Instance Segmentation of Event Camera Streams in Outdoor Monitoring Scenarios

Tobias Bolten[1][a], Regina Pohle-Fröhlich[1][b] 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, Instance Segmentation, Outdoor Environment.

Abstract: Event cameras are a new type of image sensor. The pixels of these sensors operate independently and asynchronously from each other. The sensor output is a variable rate data stream that spatio-temporally encodes the detection of brightness changes. This type of output and sensor operating paradigm poses processing challenges for computer vision applications, as frame-based methods are not natively applicable.

We provide the first systematic evaluation of different state-of-the-art deep learning based instance segmentation approaches in the context of event-based outdoor surveillance. For processing, we consider transforming the event output stream into representations of different dimensionalities, including point-, voxel-, and frame-based variants. We introduce a new dataset variant that provides annotations at the level of instances per output event, as well as a density-based preprocessing to generate regions of interest (RoI). The achieved instance segmentation results show that the adaptation of existing algorithms for the event-based domain is a promising approach.

1 INTRODUCTION

Event cameras, also known as Dynamic Vision Sensors (DVS) or silicon retinas, are a new type of image sensor. Unlike conventional frame-based image sensors, they operate completely asynchronously and independently per pixel. Following the biologically inspired ideas of neuromorphic engineering, only changes in brightness per pixel are detected and directly transmitted. The result is not a classical video frame captured at a fixed sampling frequency, but an output stream of variable data rate depending on the changes in the scene.

Each detected change in brightness above a defined threshold value results in a so-called output event and is transmitted immediately. For each output event, (a) its spatial (x,y)-position in the sensor array, (b) a very precise timestamp t of the triggering, and (c) the polarity p of the change (bright to dark and vice versa) are encoded. The technical operating paradigm of these sensors allows recordings with high temporal resolution and low data redundancy, while simultaneously offering a very high dynamic range. These are very important and advantageous factors for outdoor applications.

However, the sparse, unordered, and asynchronous output of these sensors poses challenges for processing in terms of classical computer vision approaches. In this work, we investigate the task of instance segmentation on event-based data in the context of outdoor surveillance recordings in order to gain deeper insight into the usage of the monitored areas. Additional challenges arise from unconstrained real-world factors, small object sizes, and occlusions. In summary, we contribute the first systematic evaluation of state-of-the-art deep learning approaches for instance segmentation, including different event encodings, to assess their suitability under these conditions.

The rest of this paper is structured as follows. Section 1.1 summarizes related work. Event data representations and instance segmentation networks are briefly described in Section 2. The datasets used and the preprocessing are introduced in Section 3. The results of the evaluation are discussed in Section 4. Supplemental material is available for download[1].

[a] https://orcid.org/0000-0001-5504-8472

[b] https://orcid.org/0000-0002-4655-6851

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

Bolten, T., Pohle-Fröhlich, R. and Tönnies, K.
Instance Segmentation of Event Camera Streams in Outdoor Monitoring Scenarios.
DOI: 10.5220/0012369100003660
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 3: VISAPP*, pages 452-463
ISBN: 978-989-758-679-8; ISSN: 2184-4321
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

1.1 Related Work

Segmentation is an important part of computer vision and is needed for a variety of tasks in scene understanding. Event-based research in this area is not as extensive as its frame-based counterpart. This is due to the novelty of the sensor technology itself.

Frame-Based Processing. The development of methods for traditional 2D frame-based processing is more advanced. Libraries such as *Detectron2* (Wu et al., 2019b) are available, providing state-of-the-art recognition and segmentation algorithms as well as pre-trained models. Basically, two different approaches can be distinguished here. In proposal-based approaches, objects are first detected using bounding box techniques and then segmented. A well-known example of this is *Mask R-CNN* (He et al., 2017).

On the other hand, the well-known *YOLO* family (Redmon et al., 2016; Jocher et al., 2023) directly predicts bounding boxes and class probabilities for objects in a single pass. Along with the use of pixel-level grouping or clustering techniques to form instances, such as (Xie et al., 2020; Xie et al., 2022; Wang et al., 2020), this provides proposal-free methods.

3D-Based Processing. The output stream from event cameras can also be interpreted as a three-dimensional (x,y,t) cloud. Therefore, instead of using (x,y,z) point clouds, 3D-based instance segmentation methods are also of interest in this context. Basically, 3D methods can also be distinguished into proposal-based and proposal-free approaches. The former decompose the segmentation problem into two sub-challenges: Detecting objects in 3D and refining the final object masks (Yang et al., 2019; Engelmann et al., 2020). The latter typically omit the detection part and try to obtain instances by clustering after semantic segmentation (e.g., following the assumption that instances should have similar features) (Zhao and Tao, 2020; Jiang et al., 2020; Chen et al., 2021). Typically, the processing here is point-based or voxel-based.

