2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Dynamic Threshold Spatial-Temporal Filter on FPGAs for Event-based Vision Sensors*

Ryuta Toyoda[1], Kanta Yoshioka[1,2], and Hakaru Tamukoh[1,3]

Abstract— **Event-based vision sensors are high-speed, wide dynamic range image sensors with potential applications in domains such as robotics and visual navigation. However, these sensors are sensitive to noise, particularly under low-light conditions, degrading the data quality. Therefore, developing a filter capable of detecting and removing noise from different sources with high accuracy is crucial. Moreover, removing near-edge noise in high-density areas is particularly challenging using conventional methods because of their high spatial-temporal correlation with actual events. We propose a dynamic threshold spatial-temporal filter that detects high- or low-density areas and removes noise. Detection was achieved by counting the number of events occurring within a certain period in the area surrounding each event. Applying an appropriate threshold for each density significantly enhanced noise processing accuracy, as reflected by the mean square error and peak signal-to-noise ratio metrics. Moreover, we synthesize digital circuits in a field-programmable gate array and demonstrated a notable reduction in processing time compared to that of the central processing unit-based approach, achieving up to 74-fold faster in processing speed. These findings suggest that the proposed filter can significantly enhance real-time event-based vision systems, particularly in environments with varying noise conditions.**

I. INTRODUCTION

In recent years, frame cameras have become necessary in various devices, including cell phones, robots, and autonomous cars, making them an indispensable part of our daily lives [1]. However, there are issues such as blurring when capturing moving objects due to data processing delays caused by high data volume and white-outs in bright areas due to low dynamic range [2], [3]. In this study, we focus on event-based vision sensors (EVS), which are image sensors that output only the pixels whose luminance changes [4]. An EVS has various advantages, such as a low amount of data and a wide dynamic range [5]. EVS outputs the coordinates, time, and polarity of only those pixels that experience luminance change. For example, when a camera captures a ball, the EVS output is limited to information about the moving ball, as shown in Fig. 1. Thus, in contrast to conventional frame cameras, which output all pixels, EVS reduces the data volume by outputting only those pixels whose luminance has changed.

To achieve a wide dynamic range, EVS operates according to the principle illustrated in Fig. 2. EVS has two principal units, a light-receiving unit that converts the incident light into voltage and a luminance change-detecting unit that detects the resulting luminance change. In addition, the luminance change detection unit performs a comparison between the converted and reference voltages. If the voltage difference exceeds a threshold value, which may be set as positive or negative, the event is output. By setting the threshold voltage to an optional value, it is possible to capture motion with an appropriate level of sensitivity for a given environment. In addition, as shown in Fig. 3, EVS converts incident light into voltage in a logarithmically proportional manner. This enables the detection of minor deviations in luminance at low luminance and prevents saturation of events caused by substantial differences in luminance at high luminance, thereby achieving a high dynamic range.

One application that utilizes these EVS features is human gesture recognition [6]. However, capturing images with high sensitivity, especially under low-light conditions, increases noise [7]. This causes a rise in data volume and a simultaneous decline in application performance. Therefore, it is necessary to improve the data quality by removing noise. This study proposes a noise-reduction filter that effectively removes the noise generated by various factors.

The main contributions of this study include the development of a dynamic threshold spatial-temporal filter that effectively removes noise in event-based vision systems and its successful synthesis on an FPGA for real-time processing. The remainder of the paper is organized as follows: Section II reviews related work, Section III presents the proposed method, Section IV discusses experiments, and Section V concludes the paper.

II. RELATED WORKS

A. Mode Filter

EVS outputs the information of the pixel where the event occurs in an asynchronous format, represented by the coordinates (x, y), time (t), and polarity (p) as (x, y, t, p). The polarity is set to positive (+) and negative (-) when an event occurs with a brightening and darkening luminance change in the scene, respectively. As shown in Fig. 4, the

*This research is based on results obtained from a project, JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization. This work received support from JSPS KAKENHI Grant Numbers 23H03468 and 24KJ1820, as well as from JST ALCA-Next Grant Number JPMJAN23F3.

