# Temporal Event Stereo via Joint Learning with Stereoscopic Flow
Supplementary Material

Hoonhee Cho*, Jae-Young Kang*, and Kuk-Jin Yoon

Korea Advanced Institute of Science and Technology
{gnsgnsgml, mickeykang, kjyoon}@kaist.ac.kr

In this supplementary material, we provide more details that are not included in the paper due to space limitations. This includes the conceptual comparisons (Sec. A), implementation details (Sec. B), and more experimental results and analyses (Sec. C), respectively.

## A Conceptual Comparisons

We provide a conceptual comparison to highlight the differences from existing methods. Se-CFF [7] fuses multi-density event stacks within a single timeframe. Other methods [6, 10] use recurrent architecture to aggregate feature-level temporal information. As shown in Table A1, conceptually, our method explicitly retrieves temporal information based on flow rather than relying on an implicit feature-level approach. This approach offers the advantage of reusing the computed flow in both feature and cost volume. Furthermore, a key difference lies in our cost volume aggregation, which is essential and task-dependent for stereo matching.

**Table A1:** The summary of the conceptual differences.

| Method | Target Temporal Information | | Temporal Retreieval Method |
|---|---|---|---|
| | Feature | Cost Volume | |
| DDES [8], EITNet [1] | ✗ | ✗ | ✗ |
| Se-CFF [7] | △ | ✗ | CNN-based |
| EIS-E [6], DTC [10] | ✓ | ✗ | RNN-based |
| **Ours** | ✓ | ✓ | Flow-based |

* Equal contribution.

# B Implementation Details

## B.1 Measurement of Inference Time

As mentioned in the official implementations of [3, 5], measuring the inference time of PyTorch models requires avoiding the use of 'time.time()' due to the asynchronous nature of GPU operations. Instead, two main steps must be followed to accurately measure the inference time:
**1. GPU Warm-up.** Before measuring inference time, we should warm up the GPU by running some dummy examples through the network. This step is crucial because a GPU can exist in various power states, and without warm-up, the device might not operate at its full capacity during the initial runs, leading to inaccurate timing measurements.
**2. GPU and CPU Synchronization.** To ensure accurate timing measurements, it is essential to use 'torch.cuda.synchronize()' before and after the inference calls. This function synchronizes the CPU and GPU, ensuring that all GPU tasks are completed before the time is measured. This step overcomes the potential inaccuracies caused by the asynchronous execution of GPU operations.

For accurate and fair measurement of FPS, we followed the mentioned methods. Additionally, we performed a GPU warm-up using 100 dummy examples to ensure the GPU was fully operational for the actual measurements.

## B.2 Network Architecture

The network structure and parameters for the MVSEC dataset are detailed in Table B1. We designed the network to be efficient by significantly reducing the channel dimension of features while allowing for temporal aggregation. For DSEC, we increased the channel dimensions of the multi-scale in Table B1 from 12, 24, and 36 to 32, 64, and 128, respectively.

## B.3 More Implementation Details

In the MVSEC dataset, event streams are sliced every 50ms and processed to voxels [12] of bin 5. For flow loss weights, $\lambda_t = 0.1$ and $\lambda_c = 1 \cdot 10^{-5}$ are used. For stereo loss, $\lambda_0 = 0.5$, $\lambda_1 = 0.7$, and $\lambda_f = 1.0$. The maximum disparity of cost volume is set to 48. In the train phase, four serial event voxel pairs are sequentially fed into the model, and only the last stereo pair is used for loss calculation, while previous frames are only inferenced for temporal information. Training is done in an end-to-end manner for 60 epochs and a batch size of 2. The learning rate is set to 0.0008 using Adam optimizer [4].

Faster ego-motion in DSEC tends to trigger more events per unit of time. Therefore, each 50ms event stream is processed to bin 15 voxel. Also, flow loss weights are reduced to $\lambda_t = 0.01$ and $\lambda_c = 1 \cdot 10^{-8}$ for stable training. The maximum disparity is set to 192. The model is trained for 100 epochs with batch size 4.

