} \otimes \mathbf{Y}) = \mathrm{rank}(\mathbf{X}) \cdot \mathrm{rank}(\mathbf{Y})$ for two general matrices $\mathbf{X}$ and $\mathbf{Y}$. Thus, it follows from eq. (18) that $\mathrm{rank}(\mathbf{A}_i) = \mathrm{rank}(\mathbf{X}_{w,i}^T \otimes \mathbf{B}_{l_i}) = \mathrm{rank}(\mathbf{X}_{w,i}^T) \cdot \mathrm{rank}(\mathbf{B}_{l_i}) = 1 \cdot 3 = 3$. $\qquad \square$

A direct consequence of Theorem 4.1 is that only three of the rows of $\mathbf{A}_i$ are independent and hence each correspondence provides 3 constraints. Since $\mathbf{P}$ consists of 13 uknown elements, we conclude that in total at least 4 correspondences suffice to obtain $\mathbf{P}$ up to scale. Finally, we note that there is no restriction on the choice of the $\mathbf{B}_i$ (as long as the resulting matrix is of rank 3). In our implementation, we choose $\mathbf{B}_1, \mathbf{B}_2$ and $\mathbf{B}_3$.

4.3. Extracting Rotation and Translation

In practice, the rank of $\mathbf{A}$ may vary depending on the geometric configuration of the 3D points (e.g. collinear, coplanar, etc). However, provided a sufficient number of points, the rank of $\mathbf{A}$ is 10, 11 or 12. Thus, $\mathbf{P}$ is obtained from the linear combination of the vectors spanning the null space of $\mathbf{A}$. Reshaping each null vector to a matrix, similarly to [48], we obtain:

$$\mathbf{P}(\boldsymbol{\mu}) = \sum_{i=1}^{n} \mu_i \mathbf{P}_i, \qquad (19)$$

where n is the dimension of the nullspace of $\mathbf{A}$ and μ_i are random scalars and $\boldsymbol{\mu} = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_n \end{bmatrix}^T$

In the case where $\mathrm{rank}(\mathbf{A}) = 12$, i.e. 3D points are in a general configuration, we obtain $\mathbf{P}$ up to a single scalar μ_1. By enforcing the constraints of the dual image quadric (9), we can compute μ_1 with

$$\mu_1 = \sqrt{\frac{\omega_{11}}{[\mathbf{P}\boldsymbol{\Omega}\mathbf{P}^\mathsf{T}]_{11}}}, \qquad (20)$$

where $\mathbf{A}_{ij}$ denotes the element of $\mathbf{A}$ in the i-th row and j-th column. Note that more singular vectors (i.e. $\mathrm{rank}(\mathbf{A}) = 10, 11$) can be handled similarly to [48].

3DV
#143

3DV
#143

3DV 2020 Submission #143. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

In practice, due to noise and numerical errors, it is expected that $\text{rank}(\mathbf{A}) = 13$. The goal in this case is to minimise $\|\mathbf{Ax}\|$ subject to the constraint $\|\mathbf{x}\| = 1$. We can then obtain $\mathbf{P}$ as the eigenvector of $\mathbf{A}^\mathsf{T}\mathbf{A}$ corresponding to the smallest eigenvalue, which is computed via the SVD of $\mathbf{A}$.

Given $\mathbf{P}$, the matrix $\mathbf{T}$ comprising the rotation and translation of the camera with respect to the world coordinate frame (cf. eq. (6)), is obtained as $\mathbf{T} = \mathbf{L}^{-1}\mathbf{P}$.

Note that the matrix $\mathbf{R}$ extracted from $\mathbf{T}$ is not in general orthogonal, as the orthonormality constraints were not enforced during the estimation of $\mathbf{P}$. Thus, we project $\mathbf{R}$ to $\mathcal{SO}(3)$ by solving the nearest orthogonal approximation problem [13, 53] that involves finding the orthogonal matrix that minimises the Frobenius distance from a given matrix. If $\mathbf{R} = \mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V}^T$ is the SVD, the orthogonal matrix nearest to $\mathbf{R}$ is given by $\mathbf{U}\,\mathbf{C}\,\mathbf{V}^T$, with $\mathbf{C} = \text{diag}(1, 1, \det(\mathbf{U}\,\mathbf{V}^T))$. Alternatively, the nearest orthogonal matrix can be computed without matrix factorization as in [28].

