FALCON: Fast Adaptive Lightweight Computation of Intensities and Events for Depth Estimation

Supplementary Material

1. Further investigations on δt

We aimed to conduct a principled approach to determine an optimal way of stacking, which we described in Sections 3.1, and 4.3.3 to 4.3.4 of the main paper.

The key insight of our work is that closely aligning standard deviations (σ) of the two modalities would improve our model's ability to effectively learn event representations robust to motion blur, while reducing the modality gap [7] between events and intensity images. Furthermore, our approach would have to be unlike the ones taken by previous works (such as SE-CFF [11] and EI-Stereo [10]), where they *directly* utilize the events (through short and long stacking) together, without any analysis of the intrinsic properties of the modalities. Specifically, their model is tasked with a difficult task due to: (i) the modality gap, and (ii) some scenes containing highly event-rich frames, whereas other scenes containing frames with few events.

As a result, what we proposed in the main paper, that led to state-of-the-art or competitive performances on DSEC and MVSEC, was to divide the events by an optimal δt value such that the input event frames and intensity images have similar standard deviations σ.

One can argue, instead of relying on a static *fixed* value of δt, another approach could be to *normalize* the data by scaling by a constant such that each of the inputs have $\sigma = 1$, regardless of the δt. This can be done by dividing the input event frames by their respective σ values. We note that this method is similar to the preconditioning scaling approach of [5]. This approach has the added benefit that we can use arbitrary values for δt to split the event stream, so long as the resulting event frames are normalized to have $\sigma = 1$. Furthermore, the lower variability between the event frames would attenuate the effect of motion blur as well.

δt	σ-normalized	MDE (cm, ↓)	MDisE (pix, ↓)	1PA (%, ↑)
50ms	✓	14.2	0.48	91.9
25ms	✓	13.9	0.47	92.4
10ms	✓	13.9	0.46	92.8
2ms	✓	13.3	0.46	92.6
50ms	✗	12.4	0.42	93.4
25ms	✗	12.6	0.42	**93.7**
10ms (Proposed)	✗	**11.7**	**0.41**	**93.7**
2ms	✗	13.9	0.48	91.7

Table 1. **Study on FALCON's Depth Estimation Performance when Grouping Events by Different δt Values.** On the MVSEC dataset, 10ms leads to the best performance as its σ follows intensity images (as stated Section 4.3.3 and Table 6 of the main paper).

To gauge the performance of this normalization approach, we compare its performance to our proposed approach of dividing the events by an optimal δt value, and report the results in Table 1. We attach the qualitative results of various δt values in Figure 1.

From Table 1, we make the observation that the normalization approach *does not lead to improved results*. This is not surprising, however, since some event frames would still be highly event-rich while some other frames would have very few events. These results support the claim that our proposed method of dividing the event stream by an optimal δt value is robust to motion blur and artifacts.

Another interesting observation we can make is that the normalization approach leads to similar levels of performances (although they fail to surpass the state-of-the-art results). This indicates that while normalization contributes to robustness against temporal (δt) variations and motion blur, it does not inherently improve the baseline capability.

