Events-to-Video: Bringing Modern Computer Vision to Event Cameras —Supplementary Material—

Henri Rebecq [†] René Ranftl [‡] Vladlen Koltun [‡] Davide Scaramuzza [†]

A. Architecture Details

Our reconstruction network is presented in detail in Fig. 1. It is essentially a UNet architecture [9], composed of multiple convolutional layers. Four encoder layers (blue) are followed by two residual blocks (yellow), four decoder layers (red), and a final image prediction layer. In addition, symmetric skip connections are used. The encoders are strided convolutional layers (stride of 2), with a kernel size of 5. The number of output channels of the first encoder layer ($\downarrow$conv1) is 64, and is doubled for every subsequent encoder layer, *i.e*. the sequence of output channels is $(64, 128, 256, 512)$. Both residual blocks have 512 hidden layers, and a kernel size of 3. Batch normalization is used within the residual blocks (applied before the activation). The decoders are transposed convolution layers, with a kernel size of 5. The number of output channels of the decoders starts at 256 ($\uparrow$conv1), and is divided by two for every subsequent decoder, *i.e*. the sequence of output channels is $(256, 128, 64, 32)$). The kernel size for the residual blocks is 3. ReLU is used as activation everywhere, except for the last image prediction layer, where a sigmoid activation is used instead. The skip connections are based on concatenation.

B. Initialization Phase

By analyzing the initialization phase (*i.e*. when only few events have been triggered yet) in detail we gain interesting insight into how our network operates. We see significantly different behaviour when compared to prior approaches that are based on direct event integration. Fig. 2 compares image reconstructions from our approach, HF, and MR during the initialization phase. We specifically examine the interval from $0\,\mathrm{s}$ to $0.5\,\mathrm{s}$ after the event cameras has been started.

HF and MR, which rely on event integration, can only recover the intensity up to the initial (unknown) image $\mathcal{I}_0$ (*i.e*. they can only recover $\hat{\mathcal{I}} \approx \mathcal{I} - \mathcal{I}_0$), which results in an "edge" image which does not capture the appearance of the scene correctly. In contrast, our method successfully leverages deep priors to reconstruct the scene despite the low number of events that is available.

By contrast, our method is able to leverage deep priors on what a scene might look like (learned, first, from the large amount of simulated event data, and second, embedded in the perceptual loss used (LPIPS)) to reconstruct the scene reasonably well even with a low number of events.

C. Why Use Synthetic Training Data?

Here, we expand on the reasons that motivated us to train our reconstruction network using synthetic event data. First, simulation allows to capture a large variety of scenes and motions at very little cost. Second, a conventional camera (even a high quality one) would provide poor ground truth in high-speed conditions (motion blur) and HDR scenes, which are the conditions in which event sensors excel; by contrast, synthetic data does not suffer from these issues. Last but not least, simulation allows to randomize the contrast thresholds of the event sensor, which increases the ability of the network to generalize to different sensor configurations (contrast sensitivity). To illustrate this last point, we show in Fig. 3 (left) what happens when training the network on real event data from an event camera (specifically, the sequences from the Event Camera Dataset [4] already presented in the main paper, which were recorded with a DAVIS240C sensor), and evaluating the trained network on data coming from a different event sensor (specifically, the 'outdoors_day1' sequence from the MVSEC dataset [13], which was recorded with a mDAVIS346 sensor): the reconstruction suffers from many artefacts. This can be explained by the fact that the events from the mDAVIS346 sensor have statistics that are quite different from the training events (DAVIS240C): the set of contrast thresholds are likely quite different between both sensors, and the illumination conditions are also different (outdoor lighting for the MVSEC dataset versus indoor lighting for the training event data). By contrast, the network trained on simulated event data (Fig. 3, right) generalizes well to the event data from the mDAVIS346, producing a visually pleasing image reconstruction.

[†] Dept. Informatics, Univ. of Zurich and Dept. Neuroinformatics, Univ. of Zurich and ETH Zurich

[‡] Intel Labs

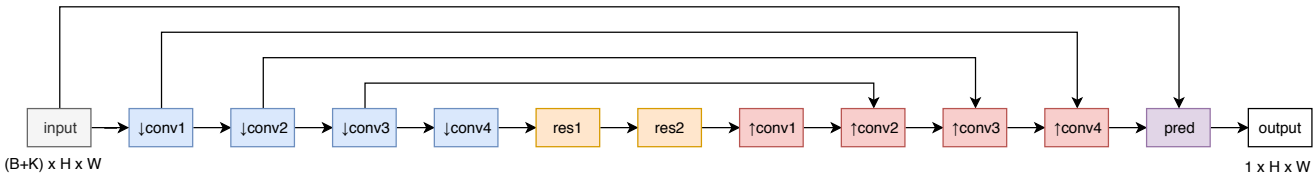


Figure 1. Detailed architecture of our reconstruction network. It is a UNet architecture [9], composed of multiple convolutional layers. Four encoder layers (blue) are followed by two residual blocks (yellow), four decoder layers (red), and a final image prediction layer. In addition, symmetric skip connections are used. More details about each layer are provided in the text.

(a) HF (b) MR (c) Ours (d) Ground truth

Figure 2. Analysis of the initialization phase (reconstruction from few events). This figure shows image reconstructions from each method, 0.5 seconds after the sensor was started. HF [11] and MR [5], which are based on event integration, cannot recover the intensity correctly, but only the intensity difference , resulting in "edge " images (first and second row), or severe "ghosting" effects (third row, where the trace of the dartboard is clearly visible). In contrast, our network successfully reconstructs most of the scene accurately, even with a low number of events.

Figure 3. Reconstruction from (i) a network trained only on real event data from the DAVIS240C sensor (left), and (ii) a network trained only on simulated event data (right). This sequence is from the MVSEC dataset, and was recorded with a mDAVIS346 sensor.

