IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 November 2022, accepted 6 December 2022, date of publication 9 December 2022,
date of current version 20 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3228041

RESEARCH ARTICLE

Spiking Cooperative Network Implemented on FPGA for Real-Time Event-Based Stereo System

JUNG-GYUN KIM, DONGHWAN SEO, AND BYUNG-GEUN LEE, (Member, IEEE)
Department of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Byung-Geun Lee (bglee@gist.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant through the Korean Government (MSIT) under Grant 2021R1A2C22013480, and in part by the National Research and Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grant NRF-2022M3H4A1A01009658.

ABSTRACT A hardware-efficient implementation of a spiking cooperative network (SCN) for a real-time event-based stereo correspondence system is presented. While fully utilizing the advantage of event data, the proposed SCN design significantly reduces the amount of hardware resources by utilizing distinct properties of the SCN, such as the repeatability of synaptic connections and operations, through physical constraints. A stereo system consisting of a field-programmable gate array (FPGA) and a pair of dynamic vision sensors (DVSs) is implemented to demonstrate the SCN design. Stereo livestreamed event data are generated from the DVSs, and the SCN is implemented on an FPGA chip to process the event data. The SCN system has four cores, each comprising an array of 32 Coincidence-Disparity units that calculate the 32-level disparity in a semi-parallel manner. The system performance was evaluated experimentally to estimate the depth of objects moving at different speeds. A rotating drum with a diameter of 8 cm was used in the test. The median relative error of the estimated depth at a rotation speed of 16.7 Hz ranged from 7.3% to 10.6%.

INDEX TERMS Event-based vision, stereo, FPGA, event-driven computation, cooperative algorithm, spiking neuron, real-time processing.

I. INTRODUCTION

The ability to process the stereo correspondence problem in real time, which solves the input data without storing it for later processing, is crucial in machine-vision systems such as robots and autonomous vehicles [1]. In particular, a high temporal resolution is required to respond quickly to the environment, and the power consumption must be low for portability. However, these requirements are difficult to satisfy using conventional frame-based image sensors. The temporal resolution is limited by the frame rate of the sensors, and redundant frame image data introduce the number of unnecessary computations.

To solve these problems, machine-vision systems with an event-based sensor, which is also known as a dynamic vision sensor (DVS), have been developed. This sensor encodes visual information into spike events. A high temporal resolution is achieved because the DVS generates event data asynchronously in response to pixel contrast changes. Because event data tend to be sparse and have far less redundant information than frame-based image data, the computation energy is low. Thus, stereo systems utilizing DVSs have recently gained popularity and have been reported in the literature [3], [4], [5], [6], [7], [8].

In frame-based stereo systems, most stereo correspondence algorithms find a matching point by comparing the similarity between the visual features of the left and right frame images and calculate the distance along the camera geometry through the disparity between the two matching points [2]. A similar idea was adopted for event-based stereo systems in previous studies [3], [4], [5], [6]. A cost function for spike events was calculated to find a matching point by collecting spike events within a specific time window. In [3], grayscale images were generated by collecting spike events over 20 ms. The normalized sum of absolute differences of local features was used as a cost function to calculate the disparity with the digital signal processor. In [4], grayscale images were generated by transforming events from a panoramic view of a

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Botella Juan.

rotating pair of DVSs for 3-D 360° vision. The disparity maps were computed using frame-based approaches [1], [9]. The methods reported in [5] and [6] do not generate frame images to compute the matching cost but compare each event with the collected events of another DVS pair within a specified spatiotemporal window. Event features, such as event polarity and timestamps, are used for event matching. Although these methods have solved stereo correspondence problems, they cannot fully exploit the event data, because the events must be saved in memory for a specific time.

To achieve the full benefit of event data, asynchronous event-driven computations must be performed [17]. Neuromorphic computing systems utilizing field-programmable analog array or field-programmable gate array (FPGA) devices have been successfully demonstrated the advantages of asynchronous computations [26], [27], [28]. In [7] and [8], a cooperative stereo algorithm employing a spiking-neuron model was presented. The cooperative stereo algorithm is a bioinspired global matching method that finds matching points using a predefined neural network [10]. The network calculates a matching belief corresponding to the disparity search range through the interaction between neurons, without repeatedly comparing the features of each pixel. Therefore, if the network operates asynchronously, each event can immediately affect the disparity map. As spiking neurons perform event-driven synaptic integration asynchronously, event data can be fully exploited by applying a spiking-neuron model to network neurons. Additionally, the spiking nature of the neuron model allows the network to maintain consistency with the DVS and allows end-to-end spike-driven computing.

However, the realization of a cooperative network system requires considerable hardware resources, because the number of neurons used in the network increases in proportion to the range of disparity to be detected, as well as the input size. For example, three DYNAPs [12] and one FPGA were used for a network with a 16 × 16 input and 16 disparity levels in [11], and six SpiNN-5 boards, which consisted of 48 SpiNNakers [13] and three FPGAs, were used to implement a network with a 106 × 106 input and 32 disparity levels in [8].

Because all neurons in the cooperative network have the same synaptic connections regardless of their positions, the same synaptic integration process is performed for each input. Finally, it is possible to implement the network simply by configuring a single event-based-processing element that performs the integration process in the system. By applying this idea, we attempted to reduce the consumption of hardware resources while implementing the event-based cooperative network architecture.

In this paper, an FPGA-based spiking cooperative network (SCN) system was presented for real-time event-based stereovision. Section II presents the background of the cooperative network. Section III presents the proposed idea for network design. The system architecture and measurement results are presented in Sections IV and V, respectively. Finally, Section VI concludes the paper.

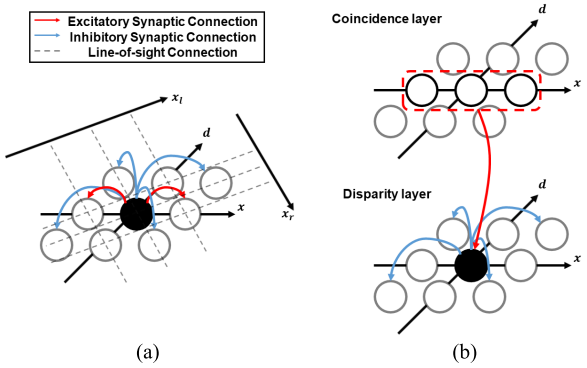


FIGURE 1. Visualization of the cooperative network: (a) original and (b) two-layer implementation.