During training, we grouped 4 sequential event voxel pairs into one clip for a training purpose, and similarly, for testing, we also used clips consisting of 4 voxels each. Additionally, to make the temporal disparity consistency more stable, we adopted the approach of pseudo ground-truth (GT), a method previously utilized in stereo research [9]. For this purpose, we first trained a single event stereo network without any temporal aggregation and performed inference of disparity maps for pseudo-GT. In areas where sparse GT was available, we used the GT, and in areas without GT, we filled in with pseudo-GT. The densified GT disparity map is only used for flow loss calculation, not for the stereo loss.

In the main paper, our explanation focuses on cost volume warping and temporal disparity consistency loss. Due to the epipolar constraint, there is no need to consider $\Delta y^R$ in cost volume warping and temporal disparity consistency, so we omitted $\Delta y^R$ for a better explanation of our core ideas. In actual implementation, $\Delta y^R$ is also estimated with other components by the single stereoscopic flow network and only used for feature warping, as mentioned in the main paper.

### B.4 Details About the Ablation Study in Table 4 of the Main Paper

In the main paper, we present Table 4 for the ablation study of stereoscopic flow. Table 4 aims to validate stereoscopic flow from two perspectives.

First, it validates the impact of sharing stereo features by comparing outcomes between sharing and not sharing stereo features as inputs to the flow network. Instead of sharing stereo features, we employ the comprehensive optical flow network, EV-Flow [11], with event voxel input. In the table, "ours" signifies feature sharing with a stereo network, and "EV-Flow" denotes a flow network independent of stereo features.

Second, we validate the relationship between the left and the right flows in stereo matching. In other words, we tested the assumption of the 'hard' epipolar constraint that matching points share the same vertical flow. In a hard constraint setup, which is our baseline, both left and right event information are fed into a single stereoscopic flow network, estimating 4-dimensional flow for stereo. Also, cost volume warping and temporal disparity consistency (TDC) loss are calculated based on the assumption that matching pixels share the same vertical flow. In contrast, in soft epipolar constraint experiments, the left and the right flows are estimated independently and calculated separately. Two twin flow networks are used for each left and right flow. $\{\underset{t\to t-1}{\Delta x^L}, \underset{t\to t-1}{\Delta y^L}\}$ are estimated from using only left event while $\{\underset{t\to t-1}{\Delta x^R}, \underset{t\to t-1}{\Delta y^R}\}$ are predicted only from right events. Moreover, cost volume and TDC loss calculation are modified to accommodate different vertical flows. As a result, computational complexity is increased for an extra flow network. However, the redundant degree of freedom on vertical flows negatively affected the stereo matching performance.

# C Additional Experimental Results and Analyses

## C.1 Hyper-parameter Analysis of Flow Loss (Eq. (8))

The flow loss (Eq. (8)) consists of the temporal disparity consistency loss, proposed for jointly learning stereoscopic flow with stereo, and a minor contrast loss [12] serving as an auxiliary loss. Table C1 provides results based on the coefficient of the loss in the flow loss. Contrast loss becomes unstable when jointly trained with the stereo network, and setting the $\lambda_c$ beyond a certain value leads to a significant decrease in the performance of the stereo network. On the other hand, our TDC loss is robust even when trained together with the stereo network, maintaining results within a certain range regardless of the scale of the weight becoming larger or smaller.

## C.2 Streaming Experiment

The main MVSEC experiments are only conducted with limited past information; 4 stereo frames at the test phase. However, in the real world, streams of events are continuously fed into the model. Therefore, we conducted additional experiments to validate the real-world application, and to verify the long-term information propagation. We inferred our model, which is trained with 4 sequential frames, with different numbers of stereo frames: 2, 4, 8, and streaming. In the streaming experiment, all event voxels are sequentially processed and the random test sets are evaluated. The results are provided in Table C2. Even if the model is trained only for 4 frames setting, it can retrieve information from further past frames to enhance current disparity prediction.

## C.3 Stereoscopic Flow

Stereoscopic flow is an auxiliary output to facilitate temporal information from the past. Even if the quality and quantity of flow results are not our main interest, visualization of the intermediate outputs is useful for understanding the model behavior. The stereoscopic flow network generates 4-dimensional flow, $\{\underset{t\to t-1}{\Delta x^L}, \underset{t\to t-1}{\Delta x^R}, \underset{t\to t-1}{\Delta y^L}, \underset{t\to t-1}{\Delta y^R}\}$, and we visualize the left camera flow $\{\underset{t\to t-1}{\Delta x^L}, \underset{t\to t-1}{\Delta y^L}\}$ among them in the Fig. C1. As the network estimates flow in 1/4 resolution of the input voxel grid, we applied bilinear upsampling for visualization.