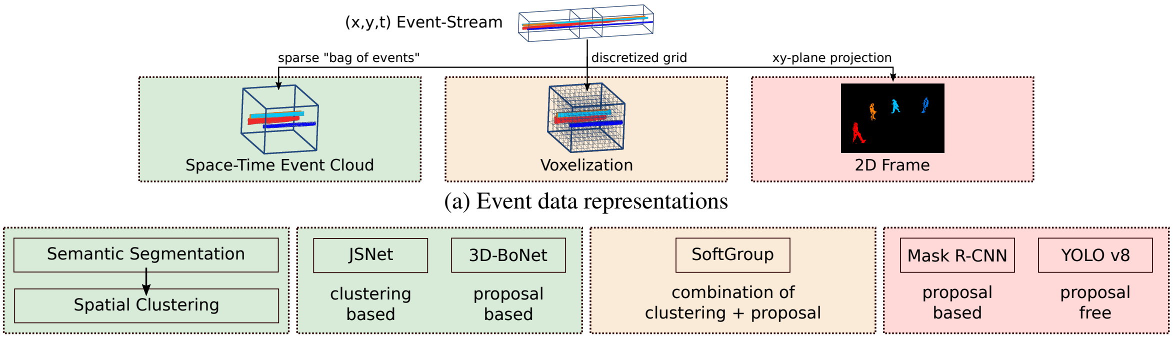
Event-Based Processing. Event clustering can be used to separate objects in simple scenes that typically do not include sensor ego-motion due to the event camera operating principle (Schraml and Belbachir, 2010; Rodríguez-Gomez et al., 2020). For more complex and unstructured scenes, clustering approaches also exist (Piątkowska et al., 2012).

In (Barranco et al., 2015), the scene is decomposed based on categorized object contours to achieve layer segmentation. In contrast, (Stoffregen and Kleeman, 2018) segments the scene into structures that move at the same velocity. Generally, event-based motion segmentation approaches can be used to distinguish objects by assigning events to objects with independent motion (Vasco et al., 2017; Mitrokhin et al., 2018; Stoffregen et al., 2019; Mitrokhin et al., 2020; Zhou et al., 2021). However, these approaches have in common that no semantic class categorization is performed for the detected objects.

Semantic segmentation fills this shortcoming. *EvNet* (Sekikawa et al., 2019) is an asynchronous, fully event-based approach for this purpose. (Biswas et al., 2022) exploits features extracted from the event stream and simultaneously acquired grayscale images, while event-only processing is considered as part of the ablation study. Approaches that rely solely on the event stream to derive a semantic segmentation are given in (Bolten et al., 2022; Bolten et al., 2023b). Here, the processing is done based on point cloud or voxel grid representations, with well-known network structures like PointNet++ (Qi et al., 2017) or UNet (Ronneberger et al., 2015). However, most semantic segmentation approaches convert the event stream into dense frame representations, such as (Alonso and Murillo, 2019; Sun et al., 2022; Wang et al., 2021). Nevertheless, it is impossible to differentiate between spatially close or even occluded objects of the same class that are moving at nearly the same speed by motion or semantic segmentation. This is particularly relevant in the context of monitoring applications, such as a group of people.

The resulting challenge of *instance* segmentation has been largely unaddressed for event-based data. In the context of robotic grasping, there are first works towards this direction, fusing the modality of RGB frames with events (Kachole et al., 2023b), or deriving a panoptic segmentation by applying graph-based network processing (Kachole et al., 2023a).

To the best of the authors' knowledge, there is currently no prior work that adapts, applies, and evaluates off-the-shelf 2D frame or 3D-based instance segmentation approaches to the event-based vision domain. However, this represents a promising way to achieve instance segmentation in this domain.



(a) Event data representations

(b) Baseline Method (c) Point-based Methods (d) Voxel-based Method (e) Frame-based Methods

Figure 1: Overview of performed experiments.

2 PROPOSED METHODS

In the following, we introduce different encodings used for event stream representations in this study. In addition, we briefly outline the deep learning frameworks used for instance segmentation.

2.1 Event Data Representations

Commonly, the event data stream from Dynamic Vision Sensors is converted into alternative representations for processing. In this work we consider the construction of 3D point clouds, also called space-time event clouds, their voxelization and conversion into classical 2D frames for subsequent processing (see Figure 1a).

Space-Time Event Cloud. A temporal window of the continuous event stream directly forms an unordered point cloud, where each 3D point represents an event defined by its (x,y,t) coordinates. This preserves the sparsity and high temporal resolution of the signal and transforms it into a geometric description.

3D Voxel-Grid. The irregularity of the event clouds can be removed by voxelization and transformed into a regular 3D grid. The voxels encode the distribution of events within the spatio-temporal domain. The sparsity of the signal is lost in this transformation. The size of the voxel bins must be chosen application specific.

2D Frame Projection. Classic 2D frames are created by projecting events onto the xy plane. This results in a dense 2D grid of fixed size defined by the pixel resolution of the sensor. It allows direct processing using classical computer vision approaches. Since events are triggered by changes, the resulting images visually resemble edge images.

There are a variety of encodings described in the literature for this projection step. In this study, we consider the following two variants:

Polarity. Each frame pixel is defined by the polarity of the last event that occurred at the corresponding (x,y) pixel position. The polarity is directly encoded in the single color values `red` for decrease, `green` for increase in brightness (see Figure 2a).

Merged-Three-Channel (MTC). This encoding was proposed by Chen et al. in (Chen et al., 2019). It incorporates three different single-channel encodings, each addressing different attributes of the underlying event stream, to create an RGB false color image (see Figure 2b):

Red Channel. *Leaky-Integrate-And-Fire* neuron model to preserve information about temporal continuity,

Green Channel. *Surface-Of-Active-Events* as a time surface containing information about the direction and speed of object motion through its gradient, and

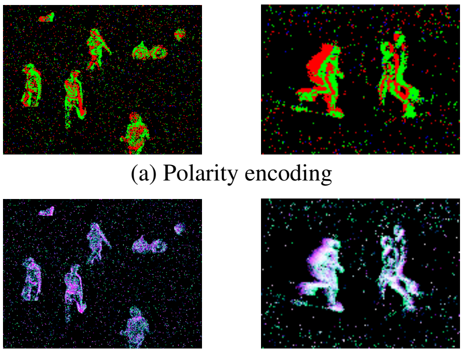
Blue Channel. *Triggering Frequency* to distinguish between noise and valid events.