2. Improved Learning Rate Schedule

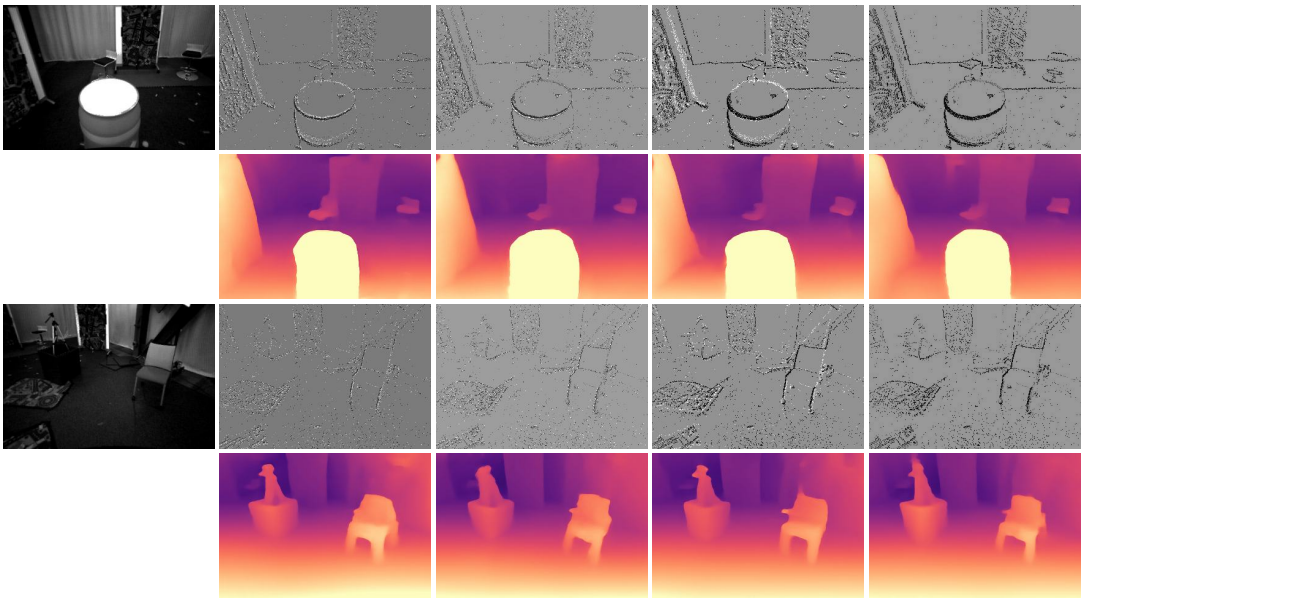
The "Cosine Annealing with Warmup Restarts" scheduler [1] provides *warm restarts* to the learning rate in a *cyclically-restarting manner*, similar to Figure 2b. It is shown that such a cyclic scheduler with warm restarts makes the model more capable of escaping any local minima, and decreases the chances of overfitting [8].

SE-CFF [11] implement this scheduler with a very small learning rate. However, interestingly, they set the restart cycle to be equal to the number of training epochs ($= 100$). This means that SE-CFF effectively does not provide any sort of cyclic restarts to the learning rate, virtually making their implementation not utilize any benefits provided the scheduler. This makes SE-CFF more prone to overfitting. Moreover, as the learning rate decreases after every epoch, the model would be incapable of escaping any local minima either. This poses a significant problem with SE-CFF's overall learning strategy.

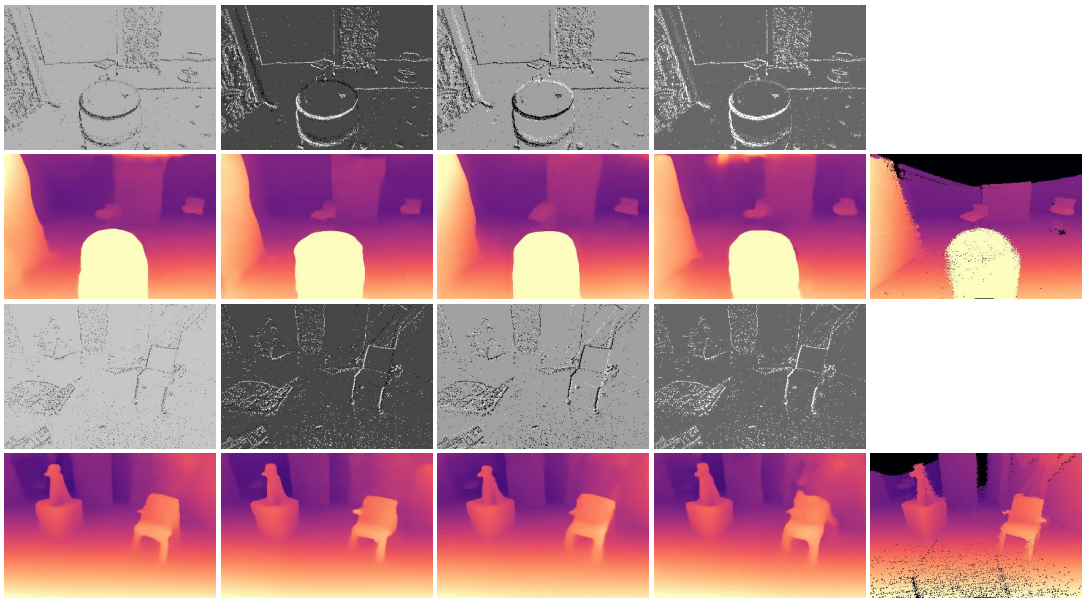
Our key realization of SE-CFF's miscalculated learning strategy motivated us to propose an improved method. We believed that their strategy goes against the original intention of the scheduler, as well.

