Sparse to Dense Scene Flow Estimation from Light Fields

Pierre David, Mikaël Le Pendu, Christine Guillemot

HAL Id: hal-02123544
https://hal.science/hal-02123544

Submitted on 8 May 2019

SPARSE TO DENSE SCENE FLOW ESTIMATION FROM LIGHT FIELDS

Pierre David⋆   *Mikaël Le Pendu*†   *Christine Guillemot*⋆

⋆Inria, Campus Universitaire de Beaulieu, 35042 Rennes, France
†Trinity College Dublin, Ireland

ABSTRACT

The paper addresses the problem of scene flow estimation from sparsely sampled video light fields. The scene flow estimation method is based on an affine model in the 4D ray space that allows us to estimate a dense flow from sparse estimates in 4D clusters. A dataset of synthetic video light fields created for assessing scene flow estimation techniques is also described. Experiments show that the proposed method gives error rates on the optical flow components that are comparable to those obtained with state of the art optical flow estimation methods, while computing a more accurate disparity variation when compared with prior scene flow estimation techniques.

Index Terms— Scene flow, optical flow, light field, sparse to dense

1. INTRODUCTION

Light fields have proven to be very useful for scene analysis, and in particular for scene depth estimation [1, 2, 3, 4, 5]. However, while a lot of work has been dedicated to scene analysis from static light fields, very little effort has been invested on the problem of dynamic scene analysis from video light fields. The concept of scene flow has first been defined in [6] as describing the 3D geometry as well as the 3D motion of each scene point. 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. However, in the recent literature (e.g. [7, 8, 9, 10]), the scene flow is instead defined as a direct extension of the optical flow, where the depth (or disparity) variation Δd of objects along time is represented in addition to the apparent 2D motion $(\Delta x, \Delta y)$.

Considering this definition, the scene flow analysis problem has first been addressed for stereo video sequences: in [7, 8, 9] a scene flow is estimated assuming that the scene can be decomposed into rigidly moving objects. Several methods based on RGB-D videos have also been developed [10], [11], [12]. The first methods for scene flow analysis from dense light fields have been proposed in [13] and [14]. Both are based on variational models. The authors in [15] proposed oriented light field windows to estimate the scene flow. All these methods rely on epipolar images which only provide sufficient information for scene flow estimation of densely sampled light fields (as those captured with plenoptic cameras). However, they do not address the case of sparse light fields (i.e. with large baselines).

In this paper, we focus on the problem of scene flow analysis from sparse video light fields. This problem is made difficult due to

the large temporal and angular occlusions. To cope with this difficulty, we propose a method for interpolating, in the 4D ray space, a sparse scene flow into a dense one. The sparse-to-dense approach naturally handles occlusions: while the sparse estimation can only be obtained on non-occluded parts of the image, the dense interpolation extends these estimations to every pixel (occluded or not). The proposed method is based on an affine model in the 4D ray space that allows us to estimate the dense flow from sparse estimates in nearby clusters. Note that an advantage of our sparse-to-dense approach is the possibility to use it as a post-processing step in other scene flow estimation methods, in the same way as EpicFlow [16] has been used in other methods (e.g. [17]) to interpolate the optical flow of pixels detected as outliers.

In order to validate the proposed algorithm, we have generated a synthetic light field video dataset based on the Sintel movie (used in the optical flow benchmark [18, 19]). This dataset is composed of two synthetic light fields ('Bamboo2' and 'Temple1') of 3×3 views of 1024×536 pixels and 50 frames. The light field views are provided with the corresponding ground truth scene flow (optical flow and disparity variation).

The proposed method has been assessed using this synthetic dataset in comparison with the oriented window method in [15], and with two methods proposed for optical flow estimation: the sparse to dense estimation method EpicFlow [16] and a state-of-the-art technique based on a deep learning architecture called PWC-Net [20]. Note that, since the depth estimation component of the method in [15] has been designed for dense light fields and is hardly applicable when the baseline is large, when using this method, we have coupled the optical flow estimation performed by the method with the ground truth disparity. The two optical flow estimation techniques [20] and [16] are used for separately estimating the optical flow in each view as well as the disparity between views. Experimental results show that the proposed method gives optical flow endpoint error rates that are comparable to those obtained with state of the art optical flow estimation methods, while computing at the same time the disparity variation with lower mean absolute errors. Although the accuracy of our optical flow is slightly lower than that of PWC-Net, the depth variation maps estimated by our method have considerably lower errors.