These encodings are selected because they represent different levels of preserved information.

2.2 Instance Segmentation Networks

2.2.1 Point Cloud-Based Processing Methods (Figure 1c)

JSNet (Zhao and Tao, 2020) (2020): clustering-based processing
JSNet consists of four main components: a shared feature encoder, two parallel branch decoders, feature fusion modules for each decoder, and a


(a) Polarity encoding

(b) MTC encoding

Figure 2: Snippets of frame encoded events (best viewed in color).

joint instance and semantic segmentation (JISS) module. High-level semantic features are learned by PointNet++ (Qi et al., 2017) and PointConv (Wu et al., 2019a) architectures and are further combined with low-level features for more discriminative values. The JISS module transforms the semantic features into an instance embedding space, where instances are formed by applying a simple mean-shift clustering.

3D-BoNet (Yang et al., 2019) (2019): proposal-based processing
3D-BoNet is designed for a single-stage, anchor-free instance segmentation in 3D point clouds. It uses a PointNet++ (Qi et al., 2017) backbone to extract local and global features, followed by two branches: one for instance-level bounding box prediction and another for point-level mask prediction. The bounding box prediction branch is a key component, generating unique, unoriented rectangular bounding boxes without predefined spatial anchors or region proposals. The subsequent point-mask prediction branch uses these boxes and features to generate point-level binary masks for valid instances, distinguishing them from the background.

2.2.2 Voxel-Based Processing Method (Figure 1d)

SoftGroup (Vu et al., 2022) (2022): clustering and proposal-based
SoftGroup attempts to combine the strengths of proposal-based and grouping-based methods while addressing their limitations. First, a bottom-up stage uses a pointwise prediction network to generate high-quality object proposals by grouping based on soft semantic scores. This stage involves processing point clouds to generate semantic labels and offset vectors, which are then refined into preliminary instance proposals using a soft grouping module. Second, the top-down refinement stage refines the generated proposals by extracting corresponding features from the backbone. These features are employed to predict final results, including classes, instance masks, and mask scores.

2.2.3 Frame-Based Processing Methods (Figure 1e)

Mask R-CNN (He et al., 2017) (2017): proposal-based processing
Proposal-based processing is considered to be the baseline technique for frame-based instance segmentation (Sharma et al., 2022). Therefore, we included Mask R-CNN in our experiments.

Mask R-CNN consists of five key components. First, a backbone network for feature extraction, followed by a Region Proposal Network that generates potential object proposals. The RoIAlign layer ensures accurate spatial alignment for RoIs. The RoI head contains two sub-networks: one for classification and bounding box regression, and another for instance mask prediction. This architecture enables object detection, classification, bounding box refinement with detailed instance masks predictions.

YOLO v8 (Jocher et al., 2023) (2023): proposal-free processing
YOLO v8 is the latest version of the popular single shot detection method and aims to improve accuracy and efficiency over previous versions. A major change is that YOLO v8 is an anchor-free model, meaning that object centers are predicted directly instead of the offset from a known anchor box. This typically results in fewer predictions and better, faster non-maximum suppression.

3 DATASETS & PREPROCESSING

Compared to the more established domain of image-based computer vision, the range of event-based datasets is currently limited. In the following, we describe the datasets used in our experiments and the preprocessing steps applied.

3.1 Datasets

Annotations at the level of semantic or even instance segmentation are only available for a few event-based datasets. For an example in autonomous driving applications see (Alonso and Murillo, 2019; Sun et al.,

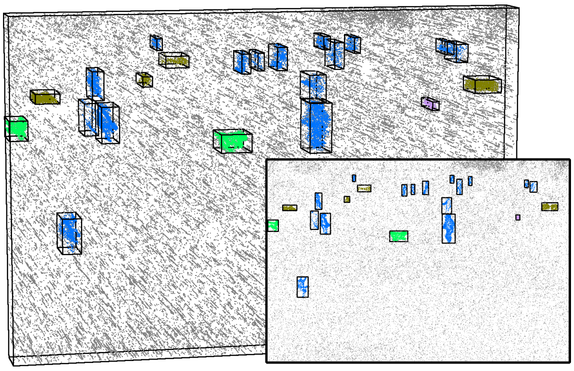


Figure 3: DVS-iOUTLAB dataset example scene (a 60 ms time window with 95,726 events is displayed as a 3D space-time event cloud and as a projected 2D label frame, with ground truth classes represented by colors and instances by bounding boxes).

2022). In the context of the monitoring scenario considered in this paper, the following datasets are relevant.

DVS-iOUTLAB. Multi-Instance DVS-OUTLAB (Bolten et al., 2021)

The DVS-OUTLAB dataset contains recordings from a multi-DVS-based monitoring scenario of an urban public children's playground. The authors provide several thousand semantically labeled patches of event data, as well as multiple hours of unlabeled raw material recorded during the process of creating the dataset. We applied a semantic segmentation based on PointNet++ (Qi et al., 2017) and the publicly available pre-trained weights from (Bolten et al., 2022) to these unprocessed, complete recordings.

Based on the originally provided labels and further semantic segmentation, we manually selected and checked object instances. Thus, we extracted 52,293 `PERSON`, 3,649 `DOG`, 7,024 `BICYCLE`, and 3,134 `SPORTSBALL` instances. From this pool, we randomly selected instances and artificially populated new challenging scenes that directly contain instance annotations. We allowed spatially close objects and prevented real occlusions based on the convex hull of the objects. In this process, we created 10,000 scenes, divided into 8,000 for training and 1,000 each for test and validation. Each scene contains a minimum of three objects and a maximum of 32 objects with 3-24 persons (average 8.76) and up to 2 dogs, 4 bicycles and 2 sportballs.

