Scene Flow Estimation from Sparse Light Fields Using a Local 4D Affine Model

Pierre David, Mikael Le Pendu, Christine Guillemot

Scene Flow Estimation from Sparse Light Fields Using a Local 4D Affine Model

Pierre David, Mikaël Le Pendu, and Christine Guillemot, *Fellow, IEEE*

***Abstract*—In this paper, we address the problem of scene flow estimation from sparsely sampled video light fields. We first propose a local 4D affine model to represent scene flows, taking into account light field epipolar geometry. The model parameters are estimated per cluster in the 4D ray space. They are derived by fitting the model on initial motion and disparity estimates obtained by using 2D dense optical flow estimation techniques. We demonstrate that the model is very effective for estimating scene flows from 2D optical flows. The model regularizes the optical flows and disparity maps, and interpolates disparity variation values in occluded regions. The proposed model allows us to benefit from deep learning-based 2D optical flow estimation methods while ensuring scene flow geometry consistency in the 4 dimensions of the light field.**

***Index Terms*—Scene flow, Optical flow, Disparity estimation, Light field.**

I. Introduction

LIGHT fields, by capturing light rays emitted by a scene along different orientations, enable a variety of computer vision applications, and in particular 3D scene modeling. While the problem of depth estimation for 3D scene modeling has already been widely investigated [1]–[5], the possibility to estimate the motion in a 3D scene from light fields remains widely open, despite the numerous applications, e.g., for robot navigation, human-computer interfaces, augmented and virtual reality.

The measured displacement of each point in the 3D scene is referred to as a dense scene flow, concept that has first been defined in [6]. Considering a multi-view set-up, the scene flow is estimated using an optical flow estimator for each view. The 3D scene flow is then computed by fitting its projection on each view to the estimated optical flows, hence is defined by the real 3D motion $(\Delta X, \Delta Y, \Delta Z)$ of each 3D point. However, in the recent literature (e.g. [7]–[10]), the scene flow is instead defined as a direct extension of the optical flow, where the depth (or disparity) d and the depth variation Δd of objects along time is represented in addition to the apparent 2D motion $(\Delta x, \Delta y)$.

The problem of scene flow analysis has first been addressed for stereo video sequences. The authors in [7]–[9] estimate a scene flow $(\Delta x, \Delta y, \Delta d, d)$ assuming that the scene can be decomposed into rigidly moving objects and using discrete-continuous optimization techniques. Several methods based on RGB-D videos have also been developed [10]–[12]. The first methods for scene flow analysis from light fields have been proposed in [13] and [14], based on variational models. The authors in [15] propose oriented light field windows to estimate the scene flow from a dense light field. All these methods rely on epipolar plane images hence are only applicable to densely sampled light fields (as those captured with plenoptic cameras). They are not suitable for sparse light fields (i.e. with large baselines), as for example those captured by rigs of cameras.

In this paper, we focus on the problem of scene flow analysis from large baseline video light fields. This problem is made difficult due to the large temporal and angular occlusions. Recent work (detailed in Section II) has shown the important benefits of using deep learning for estimating optical flows, disparity maps or scene flows from stereo images or videos. However, extending and training network architectures that would take light fields as input are challenging. First, using light fields as inputs would increase the complexity of the architecture. Then, training a deep neural network typically requires very large datasets, particularly for high dimensional data such as light fields. Only a few video light field datasets are available, which is insufficient for performing unsupervised learning. Furthermore, none of these datasets contain ground truth optical flows, disparity maps or scene flows which would be necessary in the context of supervised learning.

To cope with the above difficulties, we propose a 4D local affine model for scene flow estimation. The model is defined in the ray space and incorporates epipolar constraints to ensure consistency of the scene flow on all light field views. We show how the proposed model can be used for regularizing initial and independently computed optical flows and disparity maps in order to derive a coherent scene flow. To estimate the model, we first perform a 4D over-segmentation of the light field at time t, then we compute initial optical flows, disparity maps and disparity variation estimates between the light field at time t and $t+1$. For each 4D cluster, the parameters of the affine model are estimated by fitting the model on the initial optical flow and disparity estimates. The approach is summarized in Figure 1. The proposed method and the corresponding 4D affine model allow us to benefit from state-of-the-art deep learning-based optical flow estimation methods while ensuring scene flow geometry consistency in the 4 dimensions of the light field. Note that a simplified version of the model without the geometrical constraints, has been presented in [16] within the context of sparse-to-dense interpolation.

In order to validate the proposed model on sparse light fields, we have created synthetic video light fields based on the Sintel movie (used in the optical flow benchmark [17], [18]).

This work was supported by the EU H2020 Research and Innovation Program under grant agreement N°694122 (ERC advanced grant CLIM).

P. David and C. Guillemot are with Inria Centre de Recherche Rennes - Bretagne Atlantique, Campus universitaire de Beaulieu, 35042 Rennes Cedex (e-mail:christine.guillemot@inria.fr)

M. Le Pendu is with the School of Computer Science and Statistics, Trinity College Dublin, University of Dublin, Dublin 2, D02 PN40 Ireland.

Manuscript received ; revised .

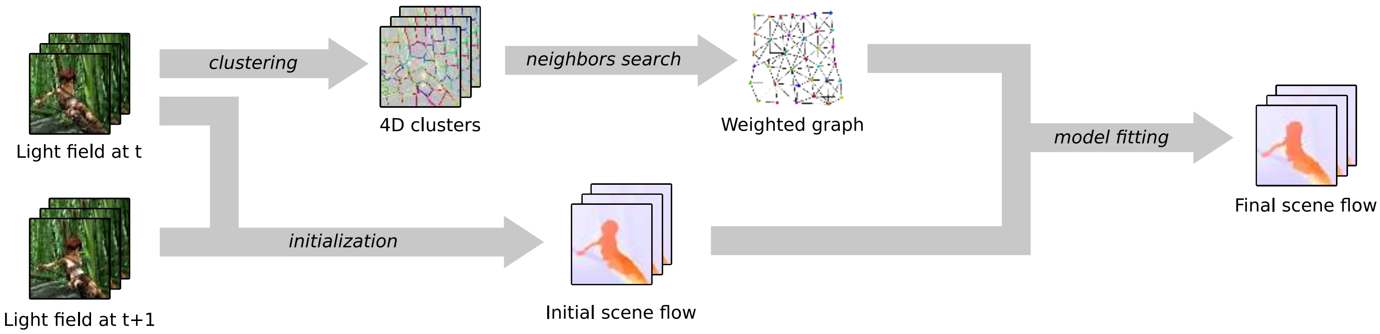

Fig. 1. Block diagram of our method. The light field at time t is first partitioned into 4D clusters. Then, we build a weighted graph where a node, an edge and the associated weight respectively represent a cluster, a connection between two adjacent clusters and the distance between their respective centroids. Using the graph, for each cluster, we search for the closest neighbors. An initial scene flow estimation is simultaneously performed using the light field at t and $t+1$. Finally, for each cluster, we fit an affine scene flow model, using the initial scene flow estimates contained inside the cluster and its neighbors.

The light field views are provided with the corresponding ground truth scene flow (optical flow, disparity and disparity variation).

Although the proposed scene flow method is designed for sparse light fields, we also assess our method on a dense light field video dataset provided by the authors of [19]. For the sparse dataset, the obtained scene flows are compared against those computed with the oriented window method in [15], with the sparse-to-dense interpolation method in [16] and with various stereo scene flow methods [7], [8]. We also compared the estimated optical flow, disparity maps and disparity variation with the one given by a state-of-the-art optical flow estimation technique based on a deep learning architecture called PWC-Net [20]. For the dense dataset, we compared our results with the full view method in [19], with the aforementioned stereo scene flow methods [7], [8] and with various light field depth estimation methods like [3], [21], [22].

Our scene flow estimation outperforms any other tested method in terms of accuracy of the estimated optical flow, disparity, and disparity variation for the sparse dataset, and achieves comparable results to state-of-the-art methods for the dense dataset.

II. Background and related work

Before reviewing prior work on scene flow estimation from multi-view captures and from light fields, this section gives a quick overview of recent methods proposed for solving two strongly related problems, i.e., scene depth estimation from light fields but also optical flow estimation from videos.

A. Scene depth estimation from light fields

With dense light fields with small baselines, pixels in the different views corresponding to the same 3D point form a line in the EPI, whose slope is proportional to the disparity between the views [23]. This observation naturally led to estimating scene depth (related to the disparity or parallax between the views) by analyzing the Epipolar Plane Images (EPI) of dense light fields. The authors in [1] use structure tensors to locally estimate these slopes, this local estimation being then placed in a global optimization framework using a variational approach. The authors in [2] propose a spinning parallelogram operator for disparity estimation from EPIs, accompanied with a confidence measure to handle ambiguities and occlusions.

While the above methods are well suited for dense light fields, they fail in the case of light fields with large baselines for which stereo matching and optical flow estimation techniques yield more accurate estimates. To give a few examples, the authors in [3] estimate disparity by computing a matching cost volume between the central sub-aperture image and sub-aperture images warped using the phase shift theorem. The approach in [5] consists in estimating disparities between the four corner views, then propagating them to the target viewpoint. The authors in [4] employ an empirical Bayesian framework to estimate scene-dependent parameters for inferring scene disparity.

We have recently seen the emergence of deep learning solutions, using in particular convolutional network architectures, for scene depth estimation from light fields. The architectures proposed in [24], [25] operate on EPI, hence are well suited for dense light fields only. A deep neural network, called Dispnet, is proposed in [26] based on the optical flow estimation network Flownet2 [27] but computing 1D correlation instead of 2D correlation to be better suited for disparity estimation. The authors in [28] propose a learning based depth estimation framework suitable for both densely and sparsely sampled light fields, that can learn depth maps for every viewpoint from any subset of input views.

B. Optical flow estimation from videos

Optical flow estimation and stereo matching have been prominent issues in computer vision for years. In order to compare the different methods, benchmark data sets have been proposed. The two most popular datasets are the MPI Sintel Dataset [17] and the KITTI Benchmark [29]. The first one consists in synthetic sequences taken from the movie Sintel. The second one consists in video sequences captured from a moving car, and is therefore better suited for autonomous driving applications.

When looking at the top ranking optical flow estimation methods with the two datasets, we can see that they are almost exclusively using a deep learning approach. To only cite a few methods, FlowNet [30] was the first end-to-end neural network to compute an optical flow from images. It is a trainable encoder-decoder network. The authors in [27] further improve the network by stacking multiple encoder-decoder networks. However, the final network is much bigger than the original one and needs to be trained sequentially for each encoder-decoder part to avoid over-fitting. To reduce the size of the network and make it easier to train, a coarse-to-fine strategy, and the corresponding network called SpyNet, were proposed in [31]. Finally, the authors in [20], as in [31], take advantage of coarse-to-fine approaches, and add a partial cost volume computation in their network, named PWC-Net. It is currently one of the top ranking optical flow methods in the MPI Sintel benchmark.

