EventPoint: Self-Supervised Interest Point Detection and Description for Event-based Camera

Ze Huang
Xiamen University

Li Sun
University of Sheffield

Cheng Zhao
Bosch Research North America

Song Li
Xiamen University

Songzhi Su*
Xiamen University

Abstract

This paper proposes a self-supervised learned local detector and descriptor, called EventPoint, for event stream/camera tracking and registration. Event-based cameras have grown in popularity because of their biological inspiration and low power consumption. Despite this, applying local features directly to the event stream is difficult due to its peculiar data structure. We propose a new time-surface-like event stream representation method called Tencode. The event stream data processed by Tencode can obtain the pixel-level positioning of interest points while also simultaneously extracting descriptors through a neural network. Instead of using costly and unreliable manual annotation, our network leverages the prior knowledge of local feature extraction on color images and conducts self-supervised learning via homographic and spatio-temporal adaptation. To the best of our knowledge, our proposed method is the first research on event-based local features learning using a deep neural network. We provide comprehensive experiments of feature point detection and matching, and three public datasets are used for evaluation (i.e. DSEC, N-Caltech101, and HVGA ATIS Corner Dataset). The experimental findings demonstrate that our method outperforms SOTA in terms of feature point detection and description.

1. Introduction

In comparison with conventional standard frame-based cameras, the bio-inspired event camera offers significant advantages of microsecond temporal resolution, low latency, very high dynamic range, and low power consumption. These revolutionary features enable some new robotics applications in extremely challenging conditions, e.g. in low-illumination scenarios and high-speed flying robot applications. The event-based cameras, e.g. DVS [20], Davis [5] and ATIS [28] can capture the event points in the corresponding pixel position when sensing the pixel brightness changes over a temporal resolution. More precisely, the event camera asynchronously measures changes in brightness of each pixel within a certain threshold in a high dynamic range from 60 dB to 140 dB. The sign of events (positive or negative) is also known as polarity.

Although event-based cameras have numerous advantages, dealing with some standard computer vision tasks directly on the event stream, e.g. local feature extraction, is challenging due to the spatio-temporal data structure. Local feature detection and representation [27], is the core technology for a variety of applications, e.g. visual odometry, place recognition, 3D reconstruction, etc. The image-based local feature extraction and description can be grouped into hand-crafted [22, 6, 32] and deep-learned [9, 40, 13, 35, 31] methods. Compared to hand-crafted features, the deep-learned features demonstrate significant advances in terms of performance on several benchmarks [34, 18].

Local feature extraction methods on image data cannot be applied to event-based data straightforwardly due to the domain variance between the traditional image and event-based data. Furthermore, the great sensitivity of the event camera creates a lot of noise, making the work more challenging [2, 39]. Some recent research [38, 24, 1, 23, 4] explore the corner points detection on the event stream. However, most of them only include interest point detection without describing the detected feature points due to the monotonic event-based data structure. Some further event-based local descriptor methods [30, 11] are evaluated using the toy or simulated event data, while the effectiveness, and robustness in large-scale realistic scenes have not been verified. Additionally, the existing methods are hand-

*Corresponding author: Songzhi Su, ssz@xmu.edu.cn

crafted rather than deep-learned, which shows weakness in noise filtering, semantic-level understanding and adaptation of sensing data and hyper-parameters.

To overcome the limitations above, our research makes the following contributions:

- We propose a deep-learning-based local detector and descriptor, i.e., EventPoint, tailored for event stream/camera.
- We propose a simple but effective events representation method, called Tencode, which significantly facilitates the feature point's representation learning.
- Our approach is delicately designed to learn spatio-temporal and homography invariant local descriptors in a self-supervised way without extra human annotations.
- We conduct a comprehensive evaluation in terms of feature detecting and feature matching on three different public benchmarks, and the experimental results show that our approach is superior to existing methods.

2. Related Work

2.1. Local Feature on Image Data

The hand-craft-based local features [22, 3, 33] are well-studied and are still the priority in real industrial applications.

Recently, the emerging deep-learned-based local features have become the dominating steam of methods. LIFT [40] is an early-stage work that firstly investigates the use of CNN to extract local features with a full pipeline of detection, orientation estimation, and feature description in a unified manner. A lightweight deep-learned local descriptor, SuperPoint [13] proposes a self-supervised learning method using pseudo-ground-truth correspondences generated by homographic transformation. It designs a dual-head network for interested point detection and description separately. Unsuperpoint [10] proposes a siamese network to learn the detector and descriptor, the interest points' positions are learned in a regression manner. R2D2 [31] learns both keypoint repeatability and a confidence for interest points reliability from relevant training data, where style transfer method is used to increase robustness against dynamic illumination change such as day-night. By leveraging implicit semantic understanding, the learning-based local features [40, 13, 10, 31] show extraordinary advances in dealing with long-term variation in real-world conditions [34, 18].

2.2. Local Feature on Event-based Data

Recently, the event-based local feature has been attracting a lot of attention from the computer vision community. EvFast [24] employs the FAST corner point detection [37] to select the interesting event points via timestamp difference. EvHarris [38] transforms the raw event stream to a Time-surface [4] representation, and further detects interesting points by Harris corner detector [12]. A more efficient Harris corner detector [17] on event-stream is designed by tuning throughout of Time-surface and refactoring the conventional Harris algorithm. In [23], a random forest is employed to extract corner interesting point, and, Speed-invariant Time-surface feature is used for training. The above methods only achieve event-based local feature detection, however, local feature description is not considered. DART [30] uses the log-polar grid of the event camera to encode the structure context to describe the interesting points. Currently, most of the event-based local features require elaborate human design, which shows limited ability to handle complex situations such as significant motion changes or high-speed motion. Most recent research [23] learns a local feature from the event data stream, but a large-scale human annotation is required.