A sample scene of the newly created DVS-iOUTLAB dataset is shown in Figure 3. This dataset composes challenges from a multi-class, multi-instance scenario combined with real sensor noise.

N-MuPeTS. Multi-Person Tracking and Segmentation (Bolten et al., 2023a)

The N-MuPeTS dataset contains ≈ 85 minutes of time-continuous labeled event data, recorded for multi-person tracking and segmentation applications. The authors provide annotations at the level of instance segmentation for four recorded individuals, as well as annotations describing the overall scene quality (judging included artifacts or label quality). In addition, the activity (e.g., `WALKING`, `RUNNING`, or `CROSSING`, ...) is labeled separately for each included individual on a 25 ms time window basis.

Although the dataset contains only the single object class `PERSON`, the processing is still challenging due to object occlusions (with infrastructure and other people), similar body shapes, different body poses, and different movement/interaction patterns. Scenes with spatially close objects (such as intersections) are particularly challenging. In addition, there is sensor noise in the data. Figure 4a shows an example of a scene from this dataset.

Since the dataset's authors haven't published a dataset split, we propose the following: the basis for training and evaluation are all recordings of the best quality level, except for time windows in which at least one person was standing (dataset annotations `KNEELING`, `STOOPED` or `STANDING`). This leads to the exclusion of ≈ 7.6 minutes of recording and is therefore negligible. This is necessary for segmentation applications, as standing persons are indistinguishable from background noise in the DVS signal. The remaining recordings were divided into consecutive 10 second segments. Based on these segments, the data was divided into training, validation, and test sets. This windowing of the data was done in order to achieve a higher variability between the splits compared to randomly sampled time windows of a few milliseconds.

By selecting a 60/20/20 % split of these time blocks, care was also taken to ensure that the remaining activity annotations of the dataset were approximately equally represented in the respective splits. The supplement to this paper provides a detailed overview of the resulting distribution of annotations per split.

The newly derived DVS-iOUTLAB dataset and the detailed split of N-MuPeTS based on the dataset files are available for download[1].

3.2 Preprocessing

3.2.1 Spatio-Temporal Filtering

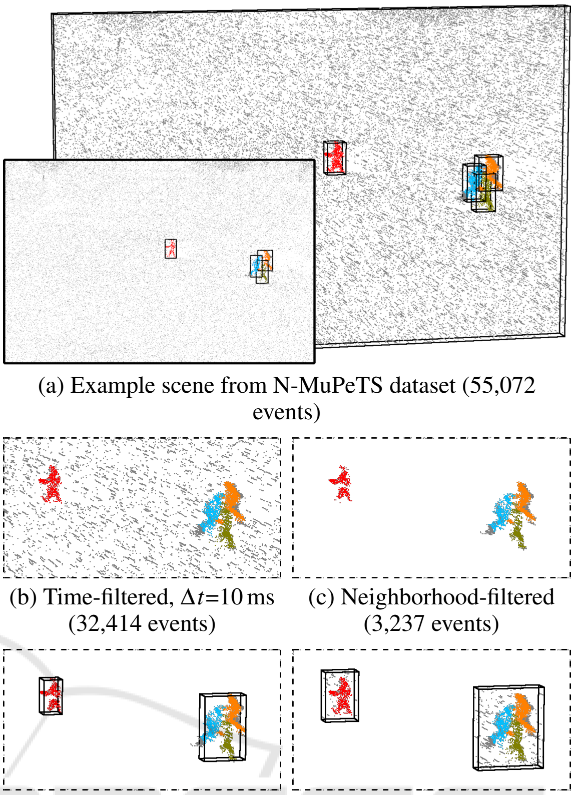
Event cameras, like all image sensors, are subject to noise. A major form of noise are background activity (BA) events, which occur when the event camera triggers an output without a corresponding change in brightness in the scene. The CeleX-IV sensor (Guo et al., 2017) used to acquire the selected datasets has a high level of background activity noise, as can be seen in Figures 3, 4a. These BA events also prevent the effective use of simple clustering approaches to segment objects.

Spatio-temporal filters are often used in preprocessing to improve the signal-to-noise ratio (SNR) of event data. Following the analysis of different spatio-temporal filters in (Bolten et al., 2021), we also apply time-filtering in the first processing stage. Each event that is not supported by another event at the same (x,y)-position within the preceding Δt ms is removed.

3.2.2 Adaptive Region-of-Interest Extraction (aRoI)

According to the function paradigm of event cameras, scene separation into foreground and background for moving objects and a static sensor is already done at the sensor level. However, a straightforward selection of events triggered by object motion is often not possible due to high noise levels. Therefore, to separate and select dense regions of events for further processing, we propose the following size-adaptive Region-of-Interest algorithm:

1. Extended spatio-temporal filtering
   First, we apply a spatio-temporal filtering stage based on the *Neighborhood-Filter* from (Bolten et al., 2021). This filter evaluates for each event a minimum threshold of other supporting events in the spatial 8-connected neighborhood. We follow the parameterization given and evaluated in (Bolten et al., 2021) for this filter. Their filter achieves an almost complete removal of BA events at the cost of events from instances. This processing step is shown in Figures 4b → 4c.
2. Hierarchical single-linkage clustering (Müllner, 2013)
   The remaining events are hierarchically clustered into regions based on the Euclidean distance of the events. Clustering is controlled by a predefined cutoff distance (d_{cut}) that prevents spatially distant clusters from merging. Resulting clusters with a number of events less than $\min_{\#\text{events}}$ are discarded in this step. An example for this step is displayed in Figure 4d.



