Research Article

Towards End-to-End Neuromorphic Voxel-based 3D Object Reconstruction Without Physical Priors

Chuanzhi Xu[1], Langyi Chen[1], Vincent Qu[1], Haodong Chen[1], Vera Chung[1]

1. University of Sydney, Australia

Neuromorphic cameras, also known as event cameras, are asynchronous brightness-change sensors that can capture extremely fast motion without suffering from motion blur, making them particularly promising for 3D reconstruction in extreme environments. However, existing research on 3D reconstruction using monocular neuromorphic cameras is limited, and most of the methods rely on estimating physical priors and employ complex multi-step pipelines. In this work, we propose an end-to-end method for dense voxel 3D reconstruction using neuromorphic cameras that eliminates the need to estimate physical priors. Our method incorporates a novel event representation to enhance edge features, enabling the proposed feature-enhancement model to learn more effectively. Additionally, we introduced *Optimal Binarization Threshold Selection Principle* as a guideline for future related work, using the optimal reconstruction results achieved with threshold optimization as the benchmark. Our method achieves a 54.6% improvement in reconstruction accuracy compared to the baseline method.

I. Introduction

3D reconstruction in VR/AR applications enables realistic restoration of scenes and objects, serving as a 3D form of information present that provides users with a more immersive experience [1][2]. Many devices can be used to collect data for 3D reconstruction, including traditional RGB cameras, RGB-D cameras, LiDAR, structured light systems, etc. However, they have different limitations, including limited dynamic range, motion blur, high power consumption, etc. [3].

Neuromorphic cameras, also known as event cameras, are bio-inspired sensors responding to local brightness changes [3]. Each pixel in a neuromorphic camera operates independently and asynchronously, which differs from traditional (frame-based) RGB cameras that capture pictures with a shutter and have a fixed interval in recording video [3][4][5]. Neuromorphic cameras report changes in brightness only when a threshold is reached. When there is a greater change in brightness in the scene or object, such as when the object is moving faster, more event data will be generated. The neuromorphic camera produces a continuous stream of events, which includes the coordinates, precise timestamp, and the polarity of brightness change.

Neuromorphic cameras in 3D reconstruction tasks can be divided into stereo and monocular types. Stereo neuromorphic cameras involve multiple rigidly connected neuromorphic cameras, providing information from multiple viewpoints.

 doi.org/10.32388/NU8FBX

These methods typically perform scene scanning and produce real-time semi-dense reconstruction results. They often follow the classical two-step stereo solution: matching disparities at the same timestamps and then triangulating 3D points to calculate the depth information of every point in the scene [6][3]. However, performing 3D reconstruction with a monocular neuromorphic camera, which lacks disparity information, often requires more complex computations. Such methods must be combined with the physical prior knowledge (e.g., camera trajectories) to achieve similar results as parallax in the stereo task. The physical prior is always required, and it can either be obtained by simultaneous mapping with other devices or be predicted by a Visual Odometer (VO) or SLAM algorithm targeting the event stream [6]. Moreover, these methods all require a complex event-to-3D pipeline [7][8][9][10]. The pipeline is a full workflow that optimally processes event data and reconstructs it into a 3D model, including steps such as event representation, prior estimation, computation of parallax, triangulation, depth estimation, and 3D model reconstruction. Event processing pipelines are a common research direction in these tasks. A recent study introduced a neural network, E2V [1], capable of directly taking the represented event stream as input and outputting voxel results. This approach made progress in simplifying the event processing pipeline. It is also the first method to perform voxel-based 3D reconstruction using a monocular neuromorphic camera. However, it leaves substantial room for improvement in reconstruction accuracy.

In this research, we propose an end-to-end method for dense voxel 3D reconstruction using neuromorphic cameras that eliminates the need to estimate physical priors. Our contributions can be summarized as:

- We propose a novel event representation method, *Sobel Event Frame*, which enhances edge features and restrains redundant data in the event stream, enabling effective learning of 3D features.
- We propose a 3D reconstruction model designed to enhance the learning of edge features in event streams with Efficient Channel Attention, building on the approach of event-based 3D reconstruction without the need for physical priors or pipelines.
- We propose *Optimal Binarization Threshold Selection Principle* and suggest it as a guideline for future research about event-based 3D reconstruction with deep learning.