C. Scene flow estimation

The most common way of estimating scene flow is by using stereo images. The authors in [7] propose a slanted-plane scene flow model for objects in a 3D scene, within the context of autonomous driving. They assume that the scene is composed of a small number of rigidly moving objects and perform a joint segmentation and scene flow estimation. In order to estimate the scene flow model, a discrete-continuous conditional random field is optimized with particle belief propagation [32]. A scene flow model representing the scene with piecewise planar and rigidly moving regions is proposed in [8]. The authors in [9] propose a conditional random field (CRF) based model for robust 3D scene flow estimation. The approach estimates so called *instance scene flows*, i.e. scene flows of 3D points that are geometrically and semantically grouped into instances, using a CNN.

While the models used in these methods are not completely specific for autonomous driving applications, they are however optimized and tested on the KITTI Benchmark [29]. Using any of these methods on other scenes than driving scenes may require some changes in the parameters of the models.

Another way to estimate a scene flow is by using RGB-D images. In [11], local and global constraints are combined in a variational framework to estimate a scene flow, assuming a locally rigid motion. The authors use the depth map to regularize the final scene flow with an adaptive total variation formulation. Similarly to [7], the authors in [10] jointly perform segmentation and 3D motion estimation. The scene is decomposed into depth layers to handle occlusions and a scene flow model is computed for each layer. The method in [12] first performs geometric segmentation and then jointly estimates odometry and scene flow by isolating the static clusters.

Scene flow estimation from densely sampled light fields was first tackled in [13]. The authors jointly estimate the disparity and the optical flow assuming piecewise smoothness of the scene flow. A preconditioned primal-dual algorithm is used to solve a convex global energy functional, which also enforces consistency between the multiple views. On the other hand, the authors in [14], [15] and [33] first estimate the geometry of the scene by computing a disparity map and then estimate the apparent motion in the scene. In [14], the disparity value in each point of the EPIs is derived by analyzing the structure tensor. The optical flow is estimated by minimizing an energy function that assumes spatio-angular smoothness and and takes into account occlusions between objects in the scene. The authors in [15] also use an EPI-based method [34] to compute the disparity maps at time t and $t+1$. Then, they use oriented light-field windows along with a coarse-to-fine strategy to minimize an energy function derived from SimpleFlow [35]. A confidence measure is computed and used to regularize the scene flow in the coarse-to-fine iterations. The authors in [33] fist estimate a disparity map using [1] and then recover a 3D scene flow solving a linear flow equation for each ray. This equation, which relies on 4D light field gradients, is under-constrained, so a global and a local approach are combined in order to solve it. The local one is derived from Lucas-Kanade [36] and the global one from Horn-Schunck [37]. The authors in [38] estimate a scene flow to construct a 4D spatio-temporally coherent representation of dynamic scenes from sparse light fields. First, a 3D point cloud is estimated, then every point is back-projected to a more densely sampled virtual light field, and the resulting EPIs are used to compute the scene flow using the oriented window approach [15]. Finally, the authors in [19] use light field super-pixels and their slanted-planes representation in 3D space to propagate and optimize an optical flow and a disparity map from the central view to every other view. The method is mostly fit for dense light fields because accurately computing the normal of the 3D slanted-plane for every super-pixels requires to have a dense set of views.

Our method computes a dense scene flow for every ray of a light field. It is based on views instead of EPIs and therefore it is suitable for sparsely sampled light fields. Using a 4D affine model to represent the scene flow in the light field, we jointly estimate an optical flow, a disparity map and a disparity variation map.

III. 4D Affine Model

Let us consider the 4D representation of Light Fields proposed in [39] and [40] to describe the radiance along the different light rays. This 4D function, at each time instant t, is denoted $LF^t(u,v,x,y)$. The pairs (u,v) and (x,y) respectively denote the angular and spatial coordinates of light rays. A view (u,v) of a light field at t is written L^t_{uv}. In the article, we assume that the vertical and horizontal baselines are the same. The scene flow can be divided in the following components:

- the optical flow: $F = \begin{pmatrix}\Delta x & \Delta y\end{pmatrix}^\top$,
- the disparity at time t: d^t,
- the disparity variation between t and $t+1$: Δd.

In this paper, we propose a local 4D affine model to represent a scene flow in a light field. The fundamental affine model can be defined as follows:

$$\Delta x(u,v,x,y) = \theta^0_1 u + \theta^0_2 v + \theta^0_3 x + \theta^0_4 y + \theta^0_5, \tag{1}$$

$$\Delta y(u,v,x,y) = \theta^0_6 u + \theta^0_7 v + \theta^0_8 x + \theta^0_9 y + \theta^0_{10}, \tag{2}$$

$$d^t(u,v,x,y) = \theta^0_{11} u + \theta^0_{12} v + \theta^0_{13} x + \theta^0_{14} y + \theta^0_{15}, \tag{3}$$

$$\Delta d(u,v,x,y) = \theta^0_{16} u + \theta^0_{17} v + \theta^0_{18} x + \theta^0_{19} y + \theta^0_{20}, \tag{4}$$

where $\boldsymbol{\theta^0} = \begin{pmatrix} \theta_1^0 & \dots & \theta_{20}^0 \end{pmatrix}^\top$ are the parameters of the model.

However, this model does not take into account epipolar geometry of light fields, i.e. the fact that a 3D point in a scene is projected on 1D lines in EPIs, the slope of these lines being directly related to inter-view disparity. Hence, we derive in this section the equations for a reduced affine model that also satisfies the epipolar constraints.

A. *Constraints on the Optical Flow*

First, let us consider the vertical epipolar constraints. Given a non-occluded point in the 3D scene, we denote by P_0, P_1, P_2 and P_3 its respective projections on the views (u, v) and $(u, v + \Delta v)$ at the time instants t and $t + 1$, as shown in Fig. 2.

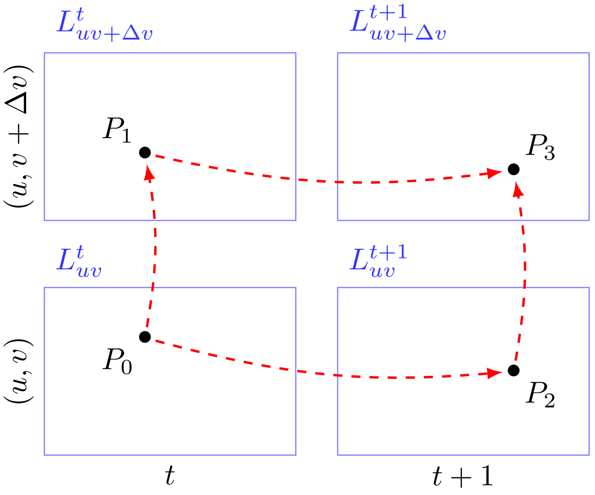

Fig. 2. Projections of one 3D scene point on 2 views of the light field at time instants t and $t + 1$

The coordinates of the points P_0, P_1, P_2 and P_3 in the 4-dimensional space (u, v, x, y) are then related with the following equation:

$$\begin{cases} P_1 = P_0 + \Delta v \cdot \boldsymbol{\nu}(P_0, t), \\ P_2 = P_0 + \boldsymbol{\phi}(P_0), \\ P_3 = P_2 + \Delta v \cdot \boldsymbol{\nu}(P_2, t + 1), \\ P_3 = P_1 + \boldsymbol{\phi}(P_1), \end{cases} \tag{5}$$

where $\boldsymbol{\nu}(P, t)$ and $\boldsymbol{\phi}(P)$ are 4-dimensional vectors representing respectively the orientation of the vertical epipolar line passing by a point P at time t, and the optical flow of P from time t to $t + 1$. These vectors are expressed as:

$$\boldsymbol{\nu}(P, t) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ d^t(P) \end{pmatrix} \text{ and } \boldsymbol{\phi}(P) = \begin{pmatrix} 0 \\ 0 \\ \Delta x(P) \\ \Delta y(P) \end{pmatrix}. \tag{6}$$

We can derive from Eq. (5) that the optical flow vectors $\boldsymbol{\phi}(P_0)$ and $\boldsymbol{\phi}(P_1)$ must satisfy the following equality to be angularly consistent:

$$\boldsymbol{\phi}(P_1) - \boldsymbol{\phi}(P_0) = \Delta v \cdot [\boldsymbol{\nu}(P_2, t + 1) - \boldsymbol{\nu}(P_0, t)]. \tag{7}$$

From the definition of $\boldsymbol{\nu}$ and $\boldsymbol{\phi}$ in Eq. (6), this equality can be rewritten:

$$\begin{cases} \Delta x(P_1) - \Delta x(P_0) = 0, \\ \Delta y(P_1) - \Delta y(P_0) = \Delta v \cdot \Delta d(P_0), \end{cases} \tag{8}$$

where we define $\Delta d(P_0) = d^{t+1}(P_2) - d^t(P_0)$.

Let us now reintegrate these constraints into the affine model. Knowing the relationship between the coordinates of P_0 and P_1 in Eq. (5) and the expressions of Δx and Δy in Eqs. (1) and (2), we can express the variation of optical flow between P_0 and P_1 (i.e. along a vertical epipolar line) as a function of the model's parameters:

$$\begin{cases} \Delta x(P_1) - \Delta x(P_0) = \Delta v \left(\theta_2^0 + \theta_4^0 \times d^t(P_0)\right), \\ \Delta y(P_1) - \Delta y(P_0) = \Delta v \left(\theta_7^0 + \theta_9^0 \times d^t(P_0)\right). \end{cases} \tag{9}$$

By combining Eqs. (8) and (9), we obtain the following constraints on the model's parameters:

$$\theta_2^0 + \theta_4^0 \times d^t(P_0) = 0, \tag{10}$$
$$\theta_7^0 + \theta_9^0 \times d^t(P_0) = \Delta d(P_0). \tag{11}$$

Similarly, horizontal epipolar constraints give:

$$\theta_1^0 + \theta_3^0 \times d^t(P_0) = \Delta d(P_0), \tag{12}$$
$$\theta_6^0 + \theta_8^0 \times d^t(P_0) = 0. \tag{13}$$

Note that one could directly replace $d^t(P_0)$ and $\Delta d(P_0)$ by their expressions in Equations (3) and (4). However, the model would lose its linearity and become more complex to solve. Instead, we choose to approximate the disparity $d^t(P_0)$ by a pre-estimated disparity value $\overline{d}$. The derivation of $\overline{d}$ is detailed in Section IV (see Equation (43)). We also eliminate $\Delta d(P_0)$ by taking the difference between Equations (11) and (12). We can then simplify our model and reduce the number of parameters as

$$\begin{aligned} \theta_2^0 &= -\theta_4^0 \times \overline{d}, \\ \theta_7^0 &= \theta_1^0 + \theta_3^0 \times \overline{d} - \theta_9^0 \times \overline{d}, \\ \theta_6^0 &= -\theta_8^0 \times \overline{d}. \end{aligned} \tag{14}$$