D. Latency and Performance Considerations

D.1. Latency

Our method operates on windows containing a fixed number of events, which incurs some latency compared to approaches that operate in an event-by-event fashion. The amount of latency depends on the duration of the event windows. Therefore, it varies through time, depending on the event rate. Since we process events in windows with a constant number of events, our method is data-driven, which means that the latency diminishes when the event rate increases, *i.e.* when something of interest occurs in the scene. Fig. 4 shows the distribution of the latency (in other words, the durations of the event windows) for the sequences used in our qualitative evaluation (Section 4 in the main paper): the latency is in the range of $3\,\mathrm{ms}$ to $300\,\mathrm{ms}$, with a me-

dian value of 26 ms. We point out that while in this paper we chose to process windows with a fixed number of events, our methodology could also be applied to windows of a constant duration (which would require, however, to retrain the network to operate with such fixed-length windows), yielding a constant, predictable latency, but at the cost of losing the adaptive output framerate.

D.2. Performance Analysis

In this section, we analyze the performance of our method, and compare it against HF [11] and MR [5]. Due to fundamental differences in the way each of these methods process event data, it is difficult to provide a direct and fair performance comparison between these three methods. HF processes the event stream in an event-by-event fashion, providing (in theory) a new image reconstruction with every incoming event. However, the raw image reconstructions from HF need to be filtered (for example, using a bilateral filter) to obtain results with reasonable quality. While MR can in principle also operate in an event-by-event fashion, its best quality results are obtained when it processes small batches of events (in our experiments, we used $N_{MR} = 1{,}000$ events per batch), thus limiting the output framerate. Our method, by contrast, processes the event stream in large batches of events (we used $N = 25{,}000$ events per batch), thus also limiting the output framerate (*i.e.* increasing the latency, as analyzed in Fig. 4). In Table 1, we report the mean event processing rate (*i.e.* the total time it takes to process a dataset divided by the number of events in the dataset) for each method. As a complementary performance measure, we also report the mean "frame synthesis time", which we define as follows:

- for our method, it is the time it takes to process N events.
- For MR, it is the time it takes to process N_{MR} events.
- For HF, it is in theory the time it takes to process a single event (since every new event triggers a new reconstruction), plus the image filtering time. However, the time to process a single event is negligible compared to the filtering time (multiple orders of magnitude less), hence we report only the filtering time. As described in the main text, we used a bilateral filter with filter size $d = 5$ and $\sigma = 25$).

We ran our method and MR on an NVIDIA GeForce RTX 2080 Ti GPU, and HF on an Intel Core i9-9900K @ 3.60 GHz CPU.

Discussion We point out that our method synthesizes fewer images per second than MR and HF, hence the numbers in Table 1 should be interpreted carefully. That being said, our method is fairly competitive in terms of performance, and, importantly, can easily run in real-time, while providing state of the art reconstructions in terms of quality.

	Event rate (Mev/s)	Frame synthesis time (ms)
HF	14.30	0.75 (filter)
MR	1.19	0.84
Ours	7.94	3.15

Table 1. Performance comparison between our method, HF, and MR.

E. Additional Results

E.1. Video

We strongly encourage the reader to view the supplemental video, which contains:

- Video reconstructions from our method on various event datasets, with a visual comparison to several state of the art methods.
- Video of the VINS-Mono visual-inertial odometry algorithm [7] running on a video reconstruction from events.
- Qualitative results on two additional downstream applications that were not presented in the main paper: object detection (based on YOLOv3 [8]), and monocular depth prediction (based on MegaDepth [3]). We point out that neither of these tasks have ever been shown with event data before this work.

E.2. Results on Synthetic Event Data

We show a quantitative comparison of the reconstruction quality of our method as well as MR and HF on synthetic event sequences in Table 2. We present qualitative reconstruction results on this dataset in Fig. 5. All methods perform better on synthetic data than real data. This is expected because simulated events are free of noise. Nonetheless, the performance gap between our method and the state of the art is preserved, and even slightly increases (24% improvement in SSIM, 56% decrease in LPIPS). We note that perfect reconstruction, even on noise-free event streams is not possible, since image reconstruction from events is only possibly up the the quantization limit imposed by the contrast threshold of the event camera.

E.3. Additional Qualitative Results on Real Data

Fig. 6 shows qualitative results on sequences from the Event Camera Dataset [4] (which we used for our quantitative evaluation). Fig 7 shows qualitative results on the sequences introduced by Bardow *et al*. [1]. Figs. 8 and 9 present HDR reconstruction results on sequences from the

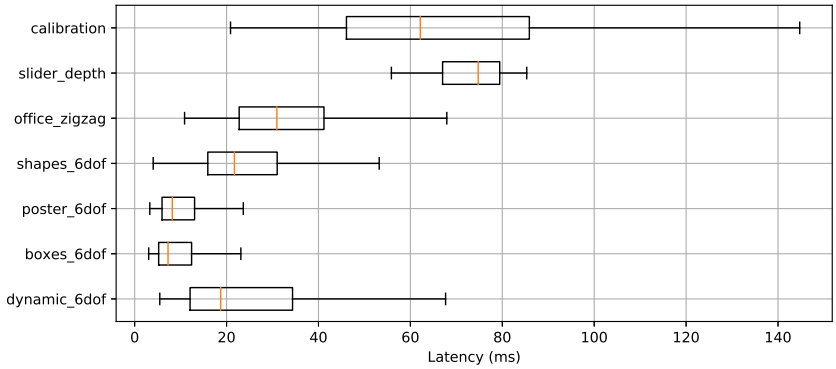


Figure 4. Distribution of latency for each dataset used in our quantitative evaluation.

Dataset	MSE			SSIM			LPIPS		
	HF	MR	Ours	HF	MR	Ours	HF	MR	Ours
synthetic_0	0.08	0.06	**0.02**	0.54	0.61	**0.83**	0.49	0.47	**0.26**
synthetic_1	0.15	0.14	**0.06**	0.38	0.45	**0.60**	0.54	0.56	**0.37**
synthetic_2	0.07	0.08	**0.02**	0.60	0.68	**0.82**	0.42	0.42	**0.26**
synthetic_3	0.07	0.05	**0.04**	0.57	0.66	**0.75**	0.45	0.43	**0.33**
synthetic_4	0.08	0.06	**0.02**	0.62	0.67	**0.85**	0.41	0.42	**0.25**
synthetic_5	0.08	0.08	**0.02**	0.50	0.61	**0.74**	0.53	0.54	**0.36**
synthetic_6	0.07	0.04	**0.03**	0.56	0.65	**0.77**	0.44	0.48	**0.30**
Mean	0.09	0.07	**0.03**	0.54	0.62	**0.77**	0.47	0.47	**0.30**

Table 2. Comparison of image quality with respect to state of the art on synthetic event sequences.

