On Linear Structure from Motion for Light Field Cameras

Ole Johannsen, Antonin Sulc and Bastian Goldluecke
University of Konstanz, Germany

Abstract

We present a novel approach to relative pose estimation which is tailored to 4D light field cameras. From the relationships between scene geometry and light field structure and an analysis of the light field projection in terms of Plücker ray coordinates, we deduce a set of linear constraints on ray space correspondences between a pair of light field cameras. These can be applied to infer relative pose of the light field cameras and thus obtain a point cloud reconstruction of the scene. While the proposed method has interesting relationships to pose estimation for generalized cameras based on ray-to-ray correspondence, our experiments demonstrate that our approach is both more accurate and computationally more efficient. It also compares favorably to direct linear pose estimation based on aligning the 3D point clouds obtained by reconstructing depth for each individual light field. To further validate the method, we employ the pose estimates to merge light fields captured with hand-held consumer light field cameras into refocusable panoramas.

1. Introduction

While the concept of light field cameras has been known since the beginning of the 20th century [19, 10], only recent progress in sensor technology and computing power paved the way to implementations in the form of market-ready digital cameras [21, 23, 31]. In contrast to conventional cameras, a light field camera records both spatial as well as angular information about incident light. This enables sophisticated post-processing, and one can for example virtually change focus or perspective [21], or estimate depth maps from a single shot [32, 14].

Given this emerging paradigm for digital photography, it is interesting to compare the alignment problem for traditional 2D images and 4D light fields. Per-pixel alignment of 2D images requires per-pixel depth, and it is impossible to find the complete 2D to 2D image transformation from sparse feature correspondence alone. In contrast, it is easy to see that the individual rays in a light field can be transformed into the ray space of a second light field using only information about relative pose, see figure 3. Not only will this allow for particularly robust pose estimation, but creating refocusable light field panoramas is also an excellent visual verification of the accuracy of the pose estimate.

This work was supported by the ERC Starting Grant "Light Field Imaging and Analysis" (LIA 336978, FP7-2014).

Figure 1. One application of the proposed method is efficient alignment of light fields obtained with a hand-held consumer plenoptic camera into a common light field panorama. Given enough views, high-quality refocusing of the panorama is possible.

Work on plenoptic camera calibration was so far mostly devoted to obtaining a calibration of the intrinsic parameters, which assigns recorded luminance information to actual rays in 3D space [6, 12, 2]. In this work, we will thus assume to have an internal calibration performed. While after calibration, camera pose with respect to a fixed camera coordinate system is usually known, the problem of computing light field camera pose given two arbitrary light fields of the same scene has so far not been dealt with explicitly. For small motion, tracking camera pose for visual odometry via the plenoptic flow was discussed in [5], while our approach is also suitable for wide-angle matching.

Nevertheless, a lot of previous work on structure-from-motion still applies to our scenario as well. We will give a short outline of the most relevant methods now, and delve into their technical details in the main part of the paper.

Related work on structure-from-motion. Techniques for estimating camera motion and scene structure from multiple images have been perfected over the past three decades [8, 25], up to the point that it now works on large

scale internet photo collections [1]. Reliable technology is available that can serve as a starting point to produce relightable models which are getting close to being indistinguishable from their real-world counterparts for human observers [27]. However, these frameworks are tailored to 2D perspective projections, and thus not directly applicable to correspondences between the ray spaces of light field camera views.

There has been comparatively less work on pose estimation beyond pinhole cameras. The maybe most general linear framework defines a generalized camera as an unordered collection of rays which is captured by its sensor elements [24]. Correspondences need to be established between rays which are assumed to intersect the same scene point, leading to a generalized epipolar constraint in terms of Plücker ray coordinates. In general configurations, 17 ray-to-ray correspondences are sufficient to allow a pose estimate [30, 18]. This also can be applied to light field cameras of course, however, we will see that we can obtain more accurate results using our approach. The main reason is that we also take into account the relationships between projections within a single light field, which contain inherent information about the 3D scene structure.

In fact, it is already possible to obtain quite accurate dense depth maps from a single light field [32, 14], and pose estimation can be performed by aligning point clouds, as common for RGB+D cameras like the Microsoft Kinect [11]. In this paper, however, the focus is on sparse methods, where we wish to avoid an expensive dense reconstruction step. Thus, we only work with a sparse set of reliable feature matches. We require multiple occurrences of each feature in both light fields, so that in principle, sparse 3D point clouds with one-to-one correspondence information can be reconstructed directly. While linear pose estimation from these registered 3D point clouds is straightforward [13, 9, 22], we show that our proposed framework easily beats this approach in accuracy.

Contributions. We investigate the problem of estimating relative pose for light field cameras, and formulate a mathematical framework for linear structure-from-motion. It is based on two key observations. First, when describing rays in Pluecker coordinates, the projection into *homogeneous* light field coordinates is a linear map. Second, a projection of a 3D scene point in a 4D light field is a two-dimensional linear subspace. Together, these two yield linear correspondence constraints between rays in the first light field and subspaces in the second. As far as we are aware, this is both a previously unexplored insight and the first systematic treatment of the structure-from-motion problem which is tailored to 4D light fields.

While pose estimation is in principle also possible using any of the previously existing approaches for pinhole or generalized cameras, we experimentally validate that using our approach, which takes into account the specific light field geometry, leads to significant increase in accuracy and robustness. Of the many possible applications, we investigate creating panoramic light fields from individual ones captured with a hand-held Lytro consumer camera, see figure 1. This type of problem ideally fits the metric we minimize for pose estimation, and gives visual confirmation of the alignment accuracy.