(a) Example scene from N-MuPeTS dataset (55,072 events)

(b) Time-filtered, Δt=10 ms (32,414 events)

(c) Neighborhood-filtered (3,237 events)

(d) Performed clustering (2 segments resulting)

(e) Extracted aRoIs (765+3,820 = 4,585 events)

Figure 4: Adaptive ROI selection (given event counts refer to complete, uncropped scene).

3. Bounding Box expansion and filter reset
   In order to account for the events of objects that may have been filtered (cf. the feet of the red actor in Figure 4), the bounding box of each cluster is expanded by $\text{bbox}_{\text{offset}}$ pixels. Within each expanded bounding box, the event stream is reset to the original time-filtered event stream, reactivating the events removed by the second restrictive filtering step. Each resulting bounding box forms a Region-of-Interest. This is shown in Figure 4e.

This processing results in Regions-of-Interest of variable spatial size, where spatially separated objects are in their own aRoI and groups of objects share a aRoI without being sliced.

4 EXPERIMENTS

Next, we describe the methodology and results of our comparative evaluation of the methods presented in Section 2.2. All event representations are based on 60 ms sliding time windows of the data. The

Table 1: Object instance statistics.

annotation	max(dist($\mathrm{NN_{inst}}$))	avg(#$\mathrm{Pixel_{inst}}$)
(a) DVS-iOUTLAB		
PERSON	$\approx$ 23.09 px	332.28 px
DOG	$\approx$ 16.49 px	360.31 px
BICYCLE	$\approx$ 21.59 px	450.28 px
SPORTSBALL	$\approx$ 5.39 px	116.05 px
(b) N-MuPeTS		
PERSON	$\approx$ 22.14 px	511.80 px

configuration for the performed preprocessing is as follows: spatio-temporal pre-filtering with threshold Δt=10 ms, Region-of-Interest generation with $d_{\mathrm{cut}} = 29.0$ px, $\mathrm{min}_{\mathrm{\#events}} = 50$ events, $\mathrm{bbox}_{\mathrm{offset}} = 10$ px.

4.1 Network Configurations

Network configurations and hyperparameters have been left at their default values where possible. The detailed configurations of the neural networks are given in the supplemental material in the format of the corresponding reference implementations (see the URLs given in the bibliography entries).

4.2 Inputs and Parameters

For point- and voxel-based methods, the temporal scaling of the input data is of particular interest, as it has a significant impact on the computation of spatio-temporal distances and neighborhoods. We represent the time information scaled in milliseconds.

Point-Based Processing. As mentioned above, space-time event clouds form naturally from the (x,y,t) coordinates of the events themselves. However, while the spatial shape of the input event cloud can vary, and therefore the generated aRoI can be used directly as a basis, the deep learning processing techniques require a fixed number of events as input for training.

Therefore, the event clouds are sampled to a fixed number of events by random choice. The sizes of 1024 and 2048 events per aRoI serve as sampling targets, since these powers of two are closest to the mean event counts of the generated aRoIs (more detailed event count statistics are given in the supplement). For aRoIs with fewer events, doublets are generated to achieve the desired number, following the original logic of PointNet++ processing that forms the basis of the point-based methods under study.

The selected grouping radii and the configuration of the point abstraction layers of the networks defined by the provided model files are adapted from (Bolten et al., 2022).

Voxel-Based Processing. The data shape is defined by the voxel grid size and not by the number of events. Therefore, there is no subsampling per aRoI performed. Instead, the time-filtered aRoIs are used directly as input.

We discretize the data per sensor pixel over a time interval of 1 ms per voxel, as this setting has already shown good results for semantic segmentation (Bolten et al., 2023b).

Frame-Based Processing. The color frame encodings are built using the full spatial sensor resolution of 768 × 640 pixels (Guo et al., 2017) and complete 60 ms window, as the frame-based processing requires a fixed input resolution.

4.3 Segmentation Baseline: Semantic Clustering

As a basic approach for comparison, we propose to utilize a hierarchical event clustering, extended by applying a prior semantic segmentation. This is shown in Figure 1b.

For this semantic segmentation step, we trained vanilla PointNets++ (Qi et al., 2017) following (Bolten et al., 2022) using the aRoIs as input. Clustering is applied separately to the events of each semantic class based on the predicted labels to group the predictions into individual instances.

The clustering cutoff distance d_{cut} in this step is individually selected per semantic class based on the maximum Euclidean distance between nearest neighbor pixels within the ground truth instances (see dist($\mathrm{NN_{inst}}$) in Table 1). This selection ensures that all events of a single instance are grouped together by this baseline approach.

4.4 Metrics

Some approaches rely on prior semantic segmentation. Therefore, we also report metrics for the quality of the semantic segmentation. For this, we report the F1 score, defined as the harmonic mean of precision and recall, as a weighted average using the given support per class on a per DVS event basis.

Regarding the instance segmentation quality, we report the standard COCO metrics, including *mean average precision* $\mathrm{mAP}_{0.5}^{0.95}$, which is the precision averaged over the intersection over union (IoU) threshold range from 0.5 to 0.95 with a step size of 0.05, as well as the $\mathrm{mAP}^{0.5}$ and $\mathrm{mAP}^{0.75}$ at fixed IoU values.

For reproducibility, we rely on the metric implementations from (Detlefsen et al., 2022) for all reported results. IoUs are calculated based on segmentation masks rather than bounding boxes. For comparability between the different methods, these masks are formed and evaluated in 2D.