MVSEC dataset [13]. Further results are shown in the supplementary video which conveys these results in a better form than still images.

F. Object Classification

Below we detail the exact modalities of our reconstruction method for each of the dataset which we used for our evaluation of object classification (Section 5.1 in the paper), as well as the specific architectures used and training modalities.

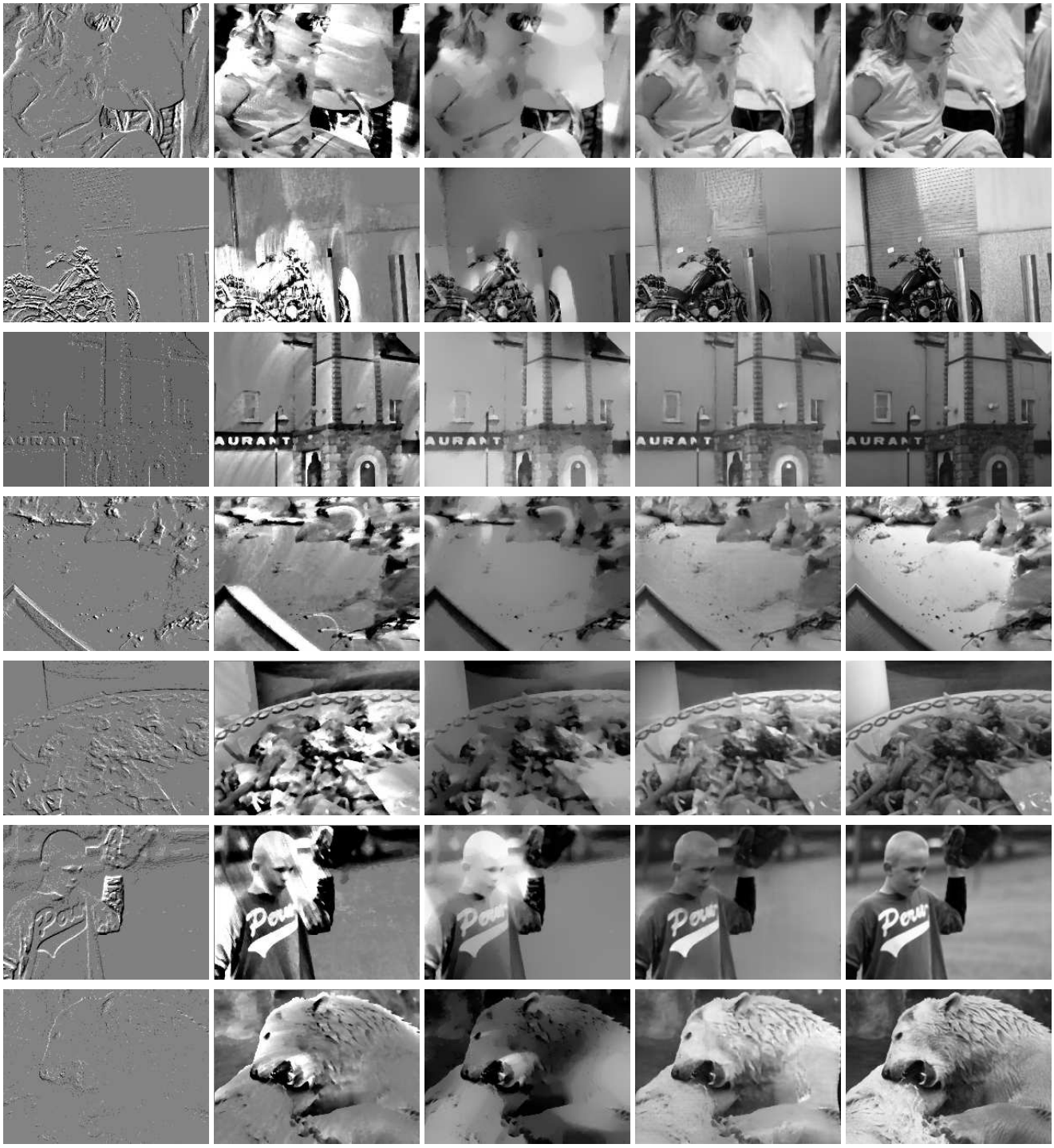
N-MNIST. To reconstruct images with our networks, we used an event window of $N = 1{,}000$ events. We passed every event sequence into our network, resulting in a video, from which we keep the final image as input for the classification network. To match the images from the original MNIST dataset, we additionally binarize the reconstructed image (whose values lie in $[0, 1]$) with a threshold of 0.5. The train and test images were normalized so that the mean value of each image is 0.1307 and the variance 0.3081. We used the official train and test split provided in the MNNIST dataset. As there is no standard state of the art architecture for MNIST, we used a simple CNN architecture as our classification network, composed of the following blocks:

- 2D convolution (stride: 5, output channels: 32) + ReLU
- 2D convolution (stride: 5, output channels: 64) + ReLU
- 2D max pooling (size: 2) + Dropout
- Fully connected layer (output size: 128 neurons) + ReLU
- Fully connected layer (output size: 10 neurons)

We used the cross entropy loss, and trained the network for 15 epochs using the ADAM optimizer, with a learning rate of 0.001.