II. BACKGROUND

A. COOPERATIVE STEREO ALGORITHM

The cooperative stereo algorithm is a global stereo correspondence approach proposed by Marr and Poggio [10] that is inspired by the biological retina network system. Each node or neuron constituting the network represents a stereo-matching belief at coordinates (x, y, d) in the disparity space. The neurons are connected systematically according to the following two physical constraints:

(1) Uniqueness constraint—Each object has a unique physical location.
(2) Continuity constraint—The disparity of an object varies smoothly.

Owing to the uniqueness constraint, each neuron has an inhibitory interaction with other neurons along the same line of sight for preventing multiple matches to the element. In addition, owing to the continuity constraint, each neuron has excitatory connections to nearby neurons with the same disparity representation. The network can be expressed iteratively using the following equation:

$$S^{n+1}_{x,y,d} = f(w_E \sum_{x',y',d' \in E} PS^{n}_{x',y',d'} - w_I \sum_{x',y'd' \in I} S^{n}_{x',y',d'} + S^{0}_{x,y,d}), \tag{1}$$

where $S^{n}_{x,y,d}$ represents the outputs of the neuron at node (x, y, d) for iteration n; E and I represent the excitatory and inhibitory connection regions, respectively; w_E and w_I represent the excitatory and inhibitory synaptic weights, respectively; and f is the sigmoid function. The initial value of neuron $S^{0}_{x,y,d}$ is defined by the input from the stereo image. Through iterations, the network converges to a stable fixed point.

Fig. 1(a) shows the structure of (1). For simplicity, the figure only shows the network on the horizontal cyclopean plane. Each pixel of the left and right cameras has its own line of sight (gray dotted line in the figure) and each neuron in the network is located at the intersection of the lines. The initial state of the neuron is determined by the line-of-sight inputs. The neuron continues exciting and inhibiting nearby

neurons according to the constraints (red and blue lines) until it converges to the local point.

B. IMPLEMENTATION OF SCN FOR EVENT-BASED SENSORS

Because (1) was designed according to frame-based images, it cannot be directly applied to DVSs. In [14], a time-correlated kernel was used for event-driven operations, but this method requires recalculating the synaptic weights for each event input, which leads to excessive computation. Instead, if each neuron performs leaky integration, spike-driven computations can be performed without using time-variant synaptic weights. Therefore, an SCN based on a leaky (LIF) neuron model was proposed:

$$\begin{aligned} S_{x,y,d}(t) &= T(u) \\ \frac{d}{dt}u(t) &= -u + w_E \sum_{x',y',d' \in E} S_{x',y',d'}(t) \\ &\quad - w_I \sum_{x',y',d' \in I} S_{x',y',d'}(t) + w_C \sum_i \delta(t - t_i), \end{aligned} \tag{2}$$

where $S_{x,y,d}(t)$ represents the output of the neuron at node (x, y, d) at time t, u is a state variable or the membrane potential of the neuron, w_C represents the synaptic weight of the input spikes, $\delta(t)$ is the Dirac delta function, index i indicates the event time of the line-of-sight input, and T is a binary threshold function. $T(x) = 1$ if x is above the threshold; otherwise, $T(x) = 0$. After a binary spike is produced, the state variable is reset to 0.

One problem with (2) is that the original network design was not aimed at handling event data. Because the DVS detects the temporal contrast changes of light, it does not output multiple spikes for a single event. Therefore, to generate a disparity spike based on (2), the SCN system must be sensitive to the event input timing of the sensor. However, the detection of temporal contrast changes is vulnerable to background noise [15]. In addition, even if no contrast change is detected, each pixel of the sensor outputs a periodic-on event because of the leakage current of the circuit [16]. Furthermore, because the sensor operates asynchronously, a timing error exists for every event between the left and right sensors in the stereovision system. Equation (2) uses these raw data directly for cooperative operations, which makes it difficult to obtain accurate results. In [7], another method was proposed for constructing the network, which separates the SCN into two layers as follows:

$$\begin{aligned} C_{x,y,d}(t) &= T(u_c) \\ \frac{d}{dt}u_c(t) &= -u_c + w_c \sum_{t_i} \delta(t - t_i) \\ D_{x,y,d}(t) &= T(u_d) \\ \frac{d}{dt}u_d(t) &= -u_d + w_E \sum_{x',y',d' \in E} C_{x',y'd'}(t) \\ &\quad - w_I \sum_{x',y',d' \in I} D_{x',y',d'}(t), \end{aligned} \tag{3}$$

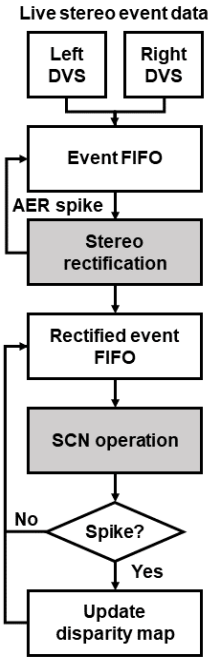


FIGURE 2. Flow chart of the end-to-end event-driven stereo correspondence system.

where $C_{x,y,d}(t)$ and $D_{x,y,d}(t)$ represent the outputs of the first layer, coincidence-layer neurons and the second layer, or disparity-layer neurons at time t. u_c and u_d represent the membrane potentials of coincidence- and disparity-layer neurons, respectively. Fig. 1(b) shows the two-layer SCN based on (3). The coincidence layer has a normal feedforward network structure. Each input spike excites coincidence-layer neurons along the line-of-sight to determine all possible matching points of the event. When a coincidence neuron is activated, it excites the disparity-layer neurons according to the continuity constraint. Finally, the disparity spikes inhibit nearby line-of-sight neurons through a uniqueness constraint. This disparity-layer inhibition process employs the Winner-Take-All mechanism, which is frequently used in stereo correspondence problems [17], [18]. By inhibiting all other disparity neurons in the line of sight, false stereo matches can be prevented. An important difference between (2) and (3) is that the raw data are not used directly for stereo matching in (3). Throughout the coincidence layer, the input events are transformed into possible matching events, which are used for cooperative operation in the disparity layer. Consequently, the effect of noise is reduced, and more accurate results are obtained.