II. Related Work

A. Physics and Geometry-based Methods

Most methods for performing spatial scanning and instantaneous 3D reconstruction using a monocular neuromorphic camera are based on physical and geometric computations. These methods establish strict event processing pipelines, requiring steps such as feature extraction and feature matching, and must compute physical prior information.

Kim et al. proposed the first method for depth estimation using a moving monocular neuromorphic camera [11], which mainly consists of three decoupled probabilistic filters that estimate the 6-DoF motion of the camera, the scene intensity gradient, and the inverse depth of the scene relative to keyframes. Rebecq et al., in 2016, proposed an event-based visual odometry algorithm (EVO) [10]. The tracking module in EVO uses Event Frame to represent event streams, generate edge images, and estimate the camera pose. Its mapping module expands the semi-dense 3D map when new events arrive and feeds the map back into the tracking module. The EMVS method proposed by Rebecq et al. in 2018,

based on a known camera trajectory [6], back-projects events into space to create a light-density volume [12]. It identifies the edge structures of the scene as the maxima of the light density to produce a semi-dense depth map. Multiple viewpoints are then used to fuse the depth maps and complete 3D reconstruction. Guan et al. introduced a new hybrid tracking and dense mapping system based on neuromorphic cameras called EVI-SAM [13]. The pipeline of EVI-SAM includes two parallel modules: an Event Visual-Inertial Odometry (EVIO) module for tracking and estimating camera poses and an event-driven mapping module.

B. Deep Learning-based Methods

Recent methods using monocular neuromorphic cameras for 3D reconstruction rely on existing synthetic or real event datasets to train deep learning models for dense 3D reconstruction results. However, pipelines and the estimation of physical priors remain indispensable.

Baudron et al. proposed the E3D method, including a pipeline with a neural network [7], E2S, to convert event frames into contours while using an additional neural branch for camera pose regression, ultimately generating multi-view mesh reconstruction results. Xiao et al. utilized the E2VID [4] deep learning method to process continuous event streams and output normalized intensity image sequences [8]. They used Structure-from-Motion (SfM) to estimate sparse intrinsic, extrinsic point clouds, followed by Multi-View Stereo (MVS) techniques to complete dense reconstruction. Wang et al. proposed a method that includes a physical prior extraction branch and a NeRF [14] rendering branch [9]. By incorporating an event warping field and a deterministic event generation model, they integrated physical priors into the NeRF pipeline. They also introduced a novel probabilistic chunk sampling strategy based on spatial event density, which helps the model learn local geometric features more robustly and efficiently.

The study done by Chen et al. in 2023 serves as the baseline for this research [1]. It is the first study to use the voxel grid to represent event-based 3D reconstruction results. The study primarily introduces a deep learning model, E2V, which consists of a 3D event frame encoder to transform the feature representation of the data and a 3D voxel decoder to convert these features into a 3D voxel grid. Additionally, this research proposed a dataset, SynthEVox3D, for event-based 3D reconstruction. This method opened up a new direction for event-based 3D reconstruction, while its reconstruction accuracy on mIoU was only 0.346, which shows room for further improvement.

III. Method

We propose an end-to-end method for dense voxel 3D reconstruction using monocular neuromorphic camera that eliminates the need to estimate physical priors.

A. Novel Event Representation: Sobel Event Frame

Event representation refers to preprocessing the data captured by the neuromorphic camera, which encodes the information of the event's coordinate, timestamp, and polarity.

Event Frame is the most commonly used event representation method, referring to all approaches that can frame event streams. Event Frame divides event data into multiple fixed-length time windows, either based on timestamp or event

count. Each time window generates an image-like frame where each pixel represents whether an event occurred within that time window. Pixels where no event occurred during the time window are filled with 0.