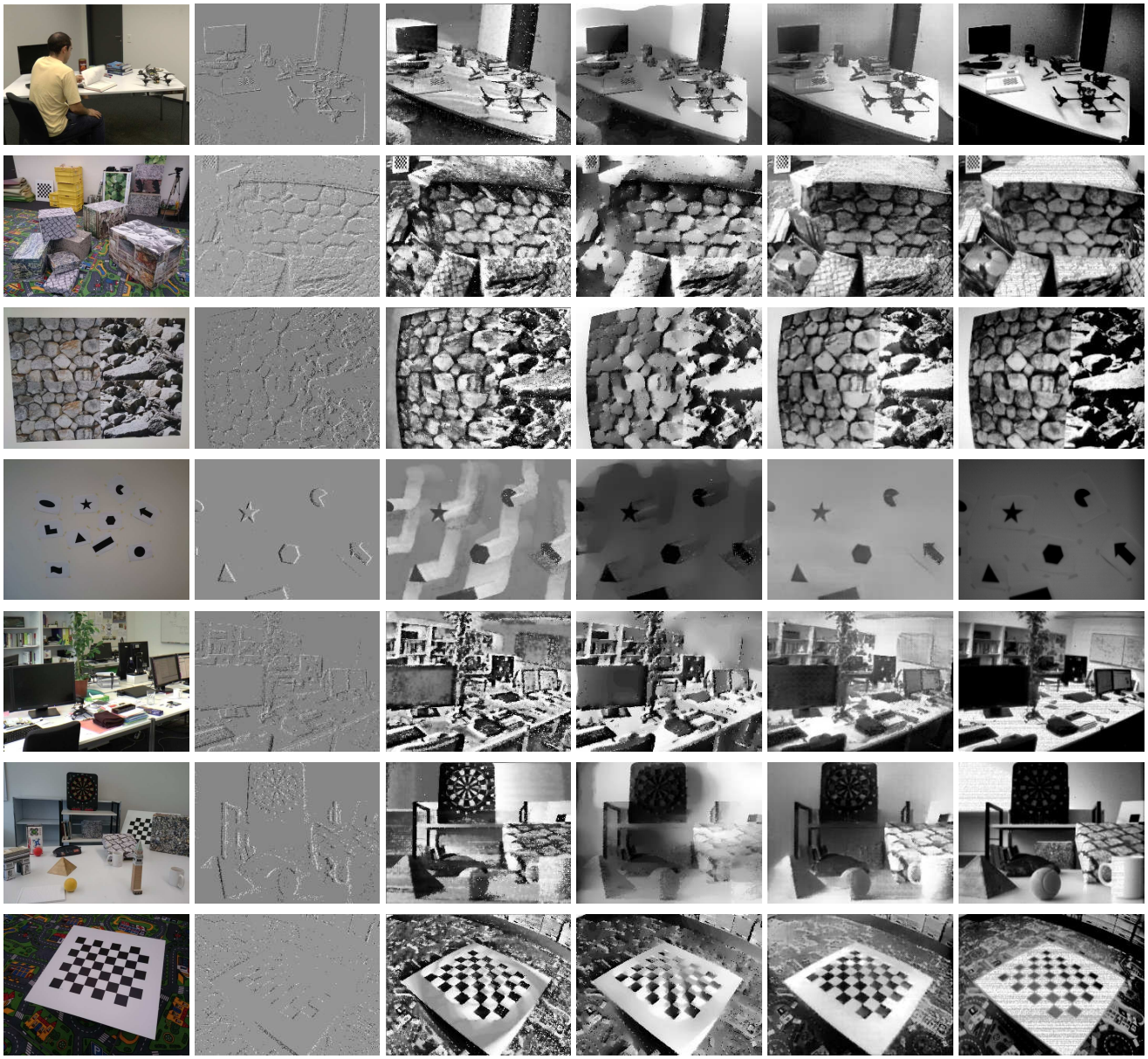
N-CARS. We used windows of events with a fixed temporal size of $20\,\mathrm{ms}$, and used the last reconstructed image from the video as input to the classification network. We used the official train and test split provided by the N-CARS dataset. We used a ResNet18 [2] architecture (with an additional fully connected final layer with 2 output neurons), initialized with weights pretrained on ImageNet [10], and fine-tuned the network using the reconstructed images from the training set for 20 epochs, using SGD with a learning rate of 0.001 (decayed by factor of 0.1 every 7 epochs), and momentum of 0.1.

N-Caltech101. For image reconstruction, we used windows of $N = 10{,}000$, events and used the last reconstructed image as input to the classification network. Since there is no official train and test split for the N-Caltech101 dataset, we



