Learning Optical Flow from Continuous Spike Streams (Supplementary Material)

Rui Zhao[1,2] **Ruiqin Xiong**[1,2*] **Jing Zhao**[3] **Zhaofei Yu**[1,2,4] **Xiaopeng Fan**[5] **Tiejun Huang**[1,2]

[1]National Engineering Research Center of Visual Technology (NERCVT), Peking University
[2]Institute of Digital Media, School of Computer Science, Peking University
[3]National Computer Network Emergency Response Technical Team
[4]Institute for Artificial Intelligence, Peking University
[5]School of Computer Science and Technology, Harbin Institute of Technology
ruizhao@stu.pku.edu.cn, {rqxiong, yuzf12, tjhuang}@pku.edu.cn,
zhaojing@cert.org.cn, fxp@hit.edu.cn

Contents

A Real Scenes with Spike and Flow Datasets **1**

A.1 Details of the Scenes 1

A.2 Statistics of Flow Fields 2

B Details of the Method **3**

B.1 Details of the Network 3

B.2 Computational Pipeline of Spike2Flow 4

C Additional Experimental Results **6**

C.1 Additional Comparative Results with Original SCFlow 6

C.2 Visual Results on RSSF, PHM, and Real Data 6

A Real Scenes with Spike and Flow Datasets

A.1 Details of the Scenes

We propose the real scenes with spike and flow (RSSF) dataset for training and evaluating spike-based optical flow. The dataset is generated based on data in Slow Flow [2] dataset that is captured by high-speed cameras. There are 31 scenes for training and 10 scenes for testing. The scenes for training are shown in Fig. 8, and the scenes for testing are shown in Fig. 9. The detailed statistics of each scene in RSSF are shown in Tab. 5. There are three kinds of training scenes with different resolutions and numbers of spike frames. There are a total of 9.6k+ flow fields and 193k+ spike frames in the training dataset. As for the 11 scenes in the evaluation dataset, we select the first 200 flow fields to balance the weights of different scenes. To standardize the evaluation data, we use center clipped to make the width of each spike frame and flow field to be 1024 and make the height

*Corresponding author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

of images whose height exceeds 768 to be 768. The totals of flow fields and spike frames are 2.2k and 44.22k respectively. Noted that the "Number of Flow Fields" only counts the flow in $dt = 20$ case. The number of flow fields in $dt = 40$ and $dt = 60$ cases is similar with that in $dt = 20$ case.

Figure 8: Scenes for generating the training set of RSSF. The indexes of the scenes are on the left. The faces wholly facing the camera are masked.

Figure 9: Scenes for generating the evaluation set of RSSF. The indexes of the scenes are on the left. The faces wholly facing the camera are masked.

A.2 Statistics of Flow Fields

To analyze the RSSF dataset further, we gather several statistics of the training and evaluation set of RSSF respectively. As shown in Fig 10. We gather the motion, speed, direction and derivatives of the flow fields $\mathbf{w}(x, y) = (u(x, y), v(x, y))$ in $dt = 20$ case. The formulations are as follows:

$$\text{speed of flow:} \quad s(x,y) = \sqrt{[u(x,y)]^2 + [v(x,y)]^2} \tag{9}$$

$$\text{direction of flow:} \quad \theta(x,y) = \arctan\left(\frac{v(x,y)}{u(x,y)}\right) \tag{10}$$

$$\text{spatial derivatives of flow:} \quad \frac{\partial\{u,v\}}{\partial\{x,y\}}(x,y) = \frac{\partial u}{\partial x}(x,y),\ \frac{\partial v}{\partial x}(x,y),\ \frac{\partial u}{\partial y}(x,y),\ \frac{\partial v}{\partial y}(x,y) \tag{11}$$

$$\text{temporal derivatives of flow:} \quad \frac{\partial\{u,v\}}{\partial t}(x,y) = u(x,y) - u_{\text{pre}}(x,y),\ v(x,y) - v_{\text{pre}}(x,y) \tag{12}$$

The u_{pre} and v_{pre} denote the flow between the previous pair of spike frames in $dt = 20$ case.