**Table C1:** The result according to hyper-paramter in Eq. (8). $\lambda_t$ and $\lambda_c$ refer to the weight of temporal disparity consistency (TDC) loss and contrast loss, respectively.

| $\lambda_t \setminus \lambda_c$ | $10^{-6}$ | | $10^{-5}$ | | $10^{-4}$ | | $10^{-3}$ | |
|---|---|---|---|---|---|---|---|---|
| | Mean Depth | 1PA | Mean Depth | 1PA | Mean Depth | 1PA | Mean Depth | 1PA |
| 0.01 | 13.8 | 92.6 | 13.4 | 92.3 | 13.5 | 92.4 | 15.5 | 90.1 |
| 0.1 | 13.1 | 92.7 | 13.0 | 92.9 | 14.4 | 91.9 | 14.6 | 91.3 |
| 1.0 | 13.3 | 92.8 | 13.5 | 92.5 | 14.3 | 91.7 | 15.7 | 90.5 |

**Table C2:** Streaming experiment with different train/test frame

| train \ test | 2 frames | | 4 frames (Base) | | 8 frames | | Streaming | |
|---|---|---|---|---|---|---|---|---|
| | Mean Depth | 1PA | Mean Depth | 1PA | Mean Depth | 1PA | Mean Depth | 1PA |
| 4 frames | 17.2 (4.2 ↑) | 89.9 (3.0 ↓) | 13.0 | 92.9 | 12.8 (0.2 ↓) | 93.0 (0.1 ↑) | 12.8 (0.2 ↓) | 93.0 (0.1 ↑) |

**Table C3:** Experimental results according to the bin size of the voxel grid.

| Bin Size | Mean Depth ↓ | Mean disp ↓ | 1PA ↑ |
|---|---|---|---|
| 1 | 13.7 | 0.47 | 92.2 |
| 5 | **13.0** | **0.46** | **92.9** |
| 10 | 13.7 | 0.48 | 92.4 |

### C.4 Experiments with Different Voxel Dimensions

Table C3 presents the results of our method with different bin sizes of the voxel grid. With the smallest bin size of 1, optimal performance is not achieved due to information loss, as much temporal information is aggregated into a single channel. However, as the bin size increases to 5, the performance improves because the discretely separated bins allow for the efficient utilization of temporal information. Nonetheless, when the bin size becomes excessively large, the events become spatially sparse, and the convolutional layers are unable to fully exploit this temporal information, resulting in a performance drop.

### C.5 Qualitative Ablation Study of Temporal Aggregation

Fig. C2 shows comparisons between our temporal event stereo network and a single stereo network, where other components are kept constant while modules related to temporal aggregation, specifically feature warping and cost volume warping, are removed. Temporal stereo, in contrast to single stereo, leverages preceding information continuously for compensation, enabling more accurate detail reconstruction of scenes. It also shows resilience in difficult conditions, including noisy night environments and instances of fewer events.