(a) Events (b) HF (c) MR (d) Ours (e) Ground truth

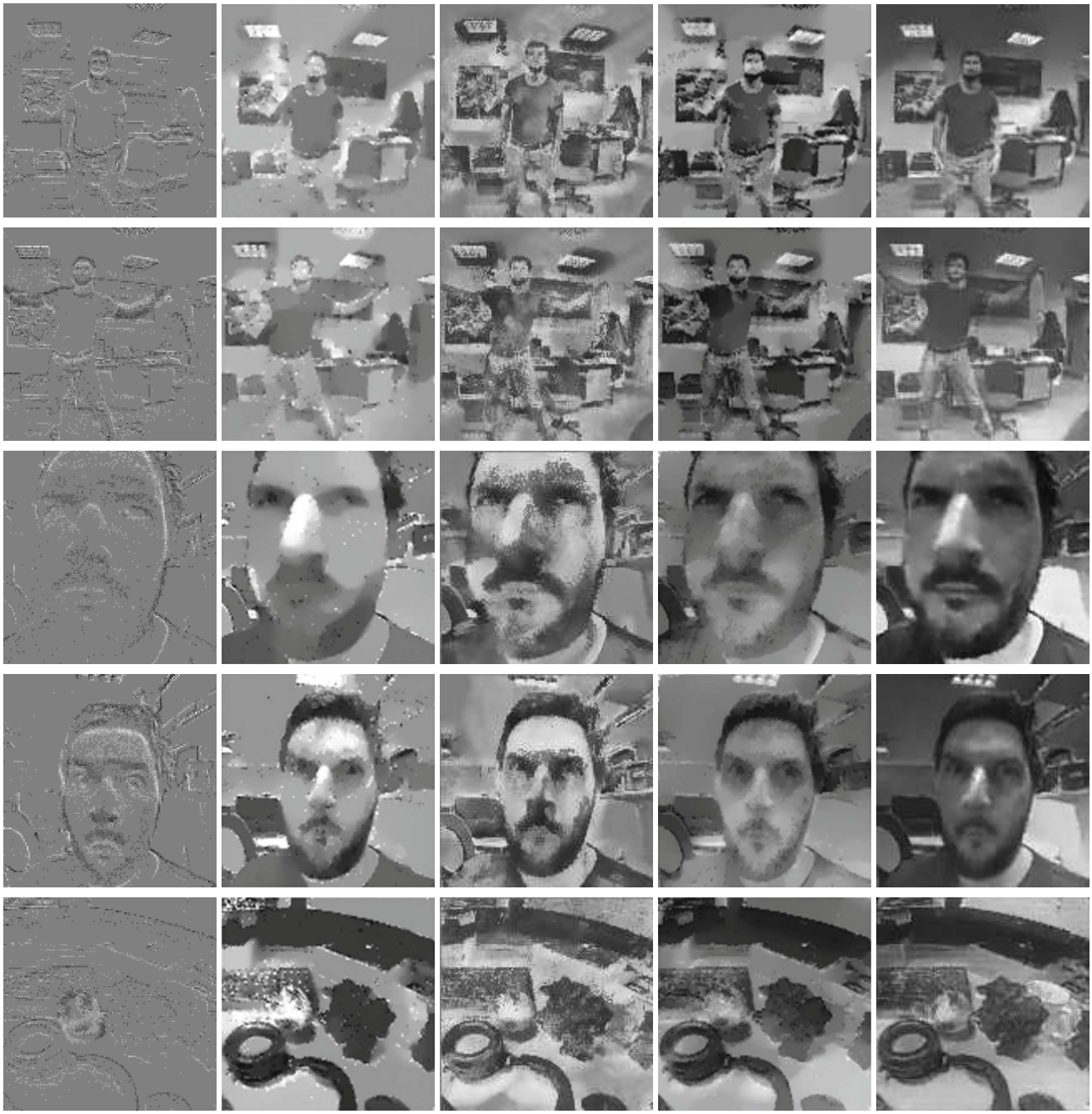
Figure 5. Qualitative comparison of our reconstruction method with HF [11] and MR [5] on synthetic sequences from the validation set. Note our method is able to reconstruct fine details such as the bear's fur (last row), which competing methods are not able to preserve.



(a) Scene Preview (b) Events (c) HF (d) MR (e) Ours (f) Ground truth

Figure 6. Qualitative comparison of our reconstruction method with two recent competing approaches, MR [5] and HF [11], on sequences from [4], which contain ground truth frames from a DAVIS240C sensor. Our method successfully reconstructs fine details (textures in the second and third row) compared to other methods, while avoiding ghosting effects (particulary visible in the shapes sequences on the fourth row).

split the dataset randomly into two third training sequences (5,863 sequences) and one third testing sequences (2,396 sequences), following the methodology used by HATS [12]. The train and test images were converted to 3-channel grayscale images (*i.e*. the three channels are the same), and normalized so that the mean value of each image is 0.485 and the variance 0.229. We also performed data augmentation at train time (random horizontal flips, and random crop of size 224). At test time, we resized all the images to 256×256 and cropped the image around the center with a size of 224. We used a ResNet18 architecture (with an additional fully-connected final layer with 101 output neurons), initialized with weights pretrained on ImageNet, and fine-tuned the network using the reconstructed images from the training set for 25 epochs using SGD with an initial learning rate of 0.001 (decayed by a factor of 0.1 every 7 epochs) and momentum of 0.1. Fig. 10 shows additional reconstruction examples from the N-Caltech101 dataset.