Table 5: Statistics of the training and evaluation set of real scenes with spike and flow (RSSF) dataset. The numbers of flow fields and spike frames that are not in the "sum" line are statistics for each scene. The data in the "sum" line are totals for all the scenes. Noted that the "Number of Flow Fields" only counts the flow fields in $dt = 20$ case.

	Scene Indexes	Number of Flow Fields	Resolution	Number of Spike Frames	Number of Scenes
Training Dataset	1, 2, 3, 14, 15, 16, 17, 18, 19, 25, 26, 27, 28, 29, 30	239	1280 x 1024	4800	15
	4, 5, 6, 7, 8, 9, 10, 11, 12, 13	340	1280 x 720	6820	10
	20, 21, 22, 23, 24	531	1024 x 576	10640	5
	Sum	9640	–	193400	30
Evaluation Dataset	1, 6, 7, 8, 10, 11	200	1024 x 768	4020	6
	2, 3, 4, 5	200	1024 x 720	4020	4
	9	200	1024 x 576	4020	1
	Sum	2200	–	44220	11

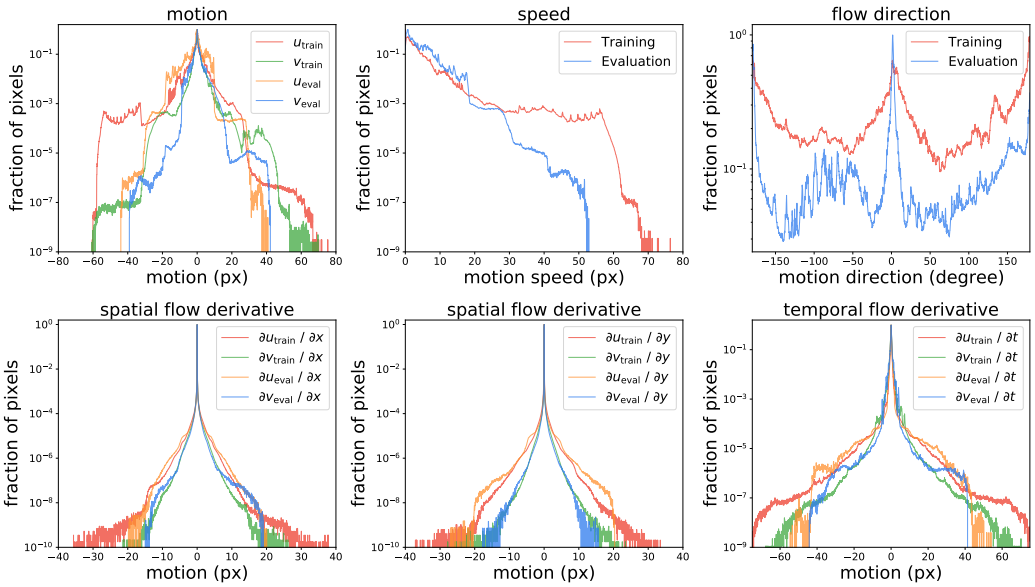

Figure 10: Statistics of the flow fields in $dt = 20$ case. The vertical axis is the ratio of the number of pixels in each bin to the total number of pixels, and the vertical axis is in the logarithmic domain.

B Details of the Method

B.1 Details of the Network

Details of the encoder. The structures of the feature encoder and context encoder are the same. Both of the feature extraction modules of them are composed of a series of convolution layers. The structure is shown in Fig. 11. The difference between the feature extraction modules of the feature encoder (FE) and the context encoder (CE) is that the output feature dimension of the feature extraction module of FE is 216, while that of the CE is 256. The feature extraction for getting the primary feature F_{p} can be formulated as:

$$F_{\mathrm{p},i} = \mathcal{F}[D_i] \tag{13}$$

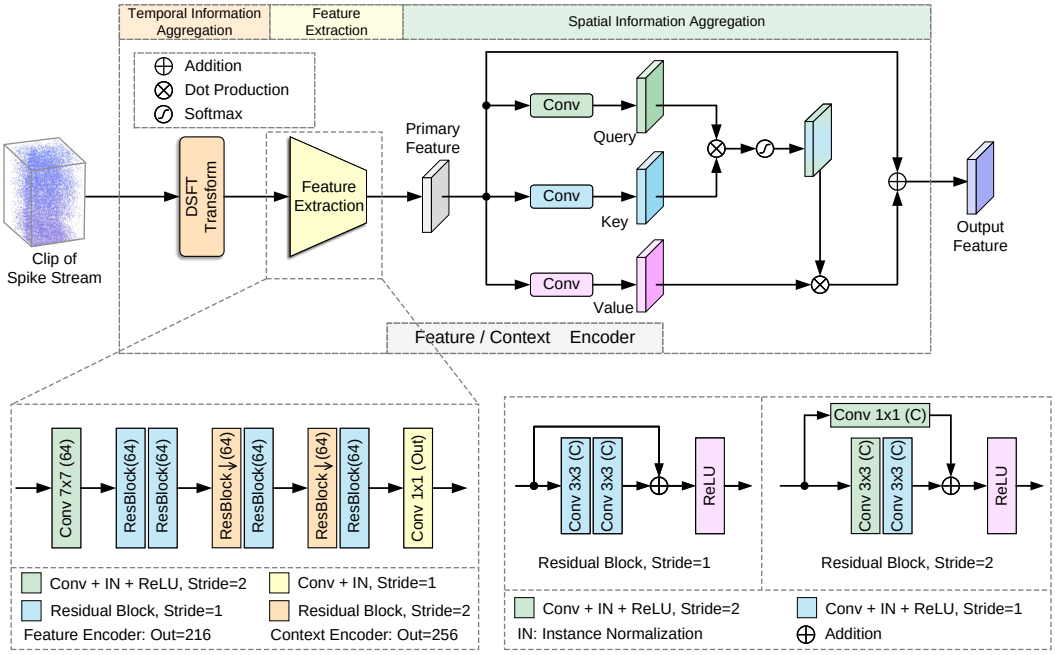

Figure 11: Illustration of the detailed structure of the feature extraction module in the feature and context encoder.

where D_i is the DSFT transform of the i-th spike sub-stream. The kernel size of the convolution layers for getting the query, key, and value feature in the spatial information aggregation module is 1×1, and the output dimension is the same as the primary feature.

Details of the recurrent decoder. Suppose that the start moment of the flow is t_{s}, and the sampling cycle of the flow is T. The motion features extracted from the local correlations $\{\mathbf{L}_i\}_{i=1}^{N}$ are $\{M_i\}_{i=1}^{N}$. The updating procedure of the two ConvGRUs in the recurrent decoder can be formulated as:

$$x_k = \mathrm{Cat}\left(F^{\mathrm{C}}, M_1, \ldots, M_N\right) \tag{14}$$
$$z_k = \mathrm{Sigmoid}\left(\mathrm{Conv}\left(\mathrm{Cat}\left(h_{k-1}, x_k\right), W_{z,k}\right)\right) \tag{15}$$
$$r_k = \mathrm{Sigmoid}\left(\mathrm{Conv}\left(\mathrm{Cat}\left(h_{k-1}, x_k\right), W_{r,k}\right)\right) \tag{16}$$
$$q_k = \mathrm{Tanh}\left(\mathrm{Conv}\left(\mathrm{Cat}\left(r_k \odot h_{k-1}, x_k\right), W_{q,k}\right)\right) \tag{17}$$
$$h_k = (1 - z_k) \odot h_{k-1} + z_k \odot q_k \tag{18}$$

where Cat denotes the concatenation operation in the channel dimension. k is the iteration index of the ConvGRUs, and $\odot$ denotes the element-wise multiplication. Different flow heads estimate the residual of flow fields $\{\Delta\tilde{\mathbf{w}}_{k,i}\}_{i=1}^{N}$ in $1/8$ downsampling resolution. Then the flow is updated and upsampled through convex combination $\mathcal{U}$ according to the mask $\mathcal{M}_k$ estimated from the hidden state h_k. Noted that the flow heads are composed of "Conv3x3(256) – ReLU – Conv3x3(2)", and the mask estimator is composed of "Conv3x3(196) – ReLU – Conv1x1(64x9)". The details about the convex combination can be found in the supplementary material of [5].