Therefore, we hypothesized that increasing the learning rate and providing the cyclic restarts slightly more often would benefit the model by better escaping any *local minima*. Unlike SE-CFF, we set the restart cycle frequency to

[1] We follow the implementation of CosineAnnealineWarmupRestarts.



(a) Grayscale (b) 2ms (Proposed) (c) 10ms (Proposed) (d) 25ms (Proposed) (e) 50ms (Proposed)

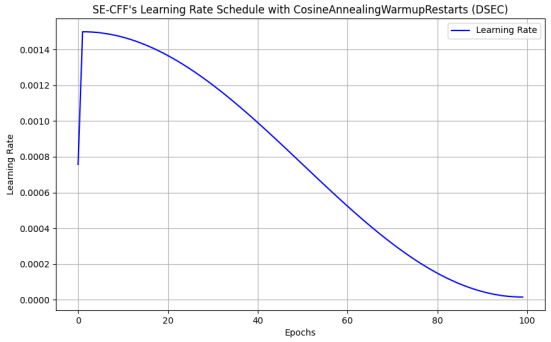


(f) 2ms w/ norm. (g) 10ms w/ norm. (h) 25ms w/ norm. (i) 50ms w/ norm. (j) GT

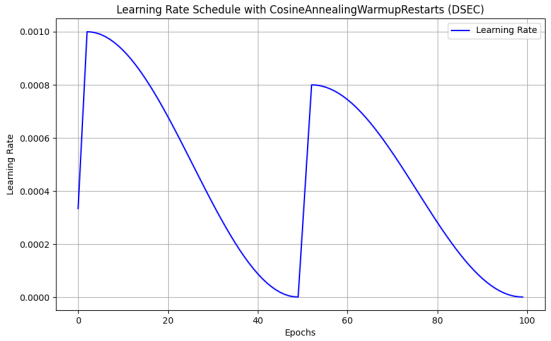
Figure 1. **Investigating the Effects of δt.** We observe that depth estimates using our approach is qualitatively better than the results from the normalization approach, supporting our claim that our approach of splitting the event stream by an optimal δt is more robust against motion blur and artifacts (Section 1). (a) is Grayscale image of scenes; (b), (c), (d) and (e) are the corresponding RepNet Stack (**odd rows**) with FALCON's disparity predictions (**even rows**) for different time-intervals δt, obtained using our proposed approach. (f), (g), (h) and (i) are results with normalized σ. (j) is the ground truth disparity map of each scene. Conducted on the MVSEC dataset [13].

50 epochs (from 100), and increase the learning rate to be in the range of $[5e-6, 1e-3]$ (from $[5e-8, 5e-4]$). We graph these learning rate schedules in Figure 2. The graph of our schedule for MVSEC is plotted in Figure 2c, for which we found that our model performs empirically better with more frequent restarts.