2. Light field correspondence

We first give a more detailed outline of our work and its contributions to establish notation and context for the remainder of the paper.

Light field cameras and coordinates. A calibrated light field camera samples luminance for a known subset of the rays passing through its aperture. We parametrize the rays which are recorded in the 4D light field captured by the camera in relative two-plane coordinates [4, 17]. In this parametrization, each ray $\boldsymbol{r}$ is described by coordinates $\boldsymbol{l} = [u, v, s, t]^T \in \mathbb{R}^4$, which encode the intersection of $\boldsymbol{r}$ with two distinct planes Π and Ω in space. Points in 3D space are denoted by $\boldsymbol{X} = [X, Y, Z]^T$.

In the standardized reference frame of the light field camera, we consider the *focal plane* Π to be the XY-plane, while the *image plane* Ω lies parallel to Π at a distance equal to the *focal length* f in positive Z-direction. The pair (s, t) is given by the first two coordinates of the intersection of $\boldsymbol{r}$ with Π. The pair (u, v) are the first two coordinates of the intersection of $\boldsymbol{r}$ with Ω, but are relative to (s, t) in the sense that the origin on Ω lies at (s, t, f), see figure 2. This corresponds to image coordinates of a pinhole camera with center of projection at $(s, t, 0)$ and optical axis parallel to the Z-axis. A view from such a camera is called a *subaperture image*, and a light field can be considered as a collection of subaperture images, i.e. standard perspective images, with slightly shifted view points.

In case we consider two light field cameras, we assume the coordinate system of the first camera to be aligned with world space. All objects related to the second camera are written with a prime symbol. We assume that the coordinate frame $\boldsymbol{X}'$ of the second camera is related to the first by a rigid motion, $\boldsymbol{X}' = R\boldsymbol{X} + t$, with rotation $R \in SO(3)$ and $t \in \mathbb{R}^3$.

Light field correspondence. In a classical pinhole camera image, a single 3D point is projected onto a unique 2D point. Thus, a correspondence between two views is a relation between one 2D point in the first view and a second 2D point in the second view. In contrast, a light field camera samples many subaperture views, and thus multiple rays emanating from the same scene point. In consequence, a light field correspondence consists of a *list of light field co-*

ordinates for each of the recorded light fields,

$$\{\boldsymbol{l}_i\}_{i=1,\dots n} \;\leftrightarrow\; \{\boldsymbol{l}'_j\}_{j=1,\dots,m}. \tag{1}$$

For a valid correspondence, all rays in both lists must come from the same scene point, so given two light fields, candidates are for example SIFT feature matches across all subaperture images. Note that if the light field cameras are internally calibrated and $n, m \geq 2$, it is already possible to triangulate a 3D point in camera coordinates for both sides of the correspondence.

Strategies for structure-from-motion. From a set of correspondences of the form (1), one can immediately formulate two promising strategies in order to recover relative camera pose:

(i) Consider each ray-to-ray correspondence individually, and apply a method based on the framework of generalized cameras [18]. This is discussed in section 3.

(ii) From the left hand and right hand side of (1), compute corresponding 3D scene points for each individual light field, then determine pose by aligning the two point clouds. This is discussed in section 4.

In section 5, we will establish a third possibility and then demonstrate its merits. We transform each ray into the respective other light field and derive a linear set of constraints from all the corresponding rays in this domain. The resulting set of equations has interesting similarities to the set of equations from the generalized epipolar constraint, but implicitly takes into account information on 3D scene structure inherent in a single light field. Thus, we believe it is an ideal unification of both ideas - indeed, we will demonstrate in section 6 that it outperforms both methods sketched above.

3. Ray correspondence in generalized cameras

A light field camera can be understood in the context of generalized camera models [24]. In this framework, a camera is described by the set of rays which it samples in its coordinate frame, and pixel correspondence is generalized to ray correspondence. It turns out that mathematically elegant correspondence equations arise from describing these rays in terms of Plücker coordinates. We will briefly review the central parts of the theory leading to the system of equations for pose estimation. Details and proofs can be found in the literature [29].

Rays in space. The Plücker coordinates of a ray $\boldsymbol{r}$ are given by a pair $(\boldsymbol{q}{:}\boldsymbol{m})$ of vectors, with direction $\boldsymbol{q} \in \mathbb{R}^3 \setminus \{0\}$ and moment $\boldsymbol{m} \in \mathbb{R}^3$. A point $\boldsymbol{X} \in \mathbb{R}^3$ lies on the ray iff

$$\boldsymbol{m} = \boldsymbol{X} \times \boldsymbol{q}.$$

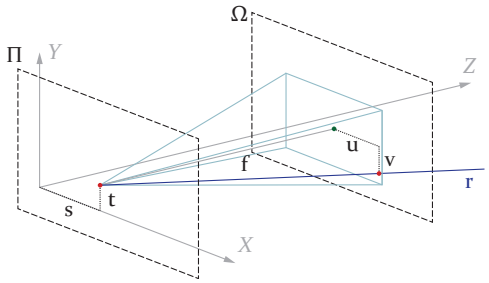

Figure 2. *Light field parametrization*. An incident ray $\boldsymbol{r}$ is parametrized by its intersections with the *focal plane* Π and the *image plane* Ω (red dots). The planes are parallel with distance equal to the focal length f. The intersection coordinates (s, t) are given in relation to the origin of the world coordinate system. The coordinates (u, v) are given relative to the intersection of the optical axis of a virtual camera placed at $(s, t, 0)$ in Z direction with the second plane (green dot). Each of these virtual cameras gives a subaperture view of the light field.