III. PROPOSED HARDWARE IMPLEMENTATION FOR SCN

An end-to-end event-driven stereo correspondence system, which employs the SCN algorithm, has been designed. A flow chart for explaining the overall system architecture and data flow is presented in Fig. 2. Whenever a spike event is generated from the DVSs, it is asynchronously assigned to an event first-in-first-out (FIFO) buffer in

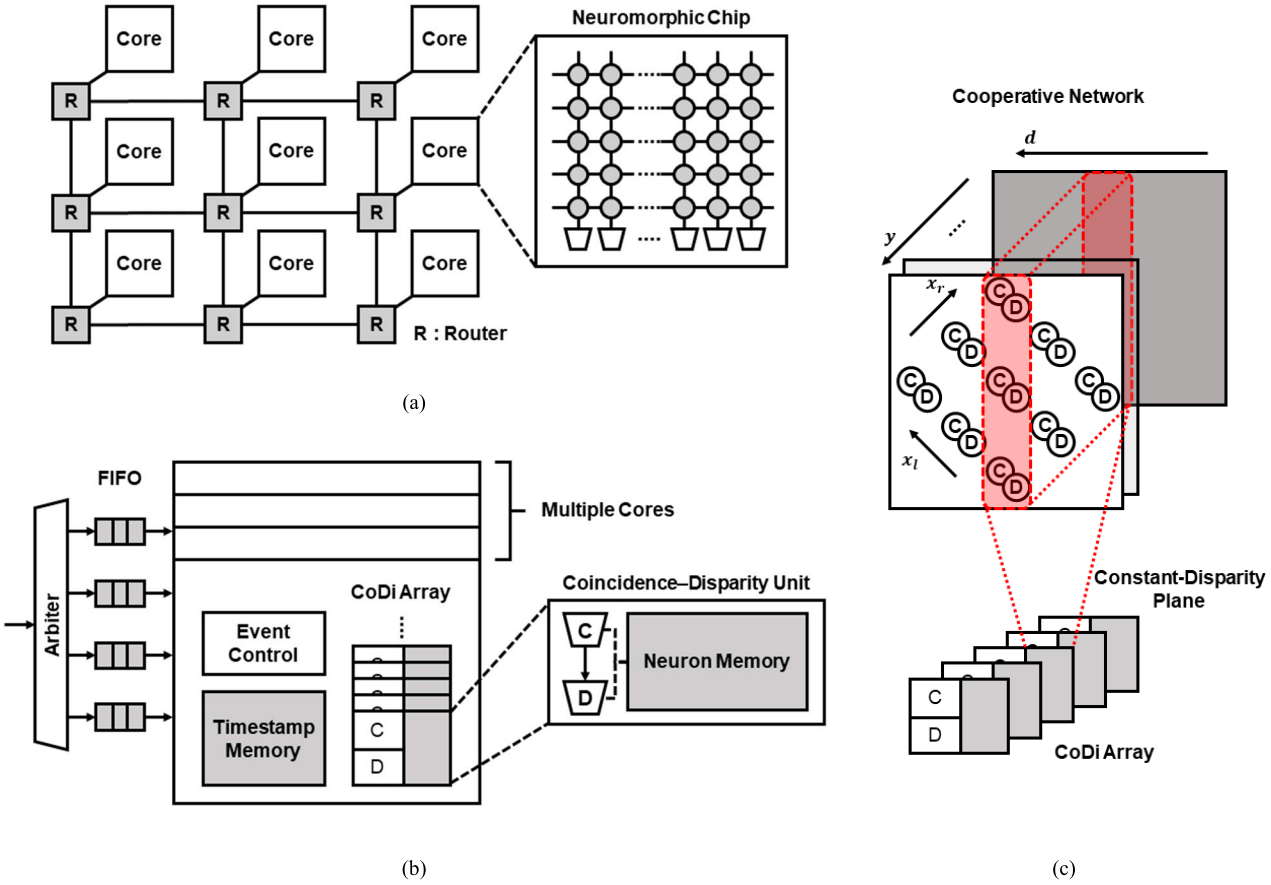


FIGURE 3. (a) SCN design with general-purpose neuromorphic chips, (b) proposed hardware design, and (c) example of SCN realization with the proposed hardware design.

address-event-representation (AER) format. The stored spike event is sent to the next FIFO buffer for SCN operation after rectification process. The SCN process is also performed asynchronously and generates the output spike, which is used to update the disparity map. As mentioned in Introduction, the SCN occupies the largest portion of hardware resources for system implementation. This section details the proposed SCN design with reduced hardware complexity.

A. LAYER IMPLEMENTATION

In general, it is reasonable to design an SNN using a neuromorphic processor to obtain a network with multiple synapses and LIF neurons. In most previous studies, a processor was used for SCN design [7], [8], [11], [17]. Fig. 3(a) shows this implementation based on neuromorphic cores. Multiple synaptic connections, which are represented as crossbar nodes in the figure, are allocated to each neuron. However, because all neurons in the SCN have fixed synaptic weights owing to physical constraints, most of the synaptic nodes in the processors are not activated. For instance, each neuron in the coincidence layer receives only inputs from the left and right pixels located along the line of sight. Therefore, it uses only two synaptic nodes, and the rest are idle. Moreover, a network-on-chip architecture is needed to transmit spike events from each core to the correct location [19], [20]. Because neurons excite and inhibit other neurons, multiple event routers with a specific addressing algorithm must be implemented.