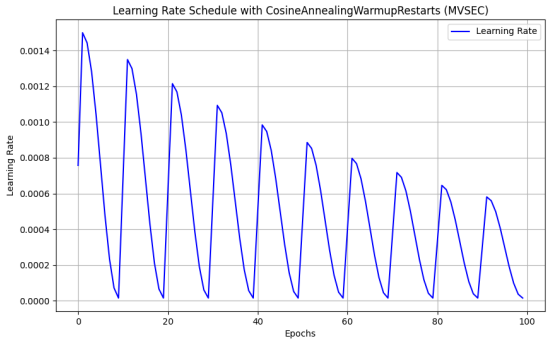
We compare our schedule to that of SE-CFF's, and report the experimental results in Table 2. As expected, the training progression and results revealed that the models may occasionally get stuck in local minima. However, our model is able to effectively escape these minima, a result of the frequent learning rate restarts employed in our approach.



(a) SE-CFF [11]'s Learning Rate Schedule across 100 epochs on DSEC. SE-CFF is more prone to getting stuck in local minima.



(b) Our Learning Rate Schedule on DSEC (Sec. 2 and 4.1). As evidenced by the results in Table 2, FALCON is less prone to getting stuck in local minima due to cyclic restarts.



(c) Our Learning Rate Schedule across epochs on MVSEC (Sec. 2 and 4.2).

Figure 2. **Comparing SE-CFF's and Our Proposed Learning Rate Schedules.** Our scheduler implements a schedule with frequent learning rate restarts to optimize the model's convergence, which we describe in detail in Sections 2, 4.1 and 4.2. (a) is SE-CFF's schedule whereas (b) is our schedule on DSEC; and (c) is our schedule on MVSEC. (We could not compare SE-CFF's schedule on MVSEC as they do not provide the training details.) Additionally, we report the performance gains achieved by implementing our schedule (a) for DSEC in Table 2.

3. Objective Function Details

The ground truth disparity map, $y \in \mathbb{R}^{H \times W}$, is originally captured by LiDAR [4, 13] and is estimated as the disparity map between the left and right maps.

Given the multi-modal inputs to our stereo matching network, X^L and X^R, we obtain the K-layered pyramid output, f. Let $n \in \{1, 2, ..., K\}$ be the indices of f's layers, where K is the last layer. Then, f_n is the estimated disparity of the n^{th} layer. Prior to calculating the loss, we bi-linearly interpolate f_n to y's size, such that $f_n \in \mathbb{R}^{H \times W}$.

Therefore, the objective becomes to minimize the weighted sum of the Smooth L_1 losses at each layer of f w.r.t. to the ground truth, y:

$$\min \sum_{n=1}^{K} w_n L\left(y, f_n\right) \tag{1}$$

where w_n is the n^{th} layer's assigned weight. We utilize the Smooth L_1 loss, as it is less sensitive to outliers and noise compared to L_2 loss [1].

4. Training Settings

4.1. DSEC Settings

DSEC [4] offers a comprehensive dataset, including event camera data of 480×640 resolution, high-resolution 1080×1440 RGB images, and 480×640 resolution LiDAR disparity maps as ground truth. For optimization, we chose the Adam [6] optimizer. The initial learning rate is set at $1e-3$, with beta values of 0.9 and 0.999, and a weight decay of $1e-4$. To enhance the training process, we employed the cosine annealing with warmup restarts [9], as mentioned in Section 2. Additionally, to address the challenge of depth perception, we set the maximum disparity to 192. Random cropping is used to crop the images to 432×576 resolution, after which they are padded to 480×640 resolution, in terms of height and width, respectively. Furthermore, for the Disparity Pyramid, we set K to 5, similar to as in StereoDRNet [2]. Moreover, note that we perform experiments only until $\delta t = 2$ms, as $\delta t = 1$ms experiments required much more intensive (over 80 GB VRAM) GPU resources.

