ANGULARLY CONSISTENT LIGHT FIELD VIDEO INTERPOLATION

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

⋆Inria, Campus Universitaire de Beaulieu, 35042 Rennes, France
†V-SENSE, School of Computer Science and Statistics, Trinity College Dublin, Ireland

ABSTRACT

In this paper, we address the problem of temporal interpolation of sparsely sampled video light fields using dense scene flows. Given light fields at two time instants, the goal is to interpolate an intermediate light field to form a spatially, angularly and temporally coherent light field video sequence. We first compute angularly coherent bidirectional scene flows between the two input light fields. We then use the optical flows and the two light fields as inputs to a convolutional neural network that synthesizes independently the views of the light field at an intermediate time. In order to measure the angular consistency of a light field, we propose a new metric based on epipolar geometry. Experimental results show that the proposed method produces light fields that are angularly coherent while keeping similar temporal and spatial consistency as state-of-the-art video frame interpolation methods.

Index Terms— Light fields, Interpolation, Optical Flow, Deep Learning

1. INTRODUCTION

Increasing the video frame rate by temporal frame interpolation has been a widely addressed problem, e.g. for compression purposes, or to create high quality videos or to produce slow motion effects. With the recent development of deep learning, this field of research has profoundly changed. Today, almost every method that occupies the top of the frame interpolation benchmarks[1] uses deep learning. In the meantime, light fields have shown that they could be useful in numerous computer vision and image processing applications such as depth estimation, optical flow, refocusing or view interpolation. In this paper, we focus on synthesizing a whole light field frame between two consecutive light fields frames. The synthesized light field should be consistent with the previous and following frames but it should also preserve the epipolar structure. In a nutshell: the generated light field sequence should be temporally and angularly consistent.

In order to synthesize an intermediate frame in a 2D video sequence, different deep learning models have been proposed. The authors in [1] and [2] both use a simple U-Net architecture which is an autoencoder with skipped connections between the encoder and decoder layers. In both methods, the input is a tensor containing the two consecutive frames. However, the network in [1] estimates a flow that is used to backwarp and merge the two input images at the intermediate time instant, whereas in [2], the network generates pixel-dependent kernels that are used to convolve and merge the input frames.

More recent approaches proposed to divide the problem into sub-problems and to train a specific network for each subproblem. For example, in [3] and [4], a network first estimates an optical flow between the two input images, and a second network then uses the estimated flow to warp and synthesize an intermediate frame. The authors in [5, 6] use multiple parallel U-Nets to extract cues such as context, disparity, optical flow or visibility and a final U-Net to generate the intermediate frame.

To the best of our knowledge, only one method has been proposed to temporally interpolate a light field sequence [7]. This approach considers a hybrid capture system composed of a plenoptic camera with a low frame-rate (3 fps) and a classical camera with a standard frame-rate (30 fps). Thanks to a neural network, the frames captured by the classical camera are warped to the different views of the the plenoptic camera.

In this article, we choose to have a full light field approach that does not require an additional single view video with higher frame rate to interpolate the light field frames. We also take into consideration that there are very few light field video datasets and therefore a neural network that would take light fields as inputs and outputs would be difficult to train. Furthermore, such an approach might require a model with a very larger number of parameters to handle the high dimensional data represented by video light fields. Instead, we enforce the angular consistency of the light field during the optical flow estimation. Then, temporal consistency is enforced by synthesizing individually every view of the light field using a neural network trained on traditional videos and taking best advantage of the optical flows obtained previously. For the neural network architecture and model, we took inspiration from the arbitrary-time flow interpolation network of [3].

In order to validate the proposed method, we use a synthetic light field video dataset based on the Sintel movie provided by [8]. We assess our algorithm in comparison with

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

[1] http://vision.middlebury.edu/flow/eval/results/results-i1.php

the Super SloMo method in [3], and with two other video frame interpolation methods, Separable Adaptive Convolution [2] and Deep Voxel Flow [1], that we separately apply on each view of the light field. For each method, we estimate the PSNR and the SSIM of each interpolated view, alongside a new metric that we introduce to quantify the angular consistency of a given light field frame. Experimental results show that the proposed method gives a better angular consistency among the synthesized light field. While our PSNR and SSIM results are comparable to state-of-the-art video interpolation methods independently applied on each view, visual improvement is also observed, in particular for areas with large motion.

