2024 IEEE International Conference on Image Processing (ICIP) | 979-8-3503-4939-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICIP51287.2024.10647465

E2SIFT: NEUROMORPHIC SIFT VIA DIRECT FEATURE PYRAMID RECOVERY FROM EVENTS

Chris Henry[1] *Paras Maharjan*[2] *Zhu Li*[3] *George York*[4]

[1,2,3]Department of Computer Science and Electrical Engineering, University of Missouri–Kansas City
[4]Academy Center for UAS Research, United States Air Force Academy

ABSTRACT

In recent years, event cameras have achieved significant attention due to their advantages over conventional cameras. Event cameras have high dynamic range, no motion blur, and high temporal resolution. Contrary to traditional cameras which generate intensity frames, event cameras output a stream of asynchronous events based on brightness change. There is extensive ongoing research on performing computer vision tasks like object detection, classification, etc via the event camera. However, due to the unconventional output format of the event camera, it is difficult to perform computer vision tasks directly on the event stream. Mostly, works reconstruct the intensity image from the event stream and then perform such tasks. An important and crucial task is feature detection and description. Scale-invariant feature transform (SIFT) is a widely-used scale-invariant keypoint detector and descriptor that is invariant to transformations like scale, rotation, noise, and illumination. In this work, given an event voxel, we directly generate the LoG pyramid for SIFT keypoint detection. We fit a 3rd-degree polynomial and calculate the polynomial roots to compute the scale-space extrema response for SIFT keypoint detection. Since the extrema computation is performed after LoG thresholding, the solution is computationally less expensive. Experimental results validate the effectiveness of our system.

Index Terms— neuromorphic vision sensor, event camera, scale-invariant feature transform, SIFT, keypoint detection

1. INTRODUCTION

Neuromorphic or dynamic vision sensors, commonly known as event cameras, are revolutionary vision sensors that offer several benefits over traditional cameras. Contrary to traditional cameras that capture intensity images at uniform intervals, event cameras produce asynchronous event streams based on brightness changes. Event cameras record changes in luminance at each pixel independently that signal an event rather than capturing a full image frame. Event cameras have high dynamic range, no motion blur, and high temporal resolution. These properties make event cameras ideal for a wide range of fast-paced environments, such as robotics for real-time motion tracking and navigation, surveillance systems for dynamic scene monitoring, autonomous vehicles, and immersive technologies for virtual and augmented reality. The event-driven approach also offers benefits in terms of power efficiency, low latency, reduced data bandwidth, and enhanced performance in challenging lighting conditions.

The last decade has witnessed a substantial amount of research on event camera-based computer vision attributable to the superiority of event cameras over conventional cameras. Ongoing research on event-based vision includes tasks like feature detection and tracking, object recognition, object segmentation, and simultaneous localization and mapping (SLAM). In regards to event camera-based feature detection, most research works only focus on corner detection instead of keypoint detection. An extremely meager amount of research has been conducted on event camera-based keypoint detection.

The scale-invariant feature transform (SIFT) [1] is a widely used and renowned feature detector and descriptor developed by David G. Lowe. This is attributed to its robustness in tackling various image transformations like variations in scale, orientation, and illumination. The scale-space pyramid is computed by using the difference of Gaussians (DoG) which is an approximation of Laplacian of Gaussian (LoG). Keypoints are identified at locations where the DoG pyramid exhibits extreme values (maxima or minima). Spatial coordinates in the image domain and at the scale level are considered while identifying the extrema in the DoG pyramid. Once the keypoints are determined, a descriptor is calculated around each point which generates a feature vector that incorporates information about the local image structure.

The computation of the LoG pyramid is computationally more expensive when compared to the DoG pyramid which is basically subtraction of matrices. Since DoG is an approximation of LoG, there is a possibility of information loss and consequently accuracy loss in keypoint detection. In

The views expressed in this article, book, or presentation are those of the author and do not necessarily reflect the official policy or position of the United States Air Force Academy, the Air Force, the Department of Defense, or the U.S. Government. Approved for public release: distribution unlimited. PA#: USAFA-DF-2024-472

This work is supported by NSF award 2148382.

this work, we extract SIFT keypoints directly from the asynchronous event stream captured via the event camera. The proposed system directly learns to generate the LoG pyramid from an event voxel. Given the LoG pyramid, we propose an alternate solution based on polynomial fitting and root finding for computing scale-space extrema. This solution is applied after eliminating non-maximum responses by LoG peak thresholding which results in an overall lower complexity as compared to the original SIFT paper [1]. The main contributions are mentioned below:

1. A direct SIFT keypoint detection scheme from event camera sensor data is proposed. The proposed system does not require conversion of events to intensity frames. Instead of using dense intensity frames, a sparse SIFT feature pyramid recovery via learning is developed.