Two sets of coordinates $(\boldsymbol{q}{:}\boldsymbol{m})$ and $(\boldsymbol{q}'{:}\boldsymbol{m}')$ define the same ray if there exists $w \neq 0$ such that

$$\boldsymbol{q} = w\boldsymbol{q}' \text{ and } \boldsymbol{m} = w\boldsymbol{m}'.$$

Thus, ray coordinates can be considered as homogeneous coordinates. In upcoming formulas, the symbol $\boldsymbol{r}$ will denote the 6D column vector obtained by stacking direction on top of moment.

Transformations of $\mathbb{R}^3$ and ray coordinates. We consider the case that space undergoes a rigid motion given by rotation R and translation t,

$$\boldsymbol{X}' = R\boldsymbol{X} + t. \tag{2}$$

In this case, transformed Plücker ray coordinates can be computed as

$$(\boldsymbol{q}'{:}\boldsymbol{m}') = \left(R\boldsymbol{q}{:}R\boldsymbol{m} + [t]_\times R\boldsymbol{q}\right), \tag{3}$$

or in block matrix form,

$$\boldsymbol{r}' = \begin{bmatrix} R & 0 \\ E & R \end{bmatrix} \boldsymbol{r} = \begin{bmatrix} I_3 & 0 \\ [t]_\times & I_3 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \boldsymbol{r}, \tag{4}$$

where one can see the decomposition into pure rotation (applied first) and translation (applied second). The matrix $E := [t]_\times R$ is called the *essential matrix*.

Generalized epipolar constraint. Two rays $(\boldsymbol{q}_1{:}\boldsymbol{m}_1)$ and $(\boldsymbol{q}_2{:}\boldsymbol{m}_2)$ given in the same coordinate frame intersect iff

$$\boldsymbol{q}_1^T\boldsymbol{m}_2 + \boldsymbol{m}_1^T\boldsymbol{q}_2 = 0. \tag{5}$$

Consider now the setting of generalized cameras, with $(\boldsymbol{q}{:}\boldsymbol{m})$ a ray in the coordinate frame of the first camera. If $(\boldsymbol{q}{:}\boldsymbol{m})$ is transformed into the coordinate frame of the second camera according to (3), it should intersect each corresponding ray $(\boldsymbol{q}'{:}\boldsymbol{m}')$ in a single 3D scene point.

Substituting (3) into (5) applied in the coordinate frame of the second camera, we obtain the *generalized epipolar constraint* [24]

$$\boldsymbol{q}'^T E \boldsymbol{q} + \boldsymbol{q}'^T R \boldsymbol{m} + \boldsymbol{m}'^T R \boldsymbol{q} = 0. \tag{6}$$

which needs to be satisfied by every ray-to-ray correspondence $(\boldsymbol{q}{:}\boldsymbol{m}) \leftrightarrow (\boldsymbol{q}'{:}\boldsymbol{m}')$.

Light field camera pose from ray correspondence. Given a list of light field correspondences of the form (1) with n rays in the first and m rays in the second light field, we obtain $n{\cdot}m$ equations from the generalized epipolar constraint (6) which are linear in the rotation and essential matrix coefficients. In [18], a method was proposed to recover the pose parameters R and t from this system of equations. In section 5, we give an overview of the implementation and also suggest an improvement regarding the numerical technique.

4. The light field projection

In this section, we will first show that the projection from Plücker rays in world space to homogeneous light field coordinates is projective linear. Second, we derive the linear 2D subspace in homogeneous light field coordinates which is the projection of a single scene point. Together, both results will be used to construct a set of two linear constraints on the rigid motion per ray in a light field correspondence in section 5.

Intersections of rays with the light field planes. Planes in space can be described by a homogeneous 4D vector $\hat{\boldsymbol{a}} = (\boldsymbol{a}; \alpha)$ with $\boldsymbol{a} \in \mathbb{R}^3$ and $\alpha \in \mathbb{R}$. A point $\boldsymbol{X}$ lies on the plane iff its homogeneous coordinates $\hat{\boldsymbol{X}}$ satisfy $\hat{\boldsymbol{a}}^T \hat{\boldsymbol{X}} = 0$. One can show that a Plücker ray $(\boldsymbol{q}{:}\boldsymbol{m})$ intersects with the plane in the point with homogeneous coordinates

$$\hat{\boldsymbol{X}} = ([\boldsymbol{a}]_\times \boldsymbol{m} - \alpha \boldsymbol{q};\ \boldsymbol{a}\boldsymbol{q}). \tag{7}$$

Note that the focal plane Π of the light field corresponds to $\boldsymbol{a}_\Pi = (\boldsymbol{e}_3; 0)$, while the image plane Ω is parametrized by $\boldsymbol{a}_\Omega = (\boldsymbol{e}_3; -f)$. Substituting into (7), we find the intersection points of a Plücker ray with these planes are

$$\boldsymbol{X}_\Pi = (-\frac{m_2}{q_3}, \frac{m_1}{q_3}, 0) \text{ and } \boldsymbol{X}_\Omega = (\frac{f q_1 - m_2}{q_3}, \frac{f q_2 + m_1}{q_3}, f) \tag{8}$$

in $\mathbb{R}^3$, respectively.