2. METHOD

Let LF^0 and LF^1 be two consecutive light fields in a video. The goal of this method is to synthesize a light field LF^t at intermediate time instant $t \in [0, 1]$. First, we compute an angularly consistent bidirectional optical flow between LF^0 and LF^1. Then, we use a convolutional neural network to refine independently the flows of each view of the light field and to synthesize the light field at an intermediate time instant t. The method is summarized in Figure 1.

2.1. 4D consistent optical flow

The first step of the proposed light field frame interpolation method is to estimate an optical flow for every view of the light field ("4D scene flow estimation" block in Fig.1). In order for the interpolation to be angularly consistent, we regularize optical flows of each view with the method described in [9]. The method first constructs clusters of light rays in the 4D space and then fits a 4D local affine model to each cluster taking into account the epipolar structure of the light field. We estimate a bidirectional optical flow, that is the optical flow $F_{0\to1}$ from LF^0 to LF^1, and the optical flow $F_{1\to0}$ from LF^1 to LF^0.

In order to interpolate the light field at frame t, we then need to compute the intermediate optical flows $F_{0\to t}$ and $F_{1\to t}$ ("Intermediate flow estimation" block in Fig.1). In [3], this step is simply performed as a weighted mean of $F_{0\to1}$ and $F_{1\to0}$. However, this approach does not estimate accurately the optical flow on the edges of an object and on the occluded areas. The authors used this simple estimation because it occurs between two convolutional neural networks (the first one estimates a bidirectional optical flow and the second one performs the frame synthesis). Therefore, they need this step to be easily differentiable in order to train simultaneously their two networks. However, in our case, we do not train a network to estimate a bidirectional optical flow. As a result, we are not constrained to use a differentialble method.

We choose to use a method similar to the one described in [10]. The algorithm is the following:

1. Forward-warp the flow $F_{0\to1}$ to time t to obtain F_t^0 where:

$$F_t^0(\text{round}(\boldsymbol{x} + tF_{0\to1}(\boldsymbol{x}))) = F_{0\to1}(\boldsymbol{x}) \quad (1)$$

The flow vectors are splatted with a splatting radius of 0.5 and when multiple flow vectors are projected to the same pixel location, we choose the vector that provides the best photo-consistency, that is the flow vector $F_{0\to1}(\boldsymbol{x})$ that minimizes the projection error $|LF^0(\boldsymbol{x}) - LF^1(\boldsymbol{x} + F_{0\to1}(\boldsymbol{x}))|$.

2. Forward-warp the flow $F_{1\to0}$ to time t to obtain F_t^1 where:

$$F_t^1(\text{round}(\boldsymbol{x} + (1-t)F_{1\to0}(\boldsymbol{x}))) = F_{1\to0}(\boldsymbol{x}) \quad (2)$$

We perform the same splatting operation and photo-consistency check as in the previous step

3. Merge the two flows into F_t:

$$F_t(\boldsymbol{x}) = \begin{cases} +F_t^0(\boldsymbol{x}) & \text{if } F_t^1(\boldsymbol{x}) \text{ is not defined} \\ -F_t^1(\boldsymbol{x}) & \text{if } F_t^0(\boldsymbol{x}) \text{ is not defined} \\ (1-t)F_t^0(\boldsymbol{x}) - tF_t^1(\boldsymbol{x}) & \text{otherwise} \end{cases} \quad (3)$$

4. Inpaint the holes in F_t using an outside-in interpolation

5. The final intermediate flows are finally estimated as:

$$F_{t\leftarrow0} = -tF_t \quad (4)$$

$$F_{t\leftarrow1} = (1-t)F_t \quad (5)$$

2.2. View-wise light field synthesis network

Once we have estimated the intermediate flows $F_{t\leftarrow0}$ and $F_{t\leftarrow1}$, we could directly use them to back-warp the views of the light field from LF^0 or LF^1. However, this simple approach produces annoying artefacts especially around motion boundaries. To tackle this issue, similarly to [3], we use a convolutional neural network with a U-Net architecture to refine independently the flows of each view of the light field and to produce visibility maps that handle temporal occlusions ("View-wise light field synthesis network" block in Fig.1). Let L_{uv}^t be a view of LF^t at angular position (u, v). For each view L_{uv}^t to estimate, we want the CNN to produce refined optical flows $F_{t\leftarrow0}^r$, $F_{t\leftarrow1}^r$ and soft visibility maps $V_{t\leftarrow0}$, $V_{t\leftarrow1}$ such that:

$$F_{t\leftarrow0}^r = F_{t\leftarrow0} + \Delta F_{t\leftarrow0} \quad (6)$$

$$F_{t\leftarrow1}^r = F_{t\leftarrow1} + \Delta F_{t\leftarrow1} \quad (7)$$

$$V_{t\leftarrow1} = 1 - V_{t\leftarrow0} \quad \text{w.r.t.} \quad V_{t\leftarrow0} \in [0, 1] \quad (8)$$