[1] All authors are with Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan toyoda.ryuta785@mail.kyutech.jp, yoshioka.kanta986@mail.kyutech.jp, tamukoh@brain.kyutech.ac.jp

[2] Kanta Yoshioka is with Japan Society for the Promotion of Science, Kojimachi Business Center Building, 5-3-1 Kojimachi, Chiyoda-ku, Tokyo 102-0083, Japan yoshioka.kanta986@mail.kyutech.jp

[3] Hakaru Tamukoh is with Research Center for Neuromorphic AI Hardware, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan tamukoh@brain.kyutech.ac.jp

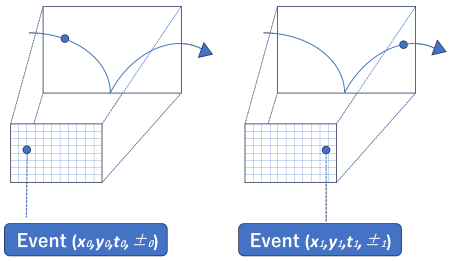

Fig. 1. EVS Output Format

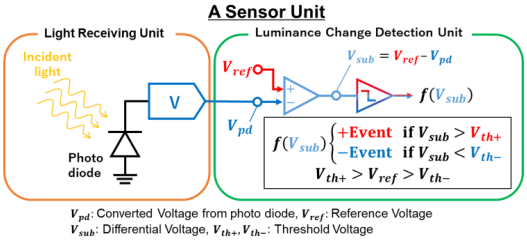

Fig. 2. How EVS processes each pixel

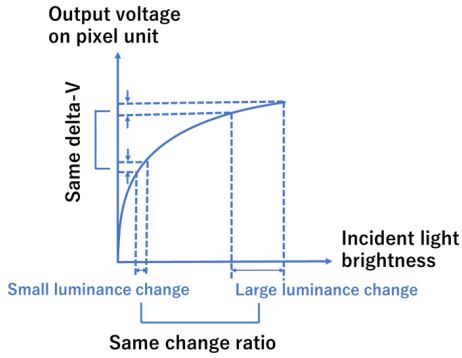

Fig. 3. Logarithmic photoelectric conversion

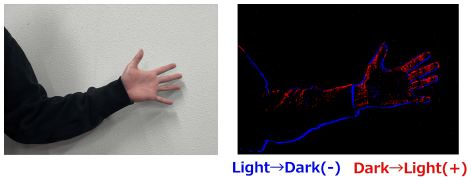

Fig. 4. Example of EVS output image

events within a certain period can be visualized by plotting the positive events in red, the negative events in blue, and no events in black at each coordinate. A mode filter that converts each pixel with the most frequent (mode) colors of the surrounding 3 × 3 kernels, as shown in Fig. 5, can be applied to obtain the mode value of the three colors in the EVS output [8]. Thus, converting the captured data into an image and applying a noise reduction filter can effectively remove noise with a low spatial correlation to events [9]. However, using a mode filter results in losing the high-speed processing ability of the EVS because it must process all the pixels, including non-occurring events.

B. Spatial-Temporal Filter

The output data of EVS is not a real image, but the data of the pixel that changes its luminance. Therefore, the conversion of EVS data into an image format requires the processing of all the pixel data, resulting in the loss of the small amount of data which is a key feature of EVS. Therefore, a spatial-temporal filter was proposed to remove noise using only the data that occurred [10], [11]. The processing algorithm is illustrated in Figs. 6 and 7. For each event, a 3D kernel centered on the event in question is created. This filter removes events with low spatial-temporal correlations as noise.

C. Types of Event Noise

When capturing images using an EVS, various factors can generate different noise types. This study focuses on background and near-edge noises that may increase the amount of data and reduce application performance [12], as explained previously. The following explanation for these noises is provided using the results captured by EVS (Fig. 8).

1) Background Noise: Background noise is a relatively low-density noise with a low spatial correlation to actual events. It is generated by the edges of moving objects or changes in the brightness of the scene. In addition, background noise can be generated by leakage current, shot noise [13], or thermal noise.

2) Near-edge Noise: Near-edge noise is a relatively high-density noise that occurs in the spatial neighborhood of an actual event. It is attributable to the output delay in the event because of edge movement. If the delays in the various pixels within the sensor are not aligned, an occurrence at the correct pixel, albeit with a delay that does not correspond to the temporal response function of the pixel, will generate noise in the edge vicinity in the opposite direction of the edge motion. The greater the velocity of motion within the scene, the more extensive the noise propagation.