5D homogeneous light field coordinates from a Plücker ray. Projecting a Plücker ray into the light field requires divisions by q_3. This is analogous to the pinhole projection, which required division by Z. Just like in this case, it is advantageous to switch to homogeneous coordinates in order to obtain linear projection equations. Thus, we parametrize a single ray in the light field with homogeneous 5D light field coordinates $\hat{\boldsymbol{l}} = (u, v, s, t, 1)^T$.

One can now read off the projection equation for a light field camera in the camera coordinate system from (8) as

$$q_3 \begin{bmatrix} u \\ v \\ s \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 & 0 & 0 \\ 0 & f & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{m} \end{bmatrix}. \tag{9}$$

Note that in order to obtain (u, v)-coordinates in the reference frame relative to (s, t), one needs to subtract $\boldsymbol{X}_\Pi$ from $\boldsymbol{X}_\Omega$. We call the 5×6 matrix above the *light field projection* $P(f)$. It depends only on the focal length.

Projections of a single scene point. Consider a point $\boldsymbol{X}$ in world space. For fixed (s, t), the coordinates (u, v) are computed according to a pinhole projection though a camera located at $(s, t, 0)$ with image plane Ω. The pinhole projection equations impose an affine relationship between (u, v) and (s, t), see figure 2 and e.g. [8, 7], which can be written in homogeneous light field coordinates as

$$\underbrace{\begin{bmatrix} 1 & 0 & \frac{f}{Z} & 0 & -\frac{fX}{Z} \\ 0 & 1 & 0 & \frac{f}{Z} & -\frac{fY}{Z} \end{bmatrix}}_{=:M(\boldsymbol{X}, f)} \begin{bmatrix} u \\ v \\ s \\ t \\ 1 \end{bmatrix} = 0. \tag{10}$$

In particular, the set of all rays intersecting a single scene point forms a linear 2D subspace of the homogeneous light field coordinate domain.

Recovering the 2D subspace from correspondences. From a set of feature correspondences of the form (1), it is straight-forward to obtain an estimate of the subspace matrices for both light fields. We solve (10) for the three unknown coefficients of M or M' given the lists of ray correspondences $\{\boldsymbol{l}_i\}_{i=1,\dots n}$ or $\{\boldsymbol{l}'_j\}_{j=1,\dots,m}$, respectively, in a least-squares sense. For greater robustness, it is advisable to employ a RANSAC scheme when using real-world data. Some outliers can also be efficiently discarded in advance, as all matches within a given light field must lie in a certain disparity range.

Recovering and aligning two 3D point clouds. From estimates of $M(\boldsymbol{X}, f)$ and $M'(\boldsymbol{X}', f)$ for a single correspondence, one immediately obtains a pair of corresponding 3D points $\boldsymbol{X} \leftrightarrow \boldsymbol{X}'$. An obvious way to estimate camera pose is thus to align the two corresponding 3D point clouds estimated from the list of correspondences. For this alignment problem, several algorithms have been proposed in the literature [13, 9, 22]. However, pose estimation via point cloud alignment turns out to be not very robust, as due to the small baseline, the estimate of 3D points is very sensitive to small errors in feature locations. Both the framework of generalized cameras as well as our novel method introduced in the next section easily beats it in terms of accuracy.

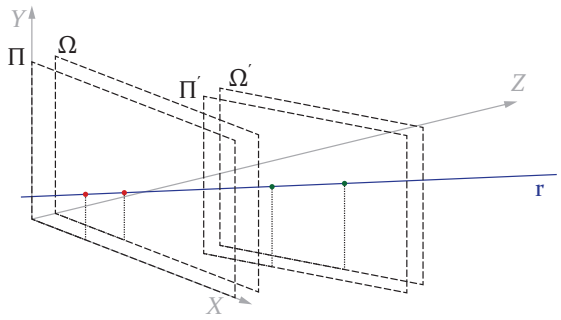

Figure 3. *Ray transformation.* A ray $\boldsymbol{r}$ given in coordinates of one light field (red dots) will also intersect the two planes describing another light field in general orientation (green dots). A key observation leading to the proposed method is that if the ray is given in Plücker coordinates, then the projection into the coordinates of the second light field is projective linear, see equations (4) and (9).

5. Recovering light field camera pose

In this section, we first describe the proposed system of equations to recover pose given a set of light field correspondences of the form (1). Although arising from a different principle, it will turn out to be of the same structure as system (6) in the related works on generalized cameras. In the second part of the section, we therefore describe how this type of system can be solved for R and t. While we could in principle employ the exact algorithm from previous work, we introduce a variant which turns out to substantially improve numerical accuracy.

Let us now consider a single correspondence, and assume we have estimated subspaces M and M' for both light fields from (10) in the last section. If a fixed ray $\boldsymbol{l}$ on the left hand side is transformed into a ray $\hat{\boldsymbol{l}}'$ by computing the projection in the second light field, then the projection must also satisfy the subspace constraint (10), i.e. $M'\hat{\boldsymbol{l}}' = 0$, as all corresponding rays intersect in the same scene point. Writing $\boldsymbol{l}$ in Plücker coordinates $(\boldsymbol{q}{:}\boldsymbol{m})$, using the ray transformation (4) and projection (9), this expands to

$$M'P(f)\begin{bmatrix} R & 0 \\ E & R \end{bmatrix}\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{m} \end{bmatrix} = 0. \tag{11}$$

We abbreviate with M_1 the first three and with M_2 the second three columns of the 2×6 matrix $M'P(f)$, which leads to the simplified form

$$M_1 R\boldsymbol{q} + M_2 R\boldsymbol{m} + M_1 E\boldsymbol{q} = 0. \tag{12}$$

Accordingly, each ray in the correspondence yields two homogeneous linear equations in the components of R and E.