So, the optical flow model becomes

$$\begin{aligned} \Delta x(P_0) &= \theta_1^0 u + \theta_3^0 x + \theta_4^0 \times (y - \overline{d}v) + \theta_5^0, \\ \Delta y(P_0) &= \theta_1^0 v + \theta_3^0 \overline{d} v + \theta_8^0 (x - \overline{d}u) + \theta_9^0 (y - \overline{d}v) + \theta_{10}^0. \end{aligned} \tag{15}$$

B. *Constraints on the Disparity and Disparity Variation*

Furthermore, we can also reduce the number of parameters of the disparity and the disparity variation models. The epipolar geometry of a light field requires that the disparity remains constant along a vertical or horizontal epipolar line. This constraint gives the following equations:

$$\begin{aligned} d^t(P_0 + \Delta v \cdot \boldsymbol{\nu}(P_0, t)) - d^t(P_0) &= 0, \\ d^t(P_0 + \Delta u \cdot \boldsymbol{\mu}(P_0, t)) - d^t(P_0) &= 0, \\ \Delta d(P_0 + \Delta v \cdot \boldsymbol{\nu}(P_0, t)) - \Delta d(P_0) &= 0, \\ \Delta d(P_0 + \Delta u \cdot \boldsymbol{\mu}(P_0, t)) - \Delta d(P_0) &= 0. \end{aligned} \tag{16}$$

By replacing the terms in Equation (16) by the expression of their model in Equations (3-4), we obtain the additional constraints:

$$\begin{aligned}\theta^0_{12} &= -\theta^0_{14} \times \bar{d},\\ \theta^0_{11} &= -\theta^0_{13} \times \bar{d},\\ \theta^0_{17} &= -\theta^0_{19} \times \bar{d},\\ \theta^0_{16} &= -\theta^0_{18} \times \bar{d}.\end{aligned} \tag{17}$$

Te disparity and disparity variation models thus become:

$$d^t(P_0) = \theta^0_{13}(x - \bar{d}u) + \theta^0_{14}(y - \bar{d}v) + \theta^0_{15}, \tag{18}$$

$$\Delta d(P_0) = \theta^0_{18}(x - \bar{d}u) + \theta^0_{19}(y - \bar{d}v) + \theta^0_{20}. \tag{19}$$

This allows us to reduce again the number of parameters from 17 to 13. We denote $\boldsymbol{\theta} = \begin{pmatrix}\theta_1 & \dots & \theta_{13}\end{pmatrix}^\top$ the new parameters. The final scene flow model is the following:

$$\begin{aligned}\Delta x(P_0) &= \theta_1 u + \theta_2 x + \theta_3(y - \bar{d}v) + \theta_4, \qquad (20)\\ \Delta y(P_0) &= \theta_1 v + \theta_2 \bar{d}v + \theta_5(x - \bar{d}u) + \theta_6(y - \bar{d}v) + \theta_7,\\ d^t(P_0) &= \theta_8(x - \bar{d}u) + \theta_9(y - \bar{d}v) + \theta_{10},\\ \Delta d(P_0) &= \theta_{11}(x - \bar{d}u) + \theta_{12}(y - \bar{d}v) + \theta_{13}.\end{aligned}$$

IV. Estimating the model parameters

A. *Initializing the scene flow*

Most recent methods to estimate optical flows or disparity maps use deep neural networks. However, these methods require a huge amount of data to train the models. Because of the limited number of video light field datasets with corresponding ground truth scene flows, extending deep scene flow estimation methods to light fields is very challenging.

Instead, in this paper, we propose to take advantage of 2D optical flow methods and then use our model to regularize the different flows in the 4D ray space in order to compute a scene flow that would be consistent across all views. For each view of the light field, we estimate an optical flow independently. We also use the same optical flow method to estimate disparity maps at t and $t{+}1$. For the experiments, we consider a state-of-the-art technique based on a deep learning architecture called PWC-Net [20]. The initial disparity variation is estimated in regions where there is no temporal or angular occlusion by computing the difference of disparity along the optical flow, using the initial optical flow and disparity maps at t and $t+1$.

This approach requires to compute an occlusion mask in order to know where we can estimate reliable disparity variation. For that purpose, similarly to [5], we compute an energy value for every point $P(u,v,x,y)$ of LF^t, as

$$E = E_c + \lambda_1 E_{\nabla c} + \lambda_2 E_f + \lambda_3 E_{\nabla f}. \tag{21}$$

The terms E_c and $E_{\nabla c}$ are respectively color and color gradient consistency terms computed between the view (u,v) and the same projected view from $t+1$ to t, and are defined as

$$E_c(P) = \left\| L^{t+1}_{uv}(P + F_{\text{init}}(P)) - L^t_{uv}(P) \right\|_2, \tag{22}$$

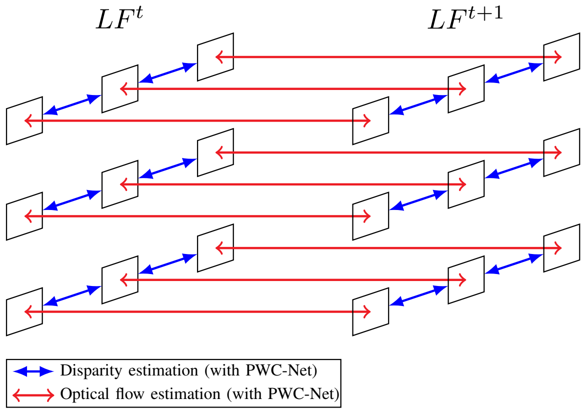
LF^t
LF^{t+1}
Disparity estimation (with PWC-Net)
Optical flow estimation (with PWC-Net)

Fig. 3. Initialization of the scene flow using a deep optical flow method (PWC-Net [20] in our case).

$$\begin{aligned}E_{\nabla c}(P) = &\left\| \nabla_x L^{t+1}_{uv}(P + F_{\text{init}}(P)) - \nabla_x L^t_{uv}(P) \right\|_2\\ &+ \left\| \nabla_y L^{t+1}_{uv}(P + F_{\text{init}}(P)) - \nabla_y L^t_{uv}(P) \right\|_2.\end{aligned} \tag{23}$$

The energy terms E_f and $E_{\nabla f}$ measure the consistency of the forward optical flow F_{init} and the backward optical flow $F^{\text{b}}_{\text{init}}$, and are defined as

$$E_f(P) = \left\| F_{\text{init}}(P) + F^{\text{b}}_{\text{init}}(P + F_{\text{init}}(P)) \right\|_2, \tag{24}$$

$$\begin{aligned}E_{\nabla f}(P) = &\left\| \nabla_x F_{\text{init}}(P) + \nabla_x F^{\text{b}}_{\text{init}}(P + F_{\text{init}}(P)) \right\|_2\\ &+ \left\| \nabla_y F_{\text{init}}(P) + \nabla_y F^{\text{b}}_{\text{init}}(P + F_{\text{init}}(P)) \right\|_2\end{aligned} \tag{25}$$

From this energy value, we can compute a confidence measure C as

$$C(P) = \exp\left(-\frac{E(P)}{2\sigma_c^2}\right) \tag{26}$$

where σ_c controls the "width" of the distribution. Finally, in order to generate a binary mask B, we threshold the confidence map as

$$B(P) = \begin{cases} 1 & \text{if } C(P) > 0.5\\ 0 & \text{otherwise}\end{cases} \tag{27}$$

For the experiments, we have chosen $\lambda_1 = 2$, $\lambda_2 = 10$, $\lambda_3 = 20$ and $\sigma_c = 0.5$.

Using this mask, we can compute an initial scene flow $F_{\text{init}}, d^t_{\text{init}}, \Delta d_{\text{init}}$ where the optical flow and the disparity map at t are completely dense and where the disparity variation is only available on non-occluded regions. We can then use our model to regularize the optical flow and disparity and to interpolate the disparity variation in occluded regions (while also regularizing it in non-occluded regions).

B. *Clustering the light field*

The model previously described works under one assumption: our model is affine, so the scene flow should not have discontinuities. As a consequence, we can partition our light field into clusters that respect the assumption and fit one model for each cluster. If the clusters correspond to the same object in the scene, the assumption will be valid. We therefore group pixels of similar color across the views and corresponding to the same scene area in 4D clusters, using the method proposed in [41]. The method is inspired by the SLIC algorithm [42]. Centroids are first initialised on a reference view. Their disparity is then estimated and used to project the centroids to all the views. A k-means clustering is then simultaneously performed on all views. Using the centroid disparity, all the rays assigned to a cluster are projected back to the reference view to update the centroids colors and spatial positions. The approach is fast, free of any strong scene geometry prior and does not require a dense depth map estimation. For these reasons, we chose this method for our model.

To estimate the model parameters for each cluster, the model is fitted to the initial optical flow and disparity estimates available in each cluster. The number of estimates may however not be sufficient in some clusters. For this reason, we propose to build a graph connecting the different clusters. This graph enables us to look for the N nearest clusters of a given cluster, adding more estimates in the computation of the scene flow model of a given cluster.

C. Connecting the clusters with a weighted graph

In order to connect the different clusters, we build an undirected weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, w\}$. $\mathcal{V}$ is the set of the K clusters computed in the clustering step (see Fig.1). A vertex i is connected to another one j if their corresponding clusters are adjacent to one another in at least one view of the light field or if they are in the same range of disparity, that is if

$$|d^t(C_i) - d^t(C_j)| < \beta \left(\max_{k \in \mathcal{V}} d^t(C_k) - \min_{k \in \mathcal{V}} d^t(C_k) \right), \quad (28)$$

when C_i and C_j denote the clusters centroids and $\beta \in [0, 1]$ a threshold coefficient. In the experiments, we fix $\beta = 0.1$.

The weight between two connected nodes i and j is defined as

$$w(i, j) = \min_{(u,v) \in \Omega_{ij}} \exp\left[-\alpha D\left(\mathcal{P}_{uv}(C_i), \mathcal{P}_{uv}(C_j)\right)\right], \quad (29)$$

where Ω_{ij} is the set of views where i and j are adjacent, $\mathcal{P}_{uv}(C)$ the projection of the centroid C on the view (u, v) and α a parameter that we empirically fix to 0.2. The distance D is based on spatial and color proximity and defined as in [41]:

$$D = \sqrt{{d_c}^2 + \frac{m^2}{S^2}{d_s}^2} \quad (30)$$

where $S = \sqrt{H \times W / K}$ with W and H the width and height of a view. The parameter m has the same value as for the clustering step, it is used in the clustering step to control the compactness of the clusters. d_c and d_s are the color and spatial distances respectively defined as euclidean distances in the CIELAB colorspace and the $[xy]$ space:

$$d_c(C_i, C_j) = \sqrt{(L_i - L_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \quad (31)$$

$$d_s(C_i, C_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (32)$$

Once the graph is computed, we can look for the N nearest neighbors of a given node i using Dijkstra's algorithm [43]. In the search, we discard every vertex whose corresponding cluster contains no scene flow estimate (which can happen when the initial scene flow is sparse). The set of N neighbors of i (including itself) is denoted $\mathcal{N}_i$. It is used to have more scene flow estimates than those inside the cluster and in particular when the cluster i has no estimate inside itself.