Each event frame $F_t(x, y)$ is a pixel value generated in the time window t, where (x, y) represents the pixel coordinates. $E_i(x, y)$ represent the i-th event in the event stream within time window t. p_i denote the polarity of event E_i (with $+1$ for positive polarity and -1 for negative polarity). n is the total number of events in the time window t.

We summarize all five design modes of Event Frame as follows:

a) *Last Positive Event (Pos):* If the last event at each pixel within the time window has a positive polarity, it is marked as 1, otherwise 0.

$$F_t(x, y) = \begin{cases} 1, & \text{if } p_n = +1, \\ 0, & \text{if } p_n \neq +1. \end{cases}$$

b) *Last Negative Event (Neg):* If the last event at each pixel within the time window has a negative polarity, it is marked as 1, otherwise 0.

$$F_t(x, y) = \begin{cases} 1, & \text{if } p_n = -1, \\ 0, & \text{if } p_n \neq -1. \end{cases}$$

c) *Last Event Polarity (Last):* If the last event at each pixel within the time window has a positive polarity, it is marked as 1, and if negative, it is marked as -1.

$$F_t(x, y) = p_n.$$

d) *Any Event (Any):* If any event occurs at a pixel within the time window, regardless of polarity, it is marked as 1.

$$F_t(x, y) = \begin{cases} 1, & \text{if } n > 0, \\ 0, & \text{if } n = 0. \end{cases}$$

e) *Separate Frames for Polarity (Sep):* For each time window, a positive event frame $F_t^+(x, y)$ is generated for positive events in a time window, and a negative event frame $F_t^-(x, y)$ is generated for negative events in a time window.

$$F_t^+(x, y) = \begin{cases} 1, & \text{if any~} p_i = +1, \\ 0, & \text{otherwise.} \end{cases}$$

$$F_t^-(x, y) = \begin{cases} 1, & \text{if any~} p_i = -1, \\ 0, & \text{otherwise.} \end{cases}$$

The Sobel operator is an image processing algorithm primarily used for edge detection. It computes the gradient of pixel intensity in both the horizontal (x-axis) and vertical (y-axis) directions using convolution, aiming to highlight the edges in the image [15].

In neuromorphic camera-related research, there has been some work on applying convolution to event data, but only a few studies have used the Sobel operator [16][17], and these are applied on initial event stream aimming to detect edge and structure in dynamic scenes, which can be considered as non-frame-based preprocessing method for events.

The Sobel operator has never been applied to frame-based event representations in the neuromorphic camera field. Our attempt introduces the Sobel operator to continuously highlight edges of patterns in the event frame for 3D reconstruction, and we name this method *Sobel Event Frame*. Applying Sobel Event Frame to all pixels continuously can be represented by the following formula:

$$O(x,y,t) = \sum_{w=-a}^{a} \sum_{h=-b}^{b} E(x+w, y+h, t) \cdot S(w,h) \tag{1}$$

Here, $E(x,y,t)$ denotes the event frame at time window t. $O(x,y,t)$ is the output frame after applying the convolution. $S(w,h)$ is the convolution kernel. a and b are half the height h and width w of the convolution kernel, respectively. We select a kernel size $m = n = 3$ in experiments. After extracting the gradient magnitude, we normalize it to the greyscale intensity range $[0, 255]$ for visualization. Referring to Figure 1, we can intuitively perceive the pattern's edges are enhanced.

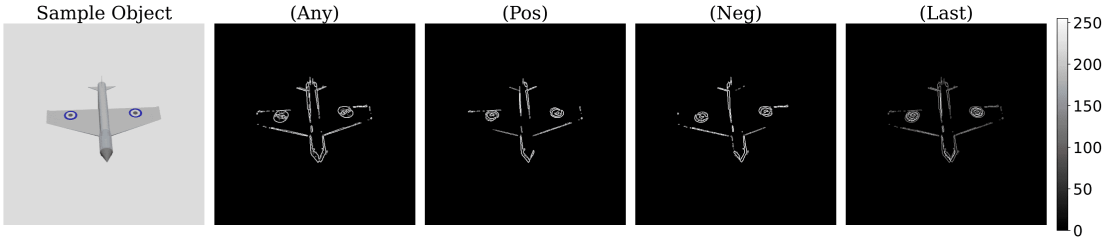

Figure 1. Visualizations of different modes of *Sobel Event Frame* applied on event stream of an airplane object.