3. Methodology

An overview of our method is given in Fig. 1.

3.1. Event Stream Representation

Event-stream data consists of four dimensional information (x, y, t, p), where x, y are the event's location, t for the timestamp, and p for the event's polarity. The mainstream event stream representation methods can be grouped into two categories, i.e., Time-window-based representation and Time-surface-based representation.

Time-window-based methods use a constant temporal resolution Δt, then accumulate all events under a time window $(T, T + \Delta t)$ to generate a single frame representation F.

$$F[x, y] = p \leftarrow (x, y, t, p), \tag{1}$$

Time-window-based representation mainly considers the polarity of events but ignores the timestamp information. i.e., the asynchrony of event-based data, which is frequently used in global feature extraction or multi-modality fusion. The Time-surface-based representation [4] transforms the spatio-temporal event stream into a frame-like representation F by normalizing the timestamps, however the events' polarity is ignored.

$$F[x, y] = t \leftarrow (x, y, t, p). \tag{2}$$

Time-surface is widely-used in tasks that require accurate local information.

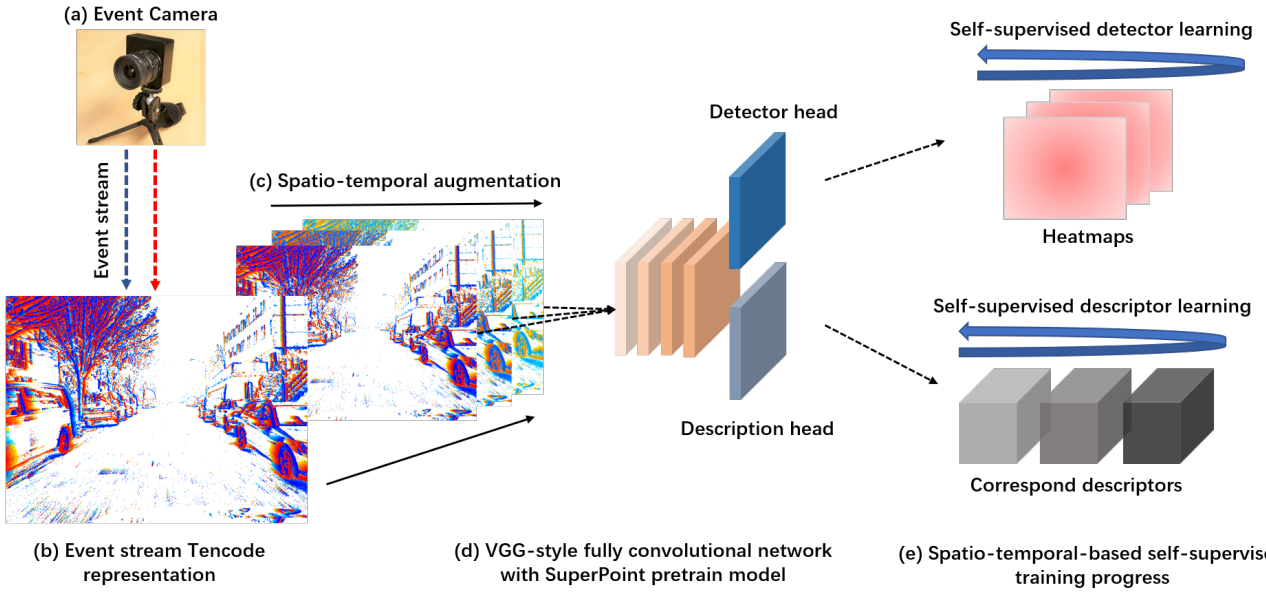

Figure 1. **Overview of our proposed method:** (a) event camera captures asynchronous event stream with binary polarity; (b) the event stream is sliced temporally and represented via Tencode; (c) spatio-temporal adaptation is used to generate the supervisory signal required for neural network training; (d) EventPoint uses SuperPoint-like architecture for local feature extraction; (e) two decoder heads, i.e. detector head and descriptor head, are trained separately in our proposed self-supervised manner.

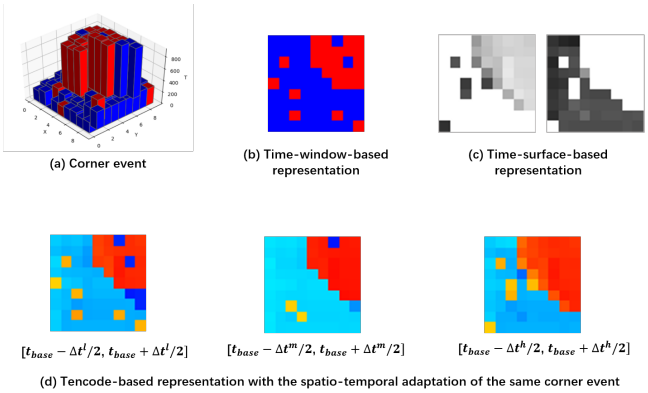

Figure 2. **Visualization of different event-stream representation methods. A local patch of a corner is shown as an example:** (a) In a raw event stream, the corner point experiences a sudden change over timestamps; (b) Time-window-based representation maps events into a fixed temporal resolution to a single frame and only the polarity is presented (it ignores the magnitude of the time difference within this time window); (c) Time-surface-based representation captures the gradient of time information for each event, but either ignores the polarity or handles different polarities separately; (d) The proposed Tencode representation considers both the polarity and the gradient of timestamp.