2. We learn recovery of the LoG pyramid directly via event stream rather than approximating the LoG pyramid via DoG. A SIFT based on a new polynomial fitting and root finding algorithm for scale-space extrema detection is proposed.

The remainder of the paper is organized as follows. Section 2 presents the related works whereas Section 3 describes our proposed approach. Experimental setup and experimental results are presented in Section 4 and Section 5, respectively. Conclusions are presented in the last section of this paper.

2. RELATED WORKS

A major portion of event-driven feature detectors focuses on event-based corner detectors. The initial event-based corner detectors can be traced back to [2, 3]. In these methods, local optical flow [4] was estimated by fitting spatiotemporal planes to the event stream. Events lying in the intersection between two planes were identified as corner-events. The studies [2, 3] can also track such corner-events via predictive velocity models. The study in [5] used artificial frames generated from events and applied the Harris corner detector [6] to a small vicinity around each event pixel location. The study in [7] proposed a more efficient corner-event detector than the one proposed in [8]. It was inspired by the FAST feature detector [9] and was locally applied to Surface of Active Events (SAE) achieving faster but slightly mediocre performance compared to [8].

Arc - a computationally efficient corner-event detector was proposed in [10]. Arc [10] ensured better corner detection repeatability than [7]. FA-Harris detector [11] built a fast and asynchronous corner detection pipeline that produced a more refined output by merging features from Arc [10], eHarris [12], and [7]. The study in [13] adapted the frame-based ACJ detector to an event camera and dubbed it as e-ACJ. It used Arc [10] to improve the speed of the original ACJ detector. [14] introduced a learning-based corner-event detector that used a motion-invariant, event-driven SAE and Random Forest to discriminate corner-events from non-corner-events. The work in [15] extracted SIFT-like descriptors [1] for the corner-events detected in [11]. It also tracked the corner-events using an approach similar to [16]. Asynchronous spatial image convolutions were introduced for event cameras by the study in [17]. These convolutions are beneficial for tasks like corner-event detection. A thorough evaluation of various corner-event detectors [10, 11, 4, 17, 12] was presented in [18].

Although there has been ongoing research on event-driven feature detectors, most of the previous works concentrate their efforts on detecting event-corners rather than event keypoints. The subtle distinction between corners and keypoints is that keypoints may include corners, edges, or other distinctive structures whereas corners can ideally only include corner points. In our work, to the best of our knowledge, we present the first-ever approach to detect SIFT keypoints directly from the event stream.

3. PROPOSED METHOD

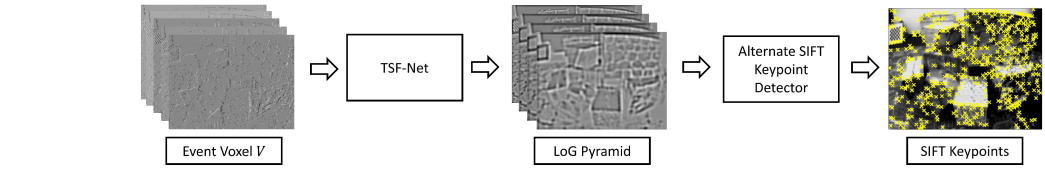
3.1. Overview

The goal of this work is to translate a continuous event stream into SIFT keypoints without intensity image reconstruction. The continuous stream of events is converted into LoG pyramid LoG_k, where LoG_k is the k-th Laplacian of Gaussian at sigma s_k. For SIFT keypoint detection, LoG_k is input to our alternate SIFT keypoint detector that is based on polynomial fitting and root finding. The overall workflow of the proposed solution is illustrated in Fig. 1a.