2. NOTATIONS AND METHOD OVERVIEW

Let us consider the 4D representation of Light Fields proposed in [21] and [22] to describe the radiance along the different light rays. This 4D function, at each time instant, is denoted $LF^t(x, y, u, v)$, where the pairs $\boldsymbol{x} = (x, y)$ and $\boldsymbol{u} = (u, v)$ respectively denote the spatial and angular coordinates of light rays. A view $\boldsymbol{u}$ of a light field at t is written $L_{\boldsymbol{u}}^t$. We denote by $C : (I, I') \rightarrow (\mathcal{S}, \mathcal{S}', f)$ the function that determines two sets $\mathcal{S}$ and $\mathcal{S}'$ of matching points in I and I' respectively. The associated bijection f is such that $f(\mathcal{S}) = \mathcal{S}'$.
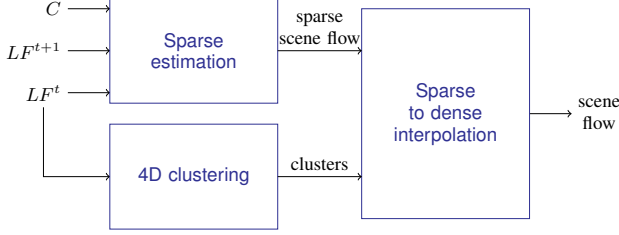
Fig. 1. Overview of the sparse to dense scene flow estimation
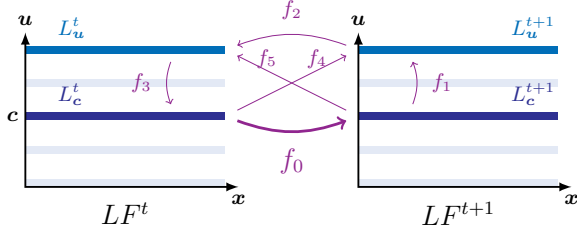
Fig. 2. Matching between a view $\boldsymbol{u}$ and the central view $\boldsymbol{c}$ at t and $t+1$, shown in epipolar images at t and t+1.

The inverse bijection is noted f^{-1}.

The proposed method takes as inputs two consecutive frames of a video light field $LF^t(x, y, u, v)$, $LF^{t+1}(x, y, u, v)$, a matching function C and estimates a consistent dense scene flow on each view. It produces disparity maps at t and t+1 and a disparity variation map between t and t+1 (in addition to the optical flow components). The algorithm proceeds as follows (see Fig.1). The first step of the method is the robust sparse scene flow estimation. All light field rays are then grouped in 4D clusters to guide the sparse to dense interpolation of the scene flow. An affine model is computed from the sparse flow estimates in neighboring clusters, and then used to compute a flow value in each light field pixel.

3. SPARSE SCENE FLOW ESTIMATION

The method pivots on the central view $L_{\boldsymbol{c}}^t$. We first compute $(\mathcal{S}_0, \mathcal{S}_0', f_0) = C(L_{\boldsymbol{c}}^t, L_{\boldsymbol{c}}^{t+1})$. Then, for every non central view $\boldsymbol{u}$, we perform three kinds of matching (see Fig. 2), angular matching to estimate the sets of points $(\mathcal{S}_1, \mathcal{S}_1', f_1) = C(L_{\boldsymbol{c}}^{t+1}, L_{\boldsymbol{u}}^{t+1})$ and $(\mathcal{S}_3, \mathcal{S}_3', f_3) = C(L_{\boldsymbol{u}}^t, L_{\boldsymbol{c}}^t)$, temporal matching to estimate the sets of points $(\mathcal{S}_2, \mathcal{S}_2', f_2) = C(L_{\boldsymbol{u}}^t, L_{\boldsymbol{u}}^t)$, and temporo-angular matching to estimate $(\mathcal{S}_4, \mathcal{S}_4', f_4) = C(L_{\boldsymbol{c}}^t, L_{\boldsymbol{u}}^{t+1})$ and $(\mathcal{S}_5, \mathcal{S}_5', f_5) = C(L_{\boldsymbol{c}}^{t+1}, L_{\boldsymbol{u}}^t)$. In the experiments, we used [23] as function C.