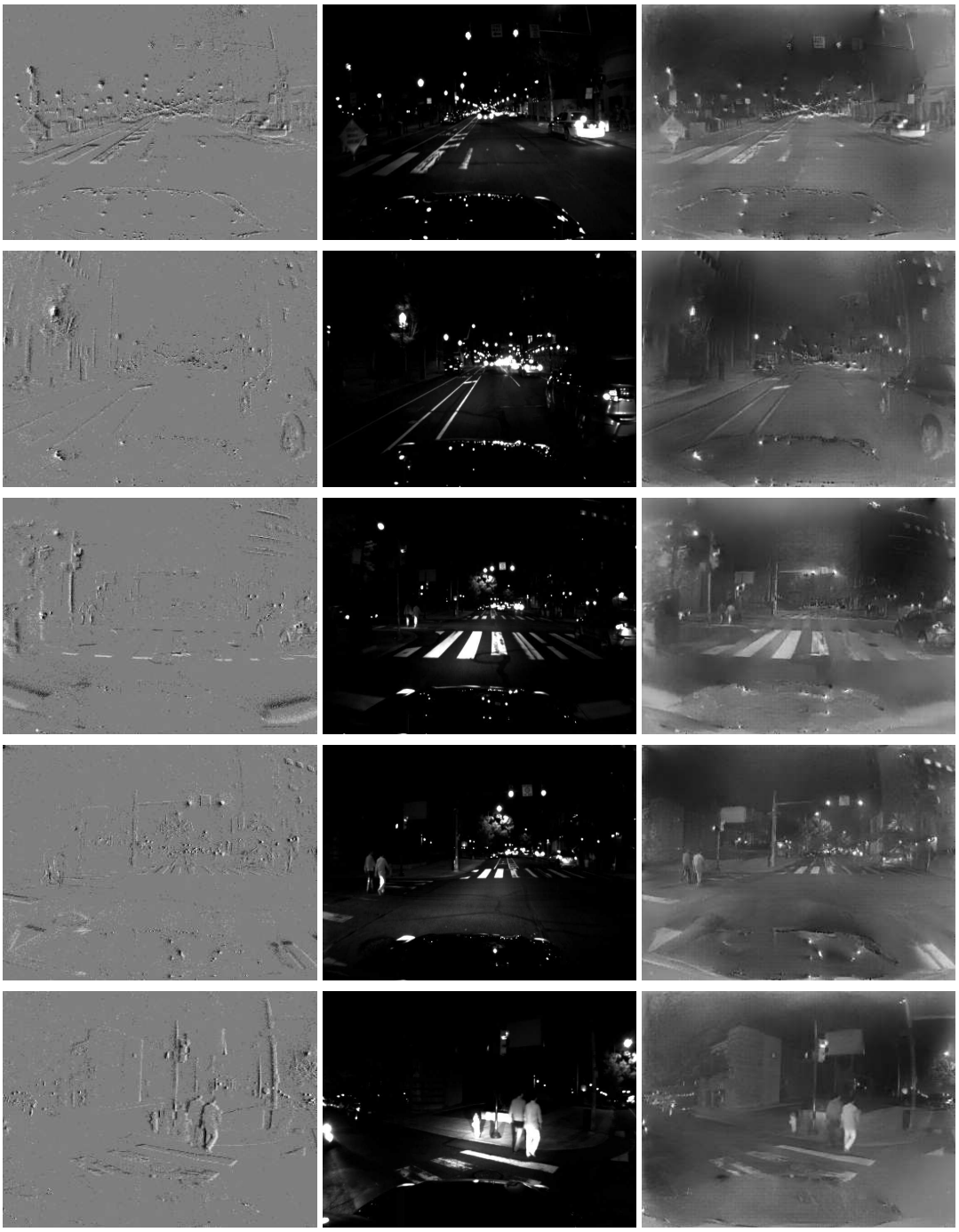


(a) Events (b) SOFIE [1] (c) HF [11] (d) MR [5] (e) Ours

Figure 7. Qualitative comparison of our reconstruction method with various competing approaches. We used the datasets from [1]. The dataset does not contain ground truth images, thus only a qualitative comparison is possible. For SOFIE and MR, we used images provided by the authors, for which the parameters were tuned for each dataset. For HF, we ran the code provided by the authors, manually tuned the parameters on these datasets to achieve the best visual quality, and additionally applied a bilateral filter to clean the high frequency noise present in the original reconstructions.

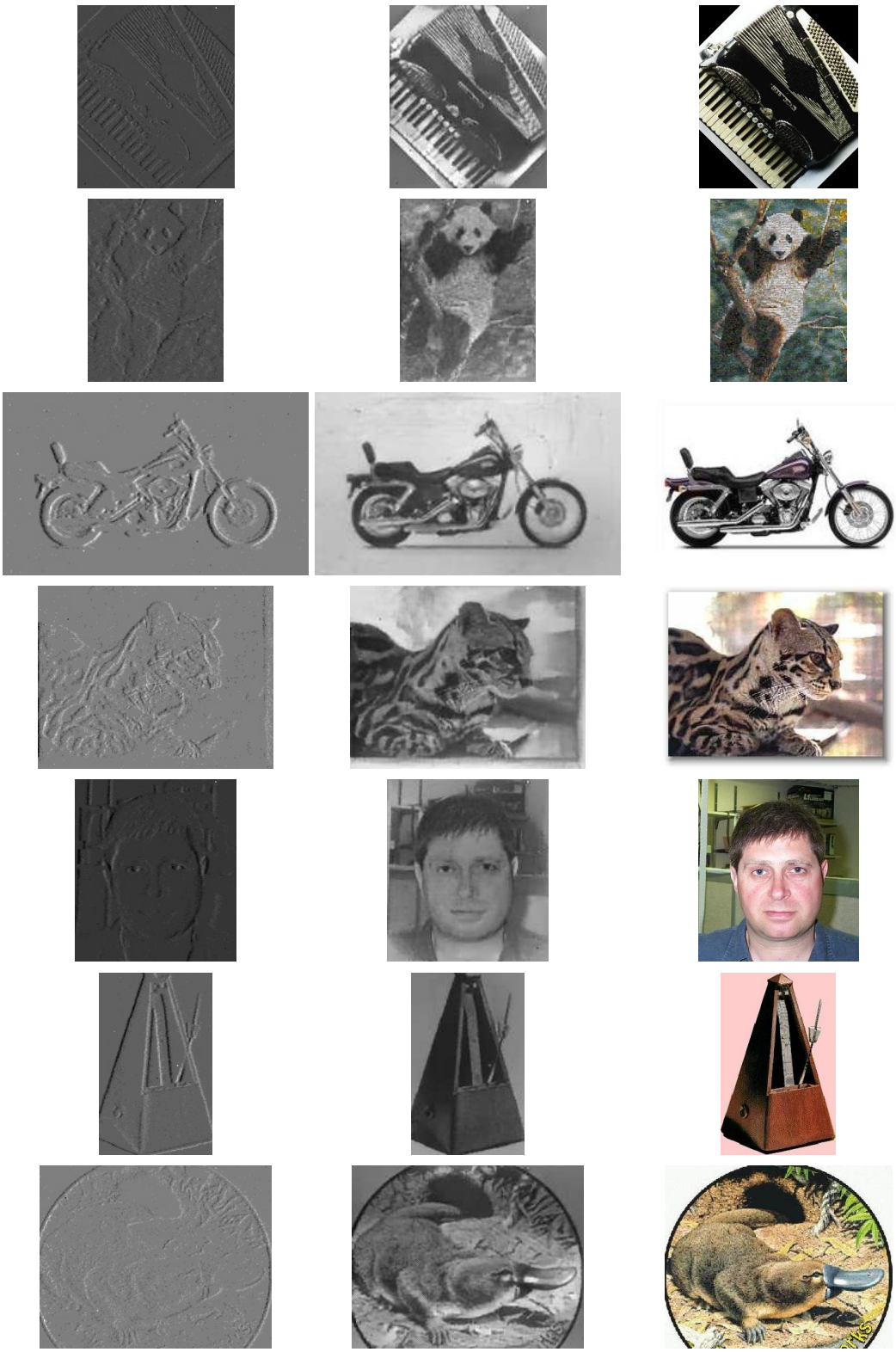
(a) Events (b) VI sensor frame (c) Our reconstruction

Figure 8. Example HDR reconstructions on the MVSEC automotive dataset [13]. The standard frames were recorded with a high-quality VI sensor with auto-exposure activated. Because the camera is facing directly the sun, the standard frames (b) are either under- or over-exposed since the limited dynamic range of the standard sensor cannot cope with the high dynamic range of the scene. By contrast, the events (a) capture the whole dynamic range of the scene, which our method successfully reconstructs to high dynamic range images (c), allow to discover details that were not visible in the standard frames.