Discussion, relation to previous work. Both the generalized epipolar constraint (6) and the new proposed subspace constraint (12) leads to a linear system of the form

$$A_E\text{vec}(E) + A_R\text{vec}(R) = 0, \tag{13}$$

where $\text{vec}(E)$ and $\text{vec}(R)$ are vectors of all components in E and R, respectively, and A_E and A_R the coefficient matrices resulting from the set of all correspondence constraints. The correct solution for R is a rotation and the essential matrix is of the form $E = [t]_\times R$.

Previous work [18] on pose estimation from generalized cameras proposes a numerical method where first the essential matrix E is recovered, from which one obtains the rotation R using a decomposition step [8]. The arising twisted pair ambiguity can be resolved uniquely by choosing the solution which leads to a smaller residual in (13). The reason given in [18] for solving for E instead of R is to avoid degeneracies from certain camera configurations. In the case of typical feature matches from light field cameras, we found that these degeneracies do not arise in our setting, and it turns out that is much more robust to recover R directly and completely ignore E. The numerical technique is the same as in [18], but applied to the other variable.

Solving for R and t. The discussion in [8] (section 9.6), shows that finding the solution to (13) subject to $||\text{vec}(R)|| = 1$ is equivalent to solving

$$(A_E A_E^+ - I)A_R\text{vec}(R) = \mathbf{0}. \tag{14}$$

Thus, to recover R, we first compute the last column of V in the SVD of $(A_E A_E^+ - I)A_R$, rearrange it into a matrix, and project the result onto the space of rotation matrices using the method in [9]. Upon publication, source code will be provided on our web page for implementation details. Substituting the resulting R into (13) now leads to a linear system in t. The least squares solution is computed again with the SVD technique, after which the initial pose estimate is complete.

Refinement iterations. The initial linear estimate (R_0, t_0) for rotation and translation usually does not exactly solve (13), since the correct solution for R and t is subject to a non-linear set of constraints. As also suggested in [18], we therefore use (R_0, t_0) as an initial estimate for the solution of the minimization problem

$$\min_{R\in SO(3), r\in\mathbb{R}^3} \left\{A_E\,[t]_\times\,\text{vec}(R) + A_R\text{vec}(R)\right\}, \tag{15}$$

and iterate the following two steps until convergence:

1. Minimize the energy with fixed t_n for unconstrained unknown R. The solution is then projected onto $SO(3)$ to obtain R_{n+1}.
2. Minimize the energy with fixed R_{n+1} for unknown t to obtain t_{n+1}.

All sub-problems are simple linear problems, which can be solved e.g. with the SVD technique above. The iteration sequence leads to a local minimizer of the energy, which fits the correspondence constraints more accurately than the linear solution.

	Correspondences	10 matches, 10 rays per point					40 matches, 10 rays per point					10 matches, 20 rays per point				
	Noise level σ_{uv}	0.2	0.4	0.6	0.8	time[s]	0.2	0.4	0.6	0.8	time[s]	0.2	0.4	0.6	0.8	time[s]
Angular rot. error [deg]	*linear methods*															
	3DPC	1.31	4.23	5.78	9.30	0.00	1.01	2.96	5.24	7.55	0.01	1.38	2.18	5.38	7.76	0.00
	R2R-O	1.55	4.81	7.31	11.65	0.07	0.70	1.35	2.68	3.55	0.51	1.34	2.38	7.11	9.34	0.56
	R2R-I	0.69	1.73	2.59	4.29	0.09	0.38	0.88	1.91	3.29	0.57	0.49	1.17	2.37	4.17	0.59
	Proposed	0.58	1.27	1.59	2.20	0.04	0.27	**0.40**	0.78	1.14	0.18	**0.29**	**0.78**	1.14	1.65	0.07
	iterative methods															
	3DPC-RANSAC	1.31	3.24	5.13	5.93	0.03	0.79	1.79	2.31	4.84	0.12	0.88	1.70	3.93	6.12	0.13
	MIN-RANSAC	1.40	3.27	3.91	6.36	49.18	1.02	2.21	3.09	3.86	194.89	1.20	2.55	4.41	4.27	197.11
	R2R-O-R20	0.68	1.63	3.13	4.27	1.51	0.37	0.91	1.79	3.54	9.48	0.49	1.17	2.49	4.42	10.62
	R2R-I-R20	0.66	1.62	2.79	4.07	1.69	0.37	0.91	1.80	3.55	10.06	0.53	1.11	2.56	4.26	10.32
	Proposed-R20	**0.49**	**1.22**	**1.54**	**2.13**	0.99	**0.23**	0.42	**0.74**	**1.05**	3.65	0.33	0.87	**1.00**	**1.62**	1.97
Angular transl. error [deg]	*linear methods*															
	3DPC	9.49	13.84	25.72	24.31	0.00	7.50	15.58	18.54	33.27	0.01	9.85	15.43	16.62	23.45	0.00
	R2R-O	2.37	4.73	6.53	15.45	0.07	0.87	1.90	4.59	3.95	0.51	1.49	1.96	6.68	13.08	0.56
	R2R-I	1.25	2.32	3.98	6.81	0.09	0.67	1.05	3.64	4.11	0.57	0.96	2.11	2.92	5.60	0.59
	Proposed	1.22	1.95	**2.36**	**3.32**	0.04	**0.52**	1.13	**1.39**	1.99	0.18	**0.59**	**1.29**	2.13	2.79	0.07
	iterative methods															
	3DPC-RANSAC	7.55	8.72	11.27	16.67	0.03	3.22	5.88	9.06	14.33	0.12	7.05	5.06	10.99	11.48	0.13
	MIN-RANSAC	3.03	4.77	5.42	10.98	49.18	2.11	3.83	4.23	5.22	194.89	2.56	4.35	6.58	6.79	197.11
	R2R-O-R20	1.19	2.27	4.47	7.41	1.51	0.61	1.26	3.40	4.76	9.48	0.86	1.94	3.45	6.23	10.62
	R2R-I-R20	1.16	2.22	3.95	6.89	1.69	0.61	1.26	3.42	4.78	10.06	0.90	1.78	3.06	5.75	10.32
	Proposed-R20	**1.15**	**1.83**	2.58	3.57	0.99	0.55	**1.01**	1.52	**1.87**	3.65	0.66	1.33	**2.05**	**2.62**	1.97

