Semantic Segmentation on Neuromorphic Vision Sensor Event-Streams Using PointNet++ and UNet Based Processing Approaches

Tobias Bolten[1][a], Regina Pohle-Fröhlich[1] and Klaus D. Tönnies[2]

[1]*Institute for Pattern Recognition, Hochschule Niederrhein, Krefeld, Germany*

[2]*Department of Simulation and Graphics, University of Magdeburg, Germany*

Keywords: Dynamic Vision Sensor, Semantic Segmentation, PointNet++, UNet.

Abstract: Neuromorphic Vision Sensors, which are also called Dynamic Vision Sensors, are bio-inspired optical sensors which have a completely different output paradigm compared to classic frame-based sensors. Each pixel of these sensors operates independently and asynchronously, detecting only local changes in brightness. The output of such a sensor is a spatially sparse stream of events, which has a high temporal resolution. However, the novel output paradigm raises challenges for processing in computer vision applications, as standard methods are not directly applicable on the sensor output without conversion.

Therefore, we consider different event representations by converting the sensor output into classical 2D frames, highly multichannel frames, 3D voxel grids as well as a native 3D space-time event cloud representation. Using PointNet++ and UNet, these representations and processing approaches are systematically evaluated to generate a semantic segmentation of the sensor output stream. This involves experiments on two different publicly available datasets within different application contexts (urban monitoring and autonomous driving).

In summary, PointNet++ based processing has been found advantageous over a UNet approach on lower resolution recordings with a comparatively lower event count. On the other hand, for recordings with ego-motion of the sensor and a resulting higher event count, UNet-based processing is advantageous.

1 INTRODUCTION

The Dynamic Vision Sensor (DVS), which stems from the research field of neuromorphic engineering, emulates key aspects of the human retina. This results in a basically different output paradigm compared to classic image sensors. A DVS does not operate at a fixed frame rate. Only local brightness changes in the scene are detected and directly transmitted. For this purpose, the pixels of a DVS work independently and asynchronously from each other. An output is generated as soon as a change in brightness above a defined threshold has been detected.

In this context, the triggering of a single DVS-pixel is called an "event". Each of these events is a tuple (x, y, t, p) which contains information about the spatial (x, y) position of the active pixel in the sensor array, a very precise timestamp t of triggering and a polarity indicator p, which encodes the direction of the brightness change (from bright to dark or vice versa). The output of a DVS is therefore an information-rich, sparse stream of events with a vari-able data rate that depends directly on the change in the scene. An example of this stream is shown in Figure 1.

This DVS operating paradigm results in advantageous characteristics, in terms of high time resolution, low data redundancy and power consumption and a very high dynamic range, which can be very useful in outdoor measurement scenarios like monitoring or autonomous driving.

The output of a Dynamic Vision Sensor is fundamentally different from a standard camera due to the described operation paradigm (synchronous and dense frame vs. asynchronous sparse event stream). Therefore, well-established computer vision approaches are not direct and natively applicable. In this work, we evaluate different event representations and deep learning networks to generate a multi-class semantic segmentation of DVS event data.

For this challenge, we consider variants of converting the DVS stream into single or multi-channel images, a 3D voxelization approach as well as the direct interpretation of the event stream as a 3D event point cloud. We summarize our main contributions as follows:
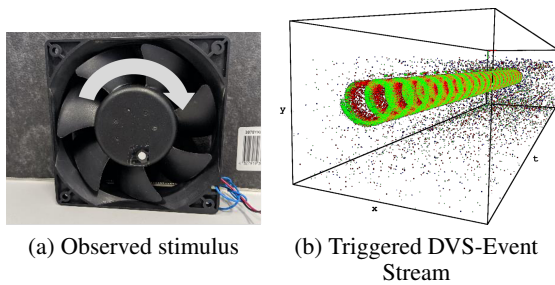
[a] https://orcid.org/0000-0001-5504-8472

(a) Observed stimulus   (b) Triggered DVS-Event Stream

Figure 1: Visualization of DVS output concept (event polarity is color-coded; "on" in green and "off" in red).

- consideration of 3D DVS space-time event cloud processing (Wang et al., 2019) and the extension of (Bolten et al., 2022) to another dataset in the application field of autonomous driving

- a systematic comparison of single-channel and high-multichannel 2D event representation, as well as 3D event stream voxelization

- and evaluation of a UNet (Ronneberger et al., 2015) network structure to generate a semantic segmentation on these representations.

The pre-processed datasets as well as the generated network predictions are available for download[1] to support further comparisons and research.

The rest of this paper is structured as follows. Section 2 outlines related work on semantic segmentations of neuromorphic event data. The evaluated event representations and deep learning approaches are explained in Section 3. Section 4 presents the used datasets, the performed data pre-processing, the training configurations and summarizes the evaluation results we obtained. Finally, a brief summary is provided in Section 5.

2 RELATED WORK

Although Dynamic Vision Sensors are a relatively new type of sensor technology, they are already being used in a variety of applications. For example, this includes real-time vibration measurements and control applications (Dorn et al., 2017) related to industrial applications, applications in the context of autonomous driving (Chen et al., 2020) or the use in space surveillance applications (McMahon-Crabtree and Monet, 2021). The goal of this work is to derive a semantic segmentation of the DVS event stream. This means that a object class label will be assigned to each DVS event.