D. Fitting a model with RANSAC

The approach we use to fit the model described in Section III to the scene flow estimates that we have is inspired from the RANSAC method [44]. The general idea is to choose m scene flow estimates, to compute the parameters of the model and then to evaluate the cost of the model.

Let SF_i be the set of initial scene flow estimates contained in the cluster i, we have

$$\mathrm{SF}_i = \begin{pmatrix} u_1 & v_1 & x_1 & y_1 & \Delta x_1 & \Delta y_1 & d_1^t & \Delta d_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{n_i} & v_{n_i} & x_{n_i} & y_{n_i} & \Delta x_{n_i} & \Delta y_{n_i} & d_{n_i}^t & \Delta d_{n_i} \end{pmatrix} \quad (33)$$

The equation (20) is linear in $\boldsymbol{\theta}$ so we build a block matrix A_i and vector b_i such that $\left\| A_i \hat{\boldsymbol{\theta}} - b_i \right\|_2$ represents the fidelity of a model $\hat{\boldsymbol{\theta}}$ to the initial scene estimates SF_i.

$$A_i = \begin{pmatrix} A_i^x & 0 & 0 \\ A_i^y & 0 & 0 \\ 0 & A_i^d & 0 \\ 0 & 0 & A_i^\Delta \end{pmatrix} \quad (34)$$

where the sub-matrices A_i^x, A_i^y, A_i^d and A_i^Δ are defined as

$$A_i^x = \begin{pmatrix} u_1 & x_1 & y_1 - \overline{d}_i v_1 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{n_i} & x_{n_i} & y_{n_i} - \overline{d}_i v_{n_i} & 1 & 0 & 0 & 0 \end{pmatrix} \quad (35)$$

$$A_i^y = \begin{pmatrix} v_1 & \overline{d}_i v_1 & 0 & 0 & x_1 - \overline{d}_i u_1 & y_1 - \overline{d}_i v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{n_i} & \overline{d}_i v_{n_i} & 0 & 0 & x_{n_i} - \overline{d}_i u_{n_i} & y_{n_i} - \overline{d}_i v_{n_i} & 1 \end{pmatrix} \quad (36)$$

$$A_i^d = A_i^\Delta = \begin{pmatrix} x_1 - \overline{d}_i u_1 & y_1 - \overline{d}_i v_1 & 1 \\ \vdots & \vdots & \vdots \\ x_{n_i} - \overline{d}_i u_{n_i} & y_{n_i} - \overline{d}_i v_{n_i} & 1 \end{pmatrix} \quad (37)$$

Let b_i be the corresponding vector to A_i:

$$b_i = \begin{pmatrix} b_i^x & b_i^y & b_i^d & b_i^\Delta \end{pmatrix}^\top \quad (38)$$

with:

$$b_i^x = \begin{pmatrix} \Delta x_1 & \cdots \Delta x_{n_i} \end{pmatrix} \quad (39)$$

$$b_i^y = \begin{pmatrix} \Delta y_1 & \cdots \Delta y_{n_i} \end{pmatrix} \quad (40)$$

$$b_i^d = \begin{pmatrix} d_1^t & \cdots d_{n_i}^t \end{pmatrix} \quad (41)$$

$$b_i^\Delta = \begin{pmatrix} \Delta d_1 & \cdots & \Delta d_{n_i} \end{pmatrix} \quad (42)$$

In order to make our model linear, we approximate the disparity in Equations (10), (11), (12) and (13) by a pre-estimated disparity value. We compute one disparity estimate per cluster $\overline{d}_i$ by averaging the disparity estimates contained in the cluster i and in its neighbors as

$$\overline{d}_i = \frac{\sum\limits_{j\in\mathcal{N}_i} e^{-\lambda w(i,j)} \sum\limits_{k=1}^{n_j} b^d_{j,k}}{\sum\limits_{j\in\mathcal{N}_i} e^{-\lambda w(i,j)} n_j} \tag{43}$$

The parameter λ controls the weight of the neighboring clusters in the average computation and $b^d_{j,k}$ denotes the kth element of vector b^d_j.

The more constant the disparity is in a cluster, the more correct the approximation is. For each cluster i, we search for the parameters $\boldsymbol{\theta}$ of the model (20) that minimize the following cost function:

$$\mathcal{L}_i(\boldsymbol{\theta}) = \sum_{j\in\mathcal{N}_i} e^{-\lambda w(i,j)} \cdot f_j(\boldsymbol{\theta}) \tag{44}$$

where $f_j(\boldsymbol{\theta})$ is the number of outliers produced by the model $\boldsymbol{\theta}$ among the estimates which are inside the cluster j, that can be formally defined as:

$$f_j(\boldsymbol{\theta}) = \sum_{k=0}^{4n_j} [\![|A_{j,k}\boldsymbol{\theta} - b_{j,k}| > \tau]\!] \tag{45}$$

The symbols $[\![\cdot]\!]$ denote the Iverson brackets, which return 1 if the proposition inside the brackets is true and 0 otherwise. $A_{j,k}$ denotes the kth row of A_j. The hyperparameter τ is analogous to the threshold defined in the classical RANSAC algorithm. It is fixed to 5 in our experiments. As with RANSAC algorithm, we generate an hypothesis $\hat{\boldsymbol{\theta}}$ for our model, we compute its cost function $\mathcal{L}_i(\hat{\boldsymbol{\theta}})$ and compare it with the best candidate $\hat{\boldsymbol{\theta}}_{\text{min}}$ that we found so far (the one with the lowest cost function). We iterate N_{iter} times.

Before iterating, we initialize our model to a constant model: we set every coordinate of $\hat{\boldsymbol{\theta}}$ to 0 except for $\hat{\theta}_4$, $\hat{\theta}_7$, $\hat{\theta}_{10}$ and $\hat{\theta}_{13}$. This way, the scene flow inside a cluster is constant and equal to the weighted average scene flow estimate.

$$\begin{aligned}
\hat{\theta}_4 &= \overline{\Delta x}_i = \frac{\sum\limits_{j\in\mathcal{N}_i} e^{-\lambda w(i,j)} \sum\limits_{k=1}^{n_j} b^x_{j,k}}{\sum\limits_{j\in\mathcal{N}_i} e^{-\lambda w(i,j)} n_j} \\
\hat{\theta}_7 &= \overline{\Delta y}_i = \frac{\sum\limits_{j\in\mathcal{N}_i} e^{-\lambda w(i,j)} \sum\limits_{k=1}^{n_j} b^y_{j,k}}{\sum\limits_{j\in\mathcal{N}_i} e^{-\lambda w(i,j)} n_j} \\
\hat{\theta}_{10} &= \overline{d}_i \\
\hat{\theta}_{13} &= \overline{\Delta d}_i = \frac{\sum\limits_{j\in\mathcal{N}_i} e^{-\lambda w(i,j)} \sum\limits_{k=1}^{n_j} b^\Delta_{j,k}}{\sum\limits_{j\in\mathcal{N}_i} e^{-\lambda w(i,j)} n_j}
\end{aligned} \tag{46}$$

What differs from classical RANSAC is the hypothesis generation. Classically, we would randomly choose 13 rows from $\{A_j \mid j \in \mathcal{N}_i\}$ to form a matrix A_s and the corresponding vector b_s, and we would compute - if possible - $\hat{\boldsymbol{\theta}} = A_s^{-1} b_s$.

In our case, the process of selection of estimates is not totally random. At every iteration, we want to generate stable parameters $\hat{\boldsymbol{\theta}}$. So, we need to form a matrix A_s with a low condition number, which means with rows that are the most linearly independent from one another.

Inspired by the work in [45], we propose to choose the samples in a careful way in order to perform our hypothesis generation step. More precisely, given a cluster i, we first build the matrix and vector U_i and v_i such that

$$U_i = \bigoplus_{j\in\mathcal{N}_i} A_j \qquad \text{and} \qquad v_i = \bigoplus_{j\in\mathcal{N}_i} b_j, \tag{47}$$

where $\bigoplus_k X_k$ denotes the vertical concatenation of the matrices X_k.

Let M be the number of rows of U_i and v_i. Our goal is to find a set $\mathcal{S}$ of 13 linearly independent rows among the M rows of U_i. The general idea is to iteratively add samples to $\mathcal{S}$ taking into account the previous added samples.

We start with an empty set $\mathcal{S}$. The first sample which is added is randomly chosen. Then, for every iteration n from 2 to 13, we add a nth sample to the set $\mathcal{S}$. To do so, we build the matrix $U_i(\mathcal{S}, \mathcal{T})$ with $\mathcal{T}$ being the range $[1, n]$. The resulting matrix of size $(n, n-1)$ is of rank $n-1$ because every row is independent from one another. The nullspace of such matrix gives us the unique vector z that is orthogonal to all rows of this matrix. Then, in the normalized version of $U_i(\mathcal{R}, \mathcal{T})$ (with $\mathcal{R} = [1, M]$), we search for the row that is the most linearly dependent on the null vector, i.e. the most linearly independent of the rows of $U_i(\mathcal{S}, \mathcal{T})$. This constitutes the new sample added to our set. We continue until we can reach our 13 samples. Once the set $\mathcal{S}$ is complete, we can build a matrix and a vector $A_s = U_i(\mathcal{S}, \mathcal{T})$ and $b_s = v_i(\mathcal{S})$ with $\mathcal{T} = [1, 13]$. We finally generate an hypothesis $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|A_s\boldsymbol{\theta} - b_s\|_2$. The hypothesis generation is fully detailed in Algorithm 1.