The evaluation is based on the events included in the constructed aRoIs. Since the spatial shape can vary between the different encoding variations (aRoI size vs. fixed and full frame resolution), only the areas covered by the aRoIs are considered for frames and included in the metric calculation.

4.5 Application Results

The evaluation was performed on datasets that originate from the application scenario of a DVS-based monitoring. The scenes considered therefore contain the typical application-oriented core challenges, such as occlusions and spatially close objects. The results focus on these challenges.

Table 3 shows the metric results for the DVS-iOUTLAB dataset. For the N-MuPeTS dataset, we report in Table 4 the results on an intentionally challenging subset of the test data. This test subset restricts the scenes to a selection in which at least one actor is occluded or they are spatially very close to each other. Details on this subset selection, as well as results on the full test set, are reported in the supplement.

Segmentation Baseline. The segmentation baseline depends on the quality of the semantic segmentation performed. It achieves very good F1 scores on both datasets. As expected, it often fails with merge errors because instances of *same* classes that are very close to each other are clustered together. This is especially true for the selected challenging test subset of the N-MuPeTS dataset. It can be clearly seen in the metric difference of this approach between the two datasets.

Point-Based Processing. Segmentation tends to fail when an aRoI is significantly larger than average. These regions occur when many objects are spatially very close to each other, so that they are clustered into a single input aRoI. The unsampled event count in these regions deviates strongly from the overall mean, so that the applied random event selection changes the spatio-temporal object densities and event neighborhoods substantially. For these error-prone aRoIs, JSNet-based processing mostly leads to interpretation as BA event noise, while 3D-BoNet predicts better semantic values, but often proposes very large and merged object instance boundaries.

Table 2: Number of network parameters for DVS-iOUTLAB network configuration.

Network	#Parameters
Baseline PointNet++	441,893
JSNet	8,098,321
3D-BoNet	1,824,582
SoftGroup	30.836.090
Mask R-CNN	44,679,088
YOLO v8	3,264,396

A simple post-processing of the obtained results seemed useful, since small errors in semantic segmentation often propagate in the form of small instances. We recommend to make sure that instances consisting of only a few events are removed and ignored before further processing.

Voxel-Based Processing. SoftGroup achieves very good results on DVS-iOUTLAB dataset which includes spatially close but not intersecting objects. Considering scenes containing occlusions of objects of the same semantic class (as in N-MuPeTS, which are considered to be particularly difficult), it can be observed that instances often merge in these cases.

Looking at the $\text{mAP}^{0.5}$ value, the best overall result is obtained for DVS-iOUTLAB, while the value for N-MuPeTS is behind all other high-level approaches. This indicates a need for further optimization of the hyperparameters used, such as voxel size and grouping radius.

Frame-Based Processing. The IoU thresholding performed for mAP calculation is more difficult for frame-based mask predictions. The low spatial resolution of the DVS sensor (the used sensor provides $768 \times 640\,\text{px}$) leads to small object sizes, as shown in Table 1. The avg(#$\text{Pixel}_{\text{inst}}$) value indicates the average projected object pixel size per instance in each dataset. Even a few mismatching pixels in the predicted masks will significantly lower the IoU score. Comparing the $\text{mAP}^{0.95}_{0.5}$and $\text{mAP}^{0.5}$ (improvements up to ≈40%) shows that the segmentation works well, but is limited by the predicted pixel mask accuracy.

When detecting and separating occluded objects of the same semantic class, the selected Mask R-CNN tends to predict a mask containing only one object in these cases. YOLO v8 predicts better partial masks at the expense of multiple false predictions.

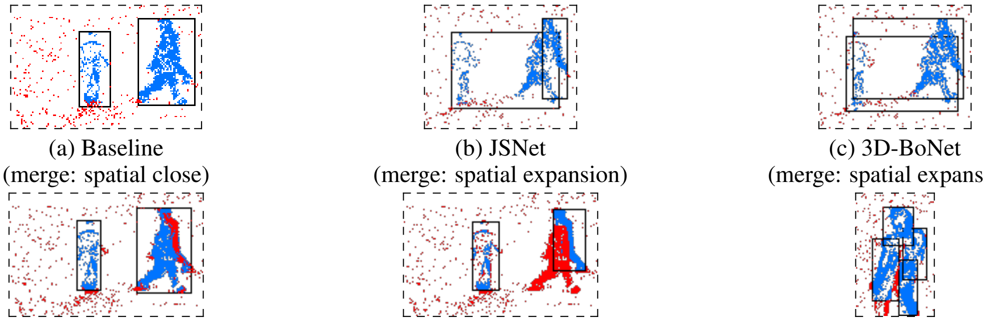
Figure 5 shows example segmentations in the form of false-color images for the N-MuPeTS dataset (corresponding examples for DVS-iOUTLAB are given in the supplement). These images highlight the typical worst-case errors.

Table 3: Segmentation results on DVS-iOUTLAB test set (60 ms event time window).