In (Sekikawa et al., 2019) the authors have introduced the so-called *EventNet*. It is as neural network designed for the processing of asynchronous event streams in an event-wise manner which is capable to produce a semantic segmentation. Their approach is based on an adaption of a single PointNet structure (Qi et al., 2017a) which is made real-time capable by recursive processing of events and precomputed look-up tables. By design it is not capable to extract hierarchical features from the data. For this reason, and because we are not targeting real-time capability, we did not consider this approach further. Instead, we examine PointNet's successor, PointNet++ in its vanilla form.

The *EvDistill* approach presented by (Wang et al., 2021) is based on a student-teacher network design to overcome the hurdle of missing large-scale, qualitatively labeled datasets. A teacher network is trained on large-scale, labeled image data where the student network learns on unlabeled and unpaired event data by knowledge distillation. Because not all datasets considered in our work provide classical images (source modality) for the teacher, we have not considered this approach further.

An Xception based encoder-decoder architecture is used in *EV-SegNet* by (Alonso and Murillo, 2019) to obtain a semantic segmentation. For this purpose, a dense 6-channel 2D frame representation of the event stream is used. They also provide a labeled dataset from the autonomous driving domain. We use this dataset and compare our UNet-based results with their results.

In (Bolten et al., 2022) an evaluation of a semantic event-wise segmentation utilizing different data scaling variations and network configurations based on PointNet++ is presented. In our work, we extend this comparison to another dataset. Furthermore, we consider more event representations and replace their MaskRCNN based 2D reference processing to UNet based approaches.

In the literature, DVS event streams are often processed by converting them to classic 2D frame representations, which are then further processed using well-known off-the-shelf computer vision techniques. For example, frame conversion is performed in (Chen et al., 2019; Jiang et al., 2019; Wan et al., 2021), and then a Yolo-based approach is used. Furthermore, a variety of other possible event representations have developed. A larger set of these representations will be considered and examined in this study. Therefore, further details as well as literature references are given in the following Subsection 3.1.
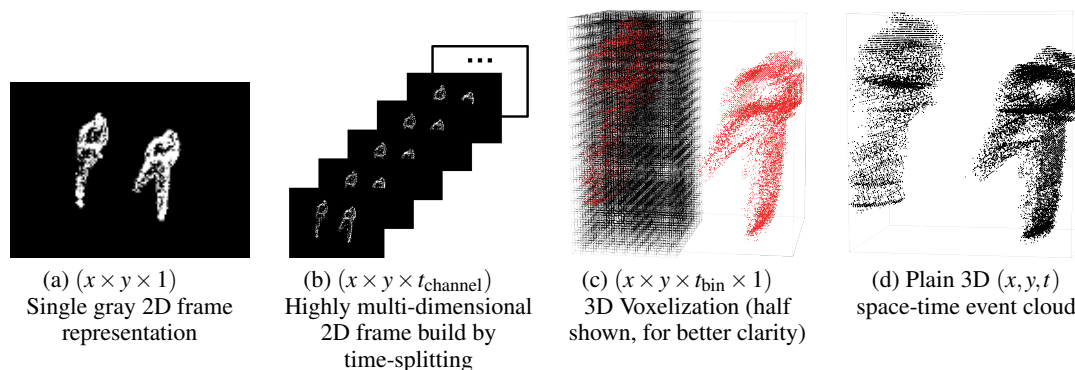
---

[1]http://dnt.kr.hsnr.de/DVS-UNetSemSeg/

(a) $(x \times y \times 1)$
Single gray 2D frame representation

(b) $(x \times y \times t_{\text{channel}})$
Highly multi-dimensional 2D frame build by time-splitting

(c) $(x \times y \times t_{\text{bin}} \times 1)$
3D Voxelization (half shown, for better clarity)

(d) Plain 3D (x, y, t)
space-time event cloud

Figure 2: Graphical rendering of the considered basic event representations ideas.

3 PROPOSED METHOD

The DVS event representations used in this work as well as the 2D and 3D deep learning network structures are presented and outlined in the following.

3.1 Event Representations

The event output stream from a Dynamic Vision Sensor is often converted into alternative representations for processing. In our work, we consider the subsequent 2D as well as 3D representations and compare the achieved results in the subsequent processing.

2D Frame Representation: The DVS event stream is converted to classic 2D frame representations by projection along the time axis. Typically, either a fixed time window or a fixed number of events is considered to construct the frame (Liu and Delbrück, 2018). For this conversion, there are a variety of encoding rules that also aim to consider the time resolution included in the DVS stream (Lagorce et al., 2017; Mitrokhin et al., 2018).

As a baseline for comparing the following encodings, we consider only the binary projection of the event stream as a pure 2D frame encoding in this work (compare with Figure 2a), resulting in a representation of shape $(x \times y \times 1)$.

Multi-Channel 2D Frame Representation: Within this projection of the DVS stream, classical 3-channel RGB images could also be generated, e.g. by color coding of the event polarities. In addition, there are also various other approaches that encode different aspects of the DVS stream in non-intuitive multichannel images. The "Merged-Three-Channel" representation defined in (Wan et al., 2021) is an example of such an encoding. In this representation, information about the event

frequency, the timestamps and continuity are represented in individual image channels.

Within our work, we aim to better represent and exploit the temporal context of the DVS stream. Therefore, we consider as an input encoding a highly multidimensional representation of the DVS data. Here, the time component of the signal is separated and stored in many channels during projection, resulting in a representation of shape $(x \times y \times t_{\text{channel}})$. A visualization is given in Figure 2b.