Instead, we propose a new representation, named Tencode, taking both polarities and timestamps of event steam into account. Firstly, a maximum temporal resolution Δt is defined to discretize the continuous events to separate frames. Then, all events falling in this temporal window will be mapped into a frame F according to polarity and timestamp information by,

$$F[x,y] = (255, \frac{255 * (t_{max} - t)}{\Delta t}, 0) \leftarrow (x, y, t, +1),$$
$$F[x,y] = (0, \frac{255 * (t_{max} - t)}{\Delta t}, 255) \leftarrow (x, y, t, -1), \tag{3}$$

where t_{max} represents the timestamp of the latest event in the temporal resolution $[t_{max} - \Delta t, t_{max}]$, and $+1$, -1 are positive and negative events respectively. The three-dimensional vector $F[x,y]$ refers to the per-pixel information of the mapped frame F at location $[x,y]$. Following the expression of Time-window [7], we set the first and third dimensions of the channel information to 255 or 0. Specifically, 255 for the first channel and 0 for the third channel when dealing with events with positive polarity, similarly, 0 for the first channel and 255 for the third channel when dealing with events with negative polarity. Fig. 2 visualizes the two mainstream event stream representation methods (i.e. Time-surface and Time-window) and the proposed Tencode.

3.2. The Neural Network Input

Given a visual landmark that appears during a small temporal resolution $[T_s^c, T_e^c]$, we believe that any reasonable segment of the event stream Ev within this temporal resolution can consistently detect the feature point and describe

it,

$$Ev[T_s^1, T_e^1] \equiv Ev[T_s^2, T_e^2], \tag{4}$$

where $Ev[T_s^n, T_e^n]$ refers to whole events with timestamps greater than T_s^n and less than T_e^n. The constraint for this formula to hold is, $\delta_{max} > T_e^i - T_s^i > \delta_{min}$, $T_e^i <= T_e^c$, and $T_s^i >= T_s^c$. We name this equivalence property the spatio-temporal consistency of the event stream. δ_{max} and δ_{min} are the maximum duration and the minimum duration to hold the spatio-temporal consistency hypothesis. We need this lower-bound to guarantee sufficient events accumulated to detect and describe the local feature. While the upper-bound eliminates the random walk of the absence of the local descriptors. The purpose of our method is to use a network to learn the spatio-temporal invariant feature point positions and descriptions given the above consistency hypothesis. The physical meaning of this hypothesis is to achieve the speed-invariant representation of landmarks in the real world.

Based on the above assumption, we used Tencode to generate 3 event frames using different temporal resolutions as the neural network inputs. For each base timestamp t_{base}, we choose a temporal resolution Δt and get the latest event's timestamp t_{max} in the temporal window $[t_{base} - \Delta t/2, t_{base} + \Delta t/2]$. The events in 3 different temporal resolutions are further mapped via Tencode with parameter t_{max} and Δt:

$$\begin{aligned} Ev[t_{base} - \frac{\Delta t^h}{2}, t_{base} + \frac{\Delta t^h}{2}] &\xrightarrow{Tencode} F_h, \\ Ev[t_{base} - \frac{\Delta t^m}{2}, t_{base} + \frac{\Delta t^m}{2}] &\xrightarrow{Tencode} F_m, \\ Ev[t_{base} - \frac{\Delta t^l}{2}, t_{base} + \frac{\Delta t^l}{2}] &\xrightarrow{Tencode} F_l, \end{aligned} \tag{5}$$

where $\Delta t^l < \Delta t^m < \Delta t^h$. As shown in Fig.2 (d), for the same patch, Tencode encodings of different temporal resolutions are distinct. We leverage the neural network to learn invariant representations over these distinct Tencode encodings.

3.3. EventPoint Network Architecture

We employ the SuperPoint-like [13] architecture consisting of a shared encoder and two heads, i.e., interest point detection and description. The detailed architecture of the network is provided in Tab.1. The VGG-style [36] encoder transforms the gray-scale Tencode representation $F \in \Re^{H*W}$ to a low-resolution and high-dimensional feature map $f \in \Re^{H/8*W/8*128}$. Then the feature map is fed into two heads: one for interest points detection and the other one for description. The interest point detector head outputs a heatmap $h \in \Re^{H/8*W/8*65}$ to give the probability of that pixel laying in an $8*8+1$ sized bin via a $Softmax$ function. The last channel value represents whether the bin presences a feature points or not. The heatmap h will be further restored to the original size through the $Reshape$ operation,