The next step is to link the different sets together based on the distance between the points computed as in [24]. We define a distance D' based on color, angular and spatial proximity in the 7D space $[labuvxy]$ as :

$$D'(P_i, P_j) = \sqrt{d_c{}^2 + \frac{m_s{}^2}{S^2}d_s{}^2 + \frac{m_a{}^2}{A^2}d_a{}^2} \quad (1)$$

with :

$$S = \sqrt{h \times w / |\mathcal{S}_0|} \qquad A = \sqrt{N_u \times N_v} \quad (2)$$

where d_c, d_s and d_a are the color, spatial and angular distances defined as euclidean distances in the CIELAB colorspace, the $[xy]$

space and the $[uv]$ space respectively. The variables w and h are the width and height of a view, N_u and N_v are the angular size of the light field and m_s is a parameter controlling the balance between color and spatial distances that we fix to 10 in the experiments. For this step, we have $d_a = 0$ because we are aiming at linking points in the same view, but this distance will be used later in the clustering step. Based on the above distances, for each point $P_0 \in \mathcal{S}_0$, we build a chain of points $(P_0, P_0', P_1, P_1', P_2, P_2', P_3, P_3')$ by searching for the nearest neighbors successively in $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ (see Eq. (3) and Fig. 3a). We call this step a forward nearest neighbor search and denote **NN** the simple nearest neighbor search function for a point $P_i \in \mathcal{S}_i$ in a set $\mathcal{S}_j$, according to the distance D.

$$P_0 \xrightarrow{f_0} P_0' \xrightarrow{\textbf{NN}} P_1 \xrightarrow{f_1} P_1' \xrightarrow{\textbf{NN}} P_2 \xrightarrow{f_2} P_2' \xrightarrow{\textbf{NN}} P_3 \xrightarrow{f_3} P_3' \quad (3)$$

To ensure the robustness of the chain of points, we then perform a backward nearest neighbor check as in Eq.(4). Given a point $P_i' \in \mathcal{S}'_i$ whose nearest neighbor in $\mathcal{S}_j$ is P_j, we check if P_i' is reciprocally the nearest neighbor of P_j in $\mathcal{S}'_i$.

$$P_3' \xrightarrow{f_3^{-1}} P_3 \xrightarrow{\textbf{NN?}} P_2' \xrightarrow{f_2^{-1}} P_2 \xrightarrow{\textbf{NN?}} P_1' \xrightarrow{f_1^{-1}} P_1 \xrightarrow{\textbf{NN?}} P_0' \xrightarrow{f_0^{-1}} P_0 \quad (4)$$

Let $\mathcal{A}_1$ be the set of points $P_0 \in \mathcal{S}_0$ that passed the backward nearest neighbor test. We have $\mathcal{A}_1 \subset \mathcal{S}_0$. We then perform a similar forward nearest neighbor search and backward nearest neighbor test for each point $P_0 \in \mathcal{S}_0$ but this time with the two temporo-angular matching sets $\mathcal{S}_4$ and $\mathcal{S}_5$, building the chain of points (P_0, P_0', P_4) and (P_0, P_0', P_5) (see Fig. 3b).

Let $\mathcal{A}_2$ be the set of points $P_0 \in \mathcal{S}_0$ that passed the backward nearest neighbor test in $\mathcal{S}_4$ and $\mathcal{S}_5$. We have $\mathcal{A}_2 \subset \mathcal{S}_0$. For each point $P_0 \in \mathcal{A} = \mathcal{A}_1 \cap \mathcal{A}_2$, we build a complete chain of points $\boldsymbol{g} = (P_0, P_1, P_2, P_3, P_4, P_5)$.

Given a chain of points, we check the consistency of each matching point $P_i' = f_i(P_i)$ with the point P_j in the same view and instant. More precisely, we discard the chain if either $D(P_0, P_3') > \tau$, $D(P_2, P_4')) > \tau$, or $D(P_3, P_5') > \tau$, where τ is a distance threshold that was fixed to 10 in our experiments.