		Semantic Quality	Instance Quality			
Network	Configuration	weighted F1-score	mIoU	mAP$^{0.95}_{0.5}$	mAP$^{0.5}$	mAP$^{0.75}$
(a) Baseline method: PointNet++ with spatial clustering						
PointNet++	in 2048 events	0.94	0.80	0.57	0.71	0.62
Clustering	in 1024 events	0.93	0.82	0.58	0.71	0.61
(b) Space-Time Event Cloud-based methods						
JSNet	4 layers in 2048 events	0.95	**0.89**	0.81	0.87	0.86
	4 layers in 1024 events	0.92	0.85	0.70	0.77	0.75
3D-BoNet	4 layers in 2048 events	0.94	0.84	0.71	0.81	0.78
	4 layers in 1024 events	0.93	0.83	0.70	0.80	0.76
(c) Voxel-based method						
SoftGroup	voxel grid (768 × 640 × 60)	**0.97**	0.86	**0.88**	**0.98**	**0.96**
(d) Frame-based methods						
Mask R-CNN	polarity in (768 × 640) px	0.92	0.78	0.62	0.96	0.72
	MTC in (768 × 640) px	0.92	0.78	0.61	0.96	0.71
YOLO v8	polarity in (768 × 640) px	0.92	0.79	0.60	0.93	0.66
	MTC in (768 × 640) px	0.91	0.80	0.58	0.89	0.65

Table 4: Segmentation results on *challenging sequences* of N-MuPeTS test subset (60 ms event time window).

		Semantic Quality		PERSON Instance Quality			
		weighted F1-score					
Network	Configuration	NOISE	PERSON	mIoU	AP$^{0.95}_{0.5}$	AP$^{0.5}$	AP$^{0.75}$
(a) Baseline method: PointNet++ with spatial clustering							
PointNet++	in 2048 events	0.91	**0.95**	0.74	0.25	0.42	0.25
Clustering	in 1024 events	0.91	**0.95**	0.74	0.25	0.41	0.24
(b) Space-Time Event Cloud-based methods							
JSNet	4 layers in 2048 events	**0.92**	**0.95**	0.82	0.54	0.79	0.57
	4 layers in 1024 events	0.91	0.94	0.80	0.46	0.70	0.48
3D-BoNet	4 layers in 2048 events	0.91	**0.95**	0.80	**0.56**	0.77	0.59
	4 layers in 1024 events	0.89	0.93	0.75	0.42	0.62	0.44
(c) Voxel-based method							
SoftGroup	voxel grid (768 × 640 × 60)	0.84	0.92	**0.83**	0.55	0.70	0.57
(d) Frame-based methods							
Mask R-CNN	polarity in (768 × 640) px	0.80	0.89	0.72	0.41	0.80	0.41
	MTC in (768 × 640) px	0.80	0.89	0.72	0.42	0.80	0.43
YOLO v8	polarity in (768 × 640) px	0.83	0.92	0.70	0.55	**0.87**	**0.61**
	MTC in (768 × 640) px	0.83	0.92	0.70	0.54	0.86	0.60



(a) Baseline (merge: spatial close)

(b) JSNet (merge: spatial expansion)

(c) 3D-BoNet (merge: spatial expansion)

(d) SoftGroup (merge: spatial close)

(e) Mask R-CNN (miss at occlusions)

(f) YOLO v8 (split: object parts)

Figure 5: Typical prediction *error cases* on N-MuPeTS displayed as false-color aRoI-montage images (best viewed in color and digital zoomed).

The proposed event representations and corresponding off-the-shelf processing approaches can effectively be used to derive an instance segmentation. From a practical point of view, the proposal-based point and voxel-based approaches require temporal normalization in addition to temporal scaling for training convergence. Our recommendation is to shift the continuous event time stamps for each input aRoI between zero and the selected sliding time window length.

The point-based approaches are inspired and built on PointNet++ as a backbone. By sharing the MLPs per point, relatively small network structures are built (see Table 2). This feature may be important when aiming for a sensor-near implementation where hardware resources are limited.

By using a submanifold sparse convolution (Graham et al., 2018), the voxel-based processing provides good results and can offer a good trade-off in terms of processing complexity. For applications where small compromises in pixel accuracy of segmentation are acceptable, classical frame-based processing seems to be a good starting point, while offering a wide range of well-established frameworks for processing.

5 CONCLUSION

We have performed a systematic evaluation of instance segmentation approaches on data from the domain of event-based vision. We included multiple state-of-the-art instance segmentation approaches that are based on deep learning, while at the same time considering event representations with varying degrees of dimensionality. Overall, very good results can be obtained by using these off-the-shelf processing approaches.

While the evaluation is scenario specific, the proposed encoding and processing combinations can easily be adopted to other applications. Real-world event-based vision projects are still uncommon. Using standard processing approaches is an appropriate way to change this.

One aspect of further work is a detailed study of the hyperparameters of the networks to fine-tune the possible results. Examples include non-maximum suppression for the frame-based approaches, or grouping radii in point or voxel-based approaches to improve the processing of very close and occluding objects. For future practical applications, it is important to consider environmental effects such as rain. These exist in the real world beyond the dimensions contained in the datasets.

ACKNOWLEDGEMENTS

We thank Laureen Klein for her preparatory work in creating the DVS-iOUTLAB dataset. We thank Noel Hanraths Pereira for his collaboration on the YOLO experiments.

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. IEEE.

Barranco, F., Teo, C. L., Fermuller, C., and Aloimonos, Y. (2015). Contour Detection and Characterization for Asynchronous Event Sensors. In *2015 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 486–494. IEEE.

Biswas, S. D., Kosta, A., Liyanagedera, C., Apolinario, M., and Roy, K. (2022). HALSIE: Hybrid Approach to Learning Segmentation by Simultaneously Exploiting Image and Event Modalities. arXiv preprint arXiv:2211.10754.

Bolten, T., Lentzen, F., Pohle-Fröhlich, R., and Tönnies, K. (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., Neumann, C., Pohle-Fröhlich, R., and Tönnies, K. (2023a). N-MuPeTS: Event Camera Dataset for Multi-Person Tracking and Instance Segmentation. In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP*, pages 290–300. INSTICC, SciTePress.