D. Event Data Processing Architecture

Recently, field-programmable gate arrays (FPGAs) have been used as processing devices for output data from EVS [14], [15], to realize real-time processing through parallel processing. Furthermore, a specific circuit that directly receives and processes event data from EVS allows for efficient processing of sparse data. Additionally, platforms that directly connect EVS and FPGAs for processing are gaining popularity [16]. The performance, power, and speed of sensors and artificial intelligence are expected to be optimized to develop next-generation edge machine vision applications.

III. PROPOSED METHOD

This study proposes a dynamic threshold spatial-temporal filter that removes background and near-edge noises using

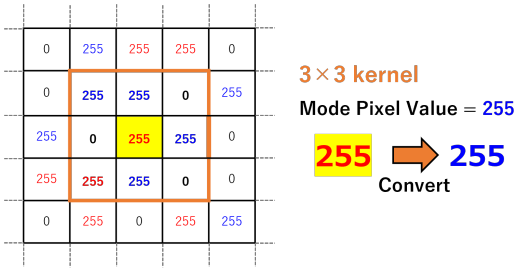

Fig. 5. Mode filter kernel

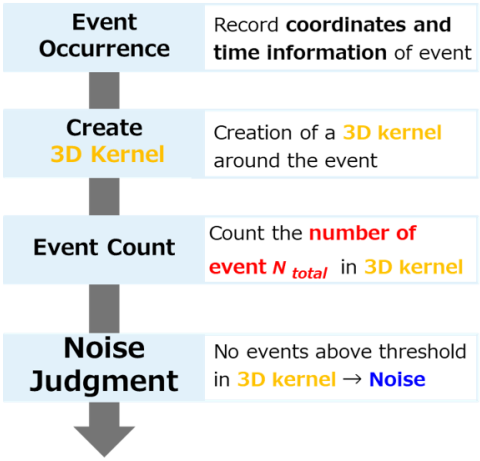

Fig. 6. Spatial-temporal filter process flowchart

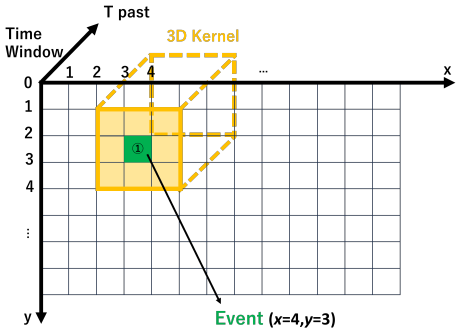

Fig. 7. Spatial-temporal filter 3D kernel

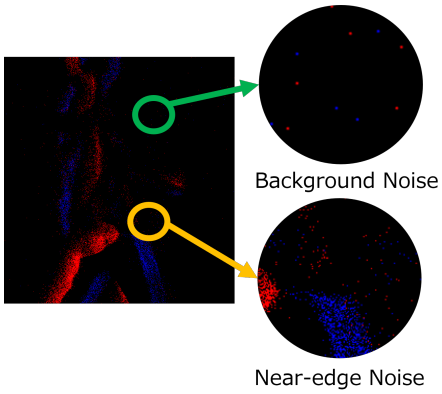

Fig. 8. Two types of noise

only event data. The objective is to perform the process efficiently and in real-time using FPGAs toward implementation in a digital circuit. In addition, the effectiveness of the noise reduction filter circuit is evaluated using a hardware simulator.

A. Dynamic Threshold Spatial-Temporal Filter Algorithm

Fig. 9 shows the algorithm of the dynamic threshold spatial-temporal filter. The algorithm is the same as that of the conventional spatial-temporal filter, from the occurrence of an event to the counting of the number of occurrences N_{total} in the 3D kernel. The filter determines whether a pixel is in a high- or low-density area based on whether N_{total} is above or below the switching threshold TH_{switch}, respectively. If the pixel is in a low-density area, the low-density threshold TH_{low} is set; if the pixel is in a high-density area, the high-density threshold TH_{high} is set. Conventional spatial-temporal filters use a fixed threshold to remove background noise in low-density areas. In contrast, the proposed method is expected to remove background and near-edge noises in the low- and high-density areas, respectively, by dynamically setting the threshold value.