5. Results

In this section we evaluate *LF-DLT* with both simulated data and real LF images. Throughout all the experiments, we compute the error between rotation matrices as the amount of rotation needed to bring one rotation matrix to align with the other (i.e., using the geodesic on the unit sphere [15]) and the error between translation vectors using the L_2 norm.

5.1. Simulation Results

Using simulated data, we study the performance of *LF-DLT* under different noise levels in Section 5.1.1, under varying point depths in Section 5.1.2 and compare it with baseline *DLT* in Section 5.1.3 as well as state-of-the-art generalised absolute solvers in Section 5.1.4.

5.1.1 Performance in the presence of noise

To evaluate our algorithm in a challenging scenario, we simulated a realistic LF camera similar to Lytro Illum, whose sub-aperture images are arranged in a 5×5 grid; each sub-aperture image is 500×400 pixels. We set the focal length of the simulated camera to 600 pixels and the baseline between two adjacent cameras to 0.5 mm. For each noise level, we carried out 200 tests, and for each test we randomly selected 50 3D points having a distance between 0.1 m and 10 m from the origin. Note that we did not consider outliers in this experiment, since the purpose of the simulation is to evaluate the performance of *LF-DLT* under different levels of noise. Sub-aperture images that are neighbouring on the grid of a LF image have a very short baseline, resulting in sub-pixel disparities. Hence, as de-
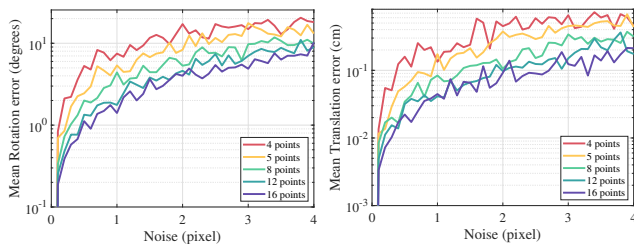

Figure 3. Mean rotation and translation estimation error for *LF-DLT* with various levels of noise and different numbers of points.

tailed in [36], many algorithms fail to correctly recover the motion in this setting.

Figure 3 illustrates the mean rotation error (in degrees) for different numbers of observed 3D points. The minimal case, *i.e.* 4 points, yields significant errors in the presence of noise exceeding 1 pixel. However, the error considerably decreases when using more points ($3°$ error for up to 2 pixels noise). The translation is estimated very accurately even in the presence of 2 pixels noise. Using as few as 5 points, we obtain a translation error on the order of 0.2 cm for up to 2 pixels noise. Considering that the proposed algorithm is linear, these results are particularly encouraging.

5.1.2 Sensitivity to different point depths

Subsequently, we examined the accuracy of the proposed method in three different scenarios for the arrangement of 3D points. This evaluation was also employed in [36] and is critical for motion estimation with small baseline cameras, especially for methods relying on 3D point triangulation from sub-aperture images. Similarly to Sec. 5.1, we randomly selected 50 3D points for which the depth was uniformly distributed in a certain interval. The intervals considered are $0.1 - 0.5$ m, $0.5 - 1$ m, $1 - 4$ m and $5 - 10$ m. Note that points lying further than 3 m result in disparities less than 0.1 pixels. We applied *LF-DLT* using 8 points.

Figure 4 illustrates the error in the estimated rotation and translation. The accuracy of the estimated rotation is not affected by the depth of the scene points. However, the accuracy of the translation estimate decreases as the depth increases. This is expected since it is well-known that larger parallax (decreased depth) results in better estimation of translation. In all cases though, we observe that the pose estimate is quite accurate.

5.1.3 Comparison with the classic DLT

We compare *LF-DLT* with the classic DLT algorithm applied to central sub-aperture images denoted as DLT_c. We consider DLT_c to be the baseline algorithm. We generated random points uniformly distributed between $0.5 - 3$ m and ran both algorithms using the same 3D points as input. For
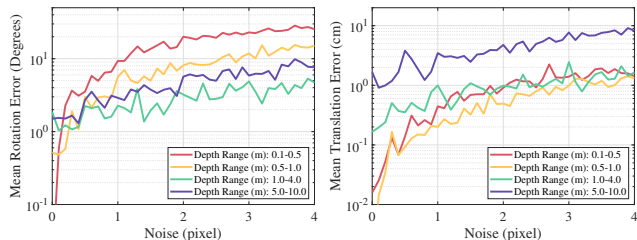
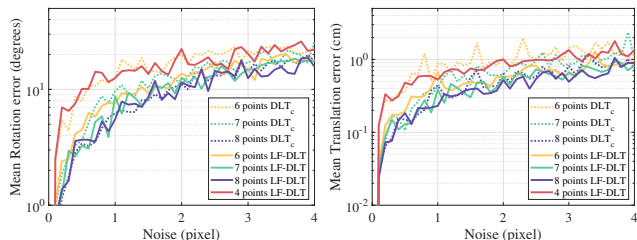Figure 4. Rotation and translation estimation error for varying scene depth for *LF-DLT* using 8 points.