Another change to the classic RANSAC algorithm is that for each iteration and for each cluster, we also evaluate the models given by the neighboring clusters. This allows us to propagate correct models among the clusters and to make the algorithm converge faster.

```
Input: Matrix $U_i$ and its corresponding vector $v_i$
Output: An hypothesis $\hat{\boldsymbol{\theta}}$ for the cluster $i$
$r_0 \leftarrow$ selectRandomRow($U_i$);
$\mathcal{S} \leftarrow \{r_0\}$;
$\mathcal{R} \leftarrow [1, M]$;
for $n = 2 \rightarrow 13$ do
    $\mathcal{T} \leftarrow [1, n]$;
    $z \leftarrow$ nullspace($U_i(\mathcal{S}, \mathcal{T})$);
    normalizeRows($U_i(\mathcal{R}, \mathcal{T})$);
    $b \leftarrow U_i(\mathcal{R}, \mathcal{T})z$;
    $r \leftarrow \arg\max_{k\in\mathcal{R}} |b_k|$;
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{r\}$
end
$A_s \leftarrow U_i(\mathcal{S}, \mathcal{T})$;
$b_s \leftarrow v_i(\mathcal{S})$;
$\hat{\boldsymbol{\theta}} \leftarrow \arg\min_{\boldsymbol{\theta}} \|A_s\boldsymbol{\theta} - b_s\|_2$;
```

Algorithm 1: Hypothesis generation for cluster i

V. Evaluation

A. Scene Flow Datasets

In order to be able to compute objective performance measures, we have generated a synthetic video Light Field dataset with the corresponding ground truth scene flow[1]. For that purpose, we have used the production files of the open source movie Sintel [46] and have modified them in the Blender 3D software [47] in order to render an array of 3x3 views. Similarly to the MPI Sintel flow dataset [17], [18], we have modified the scenes to generate not only the 'final' render, but also a 'clean' render without lighting effects, motion blur, or semi-transparent objects. Ground truth optical flow and disparity maps were also generated for each view. Since disparity variation maps could not be rendered within Blender, we have computed them using the disparity map and the optical flow. However, this process requires projecting the disparity map of a frame to the next frame using the optical flow, which results in unavailable disparity variation information in areas of temporal occlusion. We have processed two scenes of 3×3 views of 1024×436 pixels and 50 frames corresponding to the scenes 'Bamboo2' and 'Temple1' in [17]. The disparities (in pixels) between neighboring views are in the range $[-8, +52]$ for 'Bamboo2' and $[-22, +9]$ for 'Temple1'. We chose an angular configuration that is similar to the one of real light fields captured by rigs of cameras, such as e.g. in [48] and [49], which respectively provide 5×3 and 4×4 views. We also use the dataset of [48] to test our method on a real light field sequence: 'Bar'. Each frame is a 5×3 light field, in which each view has a spatial resolution of 1920×1080 pixels. The horizontal and vertical baselines of the camera setup are different, the ratio between the two is 0.625 and the horizontal disparity ranges from 22 to 75 pixels. Finally, we assess our method on a dense synthetic light field dataset provided by [19]. The light fields have an angular resolution of 9×9 views and their spatial resolution is either 1024×720 or 412×290, respectively referred to as 'Big' and 'Small' in the rest of the article. This configuration simulates light fields captured with plenoptic cameras as in [50] or captured with very dense camera arrays as in [51].

B. Influence of hyperparameters

We have various hyperparameters in our method: the most critical ones are the number of clusters K, the number of nearest neighbors N we select and the number of iterations to compute an affine model N_{iter}. For the experiments, we used three metrics to assess the scene flow estimations: the End-Point Error (EPE) for the optical flow, the Mean Absolute Error (MAE) for the disparity map d^t and the MAE for the disparity variation Δd. The latter is only computed for dis-occluded pixels because there is no ground truth on occluded pixels.

In order to search for the best combination of (N, K) hyperparameters for the scene flow estimation, we perform a grid search, using the aforementioned metrics for the whole Sintel dataset. We have tested 4 different values of $N = \{1, 2, 5, 10\}$ for 5 different values of $K = \{625, 1250, 2500, 5000, 10000\}$. The results of the grid search are shown in Figure 4.

From Figure 4, we notice that the optical flow and the disparity grid search have approximately the same profile: the errors decrease when the number of neighbors N and the number of clusters K increase. We also notice that the optical flow and disparity errors increase drastically when the number of clusters is small. This is due to some underfitting of the model: there are too many estimates and our affine model is not complex enough to fit the data.

On the other hand, we see that the disparity variation profile of the grid search behaves differently: the lowest error is obtained when the number of neighbors N is high and the number of clusters K is low. The difference with the optical flow and disparity behaviour is caused by the way we compute the initial disparity variation estimates: we compute an occlusion mask to remove outliers. If there are inaccuracies in the occlusion mask, outliers will be taken into account for the model fitting. As a consequence, taking bigger clusters and more neighbors helps reducing the errors by decreasing the weight of an outlier among the estimates.

In our experiments, we choose $K = 10000$ and $N = 10$ as it is the best combination for both optical flow and disparity estimations.

After selecting the optimal combination of N and K, we need to determine the number of iterations N_{iter} that the model needs to converge towards a stable solution. The evolution of the cost function (written in Eq.(44)) is shown in Figure 5. In the figure, we normalized with the initial cost, that is the cost of the initial model as given in Eq.(46).

The convergence rate is high and from the third iteration, the cost function is quasi-constant. So, we have taken $N_{\text{iter}} = 3$ for the following evaluations.

C. Comparison with state-of-the-art methods

TABLE I
EPE of estimated optical flow for all pixels

		Bamboo2		Temple1	
		clean	final	clean	final
Central View	OSF [7]	1.943	1.901	6.400	4.797
	PRSM [8]	1.203	1.287	1.285	1.671
	OLFW [15]	1.421	1.462	2.061	2.374
	PWC-Net [20]	0.946	1.018	1.032	1.284
	SDSF [16]	1.007	1.102	1.042	1.383
	Ours	**0.883**	**0.946**	**0.959**	**1.242**
All Views	PWC-Net [20]	0.947	1.019	1.029	1.290
	SDSF [16]	1.090	1.169	1.109	1.453
	Ours	**0.889**	**0.952**	**0.968**	**1.253**

The proposed method is first assessed using our sparse dataset. It is compared to the methods in [15], [16], respectively referred to here as OLFW (Oriented Light Field Window) and SDSF (Sparse-to-Dense Scene Flow). The OLFW method was designed for dense light fields captured with plenoptic cameras and is hardly applicable when the baseline is large. However, the optical flow searched via the oriented

[1] http://clim.inria.fr/Datasets/SyntheticVideoLF/index.html

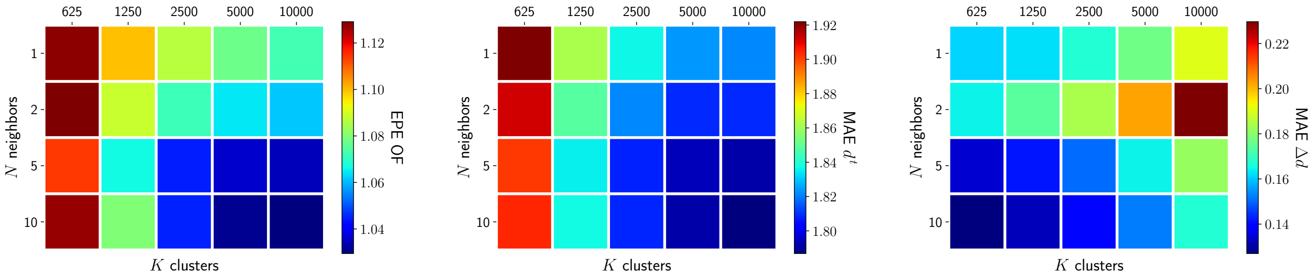

Fig. 4. Grid search to find the optimal parameters N and K for the scene flow regularization. Each image is an average error map for a set of (K, N) computed with the whole Sintel dataset. The combination of hyperparameters that gives the lowest errors is $K = 10000$ and $N = 10$.

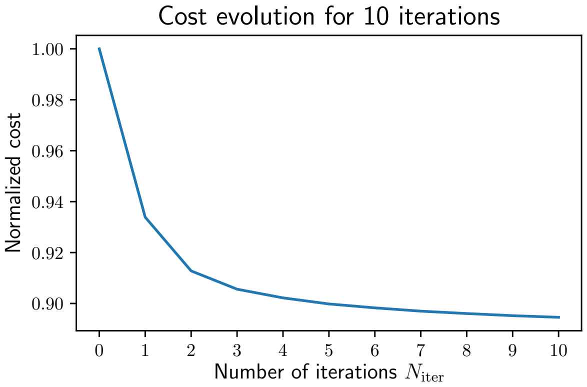

Fig. 5. Evolution of the cost in the RANSAC model fitting. The cost is normalized by the cost of the initial model computed with Eq. (46). The cost becomes almost constant from the third iteration.

TABLE II
MAE OF ESTIMATED DISPARITY FOR ALL PIXELS

		Bamboo2		Temple1	
		clean	final	clean	final
Central View	OSF [7]	2.578	2.611	19.307	16.990
	PRSM [8]	2.619	2.665	16.414	14.639
	FDE [28]	**1.598**	**1.663**	**0.250**	1.090
	PWC-Net [20]	1.888	1.985	0.384	0.689
	Ours	1.738	1.819	0.332	**0.674**
All Views	FDE [28]	2.067	2.137	0.419	1.391
	PWC-Net [20]	1.972	2.055	0.378	0.710
	Ours	**1.868**	**1.932**	**0.338**	**0.682**

window can be combined with disparity maps estimated with methods suitable for sparse light fields. In the test reported here, we have used ground truth disparity maps for this method, thus showing the best results it can give for the estimated scene flow. We also compare the disparity maps that we estimate, as part of our scene flow model, with the ones obtained with the deep learning based disparity estimation method in [28], referred to here as FDE (Flexible Depth Estimation).

Besides, our method is compared with the initial scene flow estimated as in IV-A, using PWC-Net [20]. The optical flow estimation technique [20] is used for separately estimating the optical flow in each view as well as the disparity between views. In order to have a dense disparity variation for this naive approach, we do not compute the occlusion mask. So, the disparity variation in occluded or disoccluded areas will never be consistent. Finally, we tested two stereo scene flow methods: [7], [8], denoted as OSF (Object Scene Flow) and PRSM (Piecewise Rigid Scene Model), using the central view and its right neighbour as stereo pair.

TABLE III
MAE OF ESTIMATED DISPARITY VARIATION FOR ALL UNOCCLUDED PIXELS

		Bamboo2		Temple1	
		clean	final	clean	final
Central View	OSF [7]	0.539	0.518	1.491	3.159
	PRSM [8]	0.173	0.171	0.165	0.175
	OLFW [15]	0.356	0.345	0.152	0.162
	PWC-Net [20]	0.820	0.878	0.299	0.416
	SDSF [16]	**0.136**	**0.140**	0.109	0.128
	Ours	0.146	0.153	**0.098**	**0.116**
All Views	PWC-Net [20]	0.869	0.938	0.295	0.418
	SDSF [16]	**0.140**	**0.142**	0.111	0.131
	Ours	0.150	0.157	**0.105**	**0.127**

The results are summarized in Tables I, II and III. For each successive light field frame of the four sequences (*Bamboo2* and *Temple1*, both rendered as *clean* and *final*), we compute the optical flow EPE, the disparity and disparity variation MAE on every ray of the light field (denoted 4D) and also on the central view only (denoted 2D).