Any remaining chain $\boldsymbol{g}$ is used to estimate a sparse scene flow. From $\boldsymbol{g}$, we can compute the scene flow on two points of the light field LF^t : on the central view $L_{\boldsymbol{c}}^t$ at P_0 and on the view $L_{\boldsymbol{u}}^t$ at P_2':

$$P_0' - P_0 = \begin{pmatrix} \Delta x_{\boldsymbol{c}} \\ \Delta y_{\boldsymbol{c}} \end{pmatrix} \quad \text{and} \quad P_2' - P_2 = -\begin{pmatrix} \Delta x_{\boldsymbol{u}} \\ \Delta y_{\boldsymbol{u}} \end{pmatrix} \quad (5)$$

Let d_t and d_{t+1} be the disparities at t and $t+1$. We have:

$$P_1' - P_1 = (\boldsymbol{u} - \boldsymbol{c}) \cdot d_{t+1} \quad \text{and} \quad P_3' - P_3 = -(\boldsymbol{u} - \boldsymbol{c}) \cdot d_t \quad (6)$$

With the temporo-angular matching, we also have:

$$\begin{aligned} P_4' - P_4 &= -(P_2' - P_2) - (P_3' - P_3) \\ &= (P_0' - P_0) + (P_1' - P_1) \\ P_5' - P_5 &= -(P_0' - P_0) - (P_3' - P_3) \\ &= (P_1' - P_1) + (P_2' - P_2) \end{aligned} \quad (7)$$

We solve this linear overdetermined system with a least square method in order to determine the values $\Delta x_{\boldsymbol{c}}, \Delta y_{\boldsymbol{c}}, \Delta x_{\boldsymbol{u}}, \Delta y_{\boldsymbol{u}}, d_t$ and d_{t+1} for each chain $\boldsymbol{g}$.

(a) Chain of angular and temporal correspondences | (b) Chain of temporo-angular correspondences
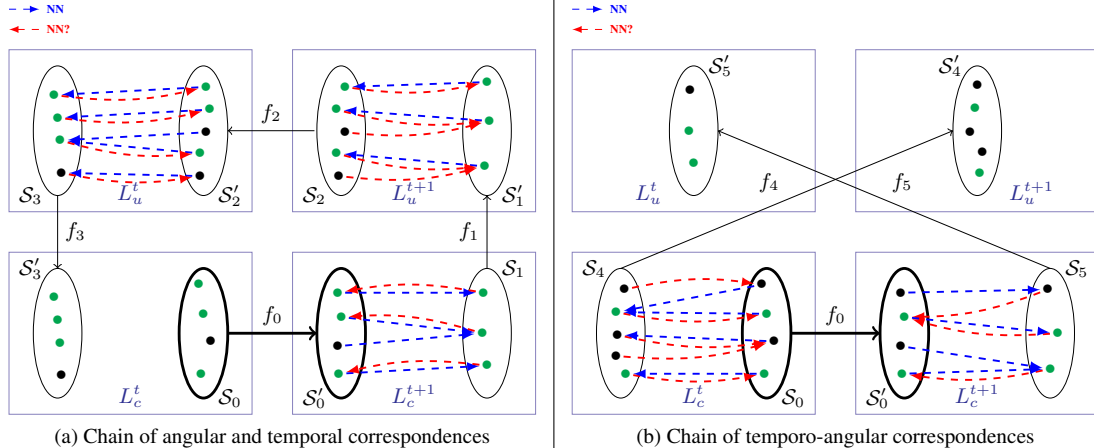
Fig. 3. Forward nearest neighbor search and backward nearest neighbor check. The green dots represent the points that passed the backward nearest neighbor check.

4. CLUSTERING THE LIGHT FIELD

In order to estimate a dense scene flow from sparse estimates, we need to make an assumption about the scene. We assume that rays which have similar colors and are located near each other in the light field should have similar motions. As a consequence, similarly to [25] but extended to 4D light fields, we propose to decompose the light field into clusters based on color, spatial and angular proximity. Then, the clustering will be used to estimate one scene flow model per cluster. We generalize in 4D the SLIC superpixels developed in [24] to cluster the light field LF^t. We use as distance D', already defined in Eq.(1). We only modify the definition of S :

$$S = \sqrt{h \times w / K} \qquad (8)$$

K denotes the number of desired clusters. We use the parameters m_s and m_a to control the balance between spatial, angular and color proximity. S (in Eq. (8)) and A (in Eq. (2)) are the maximum spatial and angular distances expected in a cluster.