Table 1. Detailed EventPoint Architecture

Encoder			
1a	ReLU(Conv2d(1,64))		
1b	ReLU(Conv2d(64,64))		
	MaxPool2d(kernel_size=2, stride=2)		
2a	ReLU(Conv2d(64,64))		
2b	ReLU(Conv2d(64,64))		
	MaxPool2d(kernel_size=2, stride=2)		
3a	ReLU(Conv2d(64,128))		
3b	ReLU(Conv2d(128,128))		
	MaxPool2d(kernel_size=2, stride=2)		
4a	ReLU(Conv2d(128,128))		
4b	ReLU(Conv2d(128,128))		
Detector head		**Descriptor head**	
cPa	Conv2d(128,256)	**dDa**	Conv2d(128,256)
	ReLU		ReLU
semi	Conv2d(256,65)	**desc**	Conv2d(256,256)

$$h \in \Re^{H/8*W/8*65} \xrightarrow[65]{Softmax} \xrightarrow{Reshape} h_{out} \in \Re^{H*W}. \tag{6}$$

The point's probability larger than a certain threshold τ is regarded as interest points. The description head firstly outputs a dense grid of descriptors $d \in \Re^{H/8*W/8*128}$, and then obtain a dense descriptor of the same size of the original frame through bi-cubic interpolation:

$$d \in \Re^{H/8*W/8*128} \xrightarrow{bi-cubic} d_{out} \in \Re^{H*W*128}. \tag{7}$$

No deconvolution operation is used to guarantee the real-time performance. In order to transferring the learned weights from pretained model, we use the same architecture with SuperPoint. We further train the detector and descriptor via the spatio-temporal correspondences.

3.4. EventPoint Network Training

3.4.1 Detector Learning

As mentioned in Sec 3.2, we have 3 Tencode frames F_h, F_m, and F_l as inputs of the network. In the detector learning step, the interest point position is firstly obtained by the detector of a pretrained SuperPoint on F_l with the lowest temporal resolution. Then more pseudo-label $label \in \Re^{H/8*W/8*65}$ of interest point position is generated via homographic adaptation [13]. To guide the detector training more effectively, we set predictions, of which probabilities are greater than a certain threshold τ, as 1 (positive labels) and other values to 0 (negative labels). The detector learning pipeline is shown in Fig. 3.

Since the number of detected interest points is much smaller than that of non-interest points, different from Su-

perPoint, we employ focal loss [21] rather than cross-entropy loss to balance the training examples. The detection loss $Loss_{kp}$ is defined as:

$$Loss_{kp} = \sum_{t=h,m,l} w_t l_p(Softmax(h_t), label), \quad (8)$$

where w_t refers to scale weights, and h_t refers to the heatmaps generated through detector head of each Tencode frame input. The detailed l_p can be described as:

$$l_p = \frac{1}{65 H_c W_c} \sum_{i=1}^{H_c} \sum_{j=1}^{W_c} \sum_{k=1}^{65} Focal(h_{ijk}, label_{ijk}), \quad (9)$$

where H_c and W_c are $1/8$ of the height and width of the original image resolution and the focal loss can be described as:

$$Focal(\wp, \urcorner) = \begin{cases} \alpha(1-\wp)^{\gamma} ln(1-\wp) & \urcorner = 1 \\ (1-\alpha)(\wp)^{\gamma} ln(\wp) & \urcorner = 0 \end{cases} \quad (10)$$

where $\wp$ refers to the predict label, $\urcorner$ refers to pseudo-label, α and γ are hyper-parameters used to balance loss [21].

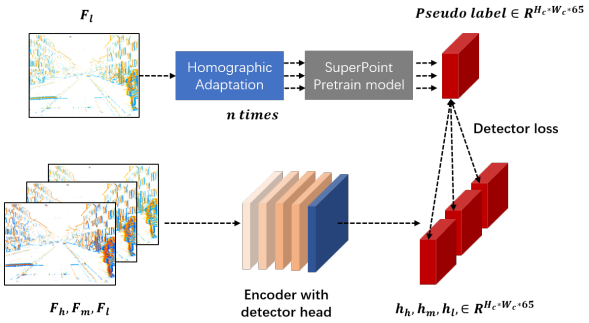

Figure 3. **Spatio-temporal-based self-supervised detector learning**: we initially apply homographic adaptations to automatically annotate F_l using SuperPoint's pretrained detector. Then, the inconsistency between the results of input and $label$ is penalized.

3.4.2 Descriptor Learning

After training the detector head, the descriptor head is further trained based on the detected interest points. Firstly, the spatio-temporal correspondences of interest points between the Tencode frame triplets are defined as:

$$s_{ij}^{i'j'} = \begin{cases} 1, & L_2(ij, i'j') < \epsilon \\ 0, & otherwise \end{cases} \quad (11)$$

Given an interest point position (i_h, j_h) in F_h and another one (i_l, j_l) in F_l. If the Euclidean distance between (i_h, j_h) and (i_l, j_l) is less than a distance threshold ϵ, we regard them as the correspond points, vice versa. The descriptor learning pipeline is shown in Fig. 4.

The $Hinge$-style loss is employed for the descriptor training:

$$Loss_{desc} = \sum_{t_1,t_2=h,m,l} w_t l_{desc}(d_{t_1}, d_{t_2}), \quad (12)$$

where w_t refers to scale weights, and d_t refers to the feature map of descriptors generated through descriptor head of each Tencode frame input. The detailed l_{desc} can be described as:

$$l_{desc} = \frac{1}{(H_c W_c)^2} \sum_{i,j=1}^{H_c,W_c} \sum_{i',j'=1}^{H_c,W_c} l_d(d_{t_1}^{ij}, d_{t_2}^{i'j'}; s_{ij}^{i'j'}), \quad (13)$$

where H_c and W_c are $1/8$ of the height and width of the original image. And l_d is defined as:

$$\begin{aligned} l_d(d, d^{'}; s) =& \lambda * s * max(0, m_p - d^T d') \\ &+ (1-s) * max(0, d^T d' - m_n), \end{aligned} \quad (14)$$

where m_p and m_n refers to positive and negative margins respectively. λ is used to balance the potential number of negative and positive correspondences.