Figure 5. Comparison of LF-DLT and classic DLT on central sub-aperture images for different noise levels and numbers of points.

the minimal case of *LF-DLT*, *i.e.* $n = 4$, we selected subsets of the 6 point samples provided to DLT_c.

Figure 5 depicts the mean rotation and translation errors. We observe that the minimal case for both algorithms behaves similarly. As a consequence, since *LF-DLT* requires less points than DLT_c for its minimal case, *LF-DLT* is more efficient than DLT_c in a RANSAC framework. Furthermore, using the same number of points, especially in the minimal case for DLT_c, *LF-DLT* provides more accurate estimates for both rotation and translation.

5.1.4 Comparison with generalised absolute pose solvers

Using a simulation scenario similar to that in Sec. 5.1, we employed synthetic data to compare the performance of several absolute pose estimation algorithms for generalised cameras. Specifically, we simulated LF frames comprising 5×5 sub-aperture images with a 0.5 mm baseline, employing 50 3D points and performing 200 random tests for each noise level. The percentage of outliers was set to 20%.

We compared the following solvers embedded in RANSAC: the minimal solver *gP3P* [20], *gPnP* [20] which is an n-point solver extending *EPnP* [25] to the non-central case, *g1P2R* [4] which employs one point-point and two point-ray correspondences, and the *UPnP* algorithm of [21]. We employed the authors' implementation[5] for *g1P2R* and OpenGV [19] for the remainder of the algorithms. Since OpenGV uses the angle between the original and re-projected rays as en error metric, we use this criterion for all algorithms including *LF-DLT*.

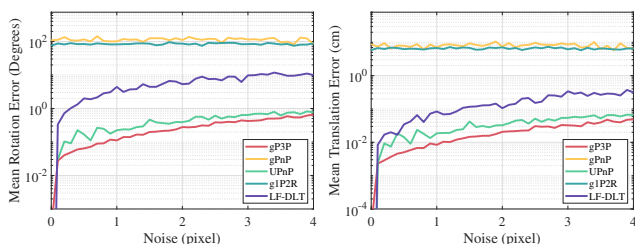---

[5] http://people.inf.ethz.ch/fcampose/publications

Figure 6. Comparison of rotation and translation errors pertaining to different generalised absolute pose estimation algorithms.

Figure 6 illustrates the performance of the algorithms with respect to the mean translation and rotation absolute pose errors. Note that *g1P2R* does not perform well since it locally triangulates a point, which in the case of a LF camera with small baseline does not yield accurate 3D points. What stands out from both figures is that *LF-DLT*, with an estimation error of around $2° - 3°$ in rotation and 0.1 cm in translation, is competitive to iterative polynomial solvers like *gP3P*, while being more computationally efficient due to its linear nature.

5.2. Experiments on real LF datasets

Next, we evaluate the accuracy of *LF-DLT* in real LF datasets from [36], using the poses recovered for them by *uLF-SfM* as reference. Furthermore, using the same data, we compare *LF-DLT* with DLT_c and a state-of-the-art absolute pose minimal solver.

5.2.1 Performance of *LF-DLT*

We evaluated *LF-DLT* on real LF datasets from [36]. Specifically, for each dataset we ran *uLF-SfM* and obtained the camera poses, 3D points and feature matches. The poses from *uLF-SfM* serve as pseudo ground truth. Then, for each dataset, we employed *LF-DLT* using 12 points within RANSAC to register each LF frame using the provided feature matches and the reconstructed 3D points. We used the reprojection error with a threshold of 1.5 pixels to determine inliers.

Table 1 depicts the mean rotation and translation error obtained from *LF-DLT* compared to the pose estimates from *uLF-SfM* for the "Octopus", "House", "Toycar" and "Chameleon" datasets. From all datasets considered, the maximum mean rotation and translation errors amount to $0.19°$ and 0.61 cm respectively. Furthermore, Fig. 7 illustrates the distribution of errors in the two largest "Toycar" and "Chameleon" datasets.