B. Event-to-Voxel 3D ResNet with ECA

We propose a model to process event data to dense voxel 3D reconstruction, effectively incorporating the event data represented by *Sobel Event Frame* and further enhancing the learning on edge feature information.

The E2V model uses a deeper model architecture, ResNet-152, as the encoder to extract complex 3D features from the represented event stream [1]. However, due to massive information in 3D data, the extracted features may include irrelevant or redundant information (e.g., the object is primarily located in the center of the event frame, while no events occur around the edges), which limits the performance of model.

Inspired by Wang et al. [18], we introduce the Efficient Channel Attention (ECA) mechanism into the encoder with deeper convolutional layers. This mechanism aims to efficiently capture inter-channel interactions by employing a 1D convolution with an adaptively determined kernel size, thus avoiding the dimensional reduction seen in other channel attention mechanisms. It achieves high performance without increasing the complexity of the deep CNN model.

Referring to Figure 2, we integrate an ECA module after each bottleneck layer of the ResNet-152 encoder, ensuring that the model can focus on more meaningful information at every stage of feature extraction. We expect edge features strengthened by the Sobel Event Frame will be enhanced again. The ECA module first applies global average pooling to each channel, capturing the global context of the input 3D feature maps. Then, a lightweight 1D convolution is applied to the pooled features to learn the dependencies among channels. A sigmoid activation function follows, normalizing the attention scores to values between 0 and 1. This process is highly efficient because it avoids fully connected layers and directly applies attention through a 1D convolution.

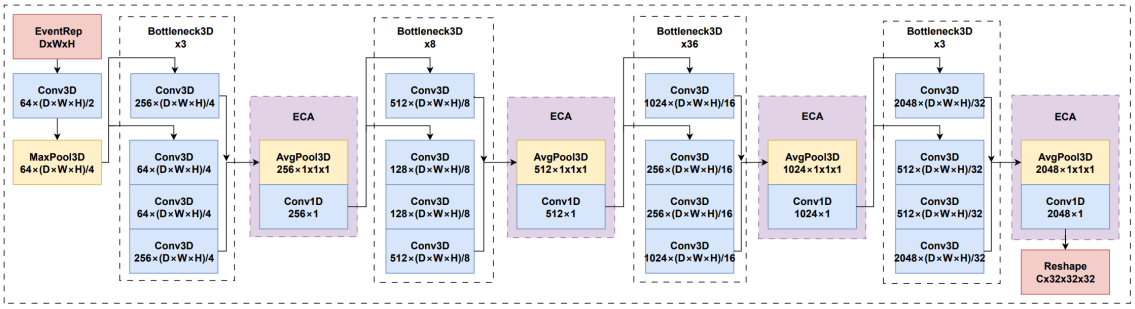

Figure 2. Enhanced 3D ResNet Encoder.

For decoder, we add a controllable output reshaping module to E2V decoder [1], learning the correlation between the event data and ground truth labels, to generate the voxel logit output.

C. Optimal Binarization Threshold Selection Principle

After generating the model, we use it to perform inference (prediction) on the testing data subset. The output of the model consists of continuous logits, which will be converted into continuous probability values $\sigma(x)$ by Sigmoid function. By setting a binarization threshold p, we can determine whether a reconstructed point is present or absent as follows:

$$\text{Output Binary Value} = \begin{cases} 1, & \text{if } \sigma(x_{\text{out}}) > p, \\ 0, & \text{if } \sigma(x_{\text{out}}) \leq p. \end{cases}$$

$$\text{where} \quad \sigma(x_{\text{out}}) = \frac{1}{1 + e^{-x_{\text{out}}}}$$

Previous studies used a fixed binarization threshold such as 20% [1] or 50% [7]. However, we found that the optimal threshold varies depending on the event representation method.