4.2. MVSEC Settings

MVSEC [13] includes event data and grayscale images of 260×346 resolution, generated using the DAVIS sensor. We use the Adam optimizer with the same configuration as that for DSEC mentioned above, except that we set the initial learning rate to $1.5e-3$. Similarly, we employ cosine annealing with warm restarts to schedule the learning rate. We set max disparity to 37 and we randomly crop to resolution 228×312 on the data. We set the pyramid levels to K to 4 unlike DSEC which was 5, as MVSEC's lower resolution led to reducing the number of layers.

Method	Our Scheduler?	Restart Cycle	Max LR	Min LR	MAE (↓)	1PE (↓)	2PE (↓)	RMSE (↓)	FPS (↑)	FLOPs (↓)
SE-CFF [11]	✗	100	5e-4	5e-8	0.364	4.844	0.840	0.818	9.3	3.94 T
FALCON	✗	100	5e-4	5e-8	0.362	4.777	0.852	0.839	**32.2**	**2.02 T**
FALCON	✓	50	1e-3	1e-6	**0.356**	**4.717**	**0.814**	**0.815**	**32.2**	**2.02 T**

Table 2. **Improving Scheduler for better convergence.** Implemented with the Cosine Annealing with Warmup Restarts Schedule. While we already exceed the benchmarks set by [11] on MAE, 1PE, FPS, and FLOPs, implementing the proposed scheduler further enhances our performance (on DSEC; E + I modality).

5. FALCON's Architecture Details

The overall details of FALCON's model structure (with input/output dimensions) are presented in Tables 3 and 4 for the DSEC dataset. Table 3 contains architecture details for RepNet, our event representation network, while Table 4 contains the architecture details for our Stereo Matching Network. We summarize the changes made to the Stereo Matching Network below.

5.1. Stereo Matching Network Details

Feature Extractor. Extracts features from the 4-channel input. This module produces a list of features with heights of $H/3$, $H/6$, and $H/12$. While the height-dimension of each of the layers of the pyramid progressively becomes smaller, the channel-dimension progressively grows larger.

Cost Volume Constructor. This module constructs the *cost volume pyramid* by feature correlation and includes convolving the features. The output size is similar to that of the Feature Extractor.

Adaptive Aggregation Network. This module is based on *Deformable Convolutional Layers* (DC layers). The reason for using these layers is due to the advantageous adaptive nature of DC layers [3]. DC layers are able to overcome the implicit dependence on the neighboring pixels that is present in traditional convolutional layers. DC Layers are ideal for event-based tasks because of their sparse nature. We make this layer much shallower (from $6 \rightarrow 3$), hence shrinking the size of the module. The output of this module is the *convolved* cost volume pyramid, without any changes in dimensions. More information about this can be obtained by referring to the AANet paper [12].

Disparity Estimation Pyramid. This module calculates the disparity estimates for each of the 3 layers in the convolved cost volume pyramid. The output continues to be of the same size.

Stereo Disparity Refinement Module. The final module of the stereo matching network consists of two upsampling layers which each refine the lower-resolution disparity maps into an intermediate resolution, and then into the original required resolution of $H \times W$. These two refined maps are appended back to the original 3-layered pyramid to create a new 5-*layered disparity pyramid*. More information about this can be obtained by reading the StereoDRNet [2]. We explain the training process for each dataset in detail in Sections 4.1 and 4.2. We will release the source code and weights to prevent any confusions while accelerating future advancements.

6. MVSEC Qualitative Results

The qualitative results for MVSEC are depicted in Figure 3. We can observe that FALCON predicts more accurate depth maps compared to SE-CFF [11], while also generating event representations that suffer from less motion blur. The objects in FALCON's event representation, RepNet Stacks are crisp and sharp.