Figure 4. *Accuracy of the different methods both before and after non-linear refinement.* Different numbers of correspondences N, projections per correspondence K, and levels of noise σ_{uv} on the (u, v)-coordinates are compared. Error metrics are the mean angular deviation from the ground truth in degrees for the estimated rotation as well translation vector for 50 random data sets. Noise standard deviation is given in units of pixels on the subaperture images. In all cases, the most accurate method (highlighted in bold) is the one proposed in this paper. Note that iterative refinement can only marginally improve the result for the methods which employ the proposed numerical scheme, while it makes a huge difference for the previous method R2R-O.

Of course, if we require more accuracy, the results of the initial estimate can be used to initialize further iterative refinement via non-linear bundle adjustment. Since this step can likewise be applied to all methods, it is not further evaluated in this paper.

6. Experimental comparison

For the experimental comparison, we generate random sets of feature matches for two light fields. We vary the number N of available correspondences of the form (1) as well as number of projections per light field K, which we assume to be the same for both light fields, i.e. $K = n = m$ in (1). The light field geometry is set to be similar to the internal calibration data of the Lytro camera, so that we obtain plausible input close to real-world scenarios. We perturb the exact light field projections with additive Gaussian noise, however only on the (u, v) domain, as correspondences are always estimated in fixed subaperture images. Rotation between the two light fields is chosen at random with a maximum angle of 45 degrees, while translation is also chosen randomly, but with the additional constraint that observation of common scene points in both light fields is possible. For each parameter set, we average results from 50 different random data sets generated in this way.

Comparison of overall accuracy. In the first run of experiments, we compare the complete set of algorithms which have been described in the previous sections. First, these are the purely linear ones denoted as follows:

3DPC	3D point cloud alignment via [13], as discussed in section 4.
R2R-O	Ray-to-ray matching, section 3, implemented as in [18].
R2R-I	Ray-to-ray matching, section 3, with our numerical improvements.
Proposed	Our proposed method, section 5.

Figure 5. The four linear methods we compare against.

For all except 3DPC, we also tested the improvement from 20 additional refinement iterations. These variants are denoted by an additional "-R20" after their descriptor. Complete results for a variety of different parameters can be found in figure 4. Note that the maximum amount of noise is quite high and usually above what one would expect, but it allows testing the limits of the methods.

Finally, we compare against two minimal estimators embedded in a RANSAC framework. The first one uses [13] with three 3D points (3DPC-RANSAC). As an example for a non-linear minimal algorithm [28, 15], we use [28] based on six ray correspondences. Source code for the core method is provided by the authors, we re-implemented the

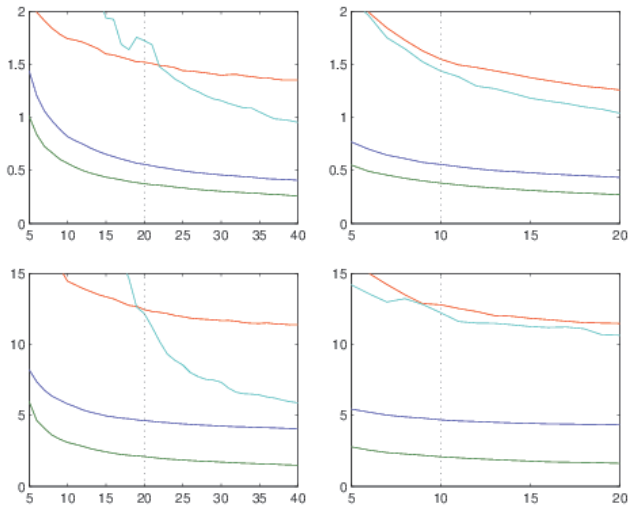

Figure 6. The graphs show how the angular error in rotation depends on the number of matches (left) and the number of rays per match (right). Compared are the four linear methods in table 5: 3DPC [13] (red) and R2R-O [18] (cyan), R2R-I with our proposed numerical improvements (blue), and finally the novel proposed method for 4D light fields (green). Top row: small amount of noise ($\sigma = 0.2$), bottom: large amount of noise ($\sigma = 1.0$).

suggested 100 RANSAC iterations (MIN-RANSAC).

For each and every parameter combination, the proposed method was the most accurate, often by a large margin. Interestingly, already the linear variant is usually sufficient. The non-linear refinement iterations, while substantially improving the results for the original method R2R-O, often give only marginal improvement and sometimes even reduce final accuracy. This is also the case for R2R-I, so we believe it is the improved numerical scheme which makes non-linear refinement obsolete and our proposed method thus much more efficient.