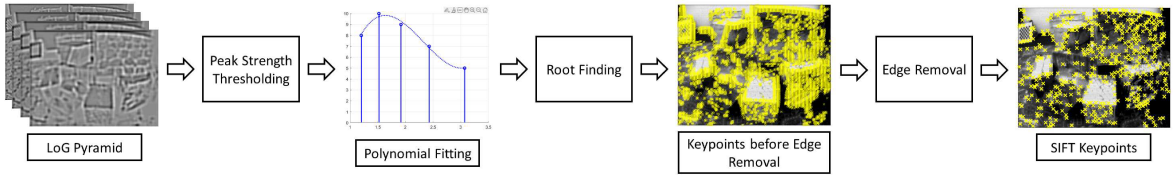
3.2. Event Representation

Each pixel in an event camera individually responds to changes in the spatio-temporal brightness signal $L(x, t)$. These changes are encoded as a stream of asynchronous events where each event $e_i = (u_i, t_i, p_i)$ is activated at pixel $u_i = (x_i, y_i)^T$ and time t_i after the change in brightness (since the last event) reaches a threshold $\pm C$. p_i denotes the polarity of the event.

We divide the event stream into equal non-overlapping spatiotemporal windows ϵ_k of fixed duration. These spatiotemporal windows ϵ_k are converted into fixed tensor representations for input to the network for learning. Inspired by E2VID [19], we choose the spatio-temporal voxel grid [20] to encode the event windows. Given an event window ϵ_k consisting of N number of events, [20] divides the timestamp range $\triangle T = t_{N-1} - t_0$ into B bins. The timestamps are then scaled to the range $[0, B-1]$ to generate the event voxel V as follows:



(a)



(b)

Fig. 1: (a) Overall workflow for our proposed system; (b) Overall workflow of the proposed alternate keypoint detection.

$$V(x,y,t) = \sum_i p_i \max\left(0, 1-|x-x_i|\right) \times \max\left(0, 1-|y-y_i|\right) \times \max\left(0, 1-|t-t_i^*|\right), \quad (1)$$

where $t_i^* = (B-1)(t_i - t_0)/\triangle T$ represents the normalized timestamp. Similar to E2VID [19], we chose $B = 5$ in our experiments.

3.3. Training Data Generation

The proposed system requires input data in the form of event voxels to generate the LoG pyramid. Event voxels are generated from event sequences by using the method mentioned in Section 3.2. Each event voxel is associated with a ground-truth intensity frame and its corresponding LoG pyramid. Event voxels are selected from each event sequence based on strict criteria that ensure the optimal event voxel and ground truth pair.

Firstly, we extract events from an event sequence within a fixed time duration t_{dur}. The ground truth intensity frame is selected based on the timestamp closest to the timestamp of the latest event within the extracted event window E_{win}. The selected intensity frame is at times almost completely black. Such event voxel/intensity frame pairs make the network difficult to converge. To remedy this, we compute SIFT keypoints [1] for the intensity frame and ignore the frame if the number of SIFT keypoints detected is below a threshold value th_{kp}.

Event voxels can have an undefined range which causes problems in network convergence. To tackle this, only voxels within a specific range $\pm r_{vox}$ are chosen. In addition, certain bins in the voxel may have almost no events which makes it almost impossible to recover the LoG pyramid. This is fixed by checking the standard deviation of events in each bin. If the standard deviation within an event bin is less than a threshold value th_{sd}, those voxels are ignored. Lastly, we also limit the number of events within a fixed-duration event window. Since the events are captured asynchronously, a 5 second window may contain fewer (e.g. 100) events or more (e.g. 100,000) events, both of which make it challenging to recover the LoG pyramid. Based on these criteria, we generate high-quality event voxel/intensity image pairs. The LoG pyramid is computed from the intensity frame and is used as ground truth while training our network. Since values in event voxels and the LoG pyramid can be in an undefined range, we clip the values in the event voxels and the LoG pyramid to $\pm c_{vox}$ and $\pm c_{log}$, respectively. This is necessary to normalize the event voxels and the LoG pyramid between 0 to 1 for input to the network.

3.4. Events to LoG Pyramid

Given an event voxel V, the goal is to learn to generate LoG pyramid LoG_k without the intensity image reconstruction. We utilize the Transformer-based Spatial and Frequency Decomposed Feature Fusion Network (TSF-Net) from the study in [21]. The TSF-Net [21] receives a total of five different inputs, out of which three are in the spatial domain while the remaining two are in the frequency domain. For our task, we modify the network to receive two inputs - one is the pixel unshuffled input and the other is the frequency-decomposed input. These two inputs are generated from the event voxel V.