We can observe that our method always yields the most accurate optical flows (see Table I), and disparity maps (see Table II). Our method is only outperformed by the FDE method [28] for the disparity of the central view on *Bamboo2 clean & final* and *Temple1 clean*. However, even for these light fields, our method provides better average results than FDE when considering the disparity maps of all the views, which indicates a better consistency between views. Furthermore, the FDE method only estimates disparity (and not optical flow or disparity variation), while our approach computes the full scene flow. Even though, we did not choose the best combination of K and N parameters for the disparity variation, the mean absolute error is the lowest for the *Temple1* sequence and the second lowest among every tested method for the *Bamboo2* sequence (see Table III). It is only outperformed by a small margin by our prior sparse-to-dense interpolation

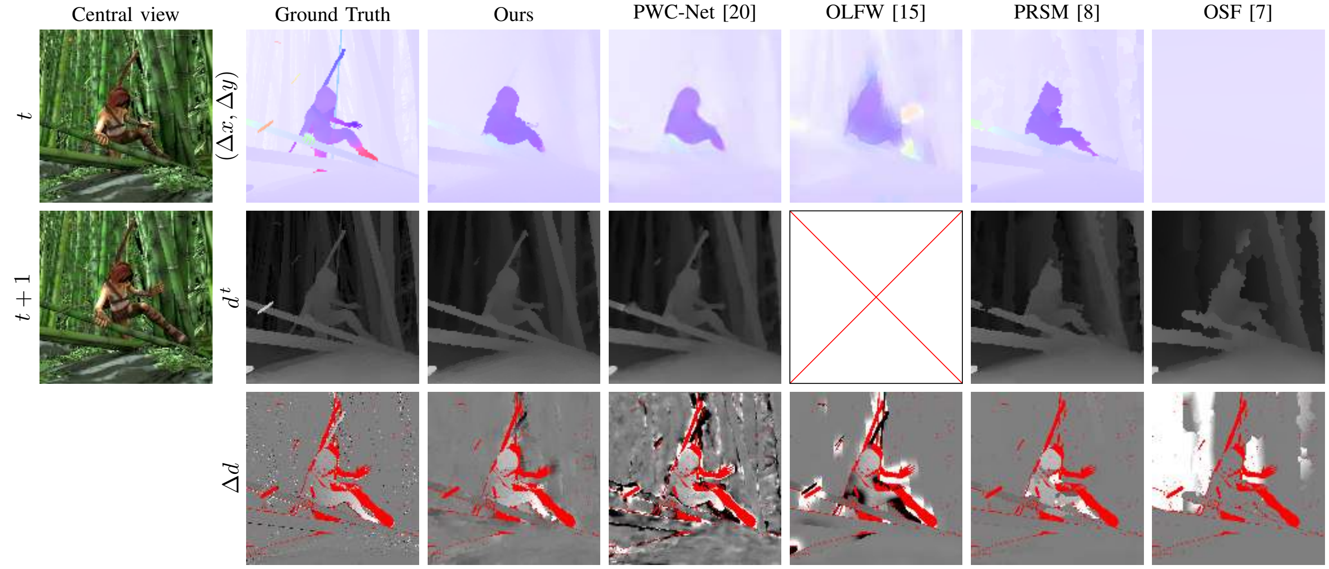

Fig. 6. Visual comparison of our method with [7], [8], [15], [20] on a frame of *Bamboo2 clean*. The optical flows are visualized with the Middlebury color code, and the disparity maps and disparity variations are visualized using a gray-scale representation. The red pixels are the occlusion mask where there is no ground truth disparity variation available.

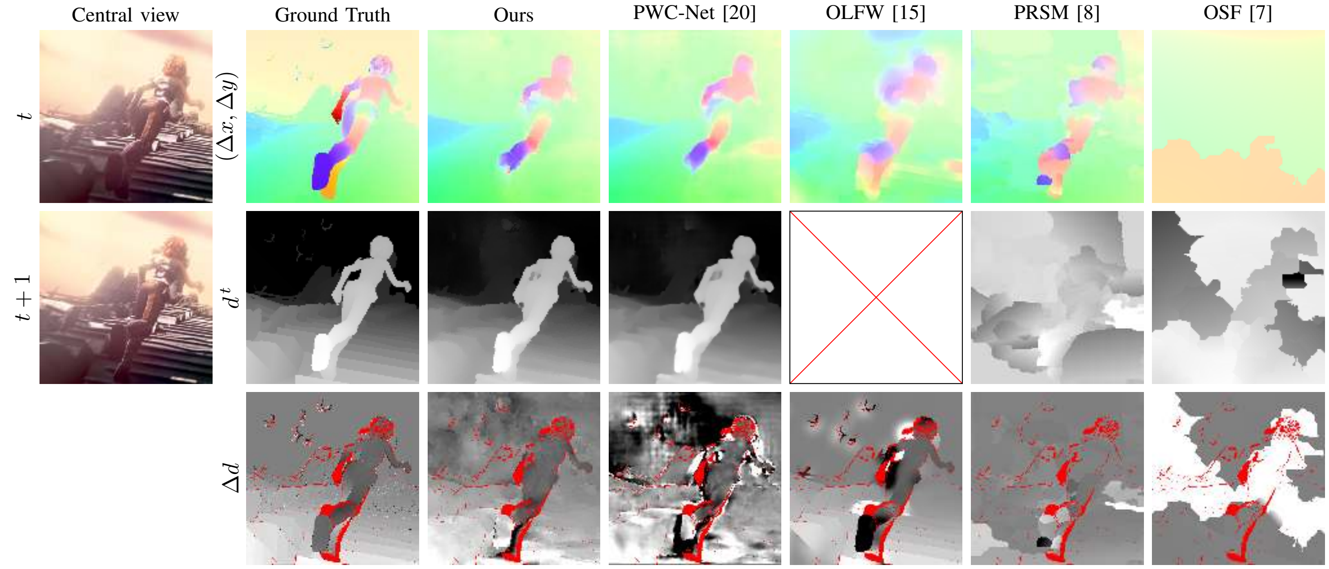

Fig. 7. Visual comparison of our method with [7], [8], [15], [20] on a frame of *Temple1 final*. The optical flows are visualized with the Middlebury color code, and the disparity maps and disparity variations are visualized using a gray-scale representation. The red pixels are the occlusion mask where there is no ground truth disparity variation available.

method (SDSF [16]). Note that the two stereo methods failed to estimate an accurate disparity in the *Temple1* sequence. These methods were mainly developed in the context of autonomous driving and their default parameters were fine-tuned for urban scenes.

We also performed some qualitative assessment of the methods on frames of *Bamboo2 clean* (Figure 6), *Temple1 final* (Figure 7) and of *Bar* (Figure 8). We can notice that our method gives sharper optical flow and disparity maps than the initial scene flow computed by [20], while correcting occlusion errors.

Finally, we tested our method on a dense dataset provided by [19]. In order to keep the complexity low, we estimated an initial scene flow on a set of nine views (central view, corner views and top, bottom, left and right views). Then, clustering every views, we were able to fit a scene flow model, interpolate and regularize the scene flow on every view of the light field.

For the disparity estimation, some light field depth estimation methods were added to compare: globally consistent depth labeling (GCDL) [21], phase-shift based depth estimation (PSDE) [3] and occlusion-aware depth estimation (OADE) [22]. Note that the metric used to compute the disparity estimation is Root Mean Square Error (RMSE) as it was in the original paper [19]. The results are given in Tables IV and V. We see that, in terms of optical flow and disparity, our method yields similar results to [19] for the central view, and that, in

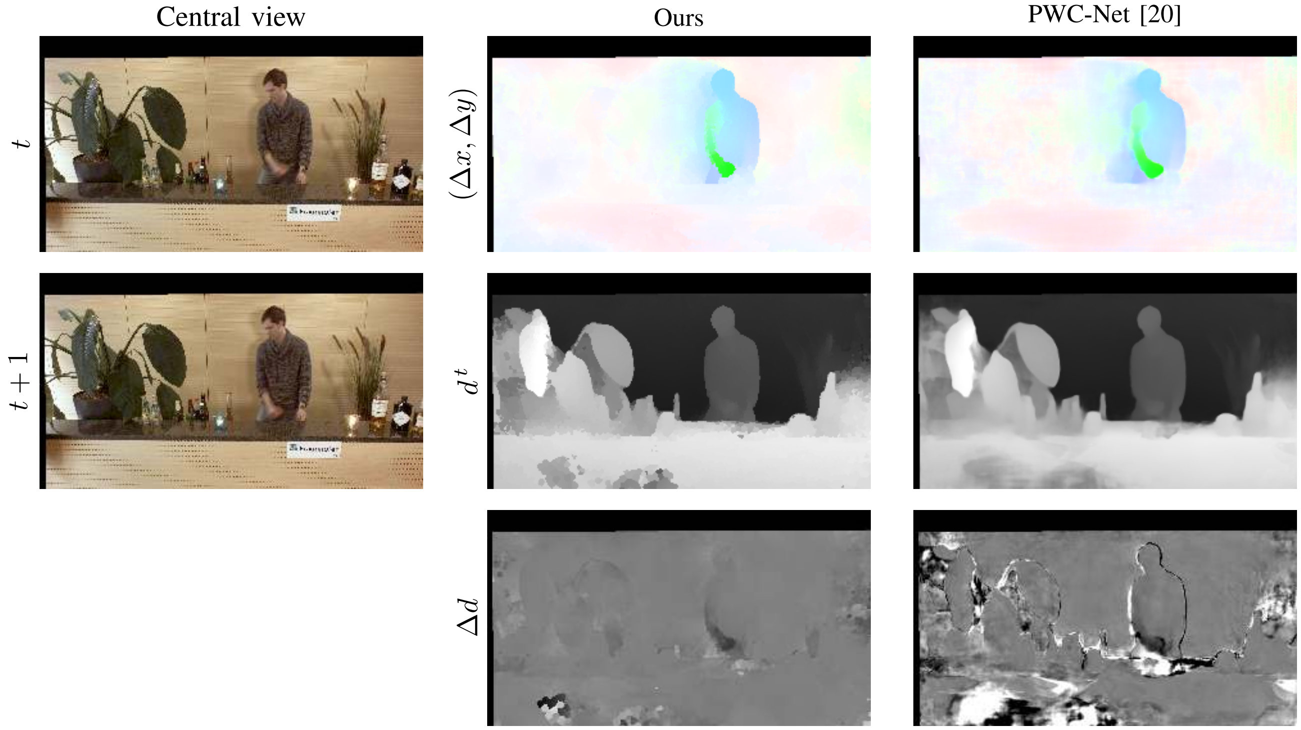

Fig. 8. Visual comparison of our methods with [20]. The optical flows are visualized with the Middlebury color code, and the disparity maps and disparity variations are visualized using a gray-scale representation. The light field frames are taken from [48].