B.2 Computational Pipeline of Spike2Flow

The computational pipeline of Spike2Flow is shown in Alg. 1. The network output the flow estimated in each iteration $\{\{\mathbf{w}_{k,i}\}_{i=1}^{N}\}_{k=1}^{N_{\mathrm{it}}}$ in the training procedure, and output the flow in the final iteration $\{\mathbf{w}_{N_{\mathrm{it}},i}\}_{i=1}^{N}$ in the evaluating procedure.

Algorithm 1 Framework of Spike2Flow

Input:

The input spike streams $S = \{S(\mathbf{x}, t) \mid \mathbf{x} \in \Omega,\ t \in \mathbb{N},\ t \leq n\}$. Start moment t_s. Sampling cycle for flow fields T. The number of correlations N. The number of input spike frames N_ISF. Pyramid level of correlation L. Iteration steps of recurrent decoder N_it. Looking-up radius for correlation r.

Output:

A series of flow fields $\{\mathbf{w}(\mathbf{x}, t_\mathrm{s} + iT \mid t_\mathrm{s})\}_{i=1}^{N}$

`/* Construct correlations of spike sub-streams */`

1: Compute the half-window length of the sub-streams $t_\mathrm{h} \leftarrow (N_\mathrm{ISF} - 1)/2$;

2: Clip the spike stream S to be sub-streams $\{S_i\}_{i=0}^{N} \leftarrow \left\{\{S(t_\mathrm{s} + iT + j)\}_{j=-t_\mathrm{h}}^{t_\mathrm{h}}\right\}_{i=0}^{N}$;

3: **for** each i **in** $0, \ldots, N$ **do**

4: Transform the sub-stream to DSFT domain: $D_i \leftarrow \mathcal{D}[S_i]$ using Eq. (3) in the body;

5: Extract the primary feature: $F_{\mathrm{p},i}$ through convolutional networks;

6: Get feature for matching: $F_i^\mathrm{M} \leftarrow \mathcal{A}[F_{\mathrm{p},i}]$ using Eq. (4) in the body;

7: **if** $i = 0$ **then**

8: Get context feature F^C and the initialization of hidden state h_0 using the same process with lines 4-6 through context encoder

9: **else**

10: Construct all-pairs correlation $\mathbf{C}_i$ using Eq. (5) in the body;

11: **end if**

12: **end for**

`/* Recurrently decode the correlations */`

13: Initialize the flow fields in low resolution $\{\tilde{\mathbf{w}}_{0,i} = \tilde{\mathbf{w}}_0(iT + t_\mathrm{s} \mid t_\mathrm{s})\}_{i=1}^{N} \leftarrow \mathbf{0}$;

14: **for** k **in** $1, \ldots, N_\mathrm{it}$ **do**

15: **for** each i **in** $1, \ldots, N$ **do**

16: Compute looking-up grid $\mathcal{N}_i$ by currently estimated flow fields $\{\mathbf{w}_{k-1}(iT + t_\mathrm{s} \mid t_\mathrm{s})\}_{i=1}^{N}$ using Eq. (6) in the body;

17: Look up from the correlation $\mathbf{C}_i$ by $\mathcal{N}_i$ to get current local correlation $\mathbf{L}_i$;

18: **end for**

19: Update the hidden state through the ConvGRUs using Eq. (14) – (18) to get h_k;

20: Estimate the residual flow in low resolution $\{\Delta\tilde{\mathbf{w}}_{k,i}\}_{i=1}^{N}$.