**Table B1:** Structure details of the networks.

| Name | Layer setting | Output Dimension |
|---|---|---|
| Input | Voxel Grid | $H \times W \times 5$ |
| **Feature Extractor** | | |
| Conv0 | $[3 \times 3, 12] \times 3$, stride=2 | $H/2 \times W/2 \times 12$ |
| Conv1 | $\begin{bmatrix} 3 \times 3, 12 \\ 3 \times 3, 12 \end{bmatrix} \times 2$ | $H/2 \times W/2 \times 12$ |
| Conv2 | $\begin{bmatrix} 3 \times 3, 24 \\ 3 \times 3, 24 \end{bmatrix} \times 3$, stride=2 | $H/4 \times W/4 \times 24$ |
| Conv3 | $\begin{bmatrix} 3 \times 3, 36 \\ 3 \times 3, 36 \end{bmatrix} \times 2$ | $H/4 \times W/4 \times 36$ |
| Conv4 | $\begin{bmatrix} 3 \times 3, 36 \\ 3 \times 3, 36 \end{bmatrix} \times 2$, dilation=2 | $H/4 \times W/4 \times 36$ |
| Avg1 | $16 \times 16$ avg. pooling<br>$3 \times 3, 12$<br>bilinear interpolate | $H/4 \times W/4 \times 12$ |
| Avg2 | $8 \times 8$ avg. pooling<br>$3 \times 3, 12$<br>bilinear interpolate | $H/4 \times W/4 \times 12$ |
| Fusion | Concat(Conv2, Conv4, Avg1, Avg2)<br>$3 \times 3, 36$<br>$3 \times 3, 12$ | $H/4 \times W/4 \times 12$ |
| Feature Warping | Spatial Warping, $\mathcal{W}_s$ (Eq.(1)) & Concat<br>$3 \times 3, 12$ | $H/4 \times W/4 \times 12$ |
| **Stereoscopic Flow** | | |
| Flow0 | Concatenate Left and Right<br>$[3 \times 3, 24] \times 8$ | $H/4 \times W/4 \times 12$ |
| Flow1 | Add Flow0 & $3 \times 3, 3$ | $H/4 \times W/4 \times 4$ |
| **Initial Cost Volume** | | |
| Cost Volume | Concatenate Left and Shifted Right | $D_{max}/4 \times H/4 \times W/4 \times 24$ |
| 3D-Conv0 | $[3 \times 3 \times 3, 12] \times 2$ | $D_{max}/4 \times H/4 \times W/4 \times 12$ |
| 3D-Conv1 | $[3 \times 3 \times 3, 12] \times 2$ | $D_{max}/4 \times H/4 \times W/4 \times 12$ |
| 3D-Conv2 | Add 3D-Conv0 & 3D-Conv1 | $D_{max}/4 \times H/4 \times W/4 \times 12$ |
| **Initial Hourglass** | | |
| 3D-Stack0-0 | $7 \times 7 \times 7, 24$, stride=3 | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack0-1 | $3 \times 3 \times 3, 24$ | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack0-2 | $[3 \times 3 \times 3, 24] \times 2$ | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack0-3 | $3 \times 3 \times 3, 24$, add 3D-Stack0-1 | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack0-4 | deconv $7 \times 7 \times 7, 24$, stride=3<br>add 3D-Conv2 | $D_{max}/4 \times H/4 \times W/4 \times 12$ |
| 3D-Output0 | $3 \times 3 \times 3, 12$<br>$3 \times 3 \times 3, 1$ | $D_{max}/4 \times H/4 \times W/4 \times 1$ |
| Output0 | bilinear interpolate & regression | $H \times W$ |
| 3D-Stack1-0 | $7 \times 7 \times 7, 24$, stride=3 | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack1-1 | $3 \times 3 \times 3, 24$, add 3D-Stack0-3 | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack1-2 | $[3 \times 3 \times 3, 24] \times 2$ | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack1-3 | $3 \times 3 \times 3, 24$, add 3D-Stack0-1 | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack1-4 | deconv $7 \times 7 \times 7, 24$, stride=3<br>add 3D-Conv2 | $D_{max}/4 \times H/4 \times W/4 \times 12$ |
| 3D-Output1 | $3 \times 3 \times 3, 12$<br>$3 \times 3 \times 3, 1$ | $D_{max}/4 \times H/4 \times W/4 \times 1$ |
| Output1 | bilinear interpolate & regression | $H \times W$ |
| **Cost Volume Refinement & Aggregation** | | |
| Warped Cost Volume | 3D Warping, $\mathcal{W}_c$ (Eq.(3)) to Prev Final Cost | $D_{max}/4 \times H/4 \times W/4 \times 12$ |
| Entropy Filter | Generate the entropy filter based on 3D-Output1 | $H/4 \times W/4 \times 1$ |
| Weight0 & Weight1 | Concatenate the entropy filters<br>$[3 \times 3, 12] \times 3$<br>$3 \times 3, 2$ & Sigmoid | $H/4 \times W/4 \times 2$ |
| Fused Cost Volume | Weight0 $\times$ 3D-Stack1-4 + Weight1 $\times$ Warped Cost | $D_{max}/4 \times H/4 \times W/4 \times 12$ |
| **Final Output** | | |
| 3D-Stack2-0 | $7 \times 7 \times 7, 24$, stride=3 | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack2-1 | $[3 \times 3 \times 3, 24] \times 2$ | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack2-2 | $[3 \times 3 \times 3, 24] \times 3$, add 3D-Stack3-1 | $D_{max}/12 \times H/12 \times W/12 \times 24$ |
| 3D-Stack2-3 | deconv $7 \times 7 \times 7, 24$, stride=3<br>add 3D-Conv2 | $D_{max}/4 \times H/4 \times W/4 \times 12$ |
| 3D-Output2 | $3 \times 3 \times 3, 12$<br>$3 \times 3 \times 3, 1$ | $D_{max}/4 \times H/4 \times W/4 \times 1$ |
| Output2 | bilinear interpolate & regression | $H \times W$ |