The TSF-Net consists of three main parts - head, body, and tail. We only keep two of the **"conv→PReLU→ conv"** blocks in the head and remove all other blocks. The output from the two **"conv→PReLU→ conv"** blocks is input to the cascade of residual fusion blocks (RFB). The RFB fuses the spatial and frequency-decomposed features. We only use the spatial feature from the cascade of RFB. Finally, the spatial feature from the cascade of RFB is input to the tail of TSF-

Net. The tail consists of a conv layer followed by a pixel-shuffle layer. The model was trained using the following loss function:

$$Loss = L1 + SSIM \tag{2}$$

where $L1$ and $SSIM$ represent L1 loss and SSIM loss, respectively. The final output is the LoG pyramid that will be used for the LoG-based SIFT keypoint detection scheme as mentioned in the following subsection.

3.5. Neuromorphic SIFT Keypoint Detector

We propose modifications to the scale-space extrema detection step in the fast LoG SIFT keypoint detector introduced in [22]. Particularly, we propose a more analytical extrema detection solution that uses roots of a quadratic function to detect maxima. The block diagram for neuromorphic SIFT can be seen in Fig. 1b.

Peak strength thresholding of the LoG response generated via TSF-Net [21] is used to reduce computation cost. Only pixels that exceed the peak threshold P_{th} are retained. For each remaining pixel or keypoint candidate, a third-degree polynomial $P(\sigma)$ is fitted to the peak responses obtained at different scales. This is followed by calculating the first-order derivative $P'(\sigma)$ of the fitted polynomial curve. The derivative of the third-degree polynomial curve is a quadratic function. To identify the maxima, we calculate the roots α and β of the quadratic function and evaluate the polynomial function at these roots. For real roots, one of the roots will give us the maxima σ_{max}. If the absolute value of the maxima is greater than threshold G_{th}, then it is considered a keypoint.

Finally, similar to the original SIFT [1], the edge removal is performed to eliminate keypoints with low contrast. The following condition is checked for edge removal:

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r} \tag{3}$$

where H is a 2×2 Hessian matrix, $\mathrm{Tr}(H)$ is the trace of H, $\mathrm{Det}(H)$ is the determinant of H, and r is the ratio between the largest magnitude eigenvalue and the smaller one.

4. EXPERIMENTAL SETUP

4.1. Dataset Details

The Event Camera Dataset [23] and Vimeo-90k dataset [24] were used in our work. Event Camera Dataset [23] is a real event camera dataset while event data is synthesized via ESIM [25] for Vimeo-90k dataset [24]. Details about these datasets are as follows:

Vimeo-90k dataset [24] is a large-scale video dataset containing 89,800 videos covering a diverse range of scenes and actions. The videos are downloaded from vimeo.com and the dataset is tailored for video processing tasks like temporal frame interpolation, video denoising, video deblocking, and video super-resolution. We scraped videos using the list containing links for full-length original videos. The list contained $5,831$ video links out of which only $3,279$ were available for download at the time of writing this paper. ESIM [25] was used to synthesize event data from these downloaded videos. The default parameters as set in the official code repository for ESIM [26] were used while generating synthetic events. The videos were upsampled to a higher frame rate using frame interpolation before event synthesis.

Event Camera Dataset [23] is a real event camera dataset captured via a DAVIS240C sensor [27]. It contains 25 sequences that include both synthetic and real environments. The dataset provides asynchronous event streams along with intensity images captured at 24 Hz. In addition, inertial measurements, ground truth camera poses, and intrinsic camera matrices are also included. All this information is accompanied by accurate timestamps. We used 6 sequences for testing our system and the remaining are used for training. The sequence 'office_zigzag' was excluded since none of the generated event voxels passed our selection criteria as mentioned in Section 3.3.

The event streams were processed to create event voxels along with the respective LoG pyramid using the procedure mentioned in Section 3.3. Training data includes both real events (Event Camera Dataset [23]) and synthetic events (Vimeo-90k dataset [24]) whereas testing is performed on only real events (a subset from Event Camera Dataset [23]). A total of $11,873$ event voxels were generated for training and $3,383$ event voxels were generated for testing our system.