3D Voxel-Grid Representation: Another approach to maintain and better preserve the high temporal resolution of the DVS event stream is the interpretation as a 3D spatio-temporal volume. Voxelization of this volume encodes the distribution of events within the spatio-temporal domain (Zhu et al., 2019; Chaney et al., 2019). This type of representation is also often used as an intermediate encoding to convert the event stream into other forms, such as graphs (Deng et al., 2022).

We form time voxels per pixel in our work, as we discretize the time dimension per pixel into t_{bin} bins to include and consider fine spatial structures. This discretization leads to a data representation of shape $(x \times y \times t_{\text{bin}} \times 1)$ which encodes the occurrence of an event per voxel. A visualization is given in Figure 2c.

3D (x, y, t) Space Time Event Cloud Representation: In the previously described representation as a voxel grid, the sparsity of the event stream is lost in the encoding. This property and the high resolution of the time information is preserved when interpreting the DVS data as a 3D (x, y, t) space-time event cloud. In this way, the spatio-temporal information is directly encoded as geometric neighborhood information (compare with Figure 2d).

(a) Farthest Point Sampling (input from Figure 2d)   (b) Subset of Ball-Query Neighborhood selections   (c) Abstracted Event Set of Layer 1   (d) Abstracted Event Set Layer 2
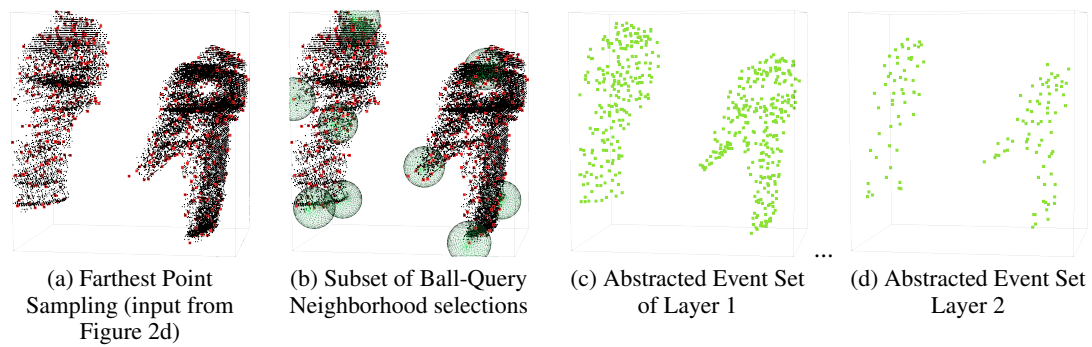
Figure 3: Summary of PointNet++ processing concept.

By applying point set processing methods, such as PointNet++ (Qi et al., 2017b), DVS data can be processed directly (Wang et al., 2019; Sekikawa et al., 2019; Mitrokhin et al., 2020; Bolten et al., 2022).

3.2 Network Architectures

In this work we compare 3D and 2D event representations and processing approaches using the following deep learning networks:

PointNet++. Hierarchical feature learning

PointNet++ (Qi et al., 2017b) learns a spatial encoding of point cloud data. For this purpose, the input data is hierarchically divided and summarized. By the respective application of a simple PointNet (Qi et al., 2017a) as a feature extractor, local and global features are built and combined. This results finally in a representation of the entire point cloud.

This hierarchical process is realized by so called *Set Abstraction Layers (SA)* of the network. First representative centroid points of local regions are selected by a farthest point sampling (Figure 3a). Subsequently local neighboring points are selected around these centroids. By default, this is performed via a ball query which finds an upper limited set of points within a defined radius (Figure 3b). The extracted pattern feature vectors of these local regions will be geometrically represented by the centroid coordinates (Figure 3c for the first and 3d for the subsequent second layer). This approach to create common structure partitions allows sharing the weights of the feature extractors per network layer. This leads to relatively small networks.

In the case of semantic segmentation, the resulting features are finally interpolated by a *Feature Propagation Layer (FP)* to produce point-wise values.

UNet. Convolutional Networks for Biomedical Image Segmentation

The UNet architecture (Ronneberger et al., 2015) has its origin in medical image segmentation, but was successfully applied to a various field of applications (Pohle-Fröhlich. et al., 2019; McGlinchy et al., 2019; Liu and Qian, 2021). It is a convolutional neural network that produces a precise pixel-by-pixel segmentation.

The architecture follows a division into an encoder and a decoder part. Within the encoder, spatial resolution is reduced by convolution and max-pooling, while the number of feature channels is increased. This extracts high-resolution and deep features about the context. In the second part, the decoder, the original resolution is restored by upsampling. By increasing the resolution of the output in this way, the decoder learns to create an output with precise localization. UNet architecture combines the feature channels from this down and upsampling by skip connections, allowing the network to propagate and combine context and localization information.

The visualization of an example configuration is given in Figure 5.

4 EXPERIMENTS

Initially, the used datasets are introduced and the performed data preprocessing is summarized. In the following, the hyper-parameters and the specific network layer and training configurations are presented. Subsequently the description of the used metric, as well as the achieved results, including a brief discussion and summary is given.