However, the implementation can be simplified by considering the structure of the SCN. Because the network is designed to solve the stereo correspondence problem, all neurons in the same layer perform the same operations in each local region. Therefore, the entire operation can be performed using a single processing element, which is called the Coincidence–Disparity (CoDi) unit in the proposed SCN design shown in Fig. 3(b). Each CoDi unit has a pair of coincidence neurons and disparity neurons with an internal state neuron memory. This configuration is possible because the coincidence and disparity layers represent the same stereo-map coordinates. By integrating two layers into one unit, the network can be controlled using a single-event controller inside the core. However, the use of a single element introduces a significant time delay. Therefore, to achieve a trade-off between the hardware area and the computation speed, a semi-parallel configuration in which the CoDi units are

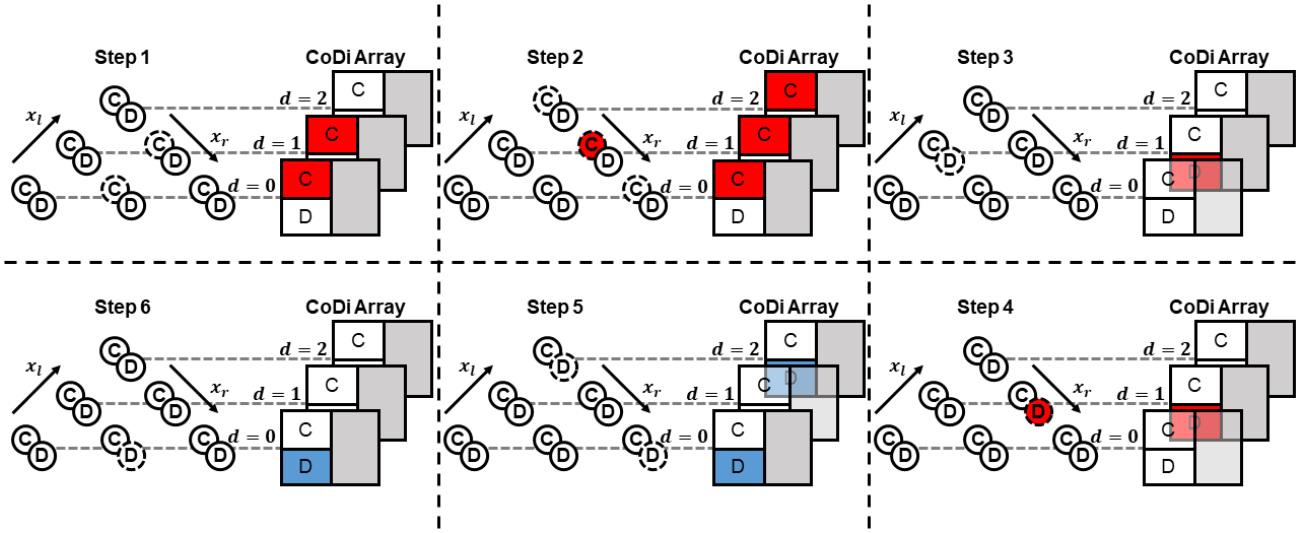


FIGURE 4. Simplified description of the CoDi array operation in six steps.

arranged in arrays is proposed. The number of CoDi units in each array determines the maximum detectable disparity range, as described in the following paragraph.

Fig. 3(c) shows how the CoDi array implements the network. Each neuron memory in the CoDi unit stores the internal state variables of the coincidence and disparity neurons in the constant-disparity plane. Then, one CoDi unit can process the operation of the neurons on the vertical plane in the figure. This memory allocation allows the CoDi unit to have one memory access for each input event in the coincidence layer, because the sensor input is connected along the line of sight. Therefore, by arranging CoDi units in accordance with the disparity range to be measured, the local operation for each input event on the coincidence layer can be processed in a single integration cycle. For disparity-layer operation, as the inhibition step also proceeds along the line of sight, efficient computation can be achieved using the proposed structure similar to the coincidence layer.

B. COOPERATIVE OPERATION OF CODI ARRAY

Fig. 4 presents the operation of the CoDi array in six steps. The description is simplified by reducing the disparity range of the network from two to zero. The horizontal gray dotted lines are the constant-disparity lines. All operations in the network can be expressed in three CoDi units by accessing the corresponding memory for each event. All synaptic weights of the same type are set to the same value so that all possible matches have the same effect on the network. In Step 1, the event generated by the second pixel of the right sensor excites the coincidence neurons along the line of sight, and the same operation is performed in Step 2 by the event from the third pixel of the left sensor. The dotted circle indicates the excited neuron at the node. The coincidence neuron at the intersection of the two lines of sight can be activated if the two events coincide. The red circle in Step 2 indicates that the neuron at the node is activated. If no spike is generated, the previous steps are repeated for the next input. From Step 3 to Step 4, nearby disparity neurons on the same disparity line as the activated coincidence neuron are excited. As the neurons on the constant-disparity line are integrated into a single CoDi unit, the excitation process must proceed step-by-step. Similar to Step 2, the dotted circle indicates the excited disparity neuron and the red circle in Step 4 indicates the activated disparity neuron. After the disparity neuron is activated, other neurons along the same line of sight are inhibited, as shown in Steps 5 and 6. Dotted circles indicate the positions of inhibited neurons. When the inhibition process ends, the operation restarts, and the steps are repeated.

TABLE 1. Hardware resource comparison.

	Neuromorphic Processor	Proposed Design	Single Neuron
Design Approach	Fully parallel	Semi parallel	Serial
Synaptic Operation Time per Event	N	N	N
Coincidence Neuron	$K^2 \times d$	d	1
Disparity Neuron	$K^2 \times d$	d	1
Coincidence-layer Excitation	N	N	$d \times N$
Disparity-layer Excitation	N	$W \times N$	$W \times N$
Disparity-layer Inhibition	N	2N	$d \times 2N$

C. COMPARISON OF IMPLEMENTATIONS

To determine the efficiency of the proposed design, a resource comparison among SCN implementations with a neuromorphic processor, the proposed design, and a single neuron

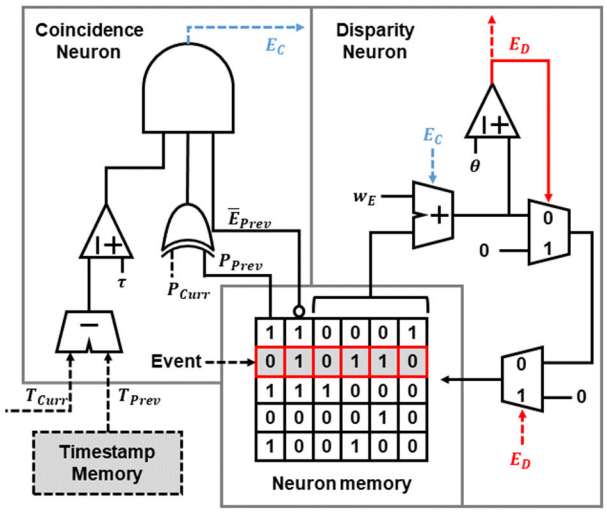


FIGURE 5. Block diagram of the CoDi unit.

was performed, and the results are presented in Table 1. For simplicity, the synaptic operation time per event was assumed to be equal to N for all the implementations. In the table, K^2 and d represent the size of the input and the detectable disparity range, respectively. W represents the excitation range of the coincidence-neuron spike. The value was set as 7 in the experiment.