4. Experiments

We evaluate the EventPoint comparing with baselines in local feature detecting and matching tasks on three public datasets, i.e., DSEC [15, 16], N-Caltech101 [25] and HVGA ATIS Corner Dataset [23].

4.1. Network Training Details

EventPoint is trained under two event stream representation methods as shown in Fig. 2(b), and Fig. 2(d), and distinguished by $EventPoint$ and $EventPoint-T$ identification. The basic model is trained on the DSEC dataset. During training, Δt^h and Δt^m are randomly generated in a fixed temporal resolution, $50ms \geqslant \Delta t^h \geqslant 35ms$, $35ms \geqslant \Delta t^m \geqslant 20ms$, and Δt^l is set to $20ms$. In the detection loss function, τ is set to 0.005. The points with a value greater than 0.005 in the heatmap are regarded as interest points in training and 0.015 in testing. The parameters α and γ in focal loss are set to 0.75 and 2.0, respectively. The weights $w_{h,m,l}$ in the loss function is set to 0.5, 0.5 and 1.0 respectively. The distance threshold ϵ is set to 8. The value of λ used to balance in descriptor loss is 0.001. The positive margin m_p and negative margin m_n are set to 1 and 0.2 respectively.

The network is implemented under PyTorch [26] framework. It is trained with batch sizes of 8 and each batch

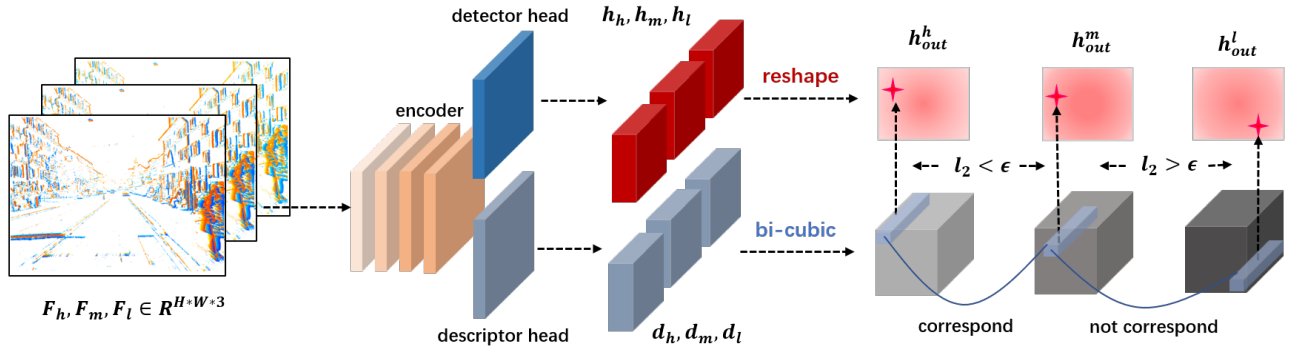

Figure 4. **Spatio-temporal-based self-supervised descriptor training:** the association of detected interest point is decided according to their locations. Then the network learns to make the descriptor distance of correspondence pairs closer and that of non-correspondence pairs farther.

contains 3 event frames. The SGD solver with default parameters of $lr = 0.001$ is used during training. The input size is cropped to $320 * 240$ for the network training. During inferring, the size is set to $640 * 480$ on the DSEC and HVGA ATIS Corner datasets. But the size is set to $320*240$ when dealing with the N-Caltech101 dataset because of its low resolution. The detector and descriptor branches are trained for around 10 epochs respectively.

For the run-time performance, the EventPoint takes about $0.1s$ to load the network and about $0.02s$ to process a single picture on an Intel(R) Core(TM) i9-9900KF CPU which achieves real-time performance similar to SuperPoint.

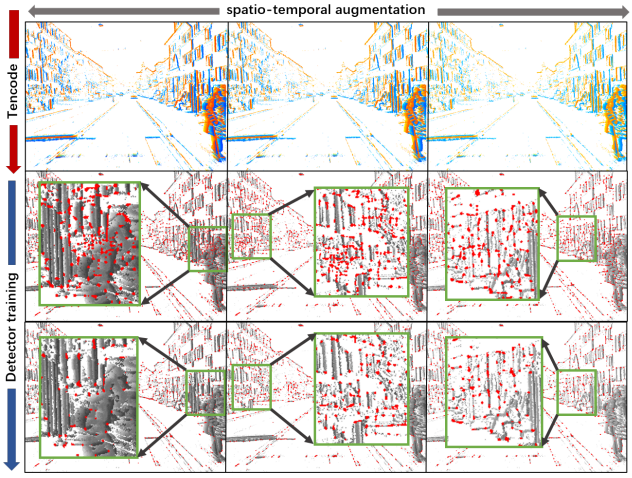

Figure 5. **Iterative detector training:** with the help of detector training, EventPoint can find more clear and more accurate interest points. Red dots represent the value greater than threshold τ in heatmaps. The red dots, i.e., interest points changes from large patches to stable positions gradually.