In practice, the neural network only gives $V_{t\leftarrow0} \in [0, 1]$ and we then compute $V_{t\leftarrow1}$ from it to ensure the validity of Equation 8. Moreover, instead of directly estimating the refined

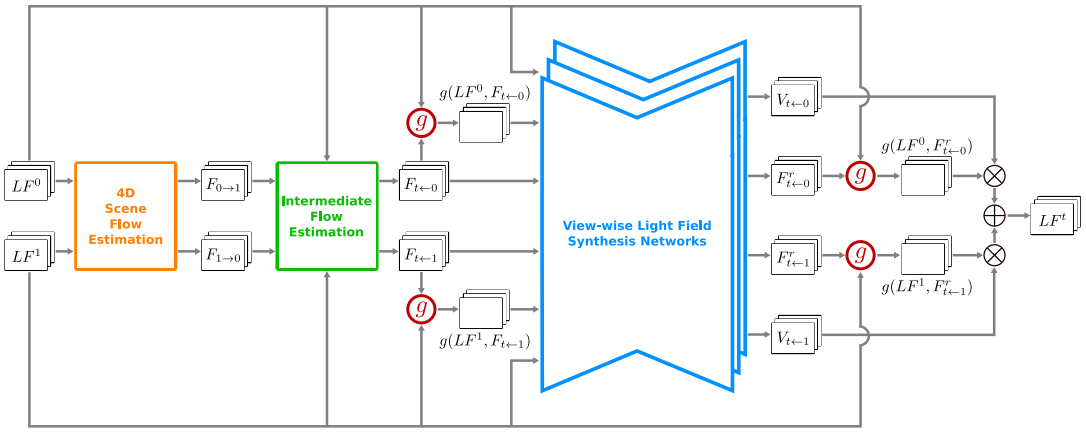

Fig. 1. Block diagram of our method.

optical flows from the network, we make it generate $\Delta F_{t\leftarrow0}$ and $\Delta F_{t\leftarrow1}$, this produces better results according to [3].

Using the refined optical flows, we produce back-warped views from L^0_{uv} and L^1_{uv} that we merge thanks to $V_{t\leftarrow0}$ and $V_{t\leftarrow1}$ to form the intermediate light field view $\hat{L}^t_{uv}$:

$$\hat{L}^t_{uv} = (\,\alpha_0 \odot g(L^0_{uv}, F^r_{t\leftarrow0}) + \alpha_1 \odot g(L^1_{uv}, F^r_{t\leftarrow1})\,) \oslash Z, \quad (9)$$

where $g(\cdot,\cdot)$, $\odot$ and $\oslash$ respectively denote differentiable back-warp operator, Hadamard product and pixel-wise division. α_0, α_1 and Z are defined as follows:

$$\alpha_0 = (1-t)V_{t\leftarrow0} \quad \text{and} \quad \alpha_1 = tV_{t\leftarrow1} \quad (10)$$

$$Z = \alpha_0 + \alpha_1 \quad (11)$$

For the architecture of the CNN, we take the same model as in [3], that is a U-Net architecture, consisting of an encoder and a decoder with skipped connections between encoder and decoder layers of the same size. The encoder consists of 6 hierarchies which are composed of two convolutional and one Leaky ReLu (with $\alpha = 0.2$) layers. Every hierarchy except the last one ends with an average pooling layer to decrease the spatial dimension by 2. The decoder consists of 5 hierarchies that start with a bilinear upsampling layer to increase the spatial dimension by a factor of 2. It is followed by two convolutional and Leaky ReLU ($\alpha = 0.2$) layers. For each convolutional layer, the kernel size is set to 3×3.