4.1 Datasets

In comparison to classical frame-based computer vision, there is currently a significantly lower number of event-based datasets available. This is particularly evident with the requirement for annotations at the level
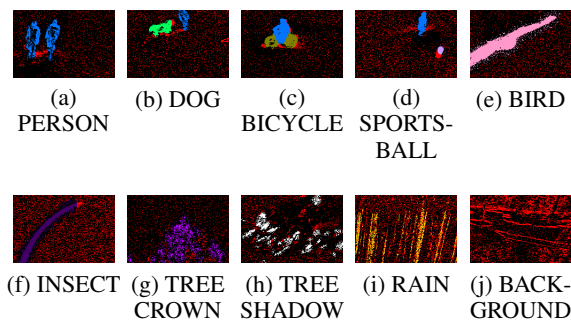
Figure 4: False-color DVS-OUTLAB class examples (reproduced from (Bolten et al., 2022) with permission from the authors).

of semantic segmentation.

Most large-scale datasets like *GEN1* (de Tournemire et al., 2020) or even smaller datasets (Miao et al., 2019) contain object labels only at the level of provided bounding boxes to achieve an object detection, but do not provide labels for a semantic segmentation.

We therefore limited our comparison to the two datasets below, which provide those annotations in a multi-class scenario.

DVS-OUTLAB: This dataset (Bolten et al., 2021) contains recordings of a DVS-based long-time monitoring of an urban outdoor place. For this purpose, three CeleX-IV sensors (Guo et al., 2017) were used. These recordings offer a total spatial resolution of 768×512 pixels.

The dataset contains semantic label annotations for about 47k regions of interest, separated into 70/15/15% sets for test, train and evaluation. Each region of interest with a spatial size of 192×128 pixels contains events and labels for a sequence of 60ms length of the underlying DVS event stream.

The labeling takes 10 different classes into account, including different object classes, as well as environmental noise originating from the outdoor-setup of the measurement (compare with Figure 4). The labels are provided on a per event-basis.

Subset of DDD17 Sequences: The authors of the Ev-SegNet approach (Alonso and Murillo, 2019) published with their work a subset of the *DDD17* dataset (Binas et al., 2017) extended by semantic labels.

The DDD17 dataset contains sequences of recordings obtained from a moving car in traffic (compare to Figure 6). These recording were taken with a DAVIS346B Dynamic Vision Sensor, offering a spatial resolution of 346×260 pixels. The data was cropped to 346×200 pixels, as the lower 60-pixel rows included the dashboard of the car.

The dataset contains 15950 sequences for training and 3890 for testing, each corresponding to a 50ms section of the event stream. For these sequences, the authors automatically generated pixel-wise semantic labels based on gray-scale images from the DAVIS sensor by applying a CNN. Thereby six different classes were considered: (1) construction/sky, (2) objects (like street signs or light poles), (3) nature (like trees), (4) humans, (5) vehicles and (6) street. These labels are provided as dense 2D frames.

4.2 Data Preprocessing

The following pre-processing was performed to prepare the datasets and generate the presented event representation:

Subset of DDD17 Sequences: The DVS event data was published by (Alonso and Murillo, 2019) in the form that only a 2D frame representation of the event stream is directly available. Furthermore, the generated labels are also only available in the form of 2D frames. Thus, they are not directly usable for the generation of our proposed multi-channel, voxel or 3D space-time event cloud representation.

Therefore, utilizing the native DDD17 event stream recordings, we first propagated the labels of the EvSegNet subset back to the original event stream. This results in annotations per event in the form of $(x, y, t, p, \text{label})$. For each 50ms of the event stream the corresponding 2D label was transferred to all underlying events at the same spatial position within this time window.

DVS-OUTLAB: The labeling of this dataset is already available in the form of a semantic annotation per event. Therefore, no adaption of the label representation was necessary.

The 3D (x, y, t) space-time event cloud representation is built natively direct from the event stream. For the remaining 2D and voxel representations, an intermediate numpy-array was calculated. For this purpose, a 3D voxel histogram was generated per pixel-position, splitting the time axis into 64 components (t_{channel} and t_{bin} in Figure 2). The labels were transformed into an equivalent voxel form. By applying numpy-operations a convenient transformation into the proposed 2D representations is possible[2]. This pre-processed data is available for download.

---

[2]The fast transformation from 3D voxels to the proposed 2D frame representations could be done by simple reshape and/or amax operations.

Figure 5: UNet 2D configuration example for DVS-OUTLAB dataset (10 object classes plus the "void" for regions without any event).

Table 1: PointNet++ configuration summary (compare to syntax used in (Qi et al., 2017b)).

DVS-OUTLAB	PNet++(4096, 3L)	$SA(2048, 9.6, [32, 32, 64]) \rightarrow SA(256, 28.8, [64, 64, 128]) \rightarrow$ $SA(16, 76.8, [128, 128, 256]) \rightarrow FP([256, 256]) \rightarrow FP([256, 128]) \rightarrow$ $FP([128, 128, 128, 128, 10])$
DDD17	PNet++(4096, 5L)	$SA(2048, 17.3, [32, 32, 64]) \rightarrow$
	PNet++(8192, 5L)	$SA(4096, 17.3, [32, 32, 64]) \rightarrow$
	followed by	$SA(1024, 34.6, [64, 64, 128]) \rightarrow SA(256, 69.2, [128, 128, 256]) \rightarrow$ $SA(64, 103.8, [256, 256, 512]) \rightarrow SA(16, 138.4, [512, 512, 1024]) \rightarrow$ $FP([256, 256]) \rightarrow FP([256, 128]) \rightarrow FP([256, 256]) \rightarrow$ $FP([256, 128]) \rightarrow FP([128, 128, 128, 128, 6])$

The division into 64-time components is selected so that in the subsequent logic of UNet processing, the choice of the input dimension as a power of two leads to integer dimensions in the downsampling and upsampling logic. Furthermore, the spatial resolution was extended by zero-padding into quadratic inputs for the UNet processing. This results in 192×192 pixel resolution for the DVS-OUTLAB data and to 346×346 pixel for the DDD17 subset. In the representations, the presence or absence of events per spatial position is encoded by the numerical value 255 or 0.

