Supplementary Material For ESOD: Event-Based Small Object Detection

Quanmin Liang*
Sun Yat-Sen University
Guangzhou, China
Pengcheng Laboratory
Shenzhen, China
liangqm5@mail2.sysu.edu.cn

Jinyi Lu*
Sun Yat-Sen University
Guangzhou, China
Pengcheng Laboratory
Shenzhen, China
lujy87@mail2.sysu.edu.cn

Qiang Li
Xpeng Motors Technology Co Ltd
Guangzhou, China
liqiang27@mail2.sysu.edu.cn

Shuai Liu
Sun Yat-Sen University
Guangzhou, China
liush376@mail2.sysu.edu.cn

Zhihao Zhao
Technical University of Munich
Munich, Germany
zhihao.zhao@tum.de

Yinzheng Zhao
Technical University of Munich
Munich, Germany
yinzheng.zhao@tum.de

Wei Zhang†
Pengcheng Laboratory
Shenzhen, China
zhangwei1213052@126.com

Kai Huang†
Sun Yat-Sen University
Guangzhou, China
huangk36@mail.sysu.edu.cn

Yonghong Tian
Peking University
Beijing, China
Pengcheng Laboratory
Shenzhen, China
yhtian@pku.edu.cn

ACM Reference Format:
Quanmin Liang, Jinyi Lu, Qiang Li, Shuai Liu, Zhihao Zhao, Yinzheng Zhao, Wei Zhang, Kai Huang, and Yonghong Tian. 2025. Supplementary Material For ESOD: Event-Based Small Object Detection. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3746027.3755486

1 Experimental Setup

1.1 Data Collection

Before collecting the dataset, we first define two key concepts: fast-moving objects and small objects. In photography, an object is considered fast-moving if its displacement on the image sensor during the exposure time exceeds 1–2 pixels [4]. Given that a typical camera has an exposure time of 1/500 seconds, an object moving at 500–1000 pixels per second qualifies as a fast-moving object. Small objects are generally defined as those with an area $\leq 32 \times 32$ pixels [3]. In practical applications, a small object is often considered one whose width and height are both smaller than 1/10 of the input image size. However, in our experiments, we found that when the object size falls below 25×25 pixels, event cameras struggle to capture sufficient information due to noise and the lack of color cues, making object classification challenging. Therefore, we constrain the bbox size of small objects in our dataset to range between 30×30 and 45×45 pixels.

We used a 1920×1080 resolution, $144Hz$ display and a Prophesee EVK4-HD event camera (1280×720 resolution) for data collection. We refer to this dataset as **ESOD-syn**. Specifically, we selected 108 video sequences from ImageNet-VID [5] as backgrounds and obtained 3D object models from Sketchfab[1] and Free3D[2]. These models cover 11 categories with a total of 58 unique objects. For each object, we applied 3D rotations from different viewpoints, projected them onto a 2D plane, and generated 240 different perspectives. To synthesize videos, we randomly selected 2–5 objects as foreground elements and randomly chose one background sequence. The foreground objects were assigned random motion trajectories, including linear, parabolic, and sinusoidal paths, moving 1–2 pixels per millisecond. The background image was updated every 20–30 milliseconds, simulating the frame rate of conventional cameras. Using this approach, we generated 320 video sequences at 1000 FPS. To prevent frame loss, we downsampled them to 70 FPS for display playback while ensuring synchronization between the event camera and the video using hardware trigger signals. We also adjusted screen brightness to simulate different lighting conditions. After recording, we scaled the event timestamps back to 1000 FPS. Since both object classes and trajectories were predefined, we only needed to determine the start and end timestamps for each sequence and locate the screen position within the event camera's field of view. This allowed us to accurately extract per-millisecond bboxes for each moving object. In practical applications, to ensure sufficient information while maintaining a high frame

*Equal Contribution
†Corresponding Author

MM '25, Dublin, Ireland.

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755486

[1] https://sketchfab.com/
[2] https://free3d.com/

rate, we accumulated event streams over 5-millisecond intervals and extracted corresponding bboxes for detection. This process resulted in a dataset containing 240,000 high-quality bboxes, which we refer to as **ESOD-syn**.