The closest competitor to our proposed method is matching based on the generalized epipolar constraint. The suggested numerical variant R2R-I consistently leads to much better results than R2R-O if used as a purely linear method. However, if enough matches are available, then non-linear refinement corrects the initial results from R2R-O, so that both lead to almost the same results.

Somewhat surprisingly, matching based on 3D point cloud alignment is no contender and completely breaks down in the presence of noise and for smaller rotation angles. In both cases, it fails to recover remotely accurate translation.

Influence of number of matches on accuracy. In a second run of experiments, we investigate how accuracy increases with the number of available correspondences. We plot angular error over the number N of available correspondences as well as over the number of projections per light field K. The results for the four linear algorithms in table 5 for two different levels of noise can be observed in figure 6. In the case that we vary N, we set $K = 10$, when we vary K, we set $N = 20$, as indicated by the dotted lines in the graphs.

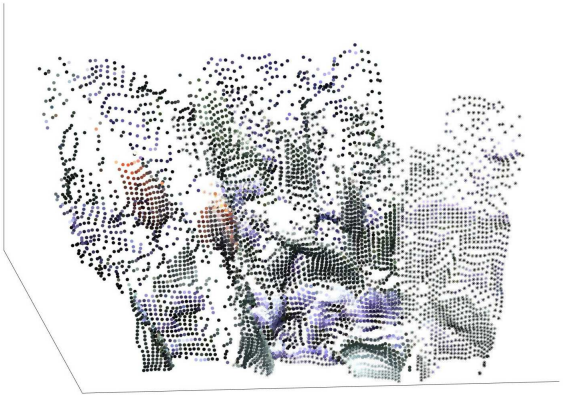

Figure 7. Rendering of a point cloud of the scene captured in figure 8. For two of the frames used for the panorama depth maps were computed. Points from the reference frame are visualized with a $*$-symbol, points from the aligned second frame with circles. See additional material for an animated high resolution version of the point cloud and further visualizations of the geometry.

Our proposed method is again the most accurate, again followed by R2R-I, i.e. [18] augmented by the suggested numerical improvements. The methods from previous work are significantly outclassed by these two. In general, it can be said that it is preferable to have fewer but more precise matches. However, with less than around 15-20 correspondences available, results become significantly less robust, even if there are in theory enough pairs of rays available - the likely reason is that all rays within the same light field belonging to a single correspondence (1) lie very close to each other in ray space due to the small baseline between the subaperture views.

Computational efficiency. Although we attain higher accuracy, for the purpose light field cameras the proposed method is computationally also more efficient than previous work on ray-to-ray correspondence [18]. For one, thanks to the change in numerical method, there is usually no real need for non-linear refinement iterations anymore, as shown above. More importantly, though, a single correspondence of the form (1) leads to $n \cdot m$ linear equations in the framework in [18], but only $2(n + m)$ linear equations for the proposed method. Thus, our method scales much better with the number of projections per correspondence, as it naturally removes a lot of the redundancy which ray-to-ray matching causes in this scenario.

7. Living panoramas with the Lytro camera

In this section, we show how to employ the pose estimates from our light field structure-from-motion pipeline to create refocusable panoramas out of several light fields

Figure 8. *Living panorama from 21 light field images*. In the regions which are in focus, one can observe the precise alignment of the individual rays. Note that a ghosting effect becomes visible in some out of focus regions, which is caused by undersampling of rays in the (s, t) domain. Without more sophisticated post-processing beyond the scope of this work, these artifacts can only be reduced by capturing more data. Since many light fields from only slightly different view points overlap, it is also possible to virtually increase the aperture. This can be observed at the center of the panorama, where the effect of depth-of-field is stronger than on the borders, where data from only one light field is available. See additional material for a video with an animated focus plane.

recorded with the Lytro Illum consumer plenoptic camera. Lytro dubbed their light fields "living pictures", so we refer to these as "living panoramas".

As far as we are aware, existing work on light field panorama stitching is so far not built on structure-from-motion principles for light field cameras. Instead, brute force search over the parameter space is performed to optimally align the individual light fields according to a photo-consistency score [3]. Often, large light fields are assembled from sequences of regular 2D images, where camera pose can be estimated by structure from motion approaches for conventional cameras [16, 26].

Determining light field features. In order to detect the necessary feature correspondences, we employ the well known SIFT algorithm [20]. We first compute SIFT features for each subaperture view of each light field individually. By searching for matches of the descriptors, we can assemble the feature locations into correspondences of the form (1) which we require for further processing. The input light field with the most overlap (measured in number of feature matches) to the other ones is selected as the reference light field.

Alignment and refocusing. After obtaining correspondences, we can run the proposed pose reconstruction pipeline, first estimating the 2D projection subspaces in equation (10) for all of the input light fields and correspondences, then estimating rotation and translation for each light field compared to the reference frame as explained in section 5. After this, we can transform each individual measured ray into a single reference coordinate system via equations (4) and (9), see figure 3.

This allows us to generate views of the assembled light field panorama with a synthetic focus setting. To create such a view, we consider a focus plane parallel to the parameterising planes Ω and Π. In order to assign a color to a pixel on this plane, we sample all available rays which pass though its area. If a point on the focus plane lies on the surface of an object, all the individual rays agree with each other, resulting in a sharp image of the scene. However, if the point lies in front or behind of a surface, the intersecting rays belong to different scene points, resulting in a blurred rendering.