Moreover, we created and tested two variations for each of the datasets. One plain version, that includes all events and one that was spatio-temporal pre-filtered to reduce included sensor background noise and to estimate the effects of noise reduction for the semantic processing. For this purpose, a time filter was applied to remove all events that were not supported by another event at the same (x, y) coordinate within the previous 10ms. This type of filter has shown a reasonable trade-off between noice reduction and preservation of object events (compare to performed evaluation in (Bolten et al., 2021)).

4.3 Training

The following network layer configurations and training hyper-parameters were used in our experiments:

PointNet++. For the PointNet++ training configura-

tion we followed the selected hyper-parameters from (Bolten et al., 2022). This results in using the Adam optimizer with a learning rate of 0.001 and an exponential decay rate of 0.99 every 200.000 trainings steps. The batch size was set to 16 space time event clouds.

For the DVS-OUTLAB dataset we follow also their data patching and scaling scheme (S^T_{native}), layer depth and set abstraction configuration. In case of the DDD17 data, we adapted the network configuration due to the larger spatial input dimension (346×200 pixel vs 192×128 pixel per region) to address the resulting higher event count. Additional we trained and tested two Point-Net++ configurations with a previous subsampling to 8192, respectively 4096 events.

The specific PointNet++ configuration used for training is summarized in Table 1.

UNet. The UNet trainings were carried out utilizing an Adam optimizer with a learning rate of 0.001 and an exponential decay with a rate of 0.99 after each epoch. The batch size was set to 6 samples. A sparse categorical cross entropy weighted by the class occurrence frequency was chosen as the loss function to address the class imbalances included in the datasets.

In all performed UNet experiments the model is built with a depth of 4 layers and a number of 16 filters in the first block, which are multiplied by 2 in each subsequent block. The kernel size in

(a) 2D Labels

(b) Corresponding Events
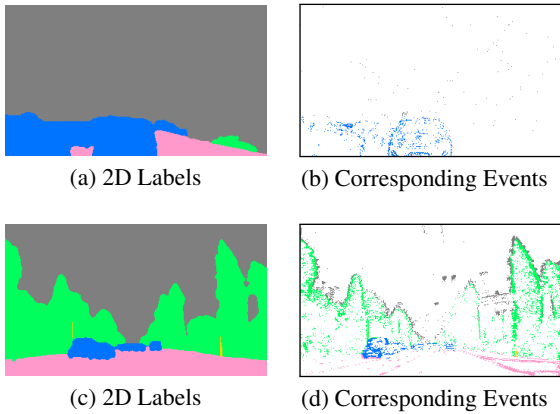
(c) 2D Labels

(d) Corresponding Events

Figure 6: Ev-SegNet DDD17 subset: dense label compared to sparse event stream.

the 2D or respective 3D-convolutions were set to three. A used 2D UNet example configuration is shown in Figure 5 as reference.

4.4 Metric

In the literature, and this is also the case for event cameras as in (Alonso and Murillo, 2019), the evaluation is often performed on the basis of dense 2D frames. For each 2D pixel of the annotation the corresponding pixel value of the network prediction is considered and compared. But this type of evaluation ignores the basic property of a Dynamic Vision Sensor that the produced event stream is spatially sparse. This is clearly illustrated by Figure 6, which shows two scenes from the DDD17 data subset. In Subfigure (a, b) a scene with limited or without movement is shown, whereas Subfigure (c, d) displays a scenes of higher speed. Within a slow scene only a few events are triggered and even with faster movements, there are many areas where no or only a few events were triggered as well.

Networks operating on a sparse representation, such as the used PointNet++, cannot predict results at positions where no events were triggered. Therefore, a proper comparison on this dense 2D basis is not possible. Furthermore, this type of 2D comparison ignores the fact, that at one spatial (x, y) position multiple events could have been triggered within the selected time window. Therefore, we consider in our evaluation only spatial positions where events were triggered. Furthermore, the number of triggered events for each predicted label is also taken into account.

In contrast to PointNet++, for the UNet based processing approaches, it is possible that a class prediction occurs at a spatial position where no DVS

Table 2: Weighted-Avg F1 results on DVS-OUTLAB dataset.

Network	Back-ground	Objects	Env-Influences	Over-All
(a) PointNet++ reference results				
PNet(4096, 3L)	**0.968**	0.816	**0.853**	**0.936**
(b) Unfiltered UNet results				
UNet 2D	0.951	0.842	0.764	0.902
UNet 2D 64ch	0.958	0.847	0.780	0.912
UNet 3D Voxel	0.941	0.843	0.775	0.895
(c) Spatio-temporal filtered (time 10ms) UNet results				
UNet 2D	0.925	0.838	0.757	0.868
UNet 2D 64ch	0.938	**0.850**	0.826	0.897
UNet 3D Voxel	0.928	0.843	0.809	0.883

event has been triggered (the "void" background). To take account for this we perform the following simple post-processing before evaluation:

If an object class prediction is made but no event is present (pred $\neq$ void $\wedge$ event $= \emptyset$), this prediction is interpreted as void and ignored. In case that no object class prediction is made but an event is present (pred $=$ void $\wedge$ event $\neq \emptyset$), this prediction is re-interpreted and considered as the dominating background class for evaluation (class background for DVS-OUTLAB, construction/sky for DDD17).