21: Update the flow in low resolution: $\{\tilde{\mathbf{w}}_{k,i}\}_{i=1}^{N} \leftarrow \{\tilde{\mathbf{w}}_{k-1,i} + \Delta\tilde{\mathbf{w}}_{k,i}\}_{i=1}^{N}$;

22: **if** training the network **then**

23: Estimate the mask and upsample the flow: $\{\mathbf{w}_{k,i}\}_{i=1}^{N} \leftarrow \{\mathcal{U}(\tilde{\mathbf{w}}_{k,i}, \mathcal{M}_k)\}_{i=1}^{N}$;

24: **else** {evaluating the network}

25: **if** $k = N_\mathrm{it}$ **then**

26: Estimate the mask and upsample the flow: $\{\mathbf{w}_{N_\mathrm{it},i}\}_{i=1}^{N} \leftarrow \{\mathcal{U}(\tilde{\mathbf{w}}_{N_\mathrm{it},i}, \mathcal{M}_{N_\mathrm{it}})\}_{i=1}^{N}$;

27: **end if**

28: **end if**

29: **end for**

30: **return** $\{\mathbf{w}_{N_\mathrm{it},i}\}_{i=1}^{N}$ **if** evaluating else $\{\{\mathbf{w}_{k,i}\}_{i=1}^{N}\}_{k=1}^{N_\mathrm{it}}$;

C Additional Experimental Results

C.1 Additional Comparative Results with Original SCFlow

We compare the performance of SCFlow [1] trained on the original SPIFT dataset in $dt = 10$ and $dt = 20$ cases and SCFlow retrained on our proposed RSSF dataset. The performance on PHM and RSSF is shown in Tab. 6 and Tab. 7 respectively. The models trained on SPIFT are official models of SCFlow, which converge well on the SPIFT dataset. The results in Tab. 6 demonstrate that our proposed RSSF dataset can make SCFlow model have the best performance, especially in $dt = 20$ case. The results in Tab. 7 demonstrate that the SCFlow model trained on RSSF performs much better than the models trained on the SPIFT dataset, which *supports our statement* that the models trained on dataset synthesized based on graphics models cannot generalize on real data well.

Table 6: Quantitative comparative results of SCFlow [1] trained on different datasets on PHM dataset. The training datasets include SPIFT in $dt = 10$ and $dt = 20$ cases, and RSSF. The best results for each group are bolded. AEPE: average end-point error. PO%: the percentage of outliers.

Training Dataset	Evaluating Dataset	$dt = 10$		$dt = 20$	
		AEPE	PO%	AEPE	PO%
SPIFT – $dt = 10$	PHM	1.077	37.12	2.347	46.52
SPIFT – $dt = 20$	PHM	1.096	40.69	2.167	47.65
RSSF – Training	PHM	**1.027**	**34.54**	**1.775**	**38.57**

Table 7: Quantitative comparative results of SCFlow [1] trained on different datasets on the evaluation set of RSSF. The training datasets include SPIFT in $dt = 10$ and $dt = 20$ cases, and RSSF. The best results for each group are bolded. AEPE: average end-point error. PO%: the percentage of outliers.

Training Dataset	Evaluating Dataset	$dt = 20$		$dt = 40$		$dt = 60$	
		AEPE	PO%	AEPE	PO%	AEPE	PO%
SPIFT – $dt = 10$	RSSF – Evaluation	1.037	34.25	4.183	58.04	8.356	67.97
SPIFT – $dt = 20$	RSSF – Evaluation	0.847	31.98	3.099	48.59	7.236	62.22
RSSF – Training	RSSF – Evaluation	**0.389**	**14.00**	**0.668**	**19.00**	**1.264**	**23.40**

C.2 Visual Results on RSSF, PHM, and Real Data

The comparative experiments in the body are based on three kinds of data: the evaluation set of RSSF, PHM, and real data captured by spike cameras. We supplement visual results on each kind of data in this part. The comparative methods include RAFT [5], GMA [3], and SCV [4] based on spikes and average image along the temporal axis respectively. SCFlow based on spikes as input is taken into consideration. The additional visual results on RSSF, PHM, and real data are shown in Fig. 12, Fig. 13, and Fig. 14 respectively.