The spatial positions of the centroids C_k of the clusters are initialized on a regular grid of step S, while their initial angular positions are randomly sampled. Then, each pixel i is assigned to the nearest cluster center C_k. We fix the search region to $N_u \times N_v \times 2S \times 2S$ because the expected approximate spatial size of each cluster is $S \times S$ and the approximate angular size is $N_u \times N_v$. Indeed, each cluster is excpected to contain pixels in all the $N_u \times N_v$ views.

Once the assignment step is over, we update the cluster centers by computing the new centroid of each cluster. We only need to iterate this second step $N_{\text{iteration}} = 10$ in order to have stabilized cluster centroids.

5. SPARSE TO DENSE INTERPOLATION

Once we have divided the light field into clusters, we estimate a model for each one of them. For each cluster, we look for the points estimated in Section 3 that are inside the cluster, we compute the mean value of the sparse estimates and associate it with the centroids of the cluster. For clusters that have no seeds inside them, we do not do anything. We end up with initialized clusters and unitialized ones.

As in [16], we build a weighted graph where each node is a cluster (initialized or not) and where an edge between two nodes

means that the two clusters are adjacent. The associated weight is defined as the distance D' (as in Eq.(1)) between the centroids of the two clusters. We then look for the N nearest initialized neighbors using Dijkstra's algorithm on the graph [26], discarding every empty cluster.

For each cluster, we estimate the parameters of the affine model in Eq.(9) in the 4D space by fitting the model on the initialized centroids using the RANSAC algorithm [27].

$$\begin{aligned} a_1 u + b_1 v + c_1 x + d_1 y + e_1 &= x + \Delta x \\ a_2 u + b_2 v + c_2 x + d_2 y + e_2 &= y + \Delta y \\ a_3 u + b_3 v + c_3 x + d_3 y + e_3 &= \Delta d \end{aligned} \qquad (9)$$

We then apply the model to every pixel belonging to the cluster to compute its $(\Delta x, \Delta y, \Delta d)$ scene flow value.

As a final post-processing step, we perform an energy minimization like in [16] for the optical flow of each subaperture image independently. To regularize the disparity variation map, we then perform joint bilateral filtering using the optical flow as a guide.

6. EVALUATION

6.1. Scene Flow Dataset

For our experiments, we have prepared a synthetic video Light Field dataset[1]. For that purpose, we have used the production files of the open source movie Sintel [28] and have modified them in the Blender 3D software [29] in order to render an array of 3x3 views. Similarly to the MPI Sintel flow dataset [18, 19], 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 and 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 1024×536 pixels and 50 frames corresponding to the scenes 'Bamboo2' and

---

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

		Bamboo2				Temple1			
		clean		final		clean		final	
		EPE OF	MAE Δd	EPE OF	MAE Δd	EPE OF	MAE Δd	EPE OF	MAE Δd
Central view only	Ours	**1.007**	**0.136**	**1.102**	**0.140**	**1.042**	**0.109**	**1.383**	**0.128**
	OW [15]	1.421	0.356	1.462	0.345	2.061	0.152	2.374	0.162
	Epic [16]	1.078	0.685	1.280	0.663	5.405	0.664	2.472	1.009
	PWC [20]	0.945	0.576	1.017	0.596	1.042	0.291	1.322	0.325
Global	Ours	1.090	**0.140**	1.169	**0.142**	1.109	**0.111**	1.453	**0.131**
	OW [15]	/	/	/	/	/	/	/	/
	Epic [16]	1.071	1.020	1.264	1.015	5.676	0.730	2.609	1.208
	PWC [20]	0.947	0.873	1.017	0.884	1.038	0.385	1.323	0.429

Table 1. Results on our light field scene flow dataset for the optical flow endpoint error EPE OF and for the disparity variation mean absolute error MAE Δd. This latter is only computed on pixels that remain visible between two adjacent frames because of the lack of ground truth in occluded and disoccluded areas. The lowest errors are in red, the second lowest errors in orange.
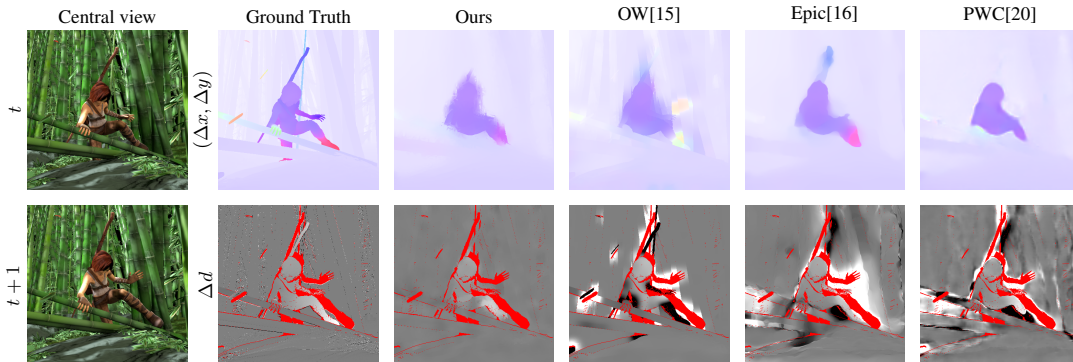