Considering the number of triggered DVS events we then calculate the F1 score, which is described as the harmonic mean of precision and recall:

$$\text{F1-Score} = \frac{TP}{TP + 0.5 \cdot (FP + FN)}$$

To summarize the results, we also calculated weighted averages taking the number of each class's support into consideration (Weighted-Avg F1).

For a fair comparison of the 3D and 2D methods (different counts of predictions and therefore a higher number of possible errors in case of higher output dimension) we equalize all generated predictions for evaluation. The 3D predictions are projected along the t-axis by considering the most frequent prediction at each spatial position for comparison with the 2D results.

4.5 Evaluation

The evaluation results of the PointNet++ processing, as well as the results for the different event representations in combination with the 2D and 3D UNet

processing for the DVS-OUTLAB dataset are summarized in Table 2. The PointNet++ based processing achieves the better segmentation results on this dataset compared to the 2D or 3D voxel UNet processing. This is consistent to the PointNet++ and 2D MaskRCNN comparison presented in (Bolten et al., 2022).

The summary of results for the subset of labeled DDD17 dataset sequences is given in Table 3. On this dataset, PointNet++ processing achieves weaker results in contrast to the UNet variations and the dataset authors' Ev-SegNet reference. A noticeable difference exists here in the results of the class "Objects" in the PointNet++ based processing. This class of the dataset contains, for example, lampposts, street signs or traffic lights. Although the PointNet++ configurations used here were adjusted in the number of points to be considered in the input cloud and the first SetAbstraction layer, as well as the layer count itself, this suggests that such fine details were not fully captured. Due to the high number of triggered events in the autonomous driving context of this dataset (compared to a static sensor in DVS-OUTLAB monitoring) and the larger spatial input (346×200 pixel vs 192×128 pixel), the encode/decoder approach to UNet processing seems to have an advantage.

The PointNet++ processing, on the other hand, relies on considering sufficient representative events selected by farthest point sampling and corresponding neighborhood formation. Please compare to Figure 3, especially (a) and b, which summarizes the basic idea of PointNet++ processing.

However, when considering the overall results on the DDD17 subset, the given quality of the ground truth label must be considered. These labels were generated by (Alonso and Murillo, 2019) through an automatic processing. Out of a total of nearly 12 hours of material from the DDD17 dataset, about 15 minutes were labeled in this way, and the GT labels obtained are not completely accurate and consistent over time. Figure 7 gives an example of included artifacts in the GT labels using two examples that are separated by a short period of time. The annotations of the included traffic sign and the tree (marked by red arrows) varies, although, for example, the UNet predictions are correct.

Overall, across both datasets, it can be observed that UNet-based processing achieves better results on unfiltered raw data than on spatio-temporal prefiltered data. In general, an improvement of the UNet based results can be observed with an increase of the dimensionality of the event representation used. Whereas the use of the 3D voxel grid brings only minor differences in comparison.

Table 3: Results on subset of DDD17 dataset.

Network	Construction	Objects	Nature	Human	Vehicle	Street	Macro-Avg	Weighted-Avg

(a) Ev-SegNet baseline (Alonso and Murillo, 2019), metric recalculated to match proposed evaluation

Network	Construction	Objects	Nature	Human	Vehicle	Street	Macro-Avg	Weighted-Avg
Ev-SegNet	**0.916**	0.229	0.712	**0.670**	0.850	**0.727**	**0.696**	**0.876**

(b) PointNet++ results

Network	Construction	Objects	Nature	Human	Vehicle	Street	Macro-Avg	Weighted-Avg
PNet(8192, 5L)	0.842	0.088	0.516	0.398	0.743	0.619	0.534	0.771
PNet(4092, 5L)	0.840	0.103	0.521	0.464	0.748	0.600	0.546	0.766

(c) Unfiltered UNet results

Network	Construction	Objects	Nature	Human	Vehicle	Street	Macro-Avg	Weighted-Avg
UNet 2D	0.886	0.266	0.686	0.577	0.835	0.703	0.659	0.829
UNet 2D 64ch	0.895	0.285	0.723	0.533	0.849	0.723	0.668	0.843
UNet 3D Voxel	0.898	**0.301**	**0.729**	0.572	0.847	0.719	0.678	0.843

(d) Spatio-temporal filtered (time 10ms) UNet results

Network	Construction	Objects	Nature	Human	Vehicle	Street	Macro-Avg	Weighted-Avg
UNet 2D	0.882	0.265	0.673	0.557	0.846	0.660	0.647	0.826
UNet 2D 64ch	0.896	0.289	0.713	0.590	**0.862**	0.681	0.672	0.846
UNet 3D Voxel	0.895	0.296	0.713	0.568	0.858	0.680	0.668	0.843