As the neuromorphic processor is designed to be fully parallel, it performs all synaptic operations of neurons within the same layer for each event in a single operation cycle. However, as the size of the SCN increases by K^2 for each disparity level, a large amount of hardware is required for high-resolution systems. The proposed design balances the hardware area and computation time with a semi-parallel architecture. While the required number of neuron units is equal to d, the computation time is not dependent on d. Therefore, the proposed method is more efficient in a high-resolution system than fully parallel or serial designs. For a quantitative comparison, we set K as 128 and d as 32 (identical to the values of our experimental environment). The proposed design with a single core required 16384 times fewer neurons than the fully parallel design, and the computation time was only increased by a factor of 3.3.

D. NEURON IMPLEMENTATION

While (3) is designed according to the LIF neuron model for generalization, the proposed neuron implementation is simplified considering the SCN structure to further reduce the amount of hardware resources. Fig. 5 shows a block diagram of the CoDi unit. The signals and memory indicated by dotted lines in the figure are controlled by an event controller external to the CoDi unit.

As no lateral interaction exists in the coincidence layer, the output of each coincidence neuron is determined only by the time correlation of the left and right events. Therefore, instead of leaky integration, the coincidence neuron simply compares the timestamps of the current input event T_{Curr} and the previous input event T_{Prev}. The current timestamp is saved in the timestamp memory for the next operation. While the operation is simple, redundant matching may occur in a single event if events rapidly come alternately from the left and right. Thus, if the time difference is within a specified time τ, an additional event blocking process is conducted. Instead of membrane potential integration, the internal state memory of the coincidence neuron is used to prevent multiple matchings to the event. A 1-bit state memory E_{Prev} is used to determine whether the neuron was activated in the previous step (1 if activated; 0 otherwise). If so, the neuron activation is blocked, and only the state variable is updated. Another 1-bit state memory P_{Prev} is used to prevent homolateral excitation, which occurs when a series of spike events are generated by a single pixel. This causes multiple false matches in the coincidence layer along the line of sight of the pixel if not prevented. P_{Prev} stores the camera position information for the previous event (0 for the left camera and 1 for the right camera). and blocks the activation if the camera position of the current event P_{Curr} is the same as that of the previous one. After the coincidence spike E_C is generated, the 4-bit membrane potential of the disparity neuron stored in the neuron memory is accumulated by fixed weight w_E. Whenever the potential exceeds the threshold θ, the disparity spike E_D is activated, and the potential is reset to 0. The spike then inhibits the other disparity neurons along the line of sight. Lateral inhibition is designed to reset the potential to 0. As it eliminates the remainder of the false matches, changing the bias for the post-events, the leakage operation is unnecessary.

IV. EVENT-BASED SCN SYSTEM

Fig. 6 shows the hardware architecture of the SCN system. Intel DE2i-150 evaluation board was used for the system implementation. It contains a Cyclone IV GX FPGA device with 149,760 logic elements and 810 kB of block RAMs. The main clock frequency of the system is 50 MHz. All the memories in the system were embedded in the FPGA chip. Stereo event data were transmitted in real time from a custom-built, non-commercial DVS with a resolution of 128 × 128. For every event generation, the DVS sent a 14-bit pixel address to the FPGA. Event polarity was ignored because it depends on the background [8]. The FPGA stored the event information using the FIFO with a 1-bit sensor position (0 for the left and 1 for the right) and 16-bit event timestamps from the internal counter. The timestamp was incremented by one every 1 ms.

In a stereo rectifier, rectified address information for all the pixels in the stereo sensors is stored as a lookup table (LUT). The Caltech Camera Calibration Toolbox in MATLAB was used for stereo rectification. Traditionally, stereo pairs of checkerboard images have been used for calibration. Because the DVS is unable to generate a frame image for the toolbox, we displayed a blinking checkerboard pattern on a PC monitor and converted the collected event data into the image [21], [22]. Figs. 7(a) and (b) show the monitor pattern and the

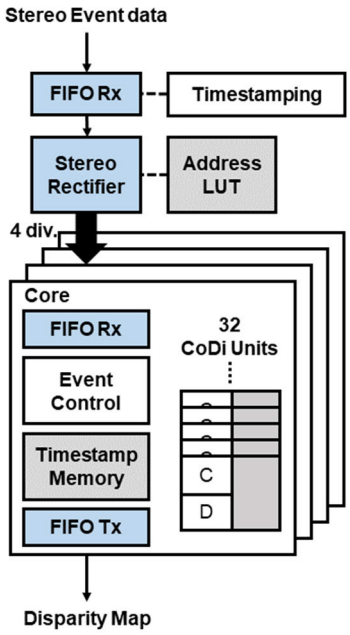


FIGURE 6. Block diagram of the FPGA-based cooperative network system.

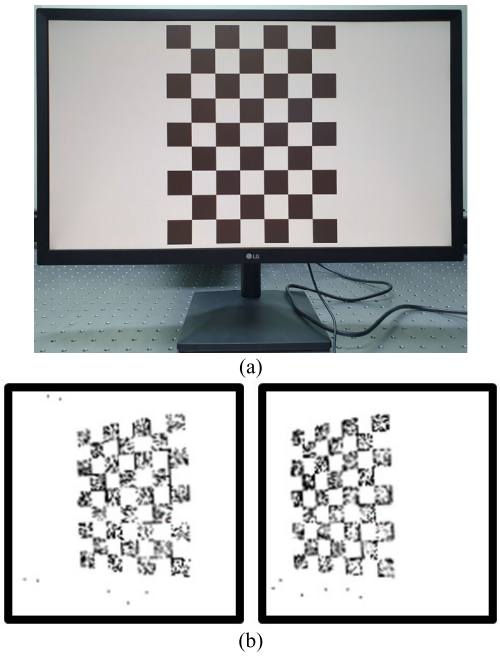


FIGURE 7. (a) Blinking checkerboard pattern on the PC monitor and (b) stereo image pair of the pattern reconstructed from DVS spike sequences.

converted images, respectively. The rectified addresses in the LUT were rounded for one-to-one event mapping.