Fig. 4. Visual comparison of our method with [15, 16, 20]. The optical flows are visualized with the Middlebury color code, and the disparity variations are visualized using a gray-scale representation. The red pixels are the occlusion mask where there is no ground truth disparity variation available.

'Temple1' in [18]. The disparities (in pixels) between neighboring views are in the range $[-8, +52]$ for 'Bamboo2' and $[-22, +9]$ for 'Temple1'.

6.2. Results

The proposed method is first assessed in comparison with the method in [15], referred to here as OW (Oriented Window). This latter was designed for dense light fields captured with plenoptic cameras. However, the optical flow searched via the oriented window can be combined with disparity maps estimated by methods suitable for sparse light fields. In the test reported here, we used for this method ground truth disparity maps, thus showing the best results it can give for the estimated scene flow. We also compare the method with a naive approach that would consist in separately estimating the disparity maps at t and t+1, the optical flows between t and t+1 and finally computing the disparity variation as the difference of disparities along the optical flows. We consider two methods proposed for optical flow estimation: the sparse to dense estimation method EpicFlow [16] and a state-of-the-art technique based on a deep learning architecture called PWC-Net [20] Note that this separate disparity and optical flow estimation does not handle occlusions. So, the disparity variation in occluded or disoccluded areas will never be consistent.

The results are summarized in Table 1. For each successive light field frame of the four sequences (*Bamboo2* and *Temple1*, both rendered as *clean* and *final*), we compute the endpoint error for the optical flow (EPE OF) and separately the mean absolute error for the disparity variation (MAE Δd). The latter is only computed for disoccluded pixels because there is no ground truth on occluded pixels. We compute these two errors on every ray of the light field (*Global*) and because the method proposed in [15] only gives the scene flow for the central view, we also compute the errors on the central view only. We can observe that although [20] yields the most accurate optical flows, our method has a lower optical flow end point error than the two other methods and that it gets close to [20] for some sequences like *Temple1 clean* and *Bamboo2 clean*. Regarding the disparity variation, our method provides much lower errors than any other tested method. In the end, we propose a method that is slightly less precise than state of the art optical flow methods but more accurate in terms of disparity variation.

7. CONCLUSION

In this paper, we have presented a new method to estimate scene flows from sparsely sampled video light fields. The method is based on three steps: first a sparse scene flow estimation, then a 4D clustering of the light field, and finally a sparse to dense scene flow interpolation 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 dataset to light fields and scene flow. Our method gives comparable performances with state-of-the art approaches only considering the horizontal and vertical displacements (i.e. the optical flow). However, significant improvements were obtained in the estimation of the disparity variation.

8. 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*. IEEE, 1999, vol. 2, pp. 722–729.

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

[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.B. 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.F. 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.W. 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] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 1164–1172.

[17] C. Bailer, B. Taetz, and D. Stricker, "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation," in *IEEE Int. Conf. on Computer Vision*, 2015, pp. 4015–4023.

[18] 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.

[19] 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.

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

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

[23] Y. Li, Y. Hu, R. Song, P. Rao, and Y. Wang, "Coarse-to-fine patchmatch for dense correspondence," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 1, pp. 1–1, 2017.

[24] 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.

[25] Y. Hu, Y. Li, and R. Song, "Robust interpolation of correspondences for large displacement optical flow," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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

[27] M.A. Fischler and R.C. 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.

[28] T. Rosendaal (producer), "Sintel," Blender Foundation, Durian Open Movie Project, 2010, https://durian.blender.org/.

[29] "Blender (3d software)," http://www.blender.org/.