4.2. Datasets Details

The DSEC [15, 16] dataset provides a set of sensory data in demanding illumination conditions. Moreover, DSEC also provides the high resolution and large-scale stereo event camera dataset. It contains 53 sequences collected by driving in a variety of illumination conditions and provides ground-truth disparity map for the evaluation of event-based stereo algorithms.

The N-Caltech101 dataset [25] is the event format of the Caltech101 dataset [14]. It consists of 101 object categories, each with a different number of samples ranging from 31 to 800. We followed the standard experimental setting of the dataset [19] and [29], which conducts feature matching evaluation by IOU matching score on up to 50 images in each category.

The HVGA ATIS Corner Dataset [23] is composed of 7 sequences of planar scenes acquired with an HVGA event sensor. We use the same evaluation metrics used in [23, 8], i.e., the reprojection error which is computed by estimating a homography from two different timestamps.

4.3. Ablation Study on DSEC

DSEC provides disparity maps corresponding to event frames under $50ms$ temporal resolution in urban-driving scenes. We selected several sequences with different brightness conditions to evaluate local feature matching quality via disparity map as a ablation study. The brightness conditions are directly related to the density of events and the number of noise. In the DSEC dataset, the number of events and noise in a dark environment is much more than that with better lighting conditions.

The value d of (x, y) in the disparity map represents that the pixel points at the left frame (x, y) correspond with the pixel points at the right frame $(x - d, y)$. For a matched pair $left(x, y)$, $right(x', y)$, if $x - x' < \sigma$, we regard it as a correct match. The σ is set to 3, 6 ,and 9 in our experiment. The matching accuracy of each event data pair

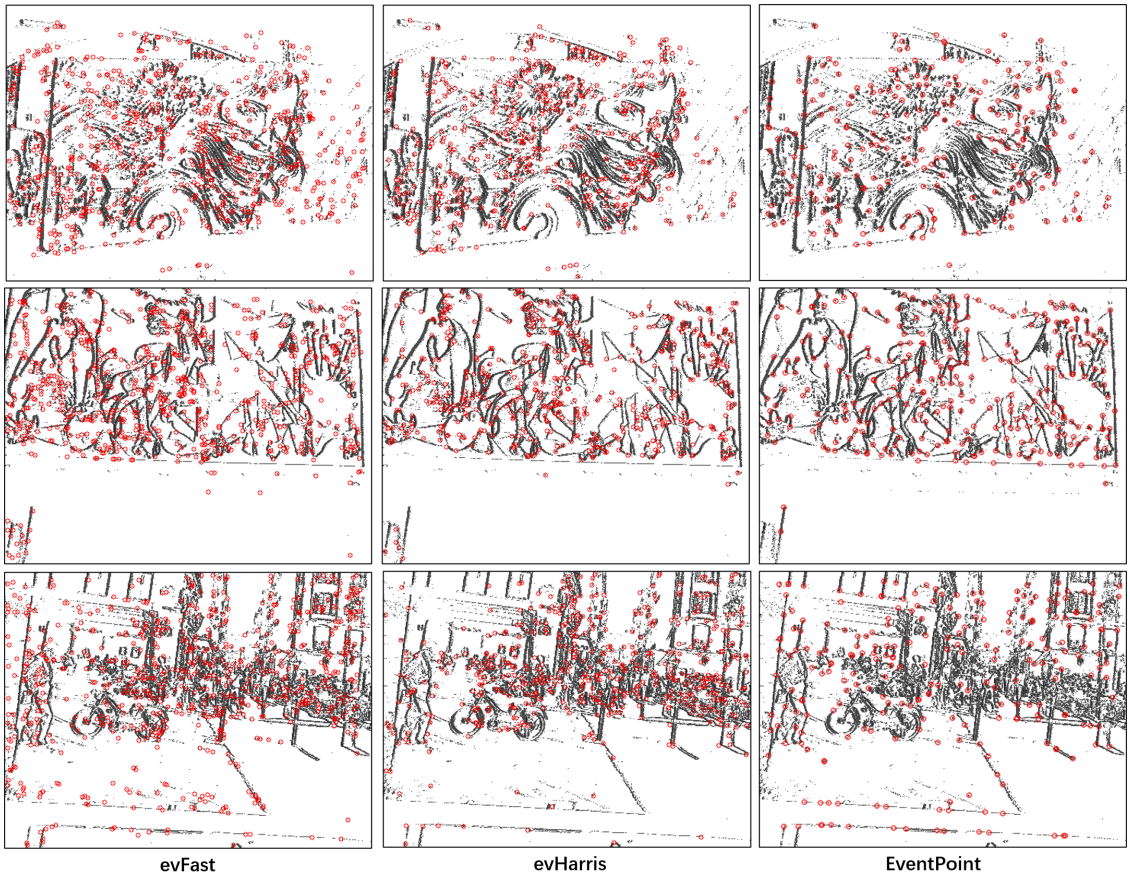

Figure 6. Qualitative results of feature point detection on HVGA ATIS Corner dataset with baselines. Events are mapped under 10ms temporal resolution with feature points detected. EventPoint can detect stable and accurate feature points and is more robust to noise.