4.2. Implementation Details

The modified TSF-Net used in this paper was implemented using the PyTorch framework and was based on the implementation from authors of TSF-Net [21]. The E2VID [19] implementation used in this work was taken from [28]. Both networks were trained and tested using a single NVIDIA RTX A5000 GPU on a desktop computer with a 12th Gen Intel Core i7-12700 processor and 64 gigabytes of RAM. TSF-Net and E2VID were trained for 200 epochs with a batch size of 32 and an initial learning rate of 0.0001. Adam optimizer with $\beta = (0.9, 0.999)$ was used and a cosine annealing learning rate scheduler was used to decrease the learning rate until $1e-6$. Both networks were trained on random crops of 160×160 and tested on center crops of 160×160. E2VID's non-recurrent version as provided in [28] was trained from scratch for the task of LoG pyramid recovery.

For voxel creation, we used a fixed time duration of 0.05 seconds, SIFT keypoint threshold th_{kp} of 50 keypoints, $\pm r_{vox}$ of ± 20, standard deviation threshold th_{sd} of 0.1, $\pm c_{vox}$ of ± 2.5, $\pm c_{log}$ of ± 0.15, and number of bins $B = 5$. The event voxels were normalized between 0 to 1 before inputting into the network. For the alternate SIFT keypoint

detection scheme, we used $P_{th} = 0.03$, $G_{th} = 0.05$, and $r = 10$.

5. EXPERIMENTS

5.1. Events to LoG Pyramid

In this subsection, we evaluate the effectiveness of our proposed events to the LoG pyramid solution. Sequences from the Event Camera Dataset [23] were used for the evaluation of LoG pyramid recovery. Peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and mean squared error (MSE) were selected as evaluation metrics to evaluate the LoG pyramid recovery process. Table 1 compares the LoG pyramid recovery by our modified TSF-Net [21] with E2VID [19]. SSIM, PSNR, and MSE for individual sequences are presented in Table 1. It is evident from Table 1, that TSF-Net [21] outperforms E2VID [19] in all the evaluation metrics. In terms of MSE (↓), TSF-Net achieves an average score of 0.0038 whereas E2VID achieves an average score of 0.053. A significantly better PSNR (↑) average of 25.0064 dB was achieved by TSF-Net [21] in contrast to E2VID's average PSNR (↑) of 23.7194 dB. In case of SSIM (↑), E2VID [19] achieved an average SSIM (↑) of 0.7969 whereas TSF-Net [21] achieved an average SSIM (↑) of 0.8407.

A sample of LoG pyramid recovery can be visualized in Fig. 2. It can be observed in Fig. 2 that the proposed LoG pyramid recovery is superior in terms of MSE (↓), PSNR (↑), and SSIM (↑) as compared to the LoG pyramid recovery by E2VID [19]. These values validate that the LoG pyramid recovered via TSF-Net [21] is closer to the ground truth than the one recovered by E2VID [19].

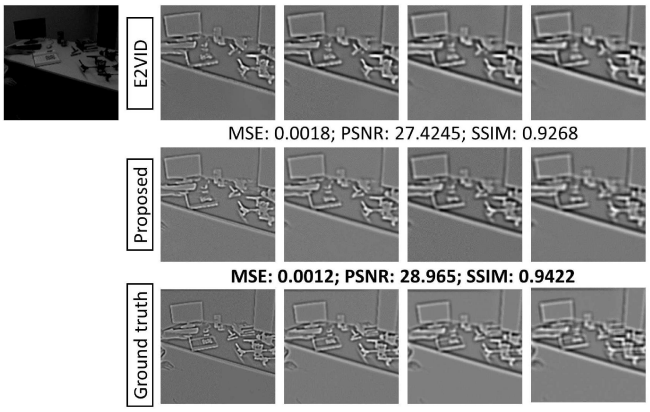


Fig. 2: LoG pyramid recovered by E2VID [19] and TSF-Net [21].

5.2. Neuromorphic SIFT Keypoint Detector

In this subsection, we evaluate the performance of our proposed event-based SIFT keypoint detector. First, we compare the SIFT keypoint repeatability between keypoints generated from the ground truth LoG pyramid and the ones generated via the predicted LoG pyramid. We chose to compare our neuromorphic SIFT to LoG-based SIFT rather than the original DoG-based SIFT. This was inspired by the study [22] which established that LoG-based SIFT keypoint detection performs better than the original DoG-based SIFT keypoint detection.