(a) Ground truth vs prediction at t_i

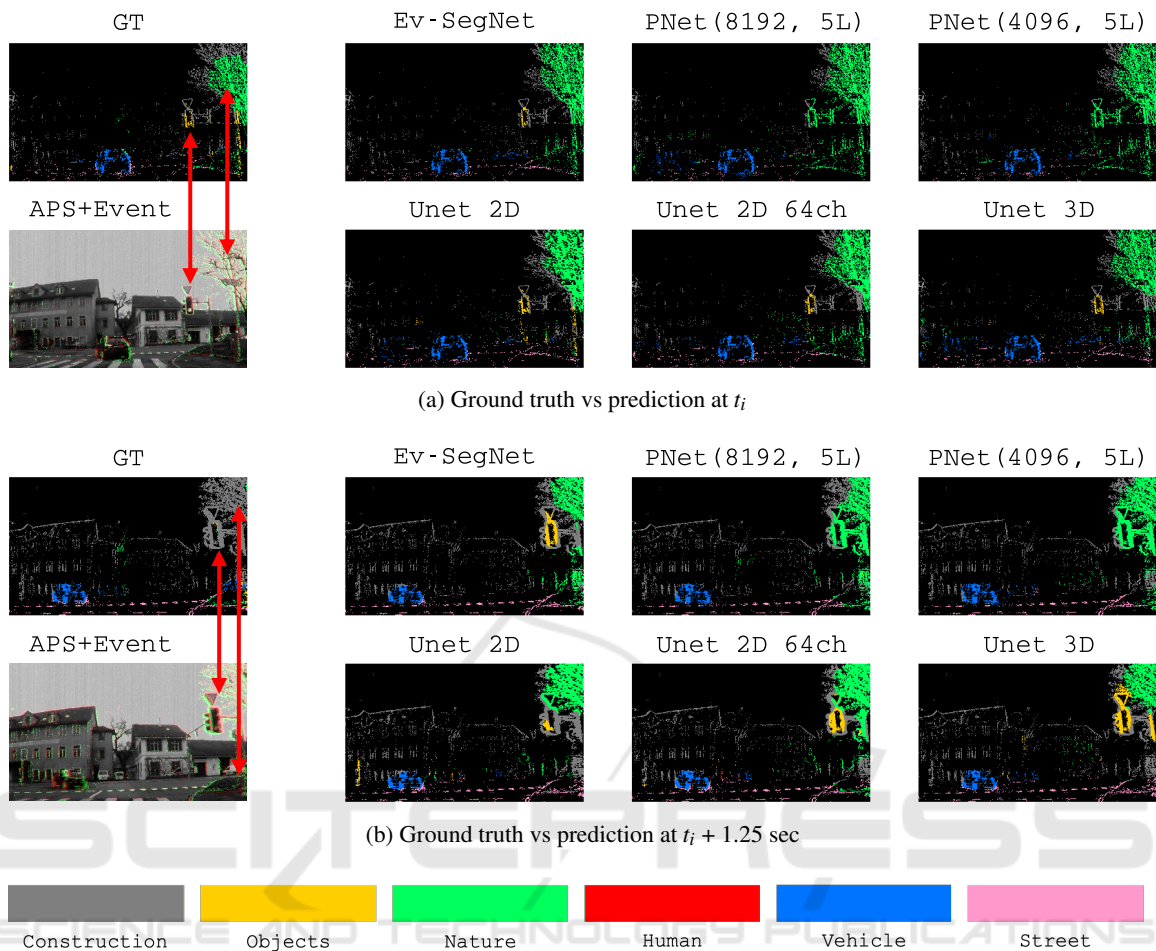(b) Ground truth vs prediction at $t_i + 1.25$ sec

Figure 7: Visualization of GT labeling quality of DDD17 subset from (Alonso and Murillo, 2019) and predictions of trained networks. Note the inconsistent GT labeling of the marked traffic sign and trees between the timestamps shown. (best viewed in color and digital zoomed)

5 CONCLUSION

The improvement in the UNet prediction quality using the highly multi-channel event representation over the single 2D frame variant indicates a benefit of the more complex representation. Whereas the use of 3D voxel grids also achieves good results (compare exemplarily with the results shown in Figure 7). Unfortunately, the associated UNet network structure is larger due to the 3D convolutions and therefore slower for inference.

The sensor property of a DVS to produce an output stream that is spatially sparse becomes particularly clear when statistically examining the used voxel representation. Over the two complete data sets, only about 0.45% of the voxels are populated by an event (corresponding 99.55% of the voxels considered are empty). A classical UNet based on simple 3D convolutions already achieves good results on this voxel representation - and this despite the fact that the application of these convolutions quickly increases and "blurs" the set of active (non-zero) features. Therefore the usage of sparse convolutions (Graham and van der Maaten, 2017) and the adaption of the UNet network into a sparse voxel network for semantic segmentation (Graham et al., 2018; Najibi et al., 2020) is an interesting task for further work.

In summary, differences between the evaluated network structures have also emerged. A PointNet++ based processing is better suited for scenes without ego-motion of the sensor, whereas for moving sensors and the inclusion of larger spatial input patches a UNet based processing has shown advantages.

ACKNOWLEDGEMENTS

REFERENCES

Alonso, I. and Murillo, A. C. (2019). EV-SegNet: Semantic Segmentation for Event-Based Cameras. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1624–1633.

Binas, J., Neil, D., Liu, S.-C., and Delbruck, T. (2017). DDD17: End-To-End DAVIS Driving Dataset. In *ICML'17 Workshop on Machine Learning for Autonomous Vehicles (MLAV 2017)*.

Bolten, T., Lentzen, F., Pohle-Fröhlich, R., and Tönnies, K. D. (2022). Evaluation of Deep Learning based 3D-Point-Cloud Processing Techniques for Semantic Segmentation of Neuromorphic Vision Sensor Event-streams. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP*, pages 168–179. INSTICC, SciTePress.