To determine the best threshold p^*, we iterate through a predefined range of thresholds P (from 0.15 to 0.50 with a step size of 0.01) and select the threshold that yields the best mIoU and F-Score as follows:

$$p^* = \arg\max_{p \in P} \begin{cases} \text{mIoU}(p), & \text{for mIoU} \\ \text{F-Score}(p), & \text{for F-Score} \end{cases} \tag{2}$$

IV. Experiments

A. Datasets and Implementation Details

Dataset: Synthetic Event Camera Voxel 3D Reconstruction Dataset (SynthEVox3D) is the only available dataset for 3D reconstruction with voxel label based on event data. SynthEVox3D consists of 39,739 event data samples in 13 categories sourced from ShapeNet [19], including Airplane, Bench, Cabinet, Car, Chair, Displayer, Lamp, Speaker, Rifle, Sofa, Table, Telephone, and Watercraft. Each category contains objects of different shapes, ensuring data differentiation. The event data are generated by all-angle scanning within $0.5s$ using a neuromorphic camera, producing event streams with a resolution of 512×512 pixels. SynthEVox3D-Tiny is a subset of SynthEVox3D, which randomly selects 80 samples from

each category, making a total of 1,040 samples. These samples include 832 training data (80%), 104 validation data (10%), and 104 testing data (10%).

Event Representation: We use a fixed time window of 5.0×10^{-3} to segment the event data into event frames, forming 100 frames (total duration of 0.5s). The karnel size of Sobel Event Frame is set to 3×3. After event representation, the event data is normalized to $[0, 1]$ and rescaled to a shape of $(n, 256, 256)$ for fitting the model input. To further prevent overfitting, we randomly apply operations such as flipping, rotation, and inversion of the event stream.

Deep Learning: We set the batch size to 5 and use the Adam optimizer with a learning rate of 1.0×10^{-6}. The optimizer is adopted with $\beta = \{0.9, 0.999\}$. The total number of training epochs is set to 100. The dropout rate is set to 0.25. The loss function used is Focal Loss due to the imbalanced data structure (polarization data).

Event Representation	Threshold	mIoU	Rifle	Chair	Car	Table	Sofa	Spkr	Airpln	Dspl	Wtrcft	Lamp	Cbn	Bench	Tel
E2V - Event Frame (Pos)	0.22	0.358	0.384	0.318	0.446	0.346	0.403	0.327	0.442	0.262	0.398	0.311	0.417	0.352	0.254
E2V - Event Frame (Any)	0.19	0.377	0.393	0.358	0.478	0.284	0.411	0.320	0.453	0.306	0.445	0.326	0.406	0.324	0.393
Ours - Sobel EvtFrm (Pos)	0.34	0.523	0.610	0.451	0.598	0.443	0.523	0.461	0.514	0.535	0.524	0.440	0.617	0.514	0.569
Ours - Sobel EvtFrm (Any)	0.33	0.512	0.655	0.437	0.569	0.451	0.539	0.458	0.570	0.607	0.515	0.372	0.548	0.372	0.560

Table I. IoU results for each category

Event Representation	Threshold	F-Score	Rifle	Chair	Car	Table	Sofa	Spkr	Airpln	Dspl	Wtrcft	Lamp	Cbn	Bench	Tel
E2V - Event Frame (Pos)	0.22	0.507	0.544	0.465	0.616	0.494	0.559	0.467	0.605	0.382	0.557	0.451	0.576	0.502	0.380
E2V - Event Frame (Any)	0.19	0.532	0.552	0.516	0.640	0.421	0.578	0.471	0.620	0.447	0.602	0.478	0.565	0.476	0.555
Ours - Sobel EvtFrm (Pos)	0.34	0.682	0.767	0.614	0.755	0.593	0.689	0.625	0.681	0.695	0.679	0.604	0.769	0.679	0.723
Ours - Sobel EvtFrm (Any)	0.33	0.667	0.796	0.596	0.726	0.616	0.702	0.610	0.728	0.757	0.675	0.528	0.695	0.520	0.719

Table II. F-Score results for each category

B. Experimental Processes & Results

First, we conducted training on the SynthEVox3D-Tiny dataset. We reproduced the E2V method using Event Frame (Pos)and evaluated it on the testing data subset as the baseline. At the optimal threshold of 0.22, the E2V method achieved mIoU of 0.358 and F-Score of 0.507.

