

PAPER • OPEN ACCESS

A retina-inspired pathway to real-time motion prediction inside image sensors for extreme-edge intelligence

To cite this article: Subhradip Chakraborty *et al* 2025 *Neuromorph. Comput. Eng.* **5** 034005

View the [article online](#) for updates and enhancements.

You may also like

- [Phasic attractors for flexible and adaptive working memory in spiking recurrent neural networks](#)
William S Girão, Thomas F Tiotto, Jelmer P Borst et al.
- [Historical perspective and opportunity for computing in memory using floating-gate and resistive non-volatile computing including neuromorphic computing](#)
Jennifer Hasler and Arindam Basu
- [Leveraging dendritic complexity for neuromorphic computing](#)
Suma G Cardwell, Mark Plagge, Luke Parker et al.



PAPER

OPEN ACCESS

RECEIVED
2 April 2025REVISED
1 July 2025ACCEPTED FOR PUBLICATION
14 July 2025PUBLISHED
23 July 2025

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



A retina-inspired pathway to real-time motion prediction inside image sensors for extreme-edge intelligence

Subhradip Chakraborty^{1,*} , Shay Snyder² , Md Abdullah-Al Kaiser¹, Maryam Parsa², Gregory Schwartz³ and Akhilesh R Jaiswal^{1,*}

¹ Department of Electrical and Computer Engineering, University of Wisconsin Madison, Madison, United States of America

² Department of Electrical and Computer Engineering, George Mason University, Fairfax, United States of America

³ Feinberg School of Medicine, Northwestern University, Evanston, United States of America

* Authors to whom any correspondence should be addressed.

E-mail: chakrabort42@wisc.edu and akhilesh.jaiswal@wisc.edu

Keywords: retina inspired sensor, motion prediction, neuromorphic sensor, image sensor, bipolar signal, 3D integration

Abstract

The ability to predict motion in real time is fundamental to many maneuvering activities in animals, particularly those critical for survival, such as attack and escape responses. Given its significance, it is no surprise that motion prediction (MP) in animals begins in the retina. Similarly, autonomous systems utilizing computer vision could greatly benefit from the capability to predict motion in real time. Therefore, for computer vision applications, MP should be integrated directly at the camera pixel level. Towards that end, we present a retina-inspired neuromorphic framework capable of performing real-time, energy-efficient MP directly within camera pixels. Our hardware-algorithm framework, implemented using GlobalFoundries' 22 nm FDSOI technology, integrates key retinal MP compute blocks, including a biphasic filter, spike adder, non-linear circuit, and a 2D array for multi-directional MP. Additionally, integrating the sensor and MP compute die using a 3D Cu–Cu hybrid bonding approach improves design compactness by minimizing area usage and simplifying routing complexity. Validated on real-world object stimuli, the model delivers efficient, low-latency MP for decision-making scenarios reliant on predictive visual computation, while consuming only 18.56 pJ/MP in our mixed-signal hardware implementation.

1. Introduction

Animal eyes are marvels of evolution, each uniquely adapted to meet ecological demands and enhance species survivability [1]. Once thought of as mere organs for detecting and filtering light, eyes are now recognized as sophisticated structures that encode and process a vast range of visual information. This specialization is particularly pronounced in the retina, a peripheral part of the central nervous system containing parallel circuits that operate to extract distinct visual features [2]. Unlike biological eyes, engineered 'eyes', such as image sensors in machine vision, are rigid and lack computational capabilities or intelligence [3]. One critical feature the retina computes is motion prediction (MP), which allows animals to predict the future position of an object from its past trajectory. [4] This ability to anticipate motion is crucial for survival, allowing animals to react quickly to dynamic environmental changes.

Prediction is challenging because much of the environmental data lacks predictive value, making it hard to separate meaningful motion cues from irrelevant ones [5]. Since much of the environmental data do not aid in accurate forecasting, it becomes hard to identify movements or changes that genuinely contribute to prediction from those that do not. Thus, to use this capacity effectively, neural circuits prioritize predictive information, discarding non-predictive data during the encoding process. [6] proposed a neuroscience MP model that relies on gain control. When a stimulus enters the edge of a retinal ganglion cell's (RGCs) receptive field (RF), it triggers photoreceptor and bipolar cells (BCs), while also activating gain control mechanisms that reduce the light response volume. Although it is simple and effective, it does not account for varying

motion speeds. To address this issue of different stimulus velocities, later neuroscience research describes a nonlinear interaction through electrical and chemical synapses via gap junctions [7], which aids in MP.

In computer-vision applications such as object tracking, motion estimation, and scene understanding, classical frame-based pipelines pair high-resolution CMOS image sensors (CIS) with deep neural networks [3]. Although accurate, these systems capture and process redundant full frames at fixed rates, which inflates data bandwidth, power consumption, and end-to-end latency—factors that make on-chip deployment difficult, especially for always-on or edge platforms. Previous work addressed motion tracking on-chip through hardware-software co-design, using traditional algorithms categorized into temporal difference methods (detecting changes in pixel intensities across frames) [8], correlation methods (calculating the product of a pixel's intensity with a neighboring delayed intensity) [9], and cluster-based methods (grouping motion events by specific criteria) [10]. Although [11] proposes a hardware-friendly algorithm which performs motion tracking but it does not account for MP, resulting in multi-megabyte memory traffic and milliwatt-level power budgets. By contrast, event-based vision sensors emit asynchronous address-event streams that report only brightness changes, dramatically reducing spatial-temporal redundancy. Yet most event-driven trackers focus on detecting motion rather than predicting it [11].