7. Companion Video

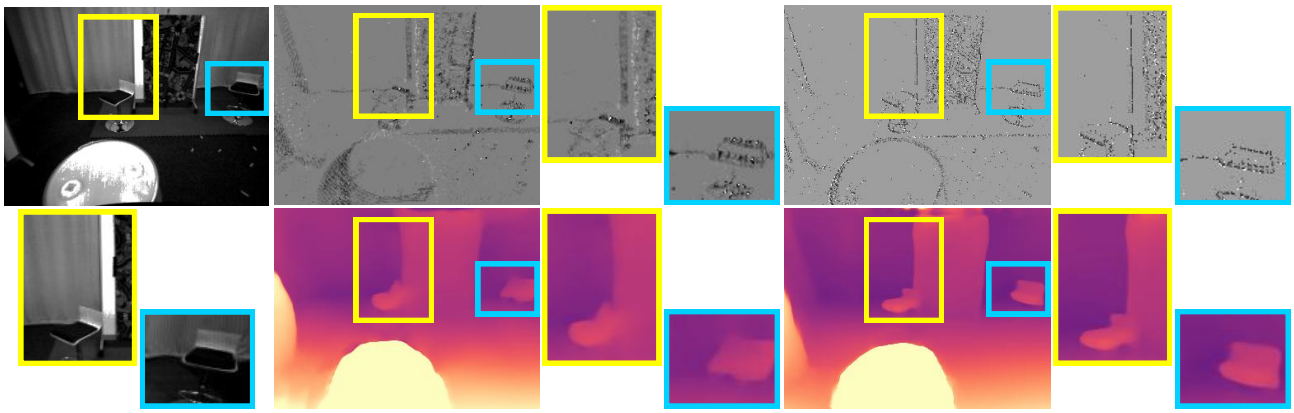
We invite interested reader to further watch our companion video, showcasing the datsets, RepNet Stack creation in action, and further qualitative results. It further includes a driving scenario comparing FALCON and SE-CFF [11] on DSEC. Note that in the video, we had to reproduce SE-CFF so the FPS rates displayed in the video, and it may slightly differ from those reported in the paper without changing the overall conclusions.

Table 3. RepNet: Event Feature Extractor Architecture Details for DSEC [4].

Layer Name	Type	Kernel	Stride	Out Channels	Activation Function
Input (Event Stack)	VoxelGrid	–	–	25	–
Conv-1	1×1 Conv $\rightarrow$ BatchNorm	1×1	1	8	–
LIF-1	Leaky Integrate and Fire (LIF)	–	–	8	snntorch.Leaky
Conv-2	1×1 Conv $\rightarrow$ BatchNorm	1×1	1	1	–
LIF-2	Leaky Integrate and Fire (LIF)	–	–	1	snntorch.Leaky
Output: RepNet Features of size $640 \times 480 \times 1$					

Table 4. Stereo Matching Network (SMNet) Architecture Details for DSEC [4]. (FPN = Feature Pyramid Network)