After rectification, events were sent to the cooperative network block. Four processing cores were used in parallel. While the entire network can be handled by a single core, multiple cores were implemented accounting for the trade-off between the hardware area and the computation speed. A quarter of the network along the y-axis was allocated

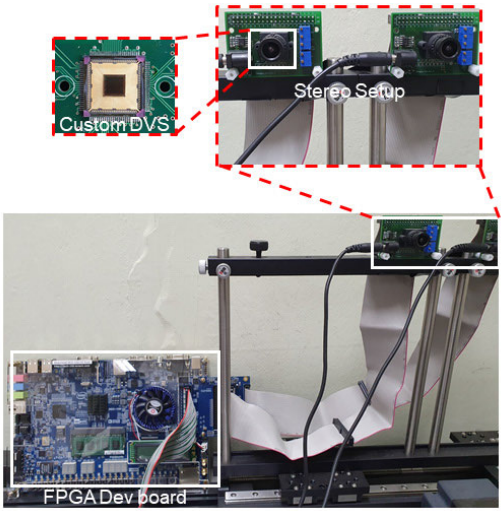


FIGURE 8. Experimental setup for the event-based stereo system.

to each core. In this implementation, owing to the limited size of the FPGA embedded memory, 32 CoDi units were arranged for each core to detect the disparity in the range of 0 to 31. Each core stored the timestamp of the most recent event for each pixel of the sensors in the timestamp memory. Since the network is divided into 4, the memory of each core stores a quarter of the total pixels. As explained in the previous section, the timestamp memory is used for synaptic operations with state variables stored in the neuron memory of each CoDi unit. For each spike input, the core loads the timestamp information of the previous event from the memory and updates the network state using the CoDi array. Whenever the core outputs a disparity spike, the address and disparity value of the spike are sent to PC for disparity map generation.

V. EXPERIMENTS

A. EXPERIMENTAL SETUP

Fig. 8 shows the setup for the stereo system experiment. The custom DVS pair was directly connected to the FPGA board through IDC cables. The FPGA received live DVS data to process the SCN in real time. A system evaluation was performed by collecting the disparity spikes of the moving object and calculating the precision. The rotating drum with a diameter of 8 cm shown in Fig. 9(a) was used to compare the performance of the system at different speeds. The accuracy of the system was expressed as the median relative error, i.e., the difference between the ground truth and the estimated depth divided by the ground truth. Ground-truth data were obtained by projecting the measured distance of the object fixed on the optical table.

Because the output of the cooperative network is the stereo-matching belief of the entire disparity range for each node of the network, the disparity value cannot be determined by a single disparity spike. To determine the disparity at a specific moment, the disparity neuron with the highest belief should

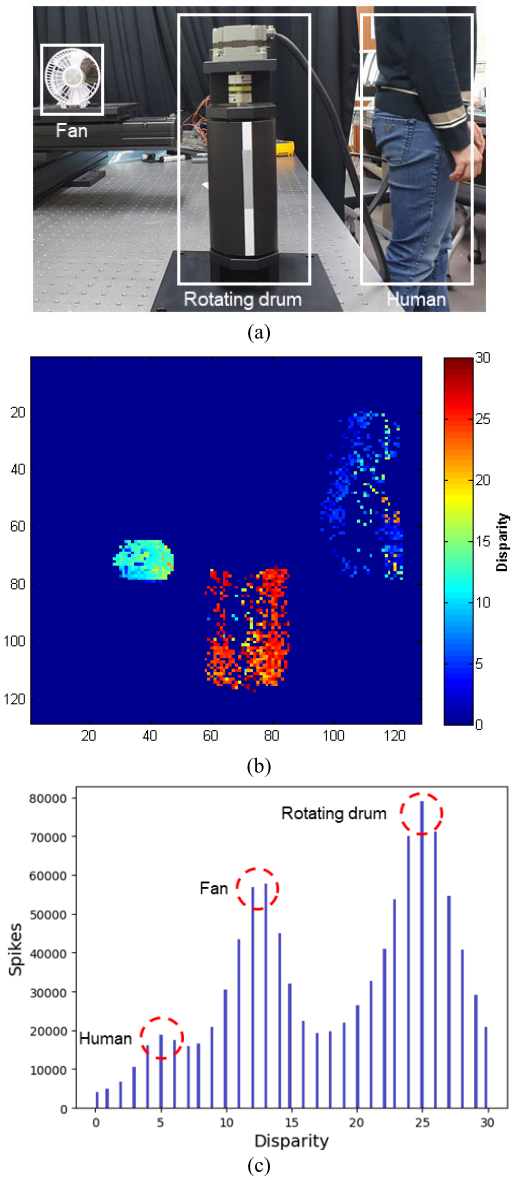


FIGURE 9. (a) Multi-object test scene, (b) disparity map of the scene and (c) histogram of spike activity from SCN.

be found through the disparity spike sequence accumulated up to that moment. We set an event time-surface map [23] for each disparity level and considered the surface with the maximum point as the disparity value at each address. If the maximum surface did not exceed the specified threshold, the disparity of that address was ignored. In this manner, a disparity image frame was obtained at a certain moment. Fig. 9(b) shows the disparity map of the multiple-object scene in Fig. 9(a). A rotating drum, small fan, and human were located in the scene. The rotating drum was closest to the stereo setup, followed by the fan and then the human. As shown in Fig. 9(b), the disparity level decreases as the distance increases. Fig. 9(c) shows a histogram of the SCN spike activity of the scene. As expected, spikes are highly activated

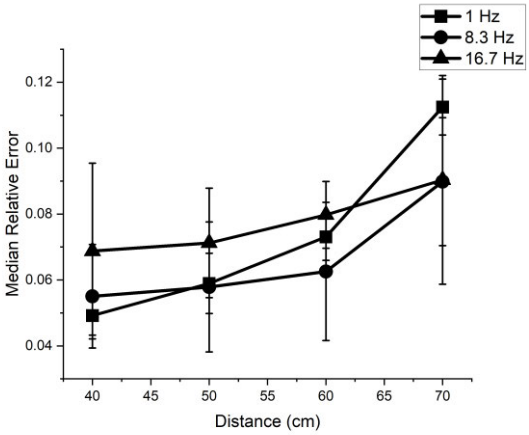


FIGURE 10. Performance of the rotating drum with three different rotation speeds.

at each disparity level of the objects. Three different peaks indicate that the system can solve the stereo correspondence problem.