B. Synthesis of Dynamic Threshold Spatial-Temporal Filter Circuit

When implementing the dynamic threshold spatial-temporal filter in FPGAs, saving a large amount of event information is necessary. In this study, we use block random access memories (BRAMs) because the available capacity of registers is limited. Without BRAMs, it would be impossible to store the data needed to operate the filter. When an event arrives, the number of events at the same coordinates N is read from one of the BRAMs, incremented, and written back to the same address. This allows for the efficient processing of event data using internal memories.

The circuit design of the dynamic threshold spatial-temporal filter using BRAMs is shown in Fig. 11. The comparator determines whether an event is in a high- or low-density area and whether it is noise, based on the total event counts N_{total} in the 3D kernel. If the incoming event is determined as noise, then the valid signal is not asserted and is not output from the filter. Using lookup tables (LUTs) and flip-flops (FFs) as memory increases the circuit size, making it challenging to implement the filter circuit. Therefore, we propose a novel circuit design with BRAMs, the memory elements in FPGAs. Furthermore, we develop an algorithm that makes it unnecessary to access more than one pixel value concurrently to use BRAMs.

In this study, we synthesize the filter on an FPGA in the environment shown in Table I. Table II lists the resource utilization of the dynamic threshold spatial-temporal filter. The proposed filter circuit is sufficiently large to be synthesized on an FPGA for edge processing (KR260), and achieving more complex processing in the future.

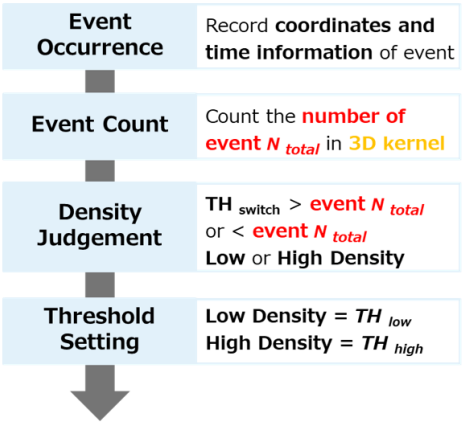

Fig. 9. Proposed dynamic threshold spatial-temporal filter process flowchart

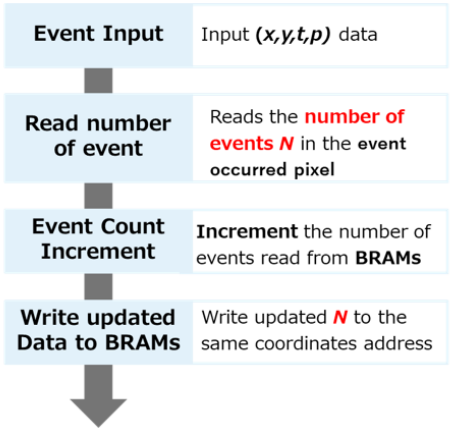

Fig. 10. Flowchart of the process using BRAMs for the proposed filter

IV. Experiments

We evaluate the developed filter circuit in terms of noise reduction accuracy and processing time. We chose the kernel size used in the experiments based on the priority of processing speed in this study. Larger kernel sizes allow the filter to consider more spatial-temporal correlations, which improves accuracy [18]. However, because using BRAMs can only read and write one pixel value per clock cycle, increasing the kernel size significantly increases the processing time. The DAVIS 240C Datasets [19], in which frame- and event-based data were captured simultaneously, was used for the experiment. We used 10 s of data captured in urban, outdoor and office environments. Ground- truth images for comparison of the removal accuracy were created using v2e software [20], which can simulate frame-based data to match the data captured by EVS, converting frame movies from the DAVIS 240C Dataset.

We evaluate the noise reduction accuracy using two most commonly used image quality metrics: mean square error (MSE) and peak signal-to-noise ratio (PSNR) [21], [22]. A lower MSE and higher PSNR indicate better image quality. We manually tuned the thresholds for noise reduction to identify the values that produced the best results. Specifically, we set the dynamic threshold for low-density areas to match the fixed threshold of conventional spatial-temporal filters for a fair comparison. The experiment was performed using the data captured by EVS for 10 s. We compared the processing time from data input to noise removal, and filtered the data output when processed using a central processing unit (CPU) and an FPGA. The CPU used was an Intel® Core™ i7-8700K CPU @ 3.70 GHz × 12, and the hardware simulation of the FPGA was conducted with a clock frequency of 100 MHz. Real-time processing is considered possible if the processing time is less than 10 s. The noise reduction results using the mode filter, conventional spatial-temporal filter, and the proposed filter are shown in Fig. 12.