For the training, the loss function is a linear combination of a reconstruction loss $\mathcal{L}_r$, a warping loss $\mathcal{L}_w$, a perceptual loss $\mathcal{L}_p$ and a smoothness loss $\mathcal{L}_s$. Compared to [3] we only change the smoothness term, imposing the smoothing constraint to the final optical flows. Originally, the smoothness constraint was applied on $F_{0\to1}$ and $F_{1\to0}$ since they were simultaneously estimated with another network.

$$\mathcal{L} = w_r\mathcal{L}_r + w_w\mathcal{L}_w + w_p\mathcal{L}_p + w_s\mathcal{L}_s \quad (12)$$

with:

$$\mathcal{L}_r = \|I^t - \hat{I}^t\|_1 \quad (13)$$

$$\mathcal{L}_w = \|I^t - g(I^0, F^r_{t\leftarrow0})\|_1 + \|I^t - g(I^1, F^r_{t\leftarrow1})\|_1 \quad (14)$$

$$\mathcal{L}_p = \|\boldsymbol{\psi}(I^t) - \boldsymbol{\psi}(\hat{I}^t)\|_2 \quad (15)$$

$$\mathcal{L}_s = \|\nabla F^r_{t\leftarrow0}\|_1 + \|\nabla F^r_{t\leftarrow1}\|_1 \quad (16)$$

where I^t, $\hat{I}^t$, ∇ and $\boldsymbol{\psi}$ respectively denote the ground truth intermediate frame, the synthesized frame estimated with Eq.9, the gradient operator and the `conv4_3` features of a pre-trained VGG16 model [11].

Like in [3], the weights are set to $w_r = 0.8$, $w_w = 0.4$, $w_p = 0.005$ and $w_s = 1$.

3. EXPERIMENTS

3.1. Training

To generate an intermediate light field frame, we choose a view-wise approach for the neural network. The reason is that finding enough light field videos to train a neural network is very challenging. Instead, having a view-wise approach enables us to use 2D videos for the training. So, in order to train our network, we use the MPI Sintel dataset [12].

We also need initial optical flows for these 2D frames. Since the purpose of the neural network is to refine the optical flows and generate corresponding visibility maps for optimal warping, we have to use estimated optical flows as inputs. Hence, we need to estimate optical flows with similar accuracy as the ones produced by the method presented in Sec. 2.1, based on the 4D consistent optical flows of [9]. Since this 4D approach cannot be applied to the 2D dataset, we use instead the PWC-Net [13] optical flow estimation network that is used for the initialization step of [9].

First, because we have the same architecture and a similar loss function as in [3], we initialize the weights of our network with those obtained by the training of [3] on the adobe240fps dataset. Then, in the training, for every clip of the MPI Sintel dataset, we use the odd frames as input frames and the even

frames as targets. During the training, we perform some data augmentation on the frames such as random cropping, horizontal flip or time inversion.

3.2. Evaluation

To test our method and compare it with other state-of-the-art methods, we use Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) between every view of the interpolated light field and those of the ground truth light field frame. While these metrics provide valuable insights on the quality of independently interpolated views, they fail to assess the angular consistency of the whole interpolated light field. So, we define a new metrics that we call Light Field Epipolar Consistency (LFEC).

3.2.1. Light Field Epipolar Consistency metric

Let $N_u \times N_v$ and $W \times H$ be the respective angular and spatial resolution of our light field. In the synthetic dataset that we use, ground truth disparity maps are provided, so we can back-warp every non-central view into the central view, taking into account the angular occlusions. We respectively denote G^t_{uv} and $\bar{G}^t$ the warped views and the view obtained when averaging every warped views. Then, for every pixel of the central view, we can compute a variance of every warped colors. This gives us a variance map σ^2:

$$\sigma^2 = \frac{1}{N_u N_v} \sum_{u,v}^{N_u,N_v} (G^t_{uv} - \bar{G}^t)^2 \tag{17}$$

The Light Field Epipolar Consistency LFEC of the synthesized light field is then computed similarly to a PSNR:

$$\text{LFEC} = 10 \log_{10} \left(\frac{d^2}{\overline{\sigma^2}} \right) \tag{18}$$

where d is the color range of the pixel values (for an 8-bit encoded light field, $d = 255$) and $\overline{\sigma^2}$ the mean of σ^2.