B. EVALUATION

Power and precision measurements were performed for the system evaluation. The power performance of the FPGA board was estimated using Powerplay power analyzer in the Intel Quartus Prime Lite Edition. The power estimation was done in three steps: First, stereo events are randomly generated at the maximum bandwidth of the system for every pixel. This allows to estimate the worst-case or maximum power dissipated by the board. Second, the signal activities are stored in a VCD file including the post place and route simulation data. Finally, the power analyzer estimates the power consumption of the system using the VCD file.

Precision measurements were performed with a rotating drum. Most previous studies demonstrated the precision error of a moving object. However, as mentioned in Section I, the DVS can achieve a high temporal resolution. Consequently, the DVS is excellent for detecting high-speed objects. Therefore, we used an object whose rotational speed can be precisely controlled by a motor, to prove that the system can still achieve reasonable accuracy in the detection of high-speed objects. The measurements were performed at rotational speeds of 1, 8.3, and 16.7 Hz at multiple distances, as shown in Fig. 10.

C. RESULTS

The results obtained are summarized in Table 2 along with other event-based stereo vision implementations. The estimated power was 188 mW and system latency was 2ms. Since the disparity data generated from SCN is frameless, the latency was estimated using the amount of SCN spikes and rate of change of edges as proposed in [7]. The median relative error ranges were 6.9 to 13.1%, 5.6 to 10.7%, and 7.3 to 10.6% at 1, 8.3, and 16.7 Hz, respectively. Compared with other ASIC approaches [3], [17], our FPGA-based

TABLE 2. Comparison with prior works.

	This Work	[17]	[3]	[6]	[25]	[5]	[3]
Algorithm	Cooperative network	Spatiotemporal similarity matching	Cooperative network	Event-based segmentation	Gabor-filtered perspective matching	Perspective matching	Normalized sum of absolute differences
Implementation Hardware	DE2i-150 FPGA board	9 TrueNorth NS1e boards (ASIC)	6 SpiNN-5 boards (ASIC)	FPGA	Spartan6 FPGA board	Pentium 4 CPU Laptop	Blackfin BF537 DSP
Asynchronous Computation	Yes	Yes	Yes	No	Partially	No	No
Sensor Size in Real-Time	16,384	10,800	11,236	16,384	16,384	16,384	16,384
Disparity Levels in Real-Time	32	21	32	36	-	128	-
Power Consumption	188 mW	626 mW	180 W	-	-	-	5 W
System Latency	2 ms	9 ms	2 ms	0.87 ms	50 ms	0.3 ms	5 ms
Median Relative Error	6.9 to 13.1% (1 Hz) 5.6 to 10.7% (8.3 Hz) 7.3 to 10.6% (16.7 Hz)	5 to 11.6% (Fan) 7.3 to 8% (Butterfly)	-	6 to 16%	6% (Pen) 2.7% (Ring) 9.9% (Cube)	-	6 to 10%

implementation achieved considerable accuracy and power performance with lesser hardware. Estimated power of our system is about 3.3 times less compared to [17]. The average median errors of the depth estimates at 1, 8.3, and 16.7 Hz were 7.3%, 6.6%, and 7.8%, respectively. Even at a speed of 16.7 Hz, which is approximately the rotational speed of the fan, high accuracy was achieved.

VI. CONCLUSION

A stereo system employing a hardware-efficient SCN design for real-time processing of event-based data was developed and experimentally evaluated. For increasing the hardware efficiency, the proposed design has a semi-parallel structure with an array of CoDi units, which utilizes the unique properties of the SCN. The system implemented on an FPGA targeting a 128×128 resolution of the DVS and 32-level disparity only consists of 128 CoDi units (4096 times less than the fully parallel design). Table 2 presents the experimental results in comparison with other real-time event-based stereo systems. The median relative error range achieved at 16.7 Hz was 7.3% to 10.6%, comparable to the other systems with less hardware resources.

Although SCN has been successfully validated, it has limitations in the detection of motion-in-depth stimuli, since the algorithm assumes a surface as fronto-parallel [24]. Implementing the existing analog-valued solutions, which can solve the stereo correspondence with slanted surfaces, in a spiking manner is required to further improve the system.

REFERENCES

[1] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze, "A fast stereo matching algorithm suitable for embedded real-time systems," *Comput. Vis. Image Understand.*, vol. 114, no. 11, pp. 1180–1202, Nov. 2010, doi: 10.1016/J.CVIU.2010.03.012.

[2] B. Tippetts, D. J. Lee, K. Lillywhite, and J. Archibald, "Review of stereo vision algorithms and their suitability for resource-limited systems," *J. Real-Time Image Process.*, vol. 11, no. 1, pp. 5–25, 2016, doi: 10.1007/S11554-012-0313-2.

[3] S. Schraml, P. Schön, and N. Milosevic, "Smartcam for real-time stereo vision—Address-event based embedded system," in *Proc. 2nd Int. Conf. Comput. Vis. Theory Appl.*, vol. 2, Mar. 2007, pp. 466–471.

[4] S. Schraml, A. N. Belbachir, and H. Bischof, "An event-driven stereo system for real-time 3-D 360° panoramic vision," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 418–428, Jan. 2016, doi: 10.1109/TIE.2015.2477265.

[5] P. Rogister, R. Benosman, S.-H. Ieng, P. Lichtsteiner, and T. Delbrück, "Asynchronous event-based binocular stereo matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 347–353, Feb. 2012, doi: 10.1109/TNNLS.2011.2180025.

[6] F. Eibensteiner, J. Kogler, and J. Scharinger, "A high-performance hardware architecture for a frameless stereo vision algorithm implemented on a FPGA platform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 637–644, doi: 10.1109/CVPRW.2014.97.