Module	Layer Name	Type (Block Structure)	I/O Channels
Feature Extractor (Input: $640 \times 480 \times 4$ intensity and representation)			
ResNetFeature	Conv-0	Conv2D $\rightarrow$ BatchNorm $\rightarrow$ ReLU	$4 \rightarrow 32$
	Layer-1 (Block 1)	Bottleneck (w/ 1×1 Downsample)	$32 \rightarrow 128$
	Layer-1 (Blocks 2-3)	$2\times$ Bottleneck	$128 \rightarrow 128$
	Layer-2 (Block 1)	DeformBottleneck (w/ 1×1 Downsample)	$128 \rightarrow 256$
	Layer-2 (Blocks 2-4)	$3\times$ DeformBottleneck	$256 \rightarrow 256$
	Layer-3 (Block 1)	DeformBottleneck (w/ 1×1 Downsample)	$256 \rightarrow 512$
	Layer-3 (Blocks 2-6)	$5\times$ DeformBottleneck	$512 \rightarrow 512$
FPN	Lateral Conv-1	1×1 Conv2D	$128 \rightarrow 128$
	Lateral Conv-2	1×1 Conv2D	$256 \rightarrow 128$
	Lateral Conv-3	1×1 Conv2D	$512 \rightarrow 128$
	FPN Conv-1	Conv2D $\rightarrow$ BN $\rightarrow$ ReLU (Scale $1/3$)	$128 \rightarrow 128$
	FPN Conv-2	Conv2D $\rightarrow$ BN $\rightarrow$ ReLU (Scale $1/6$)	$128 \rightarrow 128$
	FPN Conv-3	Conv2D $\rightarrow$ BN $\rightarrow$ ReLU (Scale $1/12$)	$128 \rightarrow 128$
Cost Volume Pyramid (Inputs: 128-channel FPN features)			
Cost Volumes	Scale 0 (1/3 Resolution)	Correlation	$\mathbf{64}$ $(\text{max\_disp} // (3 \times 2^0))$
	Scale 1 (1/6 Resolution)	Correlation	$\mathbf{32}$ $(\text{max\_disp} // (3 \times 2^1))$
	Scale 2 (1/12 Resolution)	Correlation	$\mathbf{16}$ $(\text{max\_disp} // (3 \times 2^2))$
Adaptive Aggregation (Inputs: 64 / 32 / 16 channeled Cost Volume, with num_fusions $= 3$)			
Adaptive Fusions	Fusion-0	$1\times$ DeformSimpleBottleneck (DCN)	$64 \rightarrow 64$
	Fusion-1	$1\times$ DCN	$32 \rightarrow 32$
	Fusion-2	$1\times$ DCN	$16 \rightarrow 16$
Final Conv	Scale 0 Conv Out	1×1 Conv	$64 \rightarrow 64$
	Scale 1 Conv Out	1×1 Conv	$32 \rightarrow 32$
	Scale 2 Conv Out	1×1 Conv	$16 \rightarrow 16$
Disparity Refinement (Inputs: Aggregated Cost Volume + Low Disparity + $640 \times 480 \times 4$ features)			
DR Pyramid	Disparity Estimation Layer for Scales 0, 1, and 2		
2× StereoDRNet	Conv-A (Left & Right Intensities)	Conv2D $\rightarrow$ BN $\rightarrow$ LeakyReLU	$8 \rightarrow 16$
	Conv-B (Disparity)	Conv2D $\rightarrow$ BN $\rightarrow$ LeakyReLU	$1 \rightarrow 16$
	Dilated Blocks	$6\times$ BasicBlock (Dilation: 1, 2, 4, 8, 1, 1)	$32 \rightarrow 32$
	Final Conv	Conv2D (Residual Disparity)	$32 \rightarrow 1$
Output: Disparity Prediction of size $640 \times 480 \times 1$			



(a) Intensity (b) SE-CFF [11] (c) FALCON (Ours)

Figure 3. **Qualitative Comparison on MVSEC.** (a) Intensity Images, (b) SE-CFF [11], and (c) RepNet Stacks, from our FALCON design together with the Disparity Predictions and zoomed in regions. Our model estimates better depth details.

References

[1] Jonathan T Barron. A general and adaptive robust loss function. In *CVPR*, pages 4331–4339, 2019. 3

[2] Rohan Chabra, Julian Straub, Christopher Sweeney, Richard Newcombe, and Henry Fuchs. Stereodrnet: Dilated residual stereonet. In *CVPR*, pages 11786–11795, 2019. 3, 4

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 4

[4] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021. 3, 5

[5] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 35:26565–26577, 2022. 1

[6] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[7] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *NeurIPS*, 35:17612–17625, 2022. 1

[8] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[9] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 3

[10] Mohammad Mostafavi, Kuk-Jin Yoon, and Jonghyun Choi. Event-intensity stereo: Estimating depth by the best of both worlds. In *ICCV*, pages 4258–4267, 2021. 1

[11] Yeongwoo Nam, Mohammad Mostafavi, Kuk-Jin Yoon, and Jonghyun Choi. Stereo depth from events cameras: Concentrate and focus on the future. In *CVPR*, pages 6114–6123, 2022. 1, 3, 4, 6

[12] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *CVPR*, pages 1959–1968, 2020. 4

[13] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3 (3):2032–2039, 2018. 2, 3