Table 2 presents accuracy for different sequences from the Event Camera Dataset [23]. This basically indicates how many keypoints are matched between the SIFT computed using the ground truth LoG pyramid and the predicted LoG pyramid. A tolerance level of 5 pixels was used while computing this accuracy. A mean accuracy of 58.9117% was obtained as can be seen in Table 2. Fig. 3 presents some qualitative results for our neuromorphic SIFT. As evident from Fig. 3, the proposed system is able to detect keypoints similar to the ones detected via the ground truth Log pyramid.

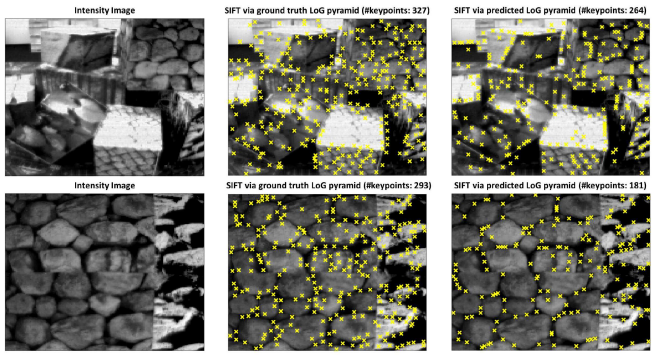


Fig. 3: Some visual results for the SIFT keypoints detected. Left: Intensity image; Center: SIFT keypoints via the ground truth LoG pyramid; Right: SIFT keypoints via the predicted LoG pyramid

Table 2: Accuracy of matched SIFT keypoints between keypoints detected via the ground truth LoG pyramid and the predicted LoG pyramid.

Dataset	Accuracy %
boxes_6dof	54.19
calibration	62.07
dynamic_6dof	61.01
poster_6dof	53.15
shapes_6dof	72.44
slider_depth	50.61
Mean	58.9117

Table 1: LoG pyramid recovery results for E2VID [19] and TSF-Net [21] on the Event Camera Dataset [23] in terms of MSE, PSNR, and SSIM. The best performance is shown in bold.

Dataset	MSE↓		PSNR↑ (in dB)		SSIM↑	
	E2VID	TSF-Net	E2VID	TSF-Net	E2VID	TSF-Net
boxes_6dof	0.0044	**0.0038**	24.0130	**24.6261**	0.7676	**0.8096**
calibration	0.0103	**0.0065**	20.0708	**22.0494**	0.7134	**0.7861**
dynamic_6dof	0.0031	**0.0027**	25.5539	**26.3663**	0.8885	**0.9114**
poster_6dof	0.0028	**0.0022**	25.9942	**27.0566**	0.8502	**0.8806**
shapes_6dof	0.0032	**0.0027**	25.7975	**26.4769**	0.9038	**0.9149**
slider_depth	0.0082	**0.0046**	20.8874	**23.4632**	0.6579	**0.7414**
Mean	0.0053	**0.0038**	23.7194	**25.0064**	0.7969	**0.8407**

In addition, we also evaluate the effectiveness of the proposed root-based extrema-finding solution. The red line in the left and right plots in Fig. 4 represent the maxima detected via [22] and via our proposed solution, respectively. It is evident from Fig. 4 that the root-based extrema solution performs similarly to the solution in [22].

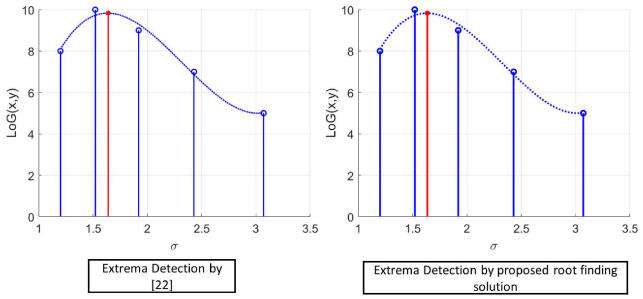


Fig. 4: Extrema detection by [22] and by our proposed root finding solution

5.3. Discussion

It is understandable that the proposed neuromorphic SIFT has room for further improvements, especially in regard to its keypoint repeatability score. For instance, similar to the original SIFT [1] work, one such improvement would be utilizing LoG pyramids at different octaves. Using the octave-based approach would also improve the edge-removal process. Another improvement would be the addition of the keypoint description. With that being said, we consider our neuromorphic SIFT to be a seminal work in event-based keypoint detection.