Subsequently, we tested E2V on the other four modes of Event Frame and evaluated all five modes of Sobel Event Frameusing our method. The (Pos) and (Any) modes, which do not contain negative data, performed better under the current experimental settings. Therefore, in Tables I and II, we only present the results of them.

Using Sobel Event Frame (Pos) with a binarization threshold of 0.34, our method achieved mIoU of 0.523 and F-Score of 0.682. This represents an improvement of 0.165 (46.1%) in mIoU and 0.175 (34.5%) in F-Score compared to the E2V method with Event Frame (Pos).

Then, we conducted a full dataset evaluation on SynthEVox3D with our model and the best-performing mode: Sobel Event Frame (Pos). As shown in Table III, our method achieved a mIoU of 0.535 on the full dataset, which represents a 54.6% improvement over E2V, demonstrating the stability of our method even with extensive training data. We also compared our method with state-of-the-art voxel reconstruction methods based on multi-view traditional images. The mIoU values of these traditional methods were obtained from 20 views of objects in the ShapeNet dataset. While the datasets are not entirely comparable, our reconstruction accuracy is approaching that of these traditional methods. We believe that if both methods were compared under the same extremely rapid scanning conditions, traditional methods will suffer from motion blur, but our method will perform better, which we plan to test as a future work.

	Training Data Size	mIoU
3D-R2N2		0.636
AttSets	~30,000	0.693
Pix2Vox++		0.706
EVolT		0.735
E2V (Tiny)	832	0.358
E2V (Full)	39,739	0.346
Ours (Tiny)	832	0.523
Ours (Full)	39,739	0.535

Table III. Comparison of mIoU scores across methods.

Figure 3 presents visualizations of 12 random reconstruction results from the testing data, showing each object, voxel reconstruction results from our method and baseline, and ground truth labels. In the voxel reconstruction results, we use green to represent correctly reconstructed voxels and red to represent incorrectly reconstructed voxels. It can be observed that our voxel outputs perform better than the baseline. There are no outlier voxels from incorrect reconstructions surrounding the object. The category of the object can be easily identified.

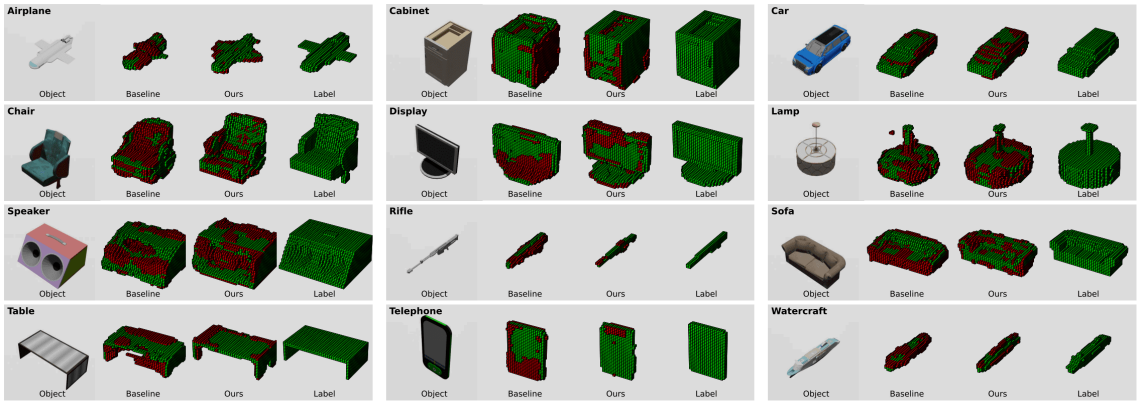

Figure 3. Visualization of 3D voxel reconstruction results.