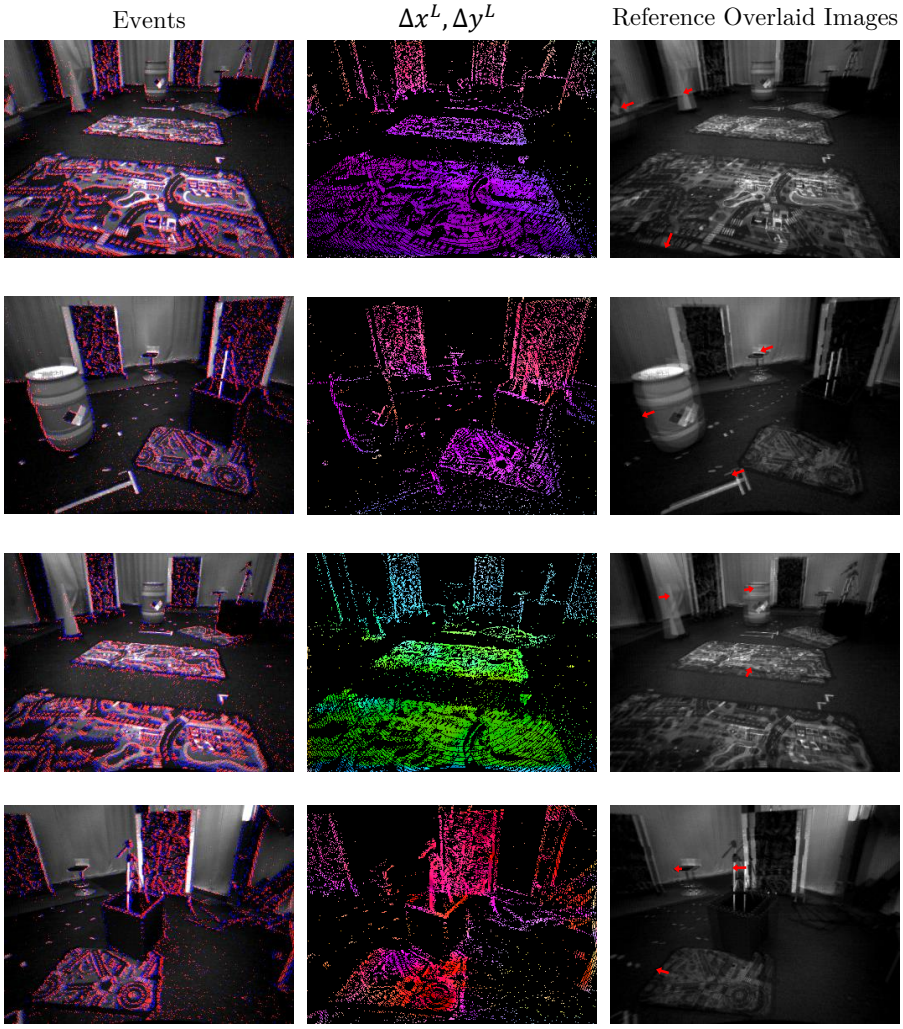


**Fig. C1:** Visualization of the components $\Delta x^L$ and $\Delta y^L$ of the left camera in stereoscopic flow. For the visualization, we set the color wheel identical to that of [2]. To make the direction of motion clear, we presented a visualization by overlaying the current image with the one from three frames earlier, indicating the direction of forward movement using red arrows. **Note that our stereoscopic flow is a backward flow**.