5.2.2 Comparison with central absolute pose solvers

Similarly to Section 5.2, we ran the classic *DLT*, indicated as DLT_c, with 12 points on the central sub-aperture image and *optDLS* [32], which is a state-of-the-art minimal
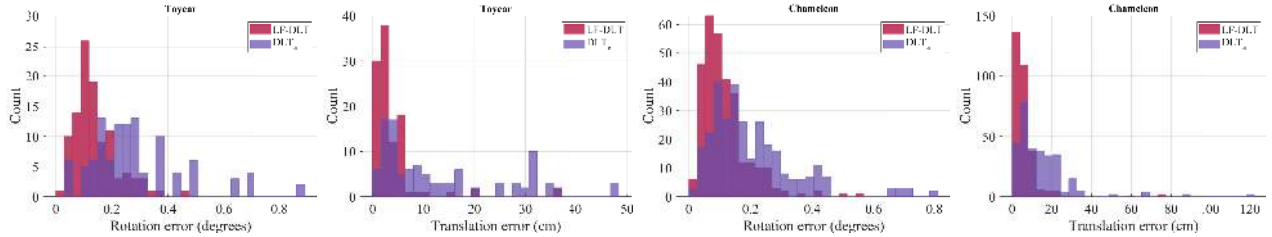
Figure 7. Histograms of rotation and translation errors for *LF-DLT* and DLT_c on "Toycar" and "Chameleon" datasets from [36].

	Rotation Difference (°)				Translation Difference (cm)			
	LF-DLT	DLT_c	*optDLS*	*LF-DLT-O*	*LF-DLT*	DLT_c	*optDLS*	*LF-DLT-O*
Octopus	0.19 ± 0.13	0.22 ± 0.16	0.04 ± 0.05	0.02 ± 0.02	0.61 ± 0.4	0.89 ± 0.63	0.11 ± 0.13	0.06 ± 0.03
House	0.19 ± 0.15	0.25 ± 0.15	0.04 ± 0.05	0.03 ± 0.03	0.45 ± 0.43	0.96 ± 0.49	0.06 ± 0.06	0.05 ± 0.04
Toycar	0.14 ± 0.11	0.25 ± 0.18	0.03 ± 0.03	0.02 ± 0.02	0.44 ± 0.57	0.95 ± 0.84	0.07 ± 0.06	0.05 ± 0.05
Chameleon	0.19 ± 0.15	0.25 ± 0.15	0.04 ± 0.05	0.03 ± 0.03	0.45 ± 0.43	0.96 ± 0.49	0.06 ± 0.06	0.05 ± 0.04

Table 1. Evaluation of *LF-DLT* on real datasets captured with a Lytro Illum camera.

	# LFs	# Registered		# 3D points		Avg. repr. error [pix]	
		uLF-SfM	*Ours*	*uLF-SfM*	*Ours*	*uLF-SfM*	*Ours*
Octopus	7	7	7	1226	978	0.25	0.29
House	16	16	16	1654	1423	0.30	0.34
Toycar	103	103	102	10793	10624	0.49	0.53
Chameleon	303	303	303	28079	27769	0.42	0.45

Table 2. Comparison of reconstruction fidelity when integrating *LF-DLT* in *uLF-SfM*.

solver for the central absolute pose problem. Both algorithms were used within a RANSAC framework with a re-projection threshold of 1.5 pixels. Furthermore, we used the solution obtained from *LF-DLT* to initialise the non-linear minimisation with the Levenberg-Marquardt algorithm of the re-projection error (for the central sub-aperture features only) corresponding to inliers, denoted by *LF-DLT-O*.

Table 1 presents the mean rotation and translation error for all methods with the corresponding variance. It is obvious that *LF-DLT* is superior to DLT_c, especially in the estimate of translation. The error distribution of *LF-DLT* and DLT_c is depicted in Fig. 7. As expected, *LF-DLT* is not as accurate as *optDLS*, since the latter uses a minimal parameterization of the rotation matrix to explicitly enforce the orthonormality constraints, which is not the case for *LF-DLT*. However, using the pose from *LF-DLT* as the initialisation to the non-linear minimisation, we obtained more accurate pose estimates than *optDLS*.