This metric measures the color consistency of every ray of the light field along the epipolar plane. If a ray has the wrong color but shares this color with the rest of rays along its epipolar plane, the metric will be high. Inversely, if none of the light field rays aligned on an epipolar plane has the same value, the metric will be low. Therefore, it enables us to measure the angular consistency of an interpolated light field.

3.2.2. PSNR, SSIM and LFEC measures

We compare our method with Super SloMo [3], Separable Adaptive Convolution [2] and Deep Voxel Flow [1], respectively denoted SSM, SAC and DVF on the dataset proposed by [8]. For each of the metrics, we separately compute them on the two scenes (Bamboo2 and Temple1) rendered with two methods (final and clean). The "clean" rendering has no lighting effect or motion blur while the "final" rendering is more photorealistic.

	Bamboo2		Temple1	
	clean	final	clean	final
DVF [1]	21.98	22.05	18.92	21.05
SAC [2]	**27.15**	**27.10**	**24.66**	27.53
SSM [3]	26.05	26.07	24.12	**27.90**
Ours	26.50	26.52	24.06	27.86

Table 1. PSNR of synthesized light field for all views

	Bamboo2		Temple1	
	clean	final	clean	final
DVF [1]	0.686	0.694	0.643	0.756
SAC [2]	**0.928**	**0.928**	**0.893**	0.932
SSM [3]	0.904	0.905	0.886	0.933
Ours	0.908	0.910	0.890	**0.935**

Table 2. SSIM of synthesized light field for all views

In terms of PSNR, our method ranks second on the Bamboo sequences and third on the Temple sequences. For these latter results, we achieve very comparable results to [3]. Our method gives a higher SSIM than [3] and [1] on every sequence and outperforms [2] on Temple1 - final. However, the PSNR and SSIM metrics are not ideal to assess the visual quality of the reconstructed views. If we look at visual comparisons for the four sequences (see Fig. 2), we can see that our method provides more natural results for objects with large motions which are effectively propagated at the expected intermediate position. On the other hand, [2] which gives the highest SSIM and PSNR, shows blurry results in such areas. As a result, the frame interpolated with [2] is visually more similar to the average of the two input frames (i.e. overlayed input in Fig. 2).

As for LFEC, our method systematically outperforms every other tested method, especially [2], which had overall better results in terms of PSNR and SSIM. On the Temple1 - clean sequence, we are even 1.5 dB ahead of [2]. If we look at the backwarp variance maps in Figure 3, our method gives lower variance on occluded areas than [1, 2, 3]. So, even though the method in [2] gives higher PSNR and SSIM on a few sequences, the angular consistency of the generated light fields is the lowest among every tested method. On the other hand, our method offers a good trade-off between angular and temporal consistency.

3.3. Multi-step prediction

We can use our method to interpolate multiple light fields between two consecutive ones. The final intermediate flows described in Subsection 2.1 and the interpolated light field (Eq. 9) can be computed for any value of $t \in [0, 1]$. To assess how our method performs on multi-step prediction, we use the dataset given in [7], which consists of light field videos of 8×8 views shot at only 3 fps. We generate 7 intermediate light fields between each original frame to have a final frame