TABLE I
Implementation Environment

Environment	
Design Tool	Vivado v2022.1
Design Language	Verilog
Target Board	KR260
Device	Zynq UltraScale+ (XCK26)

TABLE II
Resource Utilization of Proposed Dynamic Threshold Spatial-Temporal Filter

Dynamic Threshold Spatial-Temporal Filter		
Resource	**Usage**	**Utilization %**
LUT	364	0.31
FF	309	0.13
BRAM	43.5	30.2

Tables III and IV show the comparison results of the removal accuracy and processing time, respectively. The proposed filter with a dynamic threshold performed the best in all three environments for MSE and PSNR reduction accuracy. In terms of processing time, the conventional and proposed filters require less time than the mode filter. Furthermore, processing can be achieved in less than 10 s when using an FPGA compared to that using a CPU (significantly above 10 s), indicating that real-time processing can be achieved when implementing these filters on an FPGA. In addition, we confirm that implementing the filter on an FPGA allows for faster processing compared to a CPU.

V. Discussion

A. Noise Removal Accuracy

Compare to a mode filter that only considers spatial correlation, a filter that considers spatial-temporal correlation can remove noise more accurately because it refers to more event information. In addition, the proposed method dynamically changes the threshold value to effectively remove background and near-edge noises, achieving noise reduction with higher accuracy. Our findings show that the dynamic threshold spatial-temporal filter outperforms previous methods by adapting to event density, particularly in removing near-edge noise. However, the filter's effectiveness relies on accurate event density estimation, which may be less

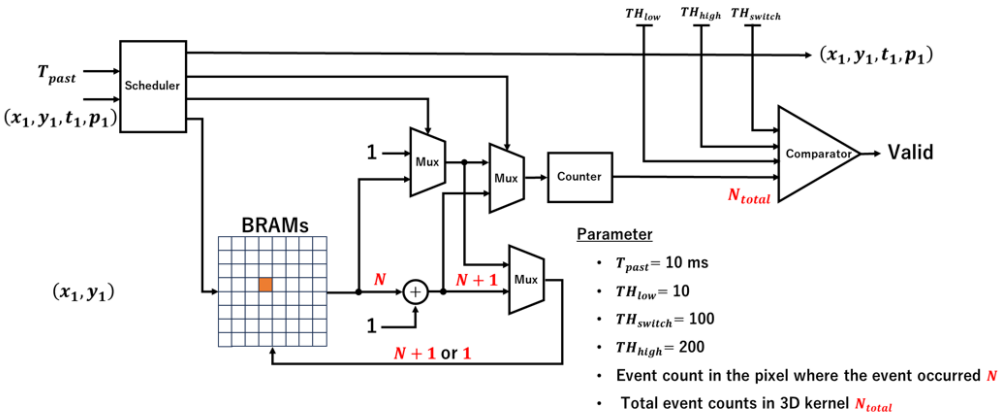

Fig. 11. Proposed circuit design of dynamic threshold spatial-temporal filter using BRAMs