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

Bolten, T., Pohle-Fröhlich, R., and Tönnies, K. (2023b). Semantic Segmentation on Neuromorphic Vision Sensor Event-Streams Using PointNet++ and UNet Based Processing Approaches. In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP*, pages 168–178. INSTICC, SciTePress.

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.

Chen, S., Fang, J., Zhang, Q., Liu, W., and Wang, X. (2021). Hierarchical Aggregation for 3D Instance Segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15467–15476. IEEE.

Detlefsen, N. S., Borovec, J., Schock, J., Jha, A. H., Koker, T., Liello, L. D., Stancl, D., Quan, C., Grechkin, M., and Falcon, W. (2022). TorchMetrics - Measuring Reproducibility in PyTorch. *Journal of Open Source Software*, 7(70):4101. https://github.com/Lightning-AI/torchmetrics.

Engelmann, F., Bokeloh, M., Fathi, A., Leibe, B., and Niessner, M. (2020). 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9028–9037. IEEE.

Graham, B., Engelcke, M., and van der Maaten, L. (2018). 3D Semantic Segmentation With Submanifold Sparse Convolutional Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9224–9232. IEEE.

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.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *2017 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE. https://github.com/matterport/Mask_RCNN.

Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.-W., and Jia, J. (2020). PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Jocher, G., Chaurasia, A., and Qiu, J. (2023). YOLO by Ultralytics. https://github.com/ultralytics/ultralytics, v8.0.0, AGPL-3.0.

Kachole, S., Alkendi, Y., Baghaei Naeini, F., Makris, D., and Zweiri, Y. (2023a). Asynchronous Events-based Panoptic Segmentation using Graph Mixer Neural Network. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4083–4092. IEEE.

Kachole, S., Huang, X., Baghaei Naeini, F., Muthusamy, R., Makris, D., and Zweiri, Y. (2023b). Bimodal SegNet: Instance Segmentation Fusing Events and RGB Frames for Robotic Grasping. arXiv preprint arXiv:2303.11228.

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. IEEE.

Müllner, D. (2013). fastcluster: Fast Hierarchical, Agglomerative Clustering Routines for R and Python. *Journal of Statistical Software*, 53(9):1–18.

Piątkowska, E., Belbachir, A. N., Schraml, S., and Gelautz, M. (2012). Spatiotemporal Multiple Persons Tracking using Dynamic Vision Sensor. In *2012 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 35–40. IEEE.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*, NIPS'17, pages 5105–5114, Red Hook, NY, USA. Curran Associates Inc. https://github.com/charlesq34/pointnet2.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE.

Rodríguez-Gomez, J. P., Eguíluz, A. G., Martínez-de Dios, J. R., and Ollero, A. (2020). Asynchronous Event-Based Clustering and Tracking for Intrusion Monitoring in UAS. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8518–8524.

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.

Schraml, S. and Belbachir, A. N. (2010). A Spatio-temporal Clustering Method Using Real-time Motion Analysis on Event-based 3D Vision. In *2010 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 57–63. IEEE.

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. IEEE.

Sharma, R., Saqib, M., Lin, C. T., and Blumenstein, M. (2022). A Survey on Object Instance Segmentation. *SN Computer Science*, 3(6):499.

Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., and Scaramuzza, D. (2019). Event-Based Motion Segmentation by Motion Compensation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7243–7252. IEEE.

Stoffregen, T. and Kleeman, L. (2018). Simultaneous Optical Flow and Segmentation (SOFAS) using Dynamic Vision Sensor. In Kodagoda, S. et al., editors, *Australasian Conference on Robotics and Automation 2017*.

Sun, Z., Messikommer, N., Gehrig, D., and Scaramuzza, D. (2022). ESS: Learning Event-based Semantic Segmentation from Still Images. In *Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 341–357. Springer.

Vasco, V., Glover, A., Mueggler, E., Scaramuzza, D., Natale, L., and Bartolozzi, C. (2017). Independent Motion Detection with Event-driven Cameras. In *2017 18th International Conference on Advanced Robotics (ICAR)*, pages 530–536.

Vu, T., Kim, K., Luu, T. M., Nguyen, T., and Yoo, C. D. (2022). SoftGroup for 3D Instance Segmentation on Point Clouds. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2708–2717. IEEE. https://github.com/thangvubk/SoftGroup.

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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 608–619. IEEE.

Wang, X., Zhang, R., Kong, T., Li, L., and Shen, C. (2020). SOLOv2: Dynamic and Fast Instance Segmentation. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33 of *NeurIPS'20*, pages 17721–17732. Curran Associates, Inc.

Wu, W., Qi, Z., and Fuxin, L. (2019a). PointConv: Deep Convolutional Networks on 3D Point Clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9613–9622. IEEE.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019b). Detectron2. https://github.com/facebookresearch/detectron2.

Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., and Luo, P. (2020). PolarMask: Single Shot Instance Segmentation With Polar Representation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Xie, E., Wang, W., Ding, M., Zhang, R., and Luo, P. (2022). PolarMask++: Enhanced Polar Representation for Single-Shot Instance Segmentation and Beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5385–5400.

Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., and Trigoni, N. (2019). Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32 of *NeurIPS'19*, pages 6737–6746. Curran Associates, Inc. https://github.com/Yang7879/3D-BoNet.

Zhao, L. and Tao, W. (2020). JSNet: Joint Instance and Semantic Segmentation of 3D Point Clouds. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12951–12958. https://github.com/dlinzhao/JSNet.

Zhou, Y., Gallego, G., Lu, X., Liu, S., and Shen, S. (2021). Event-Based Motion Segmentation With Spatio-Temporal Graph Cuts. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13.