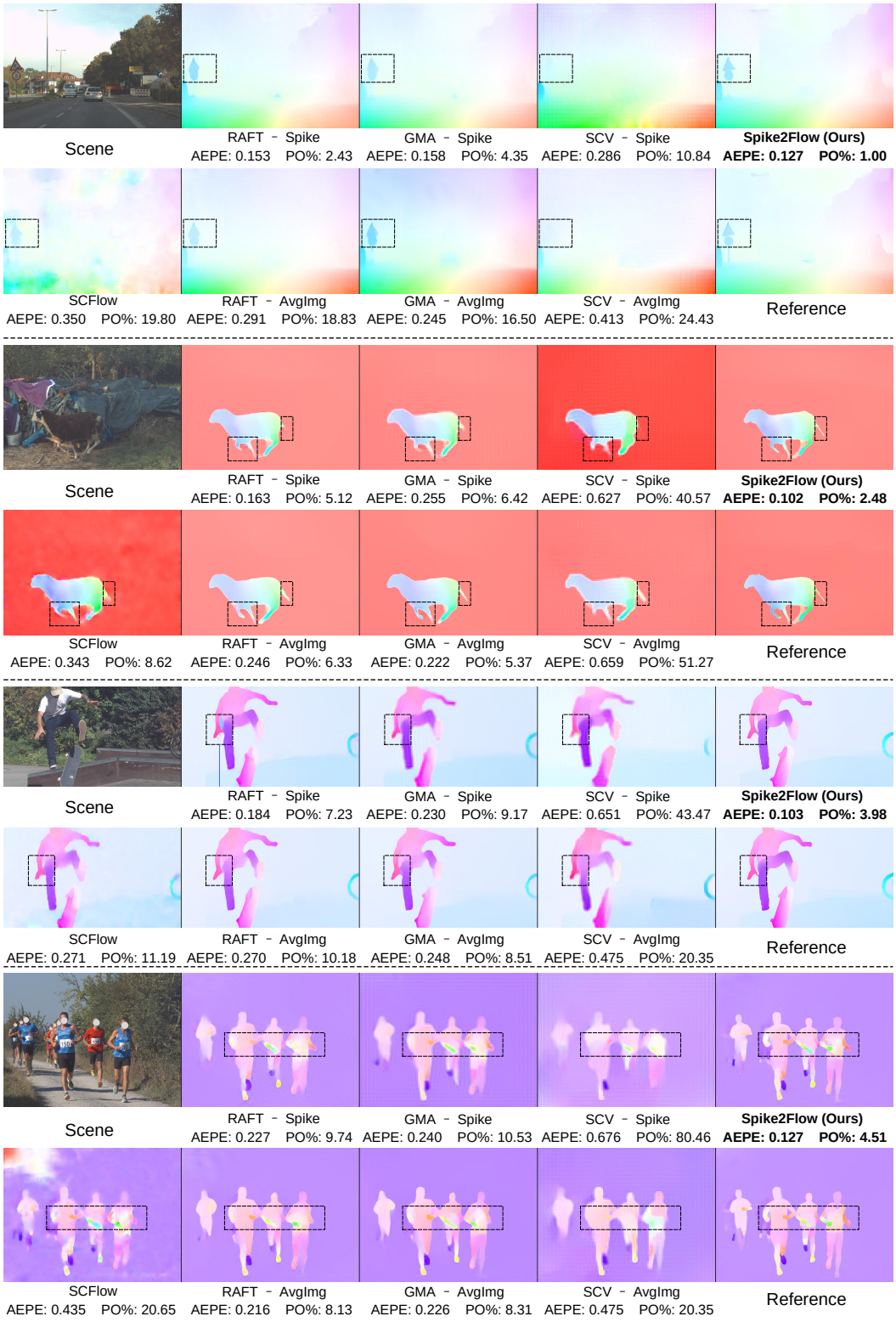

Figure 12: Visual results on the evaluation set of RSSF in $dt = 20$ case. All the model is retrained on the training set of RSSF. The model and performance are labeled at the bottom of each visual result. AEPE: average end-point error. PO%: the percentage of outliers.