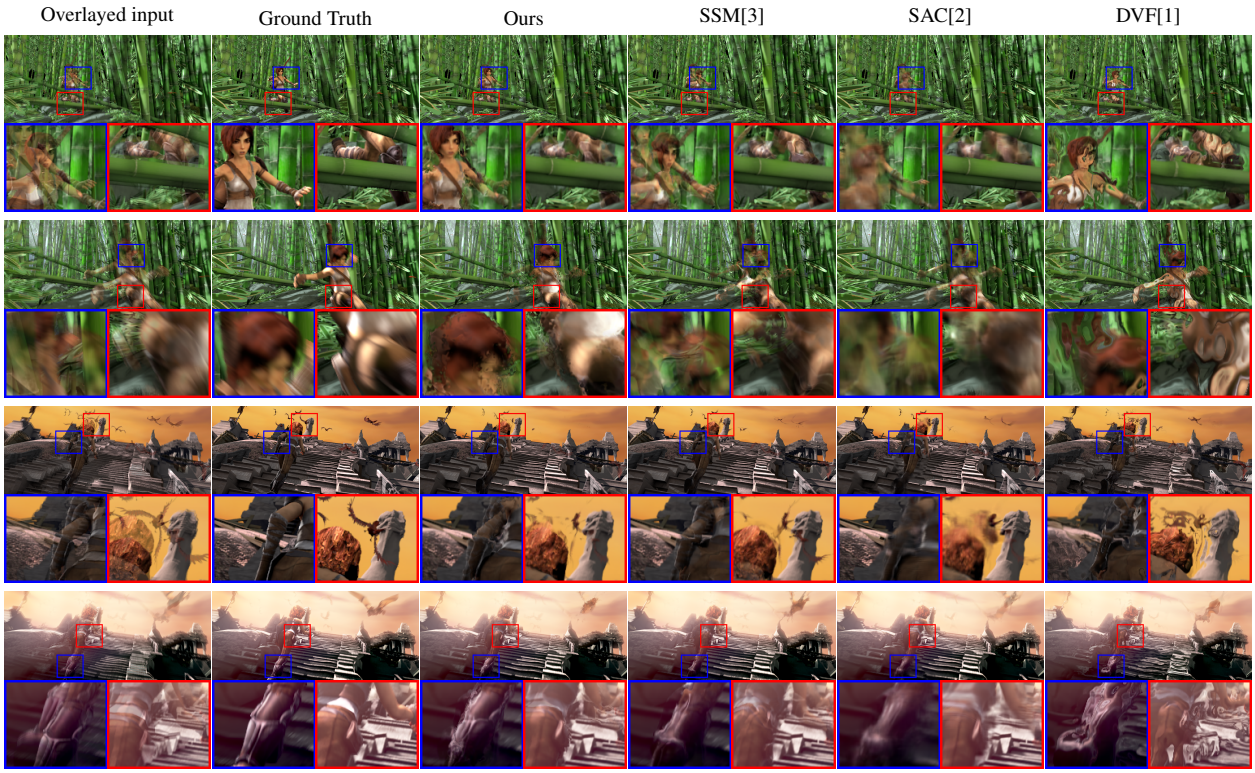

Fig. 2. Visual comparison of our method with [1, 2, 3] for the central view.

	Bamboo2		Temple1	
	clean	final	clean	final
DVF [1]	29.33	28.98	29.24	29.41
SAC [2]	28.89	28.99	28.28	29.15
SSM [3]	29.19	29.08	29.13	29.45
Ours	**29.51**	**29.48**	**29.80**	**29.71**

Table 3. LFEC of synthesized light field

rate of 24 fps. There is no ground truth available for the sequence, so we visually compare our results with those given by the previous methods [1, 2, 3]. For [1, 2], it is only possible to interpolate a frame at $t = 0.5$. So, for these methods, we recursively interpolate the intermediate frames in order to have 24 fps sequences. Some visual comparisons of the central frames and epipolar plane images (EPI) are shown on Figure 4 for the 3rd frame (among the 7 generated) and full sequences can be found on our project webpage[2]. Our method gives the sharpest results on views and EPIs. We can observe sharp epipolar lines on our EPIs unlike those generated by [2, 1]. Furthermore, we can notice that [3] produces curved and discontinuous lines on the EPIs, proving that our method is more angularly consistent than any other tested method.

[2]http://clim.inria.fr/research/LFVideoInterpolation

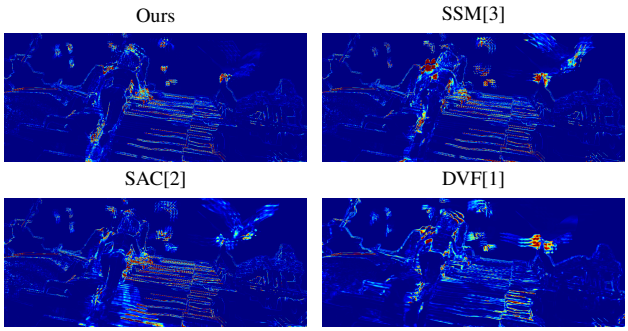

Fig. 3. Backwarp variance map σ^2 for each method.