Refocused views can be observed in figure 8, as well as in the videos in the additional material. See also figure 7 for a visualization of the corresponding 3D point cloud.

8. Conclusion

In this paper, we present a novel framework for linear structure-from-motion for light field cameras. In contrast to previous work on generalized cameras [24, 18], which employs an epipolar constraint based on ray-to-ray matches, we make use of the inherent 3D information encoded in the light field structure, and obtain a linear set of constraints from the transformation of rays in the first light field onto corresponding 2D subspaces within the ray space of the second light field.

The proposed approach not only reduces computational complexity, but also leads to pose estimates which are more accurate and more robust to noise, as can be observed in numerous numerical experiments. In addition, the precise alignment of multiple individual light fields captured with a consumer plenoptic camera is verified by stitching them into a refocusable "living panorama", which also increases the virtual aperture and the effect of depth-of-field.

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. Seitz, and R. Szeliski. Building Rome in a Day. *Commun. ACM*, 54(10):105–112, 2011. 2

[2] F. Bergamasco, A. Albarelli, L. Cosmo, A. Torsello, E. Rodolà, and D. Cremers. Adopting an Unconstrained Ray Model in Light-field Cameras for 3D Shape Reconstruction. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2015. 1

[3] C. Birklbauer and O. Bimber. Panorama Light-Field Imaging. *Computer Graphics Forum (Proc. Eurographics)*, 33(2):43–52, 2014. 8

[4] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987. 2

[5] D. G. Dansereau, I. Mahon, O. Pizarro, and S. Williams. Plenoptic flow: Closed-form visual odometry for light field cameras. In *IEEE Intelligent Robots and Systems (IROS)*, pages 4455–4462, 2011. 1

[6] D. G. Dansereau, O. Pizarro, and S. Williams. Decoding, Calibration and Rectification for Lenselet-Based Plenoptic Cameras. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 1027–1034, 2013. 1

[7] B. Goldluecke and S. Wanner. The Variational Structure of Disparity and Regularization of 4D Light Fields. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2013. 4

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 1, 4, 5

[9] B. Horn, H. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, 1988. 2, 4, 5

[10] F. E. Ives. Parallax stereogram and process of making same. *US Patent 725,567*, 04/14/1903. 1

[11] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proc. 24th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 559–568, 2011. 2

[12] O. Johannsen, C. Heinze, B. Goldluecke, and C. Perwass. On the Calibration of Focused Plenoptic Cameras. In *GCPR Workshop on Imaging New Modalities*, 2013. 1

[13] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, 32(922):827–828, 1976. 2, 4, 6, 7

[14] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross. Scene Reconstruction from High Spatio-Angular Resolution Light Fields. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 32(4), 2013. 1, 2

[15] L. Kneip, H. Li, and Y. Seo. UPnP: An Optimal O(n) Solution to the Absolute Pose Problem with Universal Applicability. In *Proc. European Conference on Computer Vision*, 2014. 6

[16] R. Koch, M. Pollefeys, B. Heigl, L. Van Gool, and H. Niemann. Calibration of Hand-held Camera Sequences for Plenoptic Modeling. In *Proc. International Conference on Computer Vision*, volume 1, pages 585–591, 1999. 8

[17] M. Levoy and P. Hanrahan. Light Field Rendering. In *Proc. SIGGRAPH*, pages 31–42, 1996. 2

[18] H. Li, R. Hartley, and J. H. Kim. A linear approach to motion estimation using generalized camera models. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2008. 2, 3, 4, 5, 6, 7, 8

[19] G. Lippmann. Épreuves réversibles donnant la sensation du relief. *J. Phys. Theor. Appl.*, 7(1):821–825, 1908. 1

[20] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 8

[21] R. Ng. *Digital Light Field Photography*. PhD thesis, Stanford University, 2006. 1

[22] N. Ohta and K. Kanatani. Optimal Estimation of Three-Dimensional Rotation and Reliability Evaluation. *IEICE Transactions on Information and Systems*, 81(11):1247–1252, 1998. 2, 4

[23] C. Perwass and L. Wietzke. The Next Generation of Photography, 2010. 1

[24] R. Pless. Using Many Cameras as One. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages II: 587–593, 2003. 2, 3, 4, 8

[25] M. Pollefeys, L. Van 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, 2004. 1

[26] C. Richardt, Y. Pritch, H. Zimmer, and A. Sorkine-Hornung. Megastereo: Constructing High-Resolution Stereo Panoramas. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 1256–1263, 2013. 8

[27] Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. Seitz. The Visual Turing Test for Scene Reconstruction. In *Proc. International Conference on 3D Vision (3DV)*, 2013. 2

[28] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to Minimal Generalized Relative Pose Problems. In *Workshop on Omnidirectional Vision (OMNIVIS)*, 2005. 6

[29] J. Stolfi. *Primitives for Computational Geometry*. PhD thesis, Stanford University, Stanford, CA, USA, 1988. AAI8826243. 3

[30] P. Sturm. Multi-View Geometry for General Camera Models. In *Proc. International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 206–212, 2005. 2

[31] K. Venkataraman, D. Lelescu, J. Duparre, A. McMahon, G. Molina, P. Chatterjee, and R. Mullis. PiCam: An Ultra-Thin High Performance Monolithic Camera Array. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 32(5), 2013. 1

[32] S. Wanner and B. Goldluecke. Variational Light Field Analysis for Disparity Estimation and Super-Resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):606–619, 2014. 1, 2