TABLE IV
EPE OF ESTIMATED OPTICAL FLOW FOR ALL PIXELS

		NewSecretaire		Mario		Drawing		Balls		NewBalls	
		Small	Big	Small	Big	Small	Big	Small	Big	Small	Big
Central View	PRSM [8]	1.261	1.809	1.120	1.395	1.257	3.093	0.289	0.495	0.670	0.892
	OSF [7]	0.877	1.513	2.749	6.239	-	4.355	0.744	1.613	0.547	0.794
	LDOF [52]	3.780	3.174	2.136	4.524	1.129	1.766	0.440	0.587	1.259	1.883
	OLFW [15]	2.265	4.441	4.893	7.166	2.495	5.005	1.167	6.073	1.206	13.713
	FVOF [19]	0.781	1.393	**0.826**	1.012	**0.928**	**1.324**	**0.284**	0.481	**0.496**	**0.693**
	Ours	**0.771**	**0.851**	1.065	**0.865**	1.146	1.661	0.390	**0.430**	0.602	0.708
All Views	FVOF (4D) [19]	1.337	1.853	**1.174**	1.299	**1.019**	**1.427**	0.397	0.555	**0.588**	0.807
	Ours (4D)	**1.247**	**1.333**	1.381	**1.201**	1.158	1.677	**0.395**	**0.435**	0.608	**0.719**

most scenes, it gives more accurate estimation when taking every view into account. Both methods outperform every other tested method based on light fields or stereo images.

D. Model validation

In order to validate the affine model, we used the ground truth scene flow as initial estimation and then we performed the clustering step as well as the fitting of the model with the same hyperparameters as in Section V-B. This gives us the minimum errors that can be obtained with our method, due to the model approximation. The results of this experiment for every scene and rendering are summarized in Table VI. In comparison with state-of-the-art methods, the endpoint errors for the optical flow and the mean absolute errors for the disparity that we obtain are substantially lower by a factor of 6. The mean absolute errors for the disparity variation also decrease but less significantly, because they are already very low.

Visual comparisons between the estimated scene flow and the corresponding ground truth are also presented in Figures 9 and 10 for the light field frames that have the highest endpoint errors in each scene rendered in *final* mode. We observe that, although these frames are the worst frames in their respective sequences, thin structures are well reconstructed. However, our estimated optical flow is inaccurate for objects whose disparity is so high that it disappears in other views of the light field, making it very difficult to accurately cluster the object and fit an affine model. This is why the optical flow of the butterfly on the *bamboo* frame in Figure 9 is visually different from the ground truth. Inaccuracies are also observed when small objects have colors that are very close to the background: this leads to a weighted graph with strong edges between the aforementioned object and the background clusters. This is the

TABLE V
RMSE OF ESTIMATED DISPARITIES FOR ALL PIXELS

		NewSecretaire		Mario		Drawing		Balls		NewBalls	
		Small	Big	Small	Big	Small	Big	Small	Big	Small	Big
Central View	GCDL [21]	0.134	0.123	0.176	0.273	0.084	0.067	0.277	0.211	0.092	0.069
	PSDE [3]	0.350	0.136	0.543	0.092	0.115	0.119	0.595	0.111	0.148	0.077
	OADE [22]	0.193	0.138	0.196	0.165	0.074	0.068	0.245	0.111	0.172	0.079
	PRSM [8]	0.136	0.125	0.139	0.102	0.079	0.061	0.051	0.036	0.059	0.048
	OSF [7]	0.131	0.120	0.216	0.103	-	0.141	0.062	0.053	0.068	0.061
	OLFW [15]	0.147	0.126	0.188	0.123	0.097	0.067	0.094	0.064	0.077	0.057
	FVOF [19]	**0.110**	**0.080**	0.136	0.073	**0.058**	**0.039**	0.068	0.036	**0.051**	0.041
	Ours (2D)	0.113	0.084	**0.084**	**0.058**	0.061	0.053	**0.049**	**0.033**	0.053	**0.039**
All Views	FVOF [19]	0.123	0.088	0.145	0.082	**0.062**	**0.045**	0.090	0.038	0.059	0.044
	Ours	**0.114**	**0.086**	**0.090**	**0.060**	0.064	0.054	**0.053**	**0.034**	**0.054**	**0.040**

case for the optical flow of the dragons on the *temple* frame in Figure 10.

TABLE VI
VALIDATION OF THE AFFINE MODEL ACCORDING TO THE SCENE

	Bamboo2		Temple1	
	clean	final	clean	final
EPE $(\Delta x, \Delta y)$	0.159	0.165	0.172	0.199
MAE d^t	0.347	0.309	0.062	0.061
MAE Δd	0.118	0.119	0.064	0.064

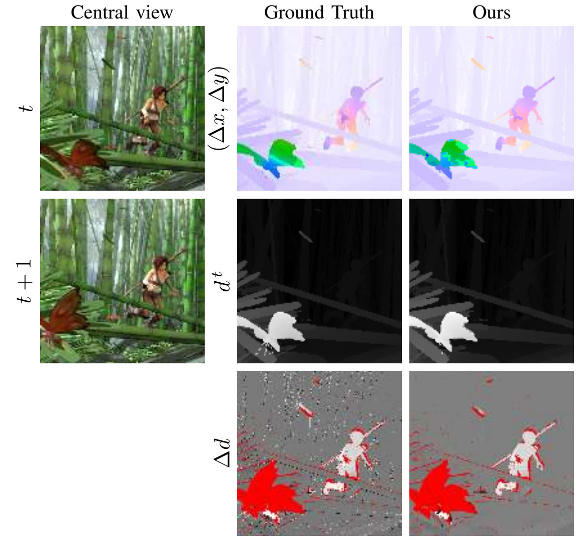

Fig. 9. Visual comparison of the ground truth scene flow and the one obtained with our method using the ground truth scene flow as initialization, with a *Bamboo2 final* frame.

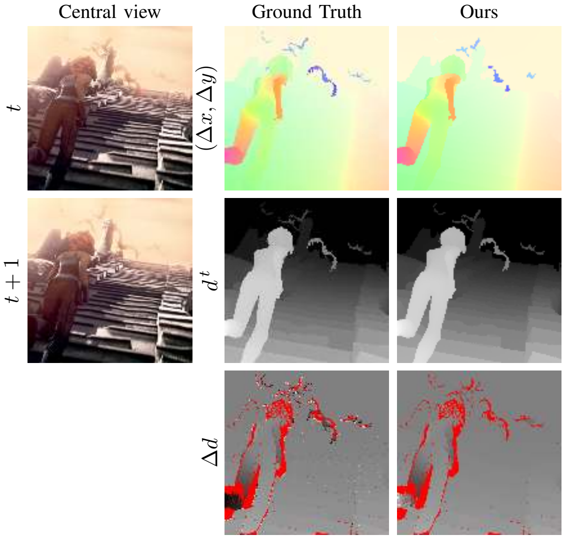

Fig. 10. Visual comparison of the ground truth scene flow and the one obtained with our method using the ground truth scene flow as initialization, with a *Temple1 final* frame.

In order to provide more insights on where our affine model fails to accurately represent the ground truth, we measured the influence of temporal occlusions, motion amplitude and disparity with the Sintel dataset. In Table VII, we computed errors on temporally non-occluded and occluded regions, respectively referred to as NOC and OCC. The last row is the ratio of each region for the whole dataset (e.g. there are 96% pixels that are not occluded in the Sintel dataset). We can notice that the errors are much higher in occluded areas. Since occlusions are typically located on objects boundaries, a bad clustering that groups pixels from different objects will cause errors during the model fitting step, thus giving a scene flow with more errors in the occluded areas.

In Table VIII, the impact of motion amplitude is measured. Let $s = \sqrt{\Delta x^2 + \Delta y^2}$ be the amplitude of motion of a pixel, s10, s10-40, s40 respectively represent the regions where $s < 10$, $s \in [10, 40]$ and $s > 40$. The results show that the error of the model is larger for objects with a very large motion $(s > 40)$.

Finally, we evaluate the influence of disparity on the errors in Table IX. Low disparity areas correspond to background objects, which tend to have lower motion amplitude than objects of the background whose disparities are higher. Therefore, high disparity and large motion are inherently related in the tested scenes, which explains why the areas with large disparity have higher errors.

TABLE VII
INFLUENCE OF OCCLUSIONS ON THE AFFINE MODEL

	NOC	OCC
EPE $(\Delta x, \Delta y)$	0.112	1.609
MAE d^t	0.174	0.671
MAE Δd	0.091	/
Ratio (%)	96	4

TABLE VIII
INFLUENCE OF THE MOTION AMPLITUDE ON THE AFFINE MODEL

	s10	s10-40	s40
EPE $(\Delta x, \Delta y)$	0.096	0.647	4.505
MAE d^t	0.202	0.084	0.475
MAE Δd	0.080	0.088	0.957
Ratio (%)	91	8	1

TABLE IX
INFLUENCE OF THE DISPARITY ON THE AFFINE MODEL

	$d^t < 10$	$d^t > 10$
EPE $(\Delta x, \Delta y)$	0.146	0.351
MAE d^t	0.172	0.333
MAE Δd	0.062	0.245
Ratio (%)	86	14

E. Complexity

Using the optimal hyperparameters, the computation takes one hour per light field frame on average for the sparse dataset, with our laptop equipped with an *Intel Core i7 - 6600U* CPU and 16 GB RAM. Note that the aforementioned duration is calculated with a non-optimal and fully sequential implementation. However, most of the steps (i.e. scene flow initialization, clustering, nearest neighbor search, model fitting) could benefit from a parallel implementation. The authors in [41] implements their clustering on a GPU. Then, once the weighted graph is built, we can simultaneously search for the nearest neighbors of each cluster. Finally, the model fitting step can be performed independently on each cluster.

Let $M_a \times M_a$ and $M_s \times M_s$ be the angular and spatial resolutions of our light field, K be the number of clusters, N the number of neighbors, I the number of iterations and S the number of initial scene flow estimates per cluster. Then, the time complexity of fitting our model to every cluster is $\mathcal{O}(13IKN^2S)$, where 13 corresponds to the number of parameters of our model. Complexity changes quadratically with the number of neighbors N due to the propagation step added in the RANSAC algorithm. In the case where we estimate an initial scene flow on every view, we have $S = M_a^2 M_s^2 / K$ and the complexity becomes $\mathcal{O}(13IN^2M_a^2M_s^2)$. However, if we want to reduce the complexity of our method, we do not need to have an initial scene flow estimate on every view. This is what we did for the dense dataset, where we took 3×3 views (instead of 9×9 views) in the initialization step. In this case, the complexity becomes $\mathcal{O}(117IN^2M_s^2)$.

F. Limitations

We further test our method (with the PWC-Net [20] initialization) on dense synthetic datasets ray-traced using POV-Ray, *Apples* and *Snails* provided by [13], that have very narrow baselines. Their angular resolution is 9×9 with a respective disparity range of $[1.1, 1.7]$ and $[0.3, 1.4]$. The scenes are photo-realistic with strong specular reflections, strong shadows and non-lambertian surfaces. Therefore they are very challenging light fields. We compare the mean square errors (MSE) produced by our estimations with those obtained by the method proposed in [13] (denoted PPDA for Preconditioned Primal-Dual Algorithm). The results are summarized in Table X.