4. CONCLUSION

In this paper, we proposed a method to interpolate an angularly-consistent intermediate light field frame from a pair of two consecutive light field frames. We divided the interpolation problem in two parts: motion estimation and view synthesis. To enforce the angular consistency of our method, we estimated an optical flow on the whole 4D light field and to enforce the temporal consistency, we independently synthesized the light field views using a convolutional neural network that we trained on a synthetic dataset. To estimate the angular consistency of the reconstructed light field, we proposed a new metrics called LFEC. We compared our method with state-of-the-art deep learning approaches and achieved comparable results in terms of PSNR and SSIM. Visual in-

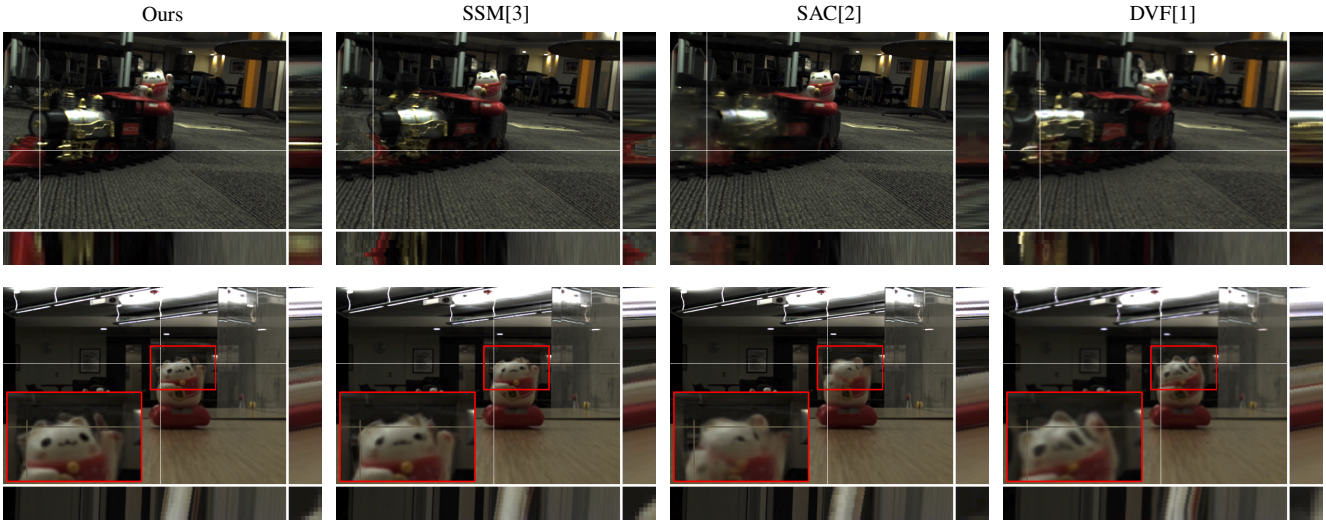

Fig. 4. Visual comparison of our method with [1, 2, 3] for the 3rd interpolated frame of the central view.

spection further indicates that our approach better keeps the temporal consistency in the case of large motions, where concurrent methods tend to produce blurred results. Furthermore, regarding the angular consistency, improved LFEC scores are obtained thanks to the use of 4D consistent optical flows, which was made possible by decoupling the optical flow estimation and the frame interpolation network.

Therefore, our results demonstrate the advantage of taking into account angular information for the temporal interpolation of video light fields. Future work in that direction would thus include a simultaneous processing of the light field views, not only for the optical flow estimation, but also for the frame interpolation.

5. REFERENCES

[1] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *IEEE International Conf. on Computer Vision*, 2017, pp. 4463–4471.

[2] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *IEEE International Conf. on Computer Vision*, 2017, pp. 261–270.

[3] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 9000–9008.

[4] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[5] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, "MEMC-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2019.

[6] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *IEEE International Conf. on Computer Vision*, 2019.

[7] T.-C. Wang, J.-Y. Zhu, N. K. Kalantari, A. A. Efros, and R. Ramamoorthi, "Light field video capture using a learning-based hybrid imaging system," *ACM Trans. on Graphics*, vol. 36, no. 4, pp. 133, 2017.

[8] P. David, M. Le Pendu, and C. Guillemot, "Sparse to dense scene flow estimation from light fields," in *IEEE International Conf. on Image Processing*, Sep. 2019, pp. 3736–3740.

[9] P. David, M. Le Pendu, and C. Guillemot, "Scene flow estimation from sparse light fields using a local 4D affine model," *IEEE Trans. on Computational Imaging*, 2020.

[10] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conf. on Learning Representations*, 2015.

[12] 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)*, A. Fitzgibbon et al. (Eds.), Ed. Oct. 2012, Part IV, LNCS 7577, pp. 611–625, Springer-Verlag.

[13] 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*, June 2018.