6. Integration in a LF-SfM pipeline

We modified the implementation[6] of *uLF-SfM* to use *LF-DLT* for LF frame registration. Specifically, we substituted *gP3P* with *LF-DLT* embedded in RANSAC (with a reprojection error threshold of 1.5 pixels) to register the subsequent LF frames after initialisation. The final pose of each

---

[6] http://www.github.com/sotnousias/uLF-SfM

LF frame is obtained by minimising the reprojection error of the inliers. Note that in this process, only central features are used for estimation, thus an outlier from RANSAC results in discarding all the features associated with this central feature. As a consequence, correct sub-aperture features may erroneously be discarded. This might explain why *uLF-SfM* reconstructs more 3D points. Furthermore, the inliers consist of central features only but there is no safeguarding against outliers in the associated sub-aperture features in this case. To account for such cases, we discard sub-aperture outliers in the triangulation step of *uLF-SfM*.

Table 2 demonstrates the fidelity of the modified reconstruction using *LF-DLT*. Note that our modifications constitute a very basic adaptation of an incremental pipeline; more sophisticated approaches in point filtering and triangulation using our new LF feature representation are beyond the scope of this paper.

7. Conclusion

This paper has proposed a linear solution to the absolute pose problem for LF cameras. The solution relies on the observation that the ratio between the disparity in different sub-aperture images and the corresponding baseline is constant. In turn, this facilitates a feature representation that allows the absolute pose of a LF frame to be estimated with the Direct Linear Transformation. The latter operates in a non-iterative manner and can accommodate an arbitrary number of corresponding input points. Comprehensive experiments with real and simulated data have demonstrated the effectiveness of the proposed solution and its superior performance compared to the classic *DLT* applied to the central sub-aperture image.

3DV
#143

3DV
#143

3DV 2020 Submission #143. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

References

[1] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20, 1991. 1

[2] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 2

[3] Y. Bok, H.-G. Jeon, and I. S. Kweon. Geometric calibration of micro-lens-based light field cameras using line features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):287–300, Feb 2017. 2

[4] F. Camposeco, T. Sattler, and M. Pollefeys. Minimal solvers for generalized pose and scale estimation from two rays and one point. In *European Conference on Computer Vision (ECCV)*, pages 202–218, 2016. 1, 7

[5] C.-S. Chen and W.-Y. Chang. On pose recovery for generalized visual sensors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(7):848–861, July 2004. 2

[6] D. G. Dansereau, B. Girod, and G. Wetzstein. Liff: Light field features in scale and depth. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[7] D. G. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenslet-based plenoptic cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1027–1034, Jun 2013. 2

[8] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003. 2

[9] R. Garg, N. Wadhwa, S. Ansari, and J. T. Barron. Learning single camera depth estimation using dual-pixels. *ICCV*, 2019. 1

[10] S. Gauglitz, C. Sweeney, J. Ventura, M. Turk, and T. Hollerer. Live tracking and mapping from both general and rotation-only camera motion. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 13–22, Nov 2012. 1

[11] M. D. Grossberg and S. K. Nayar. A general imaging model and a method for finding its parameters. In *Proceedings IEEE International Conference on Computer Vision*, volume 2, pages 108–115, 2001. 1

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003. 5

[13] N. J. Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, pages 1–27, 1989. 6

[14] N. Horanyi and Z. Kato. Generalized pose estimation from line correspondences with known vertical direction. In *International Conference on 3D Vision (3DV)*, pages 244–253, 2017. 2

[15] D. Q. Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, jun 2009. 6

[16] H. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. Tai, and I. S. Kweon. Depth from a light field image with learning-based matching costs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):297–310, 2019. 1

[17] O. Johannsen, A. Sulc, and B. Goldluecke. On linear structure from motion for light field cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 720–728, 2015. 1

[18] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Trans. Graph.*, 35(6), Nov. 2016. 1

[19] L. Kneip and P. Furgale. OpenGV: A unified and generalized approach to real-time calibrated geometric vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, May 2014. 7

[20] L. Kneip, P. Furgale, and R. Siegwart. Using multi-camera systems in robotics: Efficient solutions to the NPnP problem. In *IEEE International Conference on Robotics and Automation*, pages 3770–3776, May 2013. 2, 3, 7

[21] L. Kneip, H. Li, and Y. Seo. UPnP: An optimal O(n) solution to the absolute pose problem with universal applicability. In *European Conference on Computer Vision (ECCV)*, pages 127–142, 2014. 2, 7

[22] Z. Kukelova, J. Heller, and A. Fitzgibbon. Efficient intersection of three quadrics and applications in computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1799–1808, 2016. 2

[23] G. H. Lee, B. Li, M. Pollefeys, and F. Fraundorfer. Minimal solutions for the multi-camera pose estimation problem. *The International Journal of Robotics Research*, 34(7):837–848, 2015. 2