C. Ablation Study

Our method can be divided into three components: event representation, model, and binarization threshold selection, for conducting ablation studies to evaluate the contribution of each component to the reconstruction results.

Event Representation and Model: We conducted ablation studies on event representation and the model using the SynthEVox3D-Tiny dataset, referring to Table IV for the results. By using *Event Frame (Pos)* to process event data as input to our proposed model, we obtained a mIoU of 0.421 at a threshold of 0.21, which means our model brought about a 17.6% improvement. Using the E2V model, we also trained and tested with *Sobel Event Frame (Pos)*, achieving a mIoU of 0.471 at a threshold of 0.33. In other words, relying solely on *Sobel Event Frame* resulted in a 31.6% improvement. The combination of both led to a further enhancement, reaching a mIoU of 0.523, a 46.1% improvement over the baseline.

EventRep	Model	Threshold	mIoU
Event Frame (Pos)	E2V	0.22	0.358
Event Frame (Pos)	Ours	0.21	0.421 (17.6%↑)
Sobel EvtFrm (Pos)	E2V	0.33	0.471 (31.6%↑)
Sobel EvtFrm (Pos)	Ours	0.34	0.523 (46.1%↑)

Table IV. Ablation Study Results on Event Representation and Model Variants

Binarization Threshold: To demonstrate that the binarization threshold should be optimally selected rather than fixed, we listed the mIoU and F-Score results using *Sobel Event Frame (Pos)* at thresholds ranging from 0.16 to 0.42 (with a step size of 0.02), as shown in Table V. The best results were achieved at a threshold of 0.34, reaching a mIoU of 0.523 and an F-Score of 0.682. Performance was slightly worse at thresholds of 0.32 and 0.36. Deviating from the threshold of 0.34 in either direction led to poorer results. The E2V method suggests using 0.20 as a reference threshold [1], but this did not yield the best results in our experiments. Within this range, using the worst threshold (0.16) would result in a mIoU decrease of 0.035. Additionally, according to Table IV, applying different event representation methods had a significant impact on the optimal threshold, while the model only had a smaller effect on the optimal threshold.

Threshold	mIoU	F-Score	Threshold	mIoU	F-Score
0.16	0.488	0.653	0.30	0.521	0.681
0.18	0.496	0.660	0.32	0.522	0.682
0.20	0.503	0.666	**0.34**	**0.523**	**0.682**
0.22	0.508	0.670	0.36	0.522	0.681
0.24	0.513	0.674	0.38	0.521	0.680
0.26	0.516	0.677	0.40	0.519	0.677
0.28	0.519	0.679	0.42	0.516	0.674

Table V. The mIoU and F-Score results of *Sobel Event Frame (Pos)* under different binarization thresholds.

Therefore, this demonstrates the significant impact of the binarization threshold on model performance, indicating that the optimal threshold is not fixed but needs to be optimized based on the specific event representation method. We recommend that future related tasks refer to our *Optimal Binarization Threshold Selection Principle*, flexibly adjusting the threshold to optimize results, which serves as a guideline for future work.

V. Conclusion & Future Work

In this study, we proposed a novel event representation method, *Sobel Event Frame*, and an ECA-enhanced deep learning method for end-to-end 3D reconstruction using the monocular neuromorphic camera without relying on physical priors and pipelines. Additionally, we introduced a *Optimal Binarization Threshold Selection Principle* and advocated for it as a guideline for future event-based 3D reconstruction. Under our approach, we achieved a significant improvement in reconstruction accuracy, further bridging the gap with results from traditional camera-based 3D reconstruction.

Looking ahead, we plan to expand this project further. Regarding event representation, our aim is to explore more potential representations and extend *Sobel Event Frame* to other computer vision tasks. Although we introduced *Optimal Binarization Threshold Selection Principle*, it is still necessary to investigate the factors that influence threshold selection. Most importantly, we need to test the performance of traditional methods and event-based methods under extreme conditions, such as very fast motion, low light, or high brightness scenarios, which will demonstrate the advantages of neuromorphic cameras in the field of 3D reconstruction.