Table 2. Ablation Study on DSEC dataset

zurich city 10_b(dark)			
	< 3	< 6	< 9
EventPoint-T(ours)	42.34	**88.97**	**96.70**
EventPoint(ours)	**43.72**	85.07	93.52
Pretrained Weights[13]	35.11	56.91	70.25
zurich city 11_c(overcast)			
	< 3	< 6	< 9
EventPoint-T(ours)	65.47	**97.18**	**99.48**
EventPoint(ours)	**66.09**	96.69	99.23
Pretrained Weights[13]	25.38	44.60	60.15
Inter laken 00_c(sun)			
	< 3	< 6	< 9
EventPoint-T(ours)	**82.86**	**98.41**	**99.30**
EventPoint(ours)	67.52	90.33	94.35
Pretrained Weights[13]	19.47	14.39	22.92
Inter laken 00_d(sun)			
	< 3	< 6	< 9
EventPoint-T(ours)	**77.26**	**93.96**	**96.07**
EventPoint(ours)	67.46	91.88	94.61
Pretrained Weights[13]	19.47	35.04	46.91

is calculated as the number of correct matches divided by the number of valid matches since the disparity map is relatively sparse. The average matching precision on each sequence is reported as,

$$Precision = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{[|x - x'| < \sigma]}{M}, \qquad (15)$$

where N refers to the number of samples in a single sequence, M refers to correct matches in event data pairs, and $[\cdot]$ is the Iverson bracket.

From Tab. 2, it can be seen that the performance of the trained EventPoint significantly outperforms the initial network parameters, i.e., SuperPoint's pretrain weights. In this ablation study, accurate position detection of feature points is an important step. It shows our self-supervised detector training improves the detection performance. We visualize the heatmaps' change during detector training in Fig. 5. On the other hand, the proposed Tencode method introduces more rich timestamp cues into the training data to improve EventPoint's performance, especially in sunny times.

4.4. IOU Matching Evaluation on N-Caltech101

DART [30] provides an evaluation metric, i.e., IOU matching score, for feature matching on the N-Caltech101

Table 3. Feature Matching IOU comparing with DART

Methods	IOU
EventPoint-T(ours)	**0.83**
EventPoint(ours)	0.79
DART(FIFO size=2000)[30]	0.72
DART(FIFO size=5000)[30]	0.67

dataset. Given two event sequences with a length of about $300ms$, the global matching within the object contour is regarded as correct matching, otherwise as wrong matching. To compare with work on descriptors on event streams, EventPoint is trained on N-Caltech101 and evaluated following the DART's experiment settings.

From Tab. 3, we can see the performance of EventPoint's deep-learning based description outperforms slightly the baselines DART's handcraft description. It also shows the proposed Tencode method is more useful than the conventional encoding method even in single-objective-based datasets. Fig. 7 visualizes several samples of feature matching on N-Caltech101.

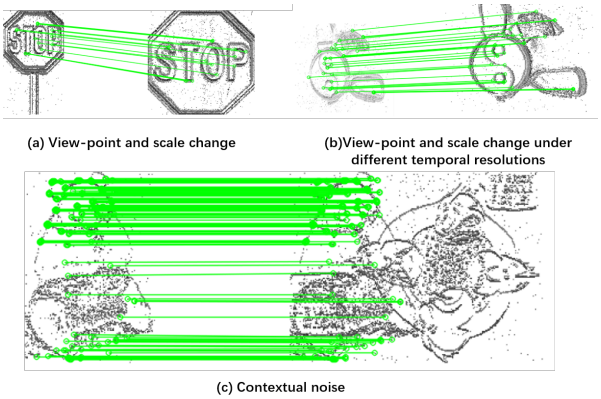

Figure 7. Qualitative results of feature matching on N-Caltech101. EventPoint is robust to view-point change, scale change, contextual background, and temporal resolution change.

4.5. Reprojection Evaluation on HVGA ATIS

HVGA ATIS Corner Dataset provides 7 sequences of planar scenes. We compute reprojection error by estimating a homography between two different timestamps as used in [23, 8]. In detail, given two timestamp T_1, T_2, and a temporal resolution δt, we firstly use Tencode represents events fall $(T_1, T_1 + \Delta t)$ and $(T_2, T_2 + \Delta t)$ into two frames, then homographic transform is estimated by matching features of this two frames. Once the homography is computed, we reproject points from time T_2 to the reference time T_1 and further compute the average distance between the reference points and the projected ones. The points detected outside the planar pattern are excluded. We compare most of the current mainstream corner detection methods [38, 24, 1, 23, 8, 4], as well as some image-based local feature extracting methods [22, 13] on the event stream.

Table 4. Reprojection Error on HVGA ATIS Corner Dataset

Methods	Representation	$25ms$	$50ms$	$100ms$
evHarris[38]	Time-surface	2.57	3.46	4.58
Arc[1]	Time-surface	3.8	5.31	7.22
evFast[24]	Time-surface	2.12	2.63	3.18
SILC[23]	Speed-invariant	2.45	3.02	3.68
SILC[23]	Time-surface	5.79	8.48	12.26
Chiberre et al.[8]	Image gradients	2.56	-	-
EventPoint(ours)	Time-surface	1.46	<u>1.57</u>	<u>1.89</u>
EventPoint(ours)	Time-window	<u>1.41</u>	1.61	2.39
EventPoint(ours)	Tencode	**1.27**	**1.41**	**1.72**

Most of the existing works are limited to corner detection without description, or only the local region around the detected corner event is considered a description. In this evaluation experiment, Δt is set to $10ms$, and the margin of two timestamp, i.e., $T_2 - T_1$ is set to $25ms$, $50ms$, $100ms$. The DSEC and HVGA ATIS Corner datasets have the same resolution of $360 * 480$. In order to verify the generalization ability of EventPoint, we train it only using the DSEC dataset but test on the HVGA ATIS Corner Dataset. To prove the impact of different event stream representations, EventPoint is trained and tested under three different representations, i.e. Time-surface, Time-window, and the proposed Tencode. We use an OpenCV implementation, i.e, $findHomography()$ and $RANSAC$, with all the matches to compute the final homography estimation.