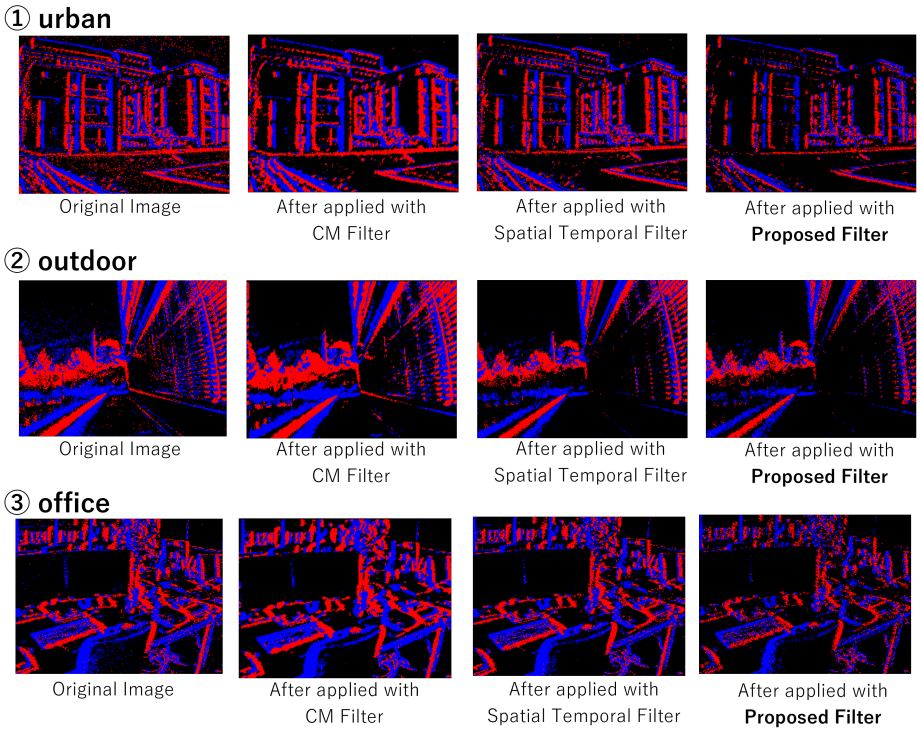

Fig. 12. Experimental results between each filter

TABLE III
NOISE REDUCTION ACCURACY EVALUATED USING MSE AND PSNR

	MSE (lower is better)			PSNR[dB] (higher is better)		
	Mode	**Conventional Spatial-Temporal**	**Proposed Spatial-Temporal**	**Mode**	**Conventional Spatial-Temporal**	**Proposed Spatial-Temporal**
urban	25.22	19.82	**11.47**	54.12	54.76	**55.80**
outdoor	27.52	24.27	**15.15**	53.55	53.55	**54.35**
office	26.93	21.98	**13.59**	53.80	54.37	**55.21**

reliable in high-noise conditions. Recently, high-efficiency memory usage approaches, such as using a single memory for multiple pixels, have been applied to high-resolution EVS [17]. However, our proposed method counts the number of events per pixel, enabling the highest accuracy.

B. Processing Time

The shorter processing time for the spatial-temporal and dynamic threshold spatial-temporal filters compared to the mode filter can be attributed to the need to process only the events that occurred. The longer CPU processing time

TABLE IV

PROCESSING TIME BETWEEN EACH FILTER USING CPU AND FPGA

	CPU[sec.]			FPGA[sec.][1]		
	Mode	**Conventional Spatial-Temporal**	**Proposed Spatial-Temporal**	**Mode**	**Conventional Spatial-Temporal**	**Proposed Spatial-Temporal**
urban	187.91 (234.51)	45.58 (56.91)	50.50 (63.05)	1.321 (1.65)	**0.801 (1.00)**	**0.801 (1.00)**
outdoor	162.02 (156.36)	69.25 (66.84)	76.81 (74.15)	1.188 (1.15)	**1.036 (1.00)**	**1.036 (1.00)**
office	151.25 (161.76)	48.92 (52.32)	58.84 (62.93)	1.202 (1.29)	**0.935 (1.00)**	**0.935 (1.00)**

[1] hardware simulation

for the dynamic threshold spatial-temporal filter compare to the spatial-temporal filter is likely due to the additional processing time required to determine the threshold. However, the FPGA implementation allows the comparator to perform processing in a single clock cycle, enabling the completion of these processes in the same amount of time. The FPGA implementation offers faster processing but may face challenges in scaling to higher resolutions, which could impact overall performance.

VI. CONCLUSION

This study presents a new approach to noise reduction in event-based vision systems and demonstrates its effectiveness over conventional methods. Conventional filters often lose the advantage of the small data volume of EVS and approach different types of noise with a single threshold, reducing the noise reduction accuracy. In contrast, the proposed filter circuit offers enhanced accuracy by considering spatial-temporal correlations using only event data. Moreover, it can capture high-quality data faster and across a wide dynamic range, which are the advantages of EVS. The findings of this study suggest that the proposed filter can significantly improve the reliability and speed of event-based vision systems, particularly in real-time applications.

In the future, we will evaluate the real-time performance of the filter circuit by implementing it in an FPGA. Additionally, we aim to develop a faster and more accurate application using the filter circuit for real-time noise reduction on EVS data. This filter circuit will be applied to both home service robots and autonomous driving systems, where real-time processing is crucial for the safe and efficient operation of these technologies [23], [24].