References

1. a, b, c, d, e, f, g *Chen H, Chung V, Tan L, Chen X. "Dense voxel 3d reconstruction using a monocular event camera." In: 2023 9th International Conference on Virtual Reality (ICVR). IEEE; 2023. p. 30-35.*

2. ^Qu Q, Liang H, Chen X, Chung YY, Shen Y (2024). "NeRF-NQA: No-Reference Quality Assessment for Scenes Generated by NeRF and Neural View Synthesis Methods". IEEE Transactions on Visualization and Computer Graphics. **2024**. Published by IEEE.
3. a, b, c, dGallego G, Delbrück T, Orchard G, Bartolozzi C, Taba B, Censi A, Leutenegger S, Davison AJ, Conradt J, Daniilidis K, et al. "Event-based vision: A survey." IEEE transactions on pattern analysis and machine intelligence. **44** (1): 154--180, 2020.
4. a, bRebecq H, Ranftl R, Koltun V, Scaramuzza D (2019). "High speed and high dynamic range video with an event camera". IEEE Transactions on Pattern Analysis and Machine Intelligence. **43** (6): 1964–1980.
5. ^Qu Q, Shen Y, Chen X, Chung YY, Liu T (2024). "E2HQV: High-Quality Video Generation from Event Camera via Theory-Inspired Model-Aided Deep Learning." In: The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24). **38** (5): 4632–4640.
6. a, b, cRebecq H, Gallego G, Mueggler E, Scaramuzza D (2018). "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time". International Journal of Computer Vision. **126** (12): 1394–1414.
7. a, b, cBaudron A, Wang ZW, Cossairt O, Katsaggelos AK (2020). "E3d: event-based 3d shape reconstruction". arXiv preprint arXiv:2012.05214. Available from: https://arxiv.org/abs/2012.05214.
8. a, bXiao K, Wang G, Chen Y, Nan J, Xie Y. "Event-based dense reconstruction pipeline." In: 2022 6th International Conference on Robotics and Automation Sciences (ICRAS). IEEE; 2022. p. 172–177.
9. a, bWang J, He J, Zhang Z, Xu R (2024). "Physical Priors Augmented Event-Based 3D Reconstruction". arXiv preprint arXiv:2401.17121.
10. a, bRebecq H, Horstschäfer T, Gallego G, Scaramuzza D (2016). "Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time". IEEE Robotics and Automation Letters. **2** (2): 593–600.
11. ^Kim H, Leutenegger S, Davison AJ. "Real-time 3D reconstruction and 6-DoF tracking with an event camera." In: Computer Vision--ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14. Springer; 2016. p. 349-364.
12. ^Conradt J. "On-board real-time optic-flow for miniature event-based vision sensors." In: 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE; 2015. p. 1858-1863.
13. ^Guan W, Chen P, Zhao H, Wang Y, Lu P (2024). "EVI-SAM: Robust, Real-Time, Tightly-Coupled Event--Visual--Inertial State Estimation and 3D Dense Mapping". Advanced Intelligent Systems. p. 2400243.
14. ^Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R (2021). "Nerf: Representing scenes as neural radiance fields for view synthesis." Communications of the ACM. **65** (1): 99–106.
15. ^Kanopoulos N, Vasanthavada N, Baker RL (1988). "Design of an image edge detection filter using the Sobel operator". IEEE Journal of Solid-State Circuits. **23** (2): 358–367.
16. ^Brändli C, Strubel J, Keller S, Scaramuzza D, Delbruck T. "ELiSeD—An event-based line segment detector." In: 2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP). IEEE; 2016. p. 1–7.
17. ^Scheerlinck C, Barnes N, Mahony R (2019). "Asynchronous spatial image convolutions for event cameras". IEEE Robotics and Automation Letters. **4** (2): 816–822.

18. ^*Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q (2020). "ECA-Net: Efficient channel attention for deep convolutional neural netw orks." In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11534–11542.*

19. ^*Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S, Savva M, Song S, Su H, et al. "Shapenet: An inf ormation-rich 3d model repository." arXiv preprint arXiv:1512.03012. 2015.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.