[7] M. Osswald, S.-H. Ieng, R. Benosman, and G. Indiveri, "A spiking neural network model of 3D perception for event-based neuromorphic stereo vision systems," *Sci. Rep.*, vol. 7, no. 1, pp. 1–12, Feb. 2017, doi: 10.1038/SREP40703.

[8] G. Dikov, M. Firouzi, F. Röhrbein, J. Conradt, and C. Richter, "Spiking cooperative stereo-matching at 2 ms latency with neuromorphic hardware," in *Proc. Conf. Biomimetic Biohybrid Syst.*, vol. 10384, Jul. 2017, pp. 119–137, doi: 10.1007/978-3-319-63537-8_11.

[9] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost volume filtering for visual correspondence and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 503–511, Feb. 2013, doi: 10.1109/CVPR.2011.5995372.

[10] D. Marr and T. Poggio, "Cooperative computation of stereo disparity: A cooperative algorithm is derived for extracting disparity information from stereo image pairs," *Science*, vol. 194, no. 4262, pp. 283–287, Oct. 1976, doi: 10.1126/science.968482.

[11] N. Risi, A. Aimar, E. Donati, S. Solinas, and G. Indiveri, "A spike-based neuromorphic architecture of stereo vision," *Frontiers Neurorobot.*, vol. 14, Nov. 2020, Art. no. 568283, doi: 10.3389/FNBOT.2020.568283.

[12] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, Feb. 2018, doi: 10.1109/TBCAS.2017.2759700.

[13] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber, "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, Aug. 2013, pp. 1943–1953, doi: 10.1109/JSSC.2013.2259038.

[14] M. Firouzi and J. Conradt, "Asynchronous event-based cooperative stereo matching using neuromorphic silicon retinas," *Neural Process. Lett.*, vol. 43, no. 2, pp. 311–326, Apr. 2016, doi: 10.1007/S11063-015-9434-5.

[15] H. K. Nishiharar, "PRISM: Practical real-time imaging stereo matcher," *Proc. SPIE*, vol. 449, pp. 134–142, Feb. 1984, doi: 10.1117/12.939233.

[16] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008, doi: 10.1109/JSSC.2007.914337.

[17] A. Andreopoulos, H. J. Kashyap, T. K. Nayak, A. Amir, and M. D. Flickner, "A low power, high throughput, fully event-based stereo system," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7532–7542, doi: 10.1109/CVPR.2018.00786.

[18] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, Apr. 2002.

[19] C. Zamarreno-Ramos, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco, "Multicasting mesh AER: A scalable assembly approach for reconfigurable neuromorphic structured AER systems. Application to ConvNets," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 1, pp. 82–102, Feb. 2013, doi: 10.1109/TBCAS.2012.2195725.

[20] J. Park, T. Yu, S. Joshi, C. Maier, and G. Cauwenberghs, "Hierarchical address event routing for reconfigurable large-scale neuromorphic systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2408–2422, Oct. 2017, doi: 10.1109/TNNLS.2016.2572164.

[21] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers Neurosci.*, vol. 9, p. 437, Nov. 2015, doi: 10.3389/FNINS.2015.00437.

[22] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-DOF pose tracking for high-speed maneuvers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 2761–2768, doi: 10.1109/IROS.2014.6942940.

[23] X. Lagorce, G. Orchard, F. Gallupi, B. E. Shi, and R. Benosman, "HOTS: A hierarchy of event-based time-surfaces for pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, Jan. 2017, doi: 10.1109/TPAMI.2016.2574707.

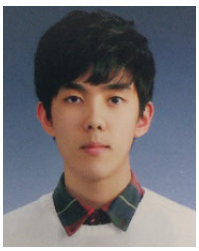
[24] M. Mahowald, "VLSI analogs of neuronal visual processing: A synthesis of form and function," Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 1992.

[25] L. A. Camuñas-Mesa, T. Serrano-Gotarredona, S. H. Ieng, R. B. Benosman, and B. Linares-Barranco, "On the use of orientation filters for 3D reconstruction in event-driven stereo vision," *Frontiers Neurosci.*, vol. 8, p. 48, Mar. 2014, doi: 10.3389/FNINS.2014.00048.

[26] S. Li, Z. Zhang, R. Mao, J. Xiao, L. Chang, and J. Zhou, "A fast and energy-efficient SNN processor with adaptive clock/event-driven computation scheme and online learning," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 4, pp. 1543–1552, Apr. 2021, doi: 10.1109/TCSI.2021.3052885.

[27] A. Sankaran, P. Detterer, K. Kannan, N. Alachiotis, and F. Corradi, "An event-driven recurrent spiking neural network architecture for efficient inference on FPGA," in *Proc. Int. Conf. Neuromorphic Syst.*, Sep. 2022, pp. 1–8, doi: 10.1145/3546790.3546802.

[28] J. Hasler, "Large-scale field-programmable analog arrays," *Proc. IEEE*, vol. 108, no. 8, pp. 1283–1302, Aug. 2020, doi: 10.1109/JPROC.2019.2950173.



JUNG-GYUN KIM received the B.S. degree in electrical engineering and computer science from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2017, where he is currently pursuing the Ph.D. degree.

His main subject is the development of the neuromorphic processor for the event-based vision sensor. His research interests include neuromorphic systems and vision application.

DONGHWAN SEO received the B.S. degree in electrical engineering from Konkuk University, Seoul, South Korea, in 2011, and the M.S. degree from the School of Mechatronics, Gwangju Institute of Science and Technology, Gwangju, South Korea, in 2013, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering and Computer Science.

His research interests include CMOS image sensors and analog-to-digital converters.

BYUNG-GEUN LEE (Member, IEEE) received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2000, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Texas at Austin, in 2004 and 2007, respectively.

From 2008 to 2010, he was a Senior Design Engineer at Qualcomm Inc., San Diego, CA, USA, where he had been involved in the development of various mixed-signal ICs. Since 2010, he has been with the Gwangju Institute of Science and Technology. He is currently an Associate Professor with the School of Electrical Engineering and Computer Science. His research interests include high-speed data converters, CMOS image sensors, and neuromorphic system design.