6. CONCLUSIONS

In this work, we presented neuromorphic SIFT that can extract SIFT keypoints from the event stream captured by neuromorphic vision sensors. The proposed solution learned to generate the LoG pyramid directly from the event stream. The learning-based LoG pyramid was then processed by the proposed alternate LoG-based SIFT keypoint detector. The proposed system was able to generate SIFT keypoints that match the keypoints generated by ground truth with a mean accuracy of 58.9117%. We consider this work to be foundational for further research on event-driven keypoint detection. The proposed solution was tested on a real event camera dataset and the experimental results affirm the efficacy of the introduced neuromorphic SIFT.

7. REFERENCES

[1] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[2] Xavier Clady, Sio-Hoi Ieng, and Ryad Benosman, "Asynchronous event-based corner detection and matching," *Neural Networks*, vol. 66, pp. 91–106, 2015.

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

[4] Elias Mueggler, Christian Forster, Nathan Baumli, Guillermo Gallego, and Davide Scaramuzza, "Lifetime estimation of events from dynamic vision sensors," in *2015 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4874–4881.

[5] Yi Zhou, Guillermo Gallego, Henri Rebecq, Laurent Kneip, Hongdong Li, and Davide Scaramuzza, "Semi-dense 3d reconstruction with a stereo event camera," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 235–251.

[6] Chris Harris, Mike Stephens, et al., "A combined corner and edge detector," in *Alvey vision conference*. Citeseer, 1988, vol. 15, pp. 10–5244.

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

[8] Valentina Vasco, Arren Glover, Elias Mueggler, Davide Scaramuzza, Lorenzo Natale, and Chiara Bartolozzi, "Independent motion detection with event-driven cameras," in *2017 18th International Conference on Advanced Robotics (ICAR)*. IEEE, 2017, pp. 530–536.

[9] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 430–443.

[10] Ignacio Alzugaray and Margarita Chli, "Asynchronous corner detection and tracking for event cameras in real time," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3177–3184, 2018.

[11] Ruoxiang Li, Dianxi Shi, Yongjun Zhang, Kaiyue Li, and Ruihao Li, "Fa-harris: A fast and asynchronous corner detector for event cameras," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6223–6229.

[12] 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)*. IEEE, 2016, pp. 4144–4149.

[13] Zhihao Liu and Yuqian Fu, "E-acj: Accurate junction extraction for event cameras," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 2603–2607.

[14] 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*, 2019, pp. 10245–10254.

[15] Ruoxiang Li, Dianxi Shi, Yongjun Zhang, Ruihao Li, and Mingkun Wang, "Asynchronous event feature generation and tracking based on gradient descriptor for event cameras," *International Journal of Advanced Robotic Systems*, vol. 18, no. 4, pp. 17298814211027028, 2021.

[16] Ignacio Alzugaray and Margarita Chli, "Ace: An efficient asynchronous corner tracker for event cameras," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 653–661.

[17] Cedric Scheerlinck, Nick Barnes, and Robert Mahony, "Asynchronous spatial image convolutions for event cameras," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 816–822, 2019.

[18] Özgün Yılmaz, Camille Simon-Chane, and Aymeric Histace, "Evaluation of event-based corner detectors," *Journal of Imaging*, vol. 7, no. 2, pp. 25, 2021.

[19] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 6, pp. 1964–1980, 2019.

[20] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis, "Unsupervised event-based optical flow using motion compensation," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[21] Birendra Kathariya, Zhu Li, and Geert Van der Auwera, "Joint pixel and frequency feature learning and fusion via channel-wise transformer for high-efficiency learned in-loop filter in vvc," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[22] Paras Maharjan, Lyle Vanfossan, Zhu Li, and Jerry Jialie Shen, "Fast log sift keypoint detector," in *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2023, pp. 1–5.

[23] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, 2017.

[24] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, pp. 1106–1125, 2019.

[25] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza, "Esim: an open event camera simulator," in *Conference on robot learning*. PMLR, 2018, pp. 969–982.

[26] Robotics and Perception Group, "Video to events: Recycling video datasets for event cameras," `https://github.com/uzh-rpg/rpg_vid2e`, Accessed: 2024-Jan-31.

[27] R Berner, C Brandli, M Yang, SC Liu, and T Delbruck, "A 240x180 130 db 3 s latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State*, 2013.

[28] Robotics and Perception Group, "High speed and high dynamic range video with an event camera," `https://github.com/uzh-rpg/rpg_e2vid`, Accessed: 2024-Jan-31.