REFERENCES

[1] A. Elmquist and D. Negrut, "Modeling Cameras for Autonomous Vehicle and Robot Simulation: An Overview," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25547-25560, 2021.

[2] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-Based Neuromorphic Vision for Autonomous Driving: A Paradigm Shift for Bio-Inspired Visual Sensing and Perception," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 34-49, 2020.

[3] S. Nuske, J. Roberts, and G. Wyeth, "Extending the dynamic range of robotic vision," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 162-167, 2006.

[4] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-Based Vision: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154-180, 2022.

[5] Sony Semiconductor Solutions Corporation. *Event-based Vision Sensors (EVS)*. Available online: `https://www.sony-semicon.com/en/technology/industry/evs.html` (accessed on 31 August 2024).

[6] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, and D. Modha, "A Low Power, Fully Event-Based Gesture Recognition System," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] S. Ding, J. Chen, Y. Wang, Y. Kang, W. Song, J. Cheng, and Y. Cao, "E-MLB: Multilevel Benchmark for Event-Based Camera Denoising," *IEEE Transactions on Multimedia*, vol. 26, pp. 65-76, 2024.

[8] C. Jian and K. Urahama, "Window Mode Filter for Image Denoising," *TENCON 2010 - 2010 IEEE area 10 Conference*, pp. 1641-1646, 2010.

[9] R. Toyoda, S. Hori, and H. Tamukoh, "High Sensitivity Motion Capture Under Low-Light Conditions Using EVS," *The 5th International Symposium on Neuromorphic AI Hardware*, 2024.

[10] T. Delbruck, "Frame-free dynamic digital vision," *Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, vol. 1, pp. 21-26, 2008.

[11] D. Czech and G. Orchard, "Evaluating noise filtering for event-based asynchronous change detection image sensors," *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 19-24, 2016.

[12] C. Yan, X. Wang, X. Zhang, and X. Li, "Adaptive Event Address Map Denoising for Event Cameras," *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3417-3429, 2022.

[13] Y. Nozaki and T. Delbruck, "Temperature and Parasitic Photocurrent Effects in Dynamic Vision Sensors," *IEEE Transactions on Electron Devices*, vol. 64, no. 8, pp. 3239-3245, 2017.

[14] Y. Gao, B. Zhang, Y. Ding, and H. K. So, "A Composable Dynamic Sparse Dataflow Architecture for Efficient Event-based Vision Processing on FPGA," *Proceedings of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pp. 246-257, 2024.

[15] Y. Zhang, T. He, L. Peng, Y. Chang, K. Huang, and G. Chen, "An ultra-high-speed hardware accelerator for image reconstruction and stereo rectification on event-based camera," *Microelectronics Journal*, vol. 119, p. 105312, 2022.

[16] T.Kryjak, "Event-based vision on FPGAs–a survey," *arXiv preprint arXiv:2407.08356*, 2024.

[17] Q. Zhao, J. Wang, Y. Ji, J. Wu, and G. Shi, "An O(m+n)-Space Spatiotemporal Denoising Filter with Cache-Like Memories for Dynamic Vision Sensors," *arXiv preprint arXiv:2410.12423*, 2024.

[18] R. Toyoda, N, Fuengfusin, and H. Tamukoh, "Developing Spatio-Temporal-Mode Filter Circuit forEvent-based Vision Sensors ," *JSME The 8th International Conference on Advanced Mechatronics*, 2024.

[19] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. 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.

[20] Y. Hu, S.-C. Liu, and T. Delbruck, "v2e: From video frames to realistic DVS events," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1312-1321, 2021.

[21] A. Horé and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," *2010 20th International Conference on Pattern Recognition*, pp. 2366-2369, 2010.

[22] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study," *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8-18, 2019.

[23] Y. Fukuda, Y. Mii, Y. Yano, H. Iwai, S. Inoue, and H. Tamukoh, "Dense Traversability Estimation System for Extreme Environments," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1-6, 2023.

[24] N. Yamaguchi, T. Shiba, K. Isomoto, and H. Tamukoh, "A Rapidly Adjustable Object Recognition System through Language Based Prompt Engineering," in *Proceedings of International Conference on Artificial Life & Robotics (ICAROB2024)*, 2024.