From Tab. 4, it can be seen that our method achieves the lowest reprojection error among all methods, and remains stable as the margin increasing between the two timestamps. The experimental results show that our network learns a temporal representation invariance of corners on the event stream. Fig. 6 visualizes the detection result comparing the baselines. We use the proposed representation method Tencode with non-polarity separation avoiding the problem of detecting redundant corners at the polarity junction by the Time-surface-based methods. Thereby the reprojection error is further reduced.

5. Conclusion

In this paper, we present a novel self-supervised local feature EventPoint, including an interest point detector and a descriptor, for the event stream data. We first represent the event stream under a temporal resolution by the proposed Tencode representation. Then the EventPoint provides pixel-wise interest point locations and matches the corresponding descriptors from two dense Tencode representations. The proposed network is end-to-end trained in a self-supervised manner via homographic and spatio-temporal adaptation without expensive human annotation. The experimental evaluations demonstrate that EventPoint achieves the SOTA performance of event feature point detection and description on DSEC, N-Caltech101, and HVGA ATIS Corner datasets.

References

[1] Ignacio Alzugaray and Margarita Chli. Asynchronous corner detection and tracking for event cameras in real time. *IEEE Robotics and Automation Letters*, 3(4):3177–3184, 2018.

[2] R Baldwin, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. Event probability mask (epm) and event denoising convolutional neural network (edncnn) for neuromorphic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2020.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[4] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE transactions on neural networks and learning systems*, 25(2):407–417, 2013.

[5] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.

[6] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[7] William Oswaldo Chamorro Hernández, Juan Andrade-Cetto, and Joan Solà Ortega. High-speed event camera tracking. In *Proceedings of the The 31st British Machine Vision Virtual Conference*, pages 1–12, 2020.

[8] Philippe Chiberre, Etienne Perot, Amos Sironi, and Vincent Lepetit. Detecting stable keypoints from events through image gradient prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1387–1394, 2021.

[9] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. *arXiv preprint arXiv:1606.03558*, 2016.

[10] Peter Hviid Christiansen, Mikkel Fly Kragh, Yury Brodskiy, and Henrik Karstoft. Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. *arXiv preprint arXiv:1907.04011*, 2019.

[11] Xavier Clady, Jean-Matthieu Maro, Sébastien Barré, and Ryad B Benosman. A motion-based feature for event-based pattern recognition. *Frontiers in neuroscience*, 10:594, 2017.

[12] Konstantinos G Derpanis. The harris corner detector. *York University*, 2, 2004.

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[14] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[15] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021.

[16] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *International Conference on 3D Vision (3DV)*, 2021.

[17] Arren Glover, Aiko Dinale, Leandro De Souza Rosa, Simeon Bamford, and Chiara Bartolozzi. luvharris: A practical corner detector for event-cameras. *arXiv preprint arXiv:2105.11443*, 2021.

[18] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021.

[19] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006.

[20] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[22] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[23] Jacques Manderscheid, Amos Sironi, Nicolas Bourdis, Davide Migliore, and Vincent Lepetit. Speed invariant time surface for learning to detect corner points with event-based cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10245–10254, 2019.

[24] Elias Mueggler, Chiara Bartolozzi, and Davide Scaramuzza. Fast event-based corner detection. 2017.

[25] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[27] Penio S Penev and Joseph J Atick. Local feature analysis: A general statistical theory for object representation. *Network: computation in neural systems*, 7(3):477–500, 1996.

[28] Christoph Posch, Daniel Matolin, Rainer Wohlgenannt, Michael Hofstätter, Peter Schön, Martin Litzenberger, Daniel Bauer, and Heinrich Garn. Live demonstration: Asynchronous time-based image sensor (atis) camera with

full-custom ae processor. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 1392–1392. IEEE, 2010.

[29] Bharath Ramesh, Cheng Xiang, and Tong H Lee. Multiple object cues for high performance vector quantization. *Pattern Recognition*, 67:380–395, 2017.

[30] Bharath Ramesh, Hong Yang, Garrick Orchard, Ngoc Anh Le Thi, Shihao Zhang, and Cheng Xiang. Dart: distribution aware retinal transform for event-based cameras. *IEEE transactions on pattern analysis and machine intelligence*, 42(11):2767–2780, 2019.

[31] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019.

[32] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[33] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[34] Johannes L Schonberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1482–1491, 2017.

[35] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE international conference on computer vision*, pages 118–126, 2015.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and vision computing*, 16(2):75–87, 1998.

[38] Valentina Vasco, Arren Glover, and Chiara Bartolozzi. Fast event-based harris corner detection exploiting the advantages of event-driven cameras. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4144–4149. IEEE, 2016.

[39] Zihao W Wang, Peiqi Duan, Oliver Cossairt, Aggelos Katsaggelos, Tiejun Huang, and Boxin Shi. Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1609–1619, 2020.

[40] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European conference on computer vision*, pages 467–483. Springer, 2016.