Bolten, T., Pohle-Fröhlich, R., and Tönnies, K. D. (2021). DVS-OUTLAB: A Neuromorphic Event-Based Long Time Monitoring Dataset for Real-World Outdoor Scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1348–1357.

Chaney, K., Zhu, A. Z., and Daniilidis, K. (2019). Learning Event-Based Height From Plane and Parallax. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1634–1637.

Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., and Knoll, A. (2020). Event-Based Neuromorphic Vision for Autonomous Driving: A Paradigm Shift for Bio-Inspired Visual Sensing and Perception. *IEEE Signal Processing Magazine*, 37(4):34–49.

Chen, G., Cao, H., Ye, C., Zhang, Z., Liu, X., Mo, X., Qu, Z., Conradt, J., Röhrbein, F., and Knoll, A. (2019). Multi-Cue Event Information Fusion for Pedestrian Detection With Neuromorphic Vision Sensors. *Frontiers in Neurorobotics*, 13:10.

de Tournemire, P., Nitti, D., Perot, E., Migliore, D., and Sironi, A. (2020). A Large Scale Event-based Detection Dataset for Automotive. *arXiv*, abs/2001.08499.

Deng, Y., Chen, H., Liu, H., and Li, Y. (2022). A Voxel Graph CNN for Object Classification With Event Cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1172–1181.

Dorn, C., Dasari, S., Yang, Y., Kenyon, G., Welch, P., and Mascareñas, D. (2017). Efficient full-field operational modal analysis using neuromorphic event-based imaging. In *Shock & Vibration, Aircraft/Aerospace, Energy Harvesting, Acoustics & Optics, Volume 9*, pages 97–103. Springer.

Graham, B., Engelcke, M., and van der Maaten, L. (2018). 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. *CVPR*.

Graham, B. and van der Maaten, L. (2017). Submanifold Sparse Convolutional Networks. *arXiv preprint arXiv:1706.01307*.

Guo, M., Huang, J., and Chen, S. (2017). Live demonstration: A 768 × 640 pixels 200meps dynamic vision sensor. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–1.

Jiang, Z., Xia, P., Huang, K., Stechele, W., Chen, G., Bing, Z., and Knoll, A. (2019). Mixed Frame-/Event-Driven Fast Pedestrian Detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8332–8338.

Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B. (2017). HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1346–1359.

Liu, M. and Delbrück, T. (2018). Adaptive Time-Slice Block-Matching Optical Flow Algorithm for Dynamic Vision Sensors. In *29th British Machine Vision Conference (BMVC)*.

Liu, M. and Qian, P. (2021). Automatic Segmentation and Enhancement of Latent Fingerprints Using Deep Nested UNets. *IEEE Transactions on Information Forensics and Security*, 16:1709–1719.

McGlinchy, J., Johnson, B., Muller, B., Joseph, M., and Diaz, J. (2019). Application of UNet Fully Convolutional Neural Network to Impervious Surface Segmentation in Urban Environment from High Resolution Satellite Imagery. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3915–3918.

McMahon-Crabtree, P. N. and Monet, D. G. (2021). Commercial-off-the-shelf event-based cameras for space surveillance applications. *Appl. Opt.*, 60(25):G144–G153.

Miao, S., Chen, G., Ning, X., Zi, Y., Ren, K., Bing, Z., and Knoll, A. (2019). Neuromorphic Vision Datasets for Pedestrian Detection, Action Recognition, and Fall Detection. *Frontiers in Neurorobotics*, 13:38.

Mitrokhin, A., Fermüller, C., Parameshwara, C., and Aloimonos, Y. (2018). Event-Based Moving Object Detection and Tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6895–6902.

Mitrokhin, A., Hua, Z., Fermüller, C., and Aloimonos, Y. (2020). Learning visual motion segmentation using event surfaces. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14402–14411.

Najibi, M., Lai, G., Kundu, A., Lu, Z., Rathod, V., Funkhouser, T., Pantofaru, C., Ross, D., Davis, L. S., and Fathi, A. (2020). DOPS: Learning to Detect 3D Objects and Predict Their 3D Shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pohle-Fröhlich., R., Bohm., A., Ueberholz., P., Korb., M., and Goebbels., S. (2019). Roof Segmentation based on Deep Neural Networks. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,*, pages 326–333. INSTICC, SciTePress.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 5105–5114, Red Hook, NY, USA. Curran Associates Inc.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

Sekikawa, Y., Hara, K., and Saito, H. (2019). EventNet: Asynchronous Recursive Event Processing. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3882–3891.

Wan, J., Xia, M., Huang, Z., Tian, L., Zheng, X., Chang, V., Zhu, Y., and Wang, H. (2021). Event-Based Pedestrian Detection Using Dynamic Vision Sensors. *Electronics*, 10(8:888).

Wang, L., Chae, Y., Yoon, S.-H., Kim, T.-K., and Yoon, K.-J. (2021). EvDistill: Asynchronous Events To End-Task Learning via Bidirectional Reconstruction-Guided Cross-Modal Knowledge Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 608–619.

Wang, Q., Zhang, Y., Yuan, J., and Lu, Y. (2019). Space-Time Event Clouds for Gesture Recognition: From RGB Cameras to Event Cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1826–1835.

Zhu, A. Z., Yuan, L., Chaney, K., and Daniilidis, K. (2019). Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 989–997.