(a) Events (b) VI sensor frame (c) Our reconstruction

Figure 9. Example HDR reconstructions on the MVSEC automotive dataset [13] at night. The standard frames were recorded with a high-quality VI sensor with auto-exposure activated. Because of low light during the night, the standard frames (b) are severely degraded. By contrast, the events (a) still can capture the whole dynamic range of the scene, which our method successfully recovers (c), allowing to discover details that were not visible in the standard frames.



(a) Events (b) Our Reconstruction (c) Original Image

Figure 10. (a) Previews of some event sequences from the N-Caltech101 dataset [6] which features event sequences converted from the Caltech101 dataset. (b) our reconstructions (from events only) preserve many of the details and statistics of the original images (c). Note that these datasets feature planar motion (since Caltech101 images were projected on white wall to record the events), which coincides with the type of motions present in the simulated data, which explains in part the outstanding visual quality of the reconstructions.

G. Visual-Inertial Odometry

Figs. 11, 12 and 13 provide additional results on the visual-inertial odometry experiments presented in the main paper. Specifically, they provide, for each sequence used in our evaluation, the evolution of the mean translation and rotation error as a function of the travelled distance for our approach, UltimateSLAM (E+I), and UltimateSLAM (E+F+I).

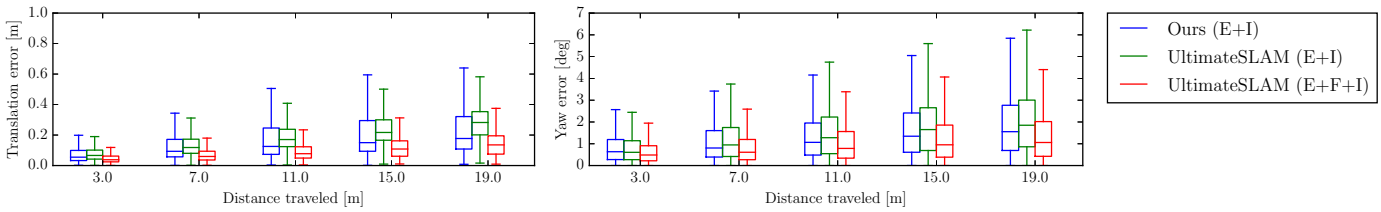


Figure 11. Evolution of the overall mean translation error (in meters) and mean rotation error (in degrees), averaged across all the datasets used in our evaluation.