ESOD-syn. As illustrated in Fig. 1, for the synthetic dataset, we first rendered 3D small objects from different viewpoints: rotating 45° clockwise and counterclockwise around the z-axis, and ±90° around the y-axis. In the generated videos, objects move either from left to right or right to left. For right-to-left motion, we horizontally flip the object to match the motion direction. The motion trajectories include sine waves, parabolas, and straight lines. To simulate shape variations during movement, the object size is randomly altered every 200 ms, keeping the bounding box area between 30×30 and 45×45 pixels. To model different lighting conditions, we adjust the screen brightness to 50, 80, and 100, representing low-light, normal, and overexposed environments, respectively. We selected 11 commonly seen small object categories: *airplane, ball, bicycle, boat, bird, car, human, insect, motorcycle, rocket, and UAV*.

ESOD-real. To simulate realistic object motion, we employed two strategies: a motorized rotating platform and hand-throwing. The hand-thrown trajectories included upward parabolas, downward parabolas, and straight-line motion. For annotation, we first manually labeled approximately 2% of the data. Then, a fine-tuned YOLOv11 [2] model was used to assist in bounding box generation. It is important to note that for each video sequence, we had prior knowledge of the object category and its approximate motion trajectory. Therefore, during auto-labeling with YOLO, we only required the model to predict bounding boxes, omitting the need for classification, which significantly simplified the labeling task. Finally, we attached the known object class to each trajectory and conducted manual verification to ensure that the bounding boxes and category labels were correct for each segment of event data.

Figure 1: Different projection views of 3D objects.

1.2 Training Settings

For training FDDNet, we set the number of training epochs to 100 and adopted the AdamW optimizer. The initial learning rate was set to 1e-3 with a weight decay of 1e-4. A warm-up phase was applied in the first 10 epochs. The event sequence length was set to $L = 8$, with a batch size of 2. Input event frames were resized to 640×640 before being fed into the network. All experiments were conducted using a single NVIDIA V100 GPU.

(C, N)	mAP	mAP@50	mAP@75	Params
(16, 1)	37.4	56.2	44.7	**4.0M**
(32, 1)	**42.3**	**63.3**	**49.8**	15.6M
(16, 2)	37.8	57.0	45.3	4.2M
(16, 3)	38.0	57.5	45.6	4.3M

Table 1: Ablation results of the base channel number C and the number of Deformable Attention layers N. Experiments are conducted on the ESOD-syn dataset.

Representation	mAP	mAP@50	mAP@75
ECM [6]	34.6	52.9	43.1
Voxid Grid [7]	36.5	55.2	44.0
EST [1]	37.1	55.8	44.5
TECM	**37.4**	**56.2**	**44.7**

Table 2: Comparison of different event representation methods on the ESOD-syn dataset.

2 Ablation Study

Hyperparameter Analysis of FDDNet. The primary factor controlling the model's parameter is the number of channels C. In addition, we investigate the effect of the number of Deformable Attention layers N used in the detection head. We conduct ablation experiments on the ESOD-syn dataset, and the results are shown in Tab. 1. Increasing the number of channels significantly boosts the detection performance, but also leads to a substantial increase in the number of parameters. Similarly, stacking more Deformable Attention layers improves detection accuracy, though the performance gain gradually saturates, exhibiting diminishing returns.

Ablation on Event Representation Methods. Event data can be represented in various forms. To investigate the impact of different representations on detection performance, we compare TECM with three widely used alternatives: Event Count Map (ECM) [6], Voxel Grid [7], and Event Spike Tensor (EST) [1]. The comparison is conducted on the ESOD-syn dataset, and the results are shown in Tab. 2. As can be observed, TECM outperforms all other representations. Notably, ECM shows inferior performance, likely due to its complete disregard of the temporal dimension in the event stream. In contrast, TECM retains richer temporal, spatial, and polarity information, which is more beneficial for accurate small object detection.

References

[1] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. 2019. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5633–5643.

[2] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. 2023. *Ultralytics YOLO.* https://github.com/ultralytics/ultralytics

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer, 740–755.

[4] Sidney Ray. 2002. *Applied photographic optics.* Routledge.

[5] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115 (2015), 211–252.

[6] Alex Zihao Zhu and Liangzhe Yuan. 2018. EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras. In *Robotics: Science and Systems*.

[7] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. 2019. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 989–997.