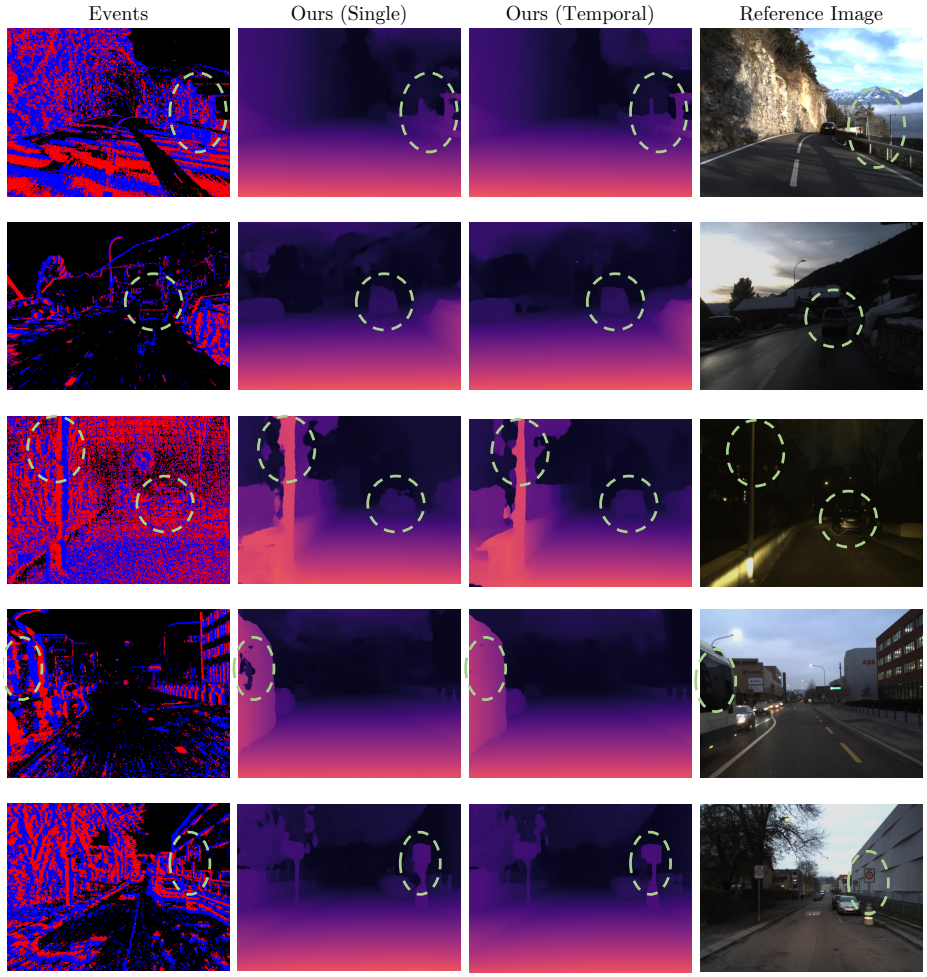


**Fig. C2:** Qualitative comparison between single and temporal event stereos on DSEC test datasets.

## References


1. Ahmed, S.H., Jang, H.W., Uddin, S.N., Jung, Y.J.: Deep event stereo leveraged by event-to-image translation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 882–890 (2021) 1
2. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International journal of computer vision **92**, 1–31 (2011) 7
3. Cho, S.J., Ji, S.W., Hong, J.P., Jung, S.W., Ko, S.J.: Rethinking coarse-to-fine approach in single image deblurring. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4641–4650 (2021) 2
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 2
5. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering (2023) 2
6. Mostafavi, M., Yoon, K.J., Choi, J.: Event-intensity stereo: Estimating depth by the best of both worlds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4258–4267 (2021) 1
7. Nam, Y., Mostafavi, M., Yoon, K.J., Choi, J.: Stereo depth from events cameras: Concentrate and focus on the future. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Patter Recognition (2022) 1
8. Tulyakov, S., Fleuret, F., Kiefel, M., Gehler, P., Hirsch, M.: Learning an event sequence embedding for dense event-based deep stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1527–1537 (2019) 1
9. Xu, H., Zhang, J.: Aanet: Adaptive aggregation network for efficient stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1959–1968 (2020) 3
10. Zhang, K., Che, K., Zhang, J., Cheng, J., Zhang, Z., Guo, Q., Leng, L.: Discrete time convolution for fast event-based stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8676–8686 (2022) 1
11. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Ev-flownet: Self-supervised optical flow estimation for event-based cameras. arXiv preprint arXiv:1802.06898 (2018) 3
12. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 989–997 (2019) 2, 4