[24] S. H. Lee and J. Civera. Triangulation: Why optimize? In *British Machine Vision Conference*, page 162. BMVA Press, 2019. 3

[25] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166, Jul 2008. 2, 7

[26] M. Levoy. Light fields and computational imaging. *Computer*, 39(8):46–55, Aug. 2006. 1

[27] H. Li, R. Hartley, and J.-H. Kim. A linear approach to motion estimation using generalized camera models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. 3

[28] M. Lourakis and G. Terzakis. Efficient absolute orientation revisited. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5813–5818, 2018. 6

[29] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992. 4

[30] P. Miraldo, H. Araujo, and N. Goncalves. Pose estimation for general cameras using lines. *IEEE transactions on cybernetics*, 45(10):2156–2164, 2015. 1, 2

[31] P. Miraldo, T. Dias, and S. Ramalingam. A minimal closed-form solution for multi-perspective pose estimation using points and lines. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *European Conference on Computer Vision (ECCV)*, pages 490–507, Cham, 2018. Springer International Publishing. 1, 2

3DV
#143

3DV
#143

3DV 2020 Submission #143. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

[32] G. Nakano. Globally optimal DLS method for PnP problem with Cayley parameterization. In *British Machine Vision Conference*, pages 78.1–78.11, 2015. 2, 7

[33] R. Ng. *Digital light field photography*. PhD thesis, Stanford, 2006. 1

[34] D. Nistér and H. Stewénius. A minimal solution to the generalised 3-point pose problem. *Journal of Mathematical Imaging and Vision*, 27(1):67–79, Jan. 2007. 2

[35] S. Nousias, F. Chadebecq, J. Pichat, P. Keane, S. Ourselin, and C. Bergeles. Corner-based geometric calibration of multi-focus plenoptic cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 957–965, 2017. 2

[36] S. Nousias, M. Lourakis, and C. Bergeles. Large-scale, metric structure from motion for unordered light fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3292–3301. Computer Vision Foundation / IEEE, June 2019. 1, 2, 4, 6, 7, 8

[37] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993. 3

[38] R. Pless. Using many cameras as one. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 587–593, 2003. 3

[39] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, Sep 2004. 1

[40] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, June 2016. 1, 2

[41] G. Schweighofer and A. Pinz. Globally optimal O(n) solution to the PnP problem for general camera models. In *British Machine Vision Conference*, pages 1–10, 2008. 2

[42] P. P. Srinivasan, R. Ng, and R. Ramamoorthi. Light field blind motion deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[43] I. E. Sutherland. Three-dimensional data input by tablet. *Proceedings of the IEEE*, 62(4):453–461, 1974. 4

[44] C. Sweeney, J. Flynn, B. Nuernberger, M. Turk, and T. Höllerer. Efficient computation of absolute pose for gravity-aware augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 19–24, 2015. 2

[45] C. Sweeney, V. Fragoso, T. Höllerer, and M. Turk. gDLS: A scalable solution to the generalized pose and scale problem. In *European Conference on Computer Vision (ECCV)*, pages 16–31, 2014. 2

[46] C. Sweeney, V. Fragoso, T. Höllerer, and M. Turk. Large scale SfM with the distributed camera model. In *Fourth International Conference on 3D Vision (3DV)*, pages 230–238, 2016. 1

[47] B. Triggs. Autocalibration and the absolute quadric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 609–614, 1997. 4

[48] B. Triggs. Camera pose and calibration from 4 or 5 known 3D points. In *International Conference on Computer Vision*, pages 278–284. IEEE Computer Society, 1999. 5

[49] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg. A minimal solution to the generalized pose-and-scale problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 422–429, 2014. 1, 2

[50] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, page 765–776, New York, NY, USA, 2005. Association for Computing Machinery. 2

[51] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu. Light field image processing: An overview. *IEEE J. of Selected Topics in Signal Processing*, 11(7):926–954, Oct 2017. 1

[52] Y. Zhang, P. Yu, W. Yang, Y. Ma, and J. Yu. Ray space features for plenoptic structure-from-motion. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4641–4649, Oct 2017. 1, 2

[53] Z. Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, Dec. 1998. 6

[54] Y. Zheng, Y. Kuang, S. Sugimoto, K. Åström, and M. Okutomi. Revisiting the PnP problem: A fast, general and optimal solution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2344–2351, 2013. 2