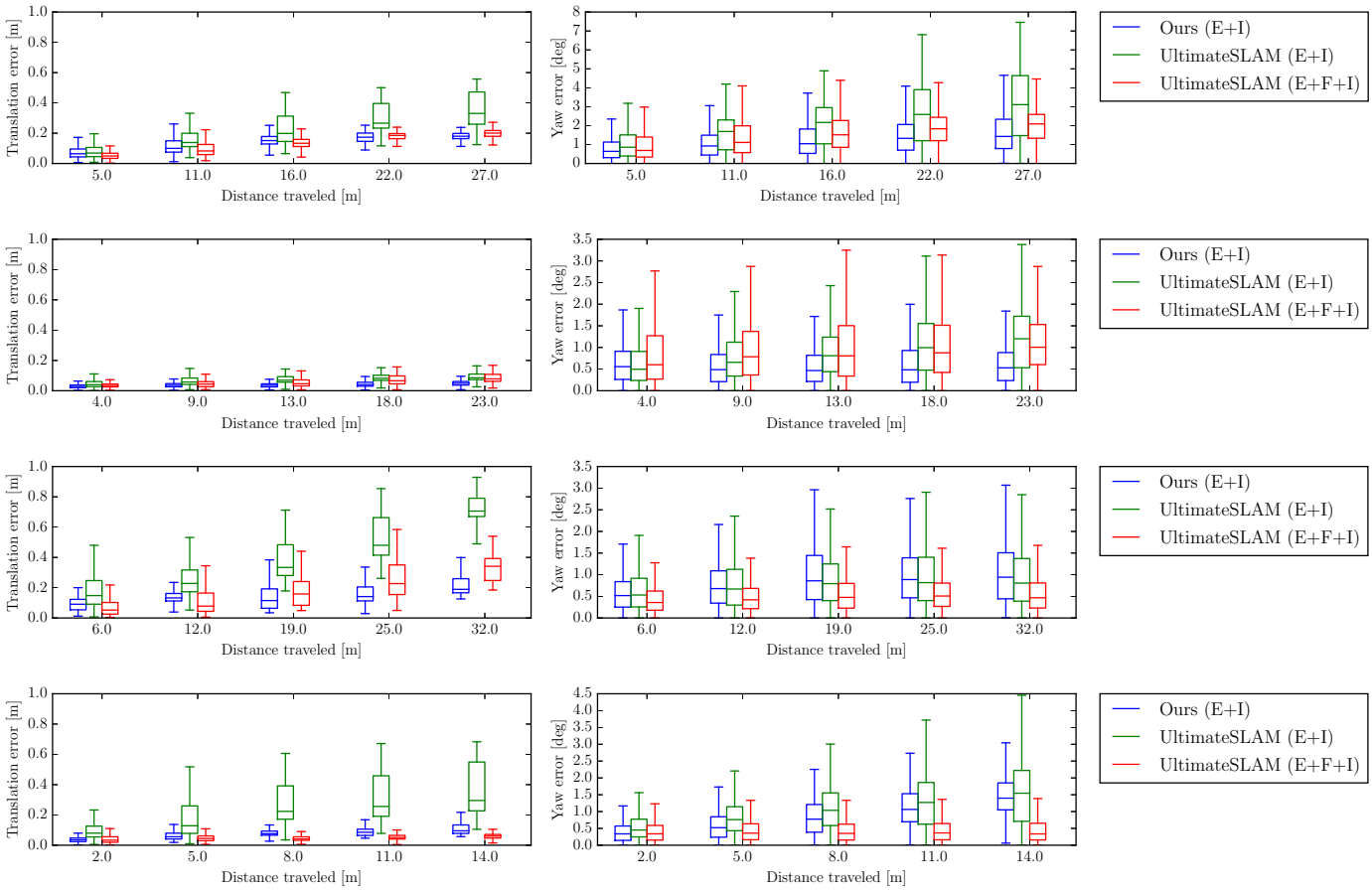


Figure 12. Evolution of the mean translation error (in meters) and mean rotation error (in degrees), as a function of the travelled distance. Sequences from top to bottom: 'shapes_translation', 'poster_translation', 'boxes_translation', 'dynamic_translation'.

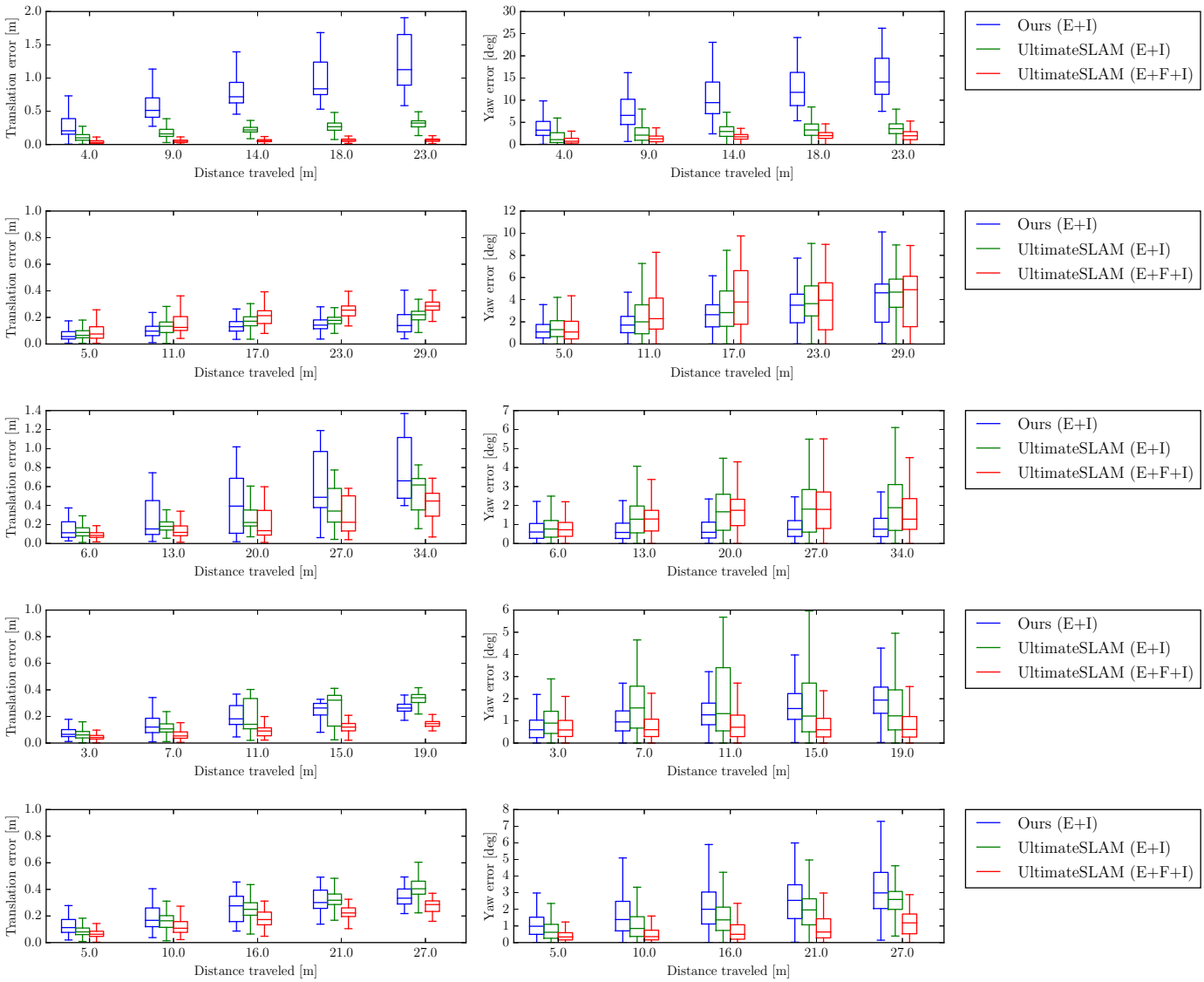


Figure 13. Evolution of the mean translation error (in meters) and mean rotation error (in degrees), as a function of the travelled distance. Sequences from top to bottom: 'shapes_6dof', 'poster_6dof', 'boxes_6dof', 'dynamic_6dof', 'hdr_boxes'.

References

[1] Patrick Bardow, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016. 3, 7

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016. 4

[3] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 3

[4] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robot. Research*, 36:142–149, 2017. 1, 3, 6

[5] Gottfried Munda, Christian Reinbacher, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *Int. J. Comput. Vis.*, 2018. 2, 3, 5, 6, 7

[6] Garrick Orchard, Ajinkya Jayawant, Gregory K. Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.*, 9:437, 2015. 10

[7] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *arXiv:1708.03852*, 2017. 3

[8] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 3

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 1, 2

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 4

[11] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conf. Comput. Vis. (ACCV)*, 2018. 2, 3, 5, 6, 7

[12] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 6

[13] Alex Zihao Zhu, Dinesh Thakur, Tolga Ozaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robot. Autom. Lett.*, 3(3):2032–2039, 2018. 1, 4, 8, 9