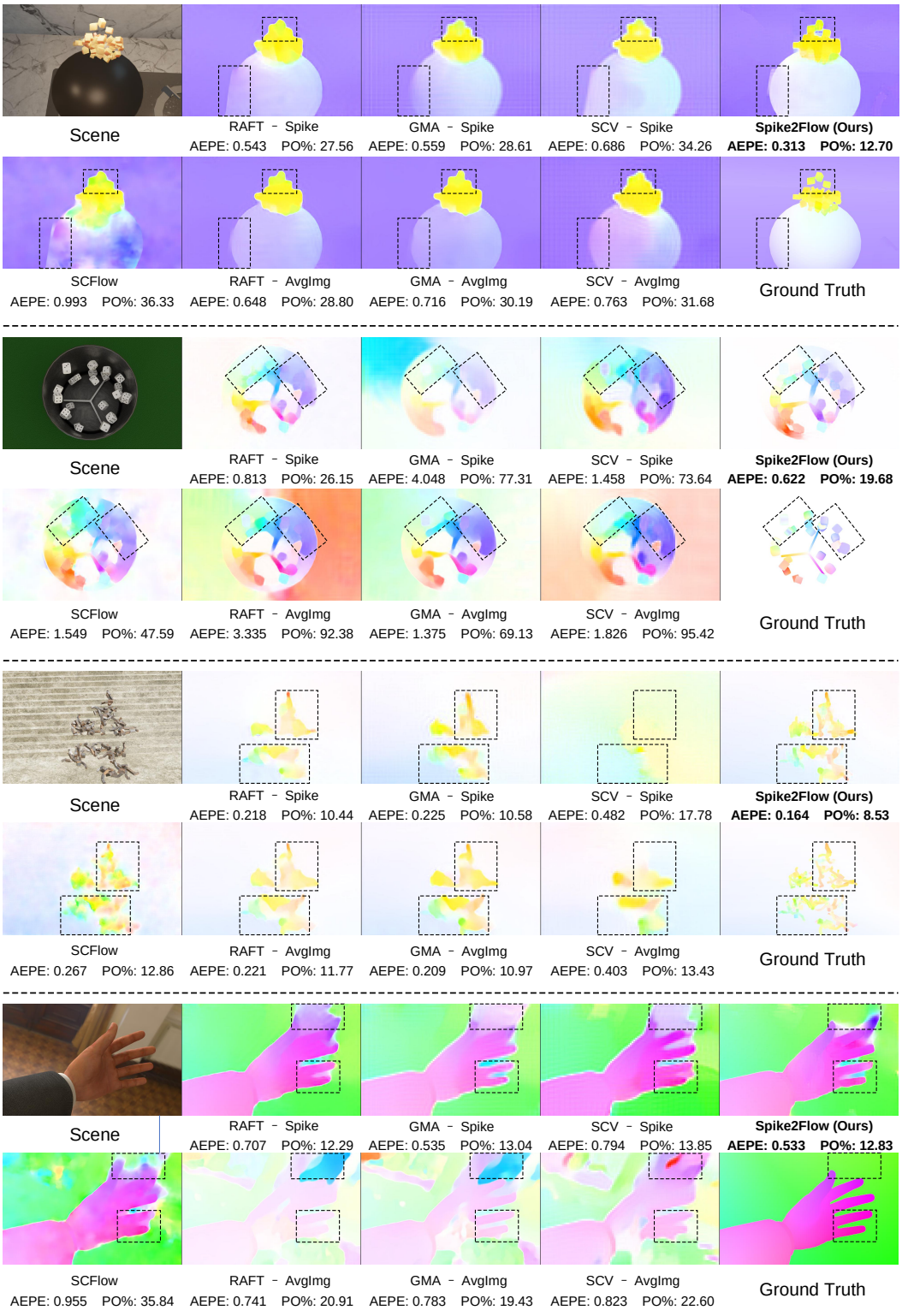

Figure 13: Visual results on the PHM in $dt = 10$ case. All the model is retrained on the training set of RSSF. The model and performance are labeled at the bottom of each visual result. AEPE: average end-point error. PO%: the percentage of outliers.