We can see that our method fails to accurately estimate the scene flow on this dataset. This failure is mostly caused by the initialization step which produces too many outliers for the fitting of the model. The method we used, i.e. PWC-Net [20], is indeed not very robust to strong specularity and does not handle ambiguous situations, e.g. when a shadow is moving, it is unclear whether the optical flow should represent the apparent motion of the shadow or the motion of the surface the shadow is projected on. The method in [13] operating on epipolar plane images is on the contrary well suited for such light fields with narrow baselines but cannot be used when the disparity is large, the case we focus on in this paper.

TABLE X
MSE OF ESTIMATED OPTICAL FLOW AND DISPARITY

		Apples	Snails
MSE Δx:	PPDA [13]	0.3114	0.0996
	Ours	0.3283	0.2652
MSE Δy:	PPDA [13]	0.0245	0.0406
	Ours	0.0321	0.1095
MSE d^t:	PPDA [13]	0.0025	0.0036
	Ours	0.00023	0.0043

VI. CONCLUSION

In this paper, we have presented a new model of scene flow that takes into account the epipolar structure of light fields. Using the developed model, we proposed a method to estimate scene flows from sparsely sampled video light fields. This method is based on the three following steps: first an initial scene flow estimation, then a 4D clustering of the light field, and finally a fitting of the model for each cluster. For the performance evaluation, we have generated a synthetic dataset from the open source movie Sintel in order to extend the popular MPI Sintel benchmark to sparsely sampled light fields and scene flow. We also assessed our method on a dense light field dataset. Some qualitative tests were finally run on real light fields using the Fraunhofer dataset. For the sparse dataset, our method had lower errors for the optical flow, the disparity and the disparity variations than any other state-of-the-art scene flow approaches. On the dense dataset, our method gave comparable performances with the state-of-the-art light field method regarding the horizontal and vertical displacements (i.e. the optical flow) and disparity of the central view while yielding more accurate results on the whole 4D light field. Using the ground truth scene flow as initialization, we have shown that the ground truth locally conform to our affine model. This model is also a light way of describing a dense scene flow on the whole light field as it requires only 13 parameters per cluster. It could therefore be incorporated in a light field coding scheme as it would provide a prediction of every view of the light field at time $t+1$, only transmitting the central view at t alongside with the scene flow parameters.

Acknowledgment

This work was supported by the EU H2020 Research and Innovation Program under grant agreement N° 694122 (ERC advanced grant CLIM).

References

[1] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 606–619, Aug. 2013.

[2] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *J. of Computer Vision and Image Understanding*, vol. 145, pp. 148–159, Apr. 2016.

[3] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. So Kweon, "Accurate depth map estimation from a lenslet light field camera," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1547–1555.

[4] C.-T. Huang, "Empirical bayesian light-field stereo matching by robust pseudo random field modeling," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1–1, Feb. 2018.

[5] X. Jiang, M. Le Pendu, and C. Guillemot, "Depth estimation with occlusion handling from a sparse set of light field views," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2018, pp. 634–638.

[6] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *IEEE Int. Conf. on Computer Vision*, vol. 2. IEEE, 1999, pp. 722–729.

[7] M. Menze, C. Heipke, and A. Geiger, "Object scene flow," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 60–76, 2018.

[8] C. Vogel, K. Schindler, and S. Roth, "3d scene flow estimation with a piecewise rigid scene model," *Int. Journal of Computer Vision*, vol. 115, no. 1, pp. 1–28, 2015.

[9] A. Behl, O. Hosseini Jafari, S. Karthik Mustikovela, H. Abu Alhaija, C. Rother, and A. Geiger, "Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios?" in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 2574–2583.

[10] D. Sun, E. Sudderth, and H. Pfister, "Layered rgbd scene flow estimation," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 548–556.

[11] J. Quiroga, F. Devernay, and J. Crowley, "Local/global scene flow estimation," in *IEEE Int. Conf. on Image Processing (ICIP),*. IEEE, 2013, pp. 3850–3854.

[12] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, "Fast odometry and scene flow from rgb-d cameras based on geometric clustering," in *IEEE Int. Conf. on Robotics and Automation (ICRA),*. IEEE, 2017, pp. 3992–3999.

[13] S. Heber and T. Pock, "Scene flow estimation from light fields via the preconditioned primal-dual algorithm," in *German Conf. on Pattern Recognition*. Springer, 2014, pp. 3–14.

[14] J. Navarro and J. Garamendi, "Variational scene flow and occlusion detection from a light field sequence," in *Int. Conf. on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2016, pp. 1–4.

[15] P. Srinivasan, M. Tao, R. Ng, and R. Ramamoorthi, "Oriented light-field windows for scene flow," in *IEEE Int. Conf. on Computer Vision*, 2015, pp. 3496–3504.

[16] P. David, M. Le Pendu, and C. Guillemot, "Sparse to dense scene flow estimation from light fields," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2019.

[17] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conf. on Computer Vision (ECCV)*, Oct. 2012, pp. 611–625.

[18] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black, "Lessons and insights from creating a synthetic optical flow benchmark," in *ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation*, Oct. 2012, pp. 168–177.

[19] H. Zhu, X. Sun, Q. Zhang, Q. Wang, A. Robles-Kelly, H. Li, and S. You, "Full view optical flow estimation leveraged from light field superpixel," *IEEE Transactions on Computational Imaging*, 2019.

[20] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8934–8943.

[21] S. Wanner, C. Straehle, and B. Goldluecke, "Globally consistent multi-label assignment on the ray space of 4d light fields," in *CVPR*, 2013, pp. 1011–1018.

[22] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3487–3495.

[23] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision*, vol. 1, no. 1, pp. 7–55, Mar 1987.

[24] S. Heber, W. Yu, and T. Pock, "Neural epi-volume networks for shape from light field," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2252–2260.

[25] C. Shin, H.-G. Jeon, Y. Yoon, I. So Kweon, and S. Joo Kim, "Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4748–4757.

[26] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[27] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[28] J. Shi, X. Jiang, and C. Guillemot, "A framework for learning depth from a flexible subset of dense and sparse light field views," *IEEE Transactions on Image Processing*, 2019.

[29] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3061–3070.

[30] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[31] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4161–4170.

[32] J. Pacheco, S. Zuffi, M. Black, and E. Sudderth, "Preserving modes and messages via diverse particle selection," in *International Conference on Machine Learning*, 2014, pp. 1152–1160.

[33] S. Ma, B. M. Smith, and M. Gupta, "3d scene flow from 4d light field gradients," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[34] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 673–680.

[35] M. Tao, J. Bai, P. Kohli, and S. Paris, "Simpleflow: A non-iterative, sublinear optical flow algorithm," in *Computer graphics forum*, vol. 31. Wiley Online Library, 2012, pp. 345–353.

[36] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[37] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[38] A. Mustafa, M. Volino, J.-Y. Guillemaut, and A. Hilton, "4d temporally coherent light-field video," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 29–37.

[39] M. Levoy and P. Hanrahan, “Light field rendering,” in *23rd Annual Conf. on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH, 1996, pp. 31–42.

[40] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, “The lumigraph,” in *Proc. SIGGRAPH*, 1996, pp. 43–54.

[41] M. Hog, N. Sabater, and C. Guillemot, “Superrays for efficient light field processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1187–1199, Oct. 2017.

[42] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk *et al.*, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[43] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[44] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[45] D. E. O. Tzamarias, P. Akyazi, and P. Frossard, “A novel method for sampling bandlimited graph signals,” in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 126–130.

[46] T. R. (producer), “Sintel,” 2010, https://durian.blender.org/.

[47] T. Roosendaal, “Blender (3d software),” Blender Foundation, Blender Institute, Amsterdam, 1998, http://www.blender.org/.

[48] L. Dabała, M. Ziegler, P. Didyk, F. Zilly, J. Keinert, K. Myszkowski, H.-P. Seidel, P. Rokita, and T. Ritschel, “Efficient Multi-image Correspondences for On-line Light Field Video Processing,” *Computer Graphics Forum*, 2016.

[49] N. Sabater, G. Boisson, B. Vandame, P. Kerbiriou, F. Babon, M. Hog, T. Langlois, R. Gendrot, O. Bureller, A. Schubert, and V. Allie, “Dataset and pipeline for multi-view light-field video,” in *CVPR Workshops*, 2017.

[50] X. Jiang, M. Le Pendu, R. A. Farrugia, and C. Guillemot, “Light field compression with homography-based low-rank approximation,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1132–1145, 2017.

[51] 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*, 2005, pp. 765–776.

[52] T. Brox and J. Malik, “Large displacement optical flow: descriptor matching in variational motion estimation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 500–513, 2010.

Pierre David received the Engineering degree from the Institut d’Optique Graduate School - SupOptique, Palaiseau, France, and the M.Sc. degree in signal processing from University of Bordeaux, Bordeaux, France, both in 2016. He is currently a Ph.D. student in signal and image processing in the University of Rennes 1 at INRIA, Rennes. His research interests include light field imaging, motion estimation and frame interpolation.

Mikaël Le Pendu received the Engineering degree from the Ecole Nationale Supérieure des Mines de Nantes, Nantes, France, in 2012, and the Ph.D. degree in computer science from the University of Rennes 1, Rennes, France, in 2016. His Ph.D. studies were conducted in conjunction between the Institut National de Recherche en Informatique et en Automatique (INRIA) and Technicolor in Rennes, France, and addressed the compression of High-Dynamic Range video content. After pursuing post-doctoral research at INRIA on light field image processing, he is now a Post-Doctoral Researcher at Trinity College Dublin. His research interests include High Dynamic Range as well as Light Field imaging and covers the full processing chain from capture to compression, including editing tasks.

Christine Guillemot, IEEE FELLOW, is Director of Research at INRIA. She holds a Ph.D. degree from ENST (École Nationale Supérieure des Télécommunications), Paris, and an Habilitation for Research Direction from the University of Rennes. From 1985 to Oct. 1997, she has been with FRANCE TELECOM, where she has been involved in various projects in the area of image and video coding and processing for TV, HDTV and multimedia. From Jan. 1990 to mid 1991, she has worked at Bellcore, NJ, USA, as a visiting scientist. Her research interests are signal and image processing, and computer vision.

She has served as Associate Editor for IEEE TRANS. ON IMAGE PROCESSING (from 2000 to 2003, and from 2014-2016), for IEEE TRANS. ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (from 2004 to 2006), and for IEEE TRANS. ON SIGNAL PROCESSING (2007-2009). She has served as senior member of the editorial board of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING (2013-2015) and is currently senior area editor of IEEE TRANS. ON IMAGE PROCESSING.