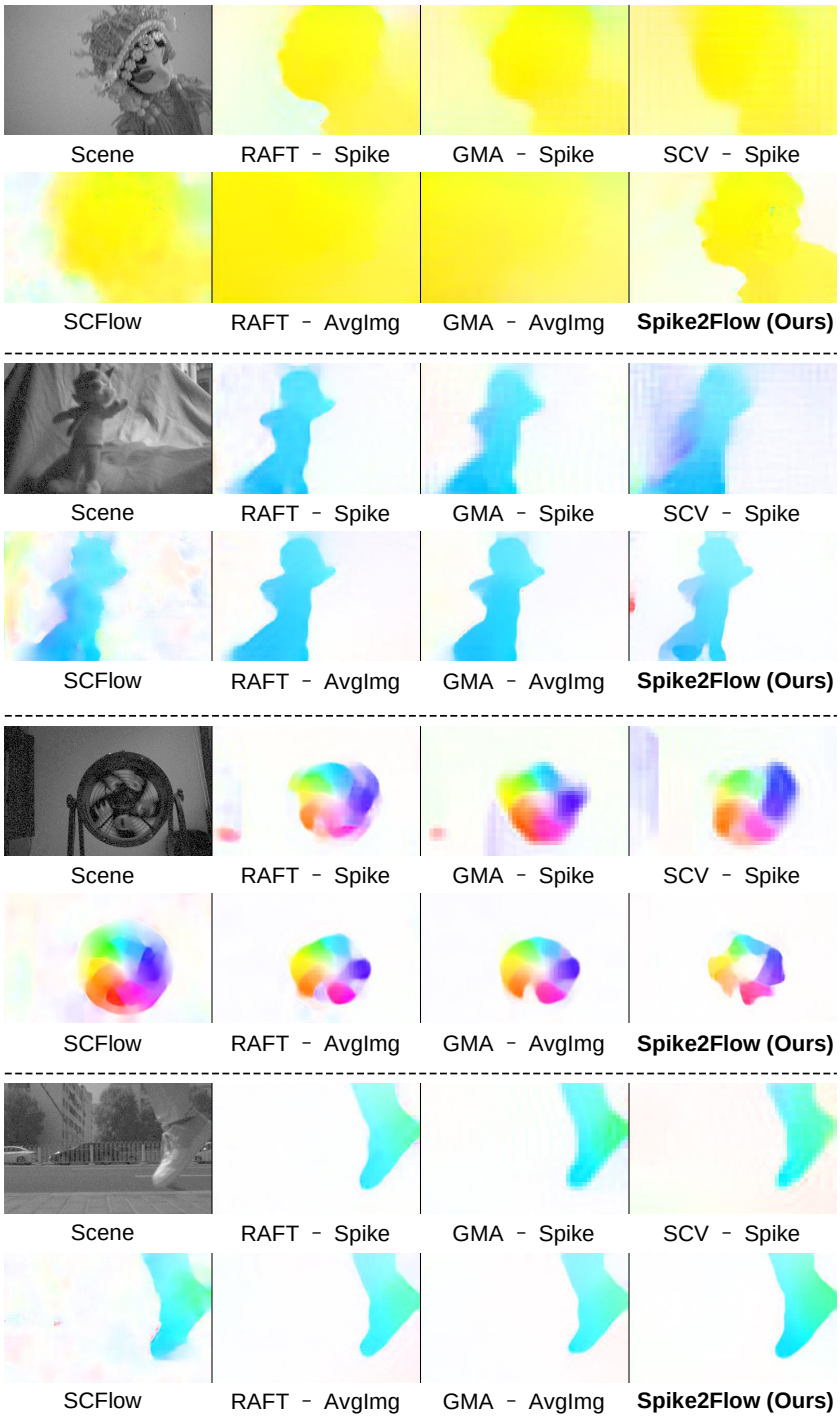

Figure 14: Visual results on real data captured by spike cameras in $dt = 20$ case. All the model is retrained on the training set of RSSF. The model and performance are labeled at the bottom of each visual result. AEPE: average end-point error. PO%: the percentage of outliers.

References

[1] Liwen Hu, Rui Zhao, Ziluo Ding, Lei Ma, Boxin Shi, Ruiqin Xiong, and Tiejun Huang. Optical flow estimation for spiking camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[2] Joel Janai, Fatma Guney, Jonas Wulff, Michael J Black, and Andreas Geiger. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3597–3607, 2017.

[3] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9772–9781, 2021.

[4] Shihao Jiang, Yao Lu, Hongdong Li, and Richard Hartley. Learning optical flow from a few matches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16592–16600, 2021.

[5] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419, 2020.