To implement these above motion tracking algorithms on-chip while accounting for MP, efficient hardware architectures are required. Although efforts to mimic retinal circuits using solid-state technology date back to the 1980s [12], interest in integrating bio-inspired functionalities into electronic image sensors grew with the advent of neuromorphic sensors [13, 14]. Recent work demonstrates retinal features like differential motion detection [15] and motion tracking [16], achieved through electronics and 2D materials [17]. These works primarily focus on detecting and tracking past movement but cannot predict future motion. Additionally, some of them use 2D materials, which necessitate significant alterations to the existing CMOS foundry and face scaling and variability challenges. Some studies have mimicked retinal characteristics using complex CMOS based biphasic circuits [18], which, however, consume significant power and occupy substantial area (due to power hungry amplifier). Notably, these approaches focus on replicating the detailed electrochemical behavior of MP using mixed-signal circuits, rather than on image sensor or camera technology using retinal computation capabilities.

MP is one of several features extracted within the mammalian retina [4]. Prior work has introduced spiking models for motion detection, such as the spiking elementary motion detector [19], a biologically inspired analog to the Reichardt Detector [20], which detects past motion trajectories. Similarly, temporal difference encoding has been used for motion detection in spiking neural networks [21]. These methods focus on identifying where motion has occurred in the past. In contrast, our MP circuit projects motion into future time steps, enabling prediction of an object's future position rather than merely detecting past motion. Other retinal-inspired systems [22–25] focus on distinct features such as object motion sensitivity (OMS) and looming detection (LD). OMS detects object motion during ego-motion (camera or observer movement), while LD identifies rapidly approaching objects indicative of potential threats. MP, by contrast, captures fundamentally different information by enabling forward-time extrapolation of motion trajectories. To the best of our knowledge, our proposed MP circuit represents the first biologically inspired software-hardware co-design for real-time MP.

In this work, we propose a retina-inspired neuromorphic vision sensor that leverages a hardware-algorithm framework based on retinal neuroscience insights to perform pixel-level MP. Rather than simply replicating retinal functions, our approach develops a model, an algorithmic workflow, and a dedicated hardware architecture that seamlessly integrates the MP computational capabilities of the retina with those of a CMOS image sensor. To the best of our knowledge, this is the first work to present a retina-inspired neuromorphic camera for MP. In summary, the key contributions of this paper include:

- (i) We identify and select only the essential critical features from the broad set of characteristics exhibited by biological retinal cells that are necessary to implement MP functionality. This feature-specific approach minimizes design complexity, reduces energy consumption, and reduces area overhead, allowing integration of the proposed MP circuits into CMOS image sensors.
- (ii) We develop an algorithmic model inspired by neurological processes to enable precise MP at the pixel level, providing real-time MP capabilities.
- (iii) We propose both a mixed-signal and a fully digital design for MP using GlobalFoundries' 22 nm FDSOI technology. The mixed-signal design emulates essential retinal functions for MP and incorporates key compute elements, including a biphasic filter, signal divider, spike adder, and a non-linear circuit. It also features a 2D retinal circuit array capable of performing MP in all four directions. The fully digital implementation of the 2D MP architecture offers more flexible scaling characteristics using automated design tools.

- (iv) We proposed a 3D integration approach using Cu–Cu hybrid bonding, featuring one hybrid bond per pixel. This method reduces both area overhead and routing complexity, facilitating a more compact design. We validated the functionality of our MP model on real-world stimulus [26, 27], demonstrating its practical implementation and pixel-level operating efficiency.

The rest of the paper is organized as follows: section 2 explains the mechanics of MP in retina. Sections 3 and 4 presents the proposed retina inspired neuromorphic camera algorithm and hardware design. Section 5 shows the simulation results of the architecture. Section 6 concludes the paper.

2. Motion prediction circuit in the retina

In the vertebrate retina, photoreceptors form the initial layer, where they transduce light into analog voltage signals. These signals are then relayed to BC in the inner retina. There are approximately 15 types of BC [2], which in turn connect to around 60 types of amacrine cells [28], refining the signal for specialized computations. These refined signals are integrated by over 40 types of RGCs [29], each with distinct RFs. RGCs transmit analog spikes that encode specific visual information, such as object motion, direction, orientation, and color contrast. Each RGC corresponds to a unique visual feature transmitted to the brain, enabling extraction of key features like MP [30]. Retinal cells send signals (chemical and electrical) to each other through gap junctions, which helps in performing MP. [5]

The implementation of MP, as shown in figure 1(c), occurs in three main Stages. The first Stage involves BCs, which generate a signal upon luminance changes. Each BC includes a biphasic filter (BPF) followed by a nonlinear activation function. The BPF allows the forward (in the direction of motion) signal flow during the positive phase but prevents backward (opposite to the direction of motion) signal flow into the BCs during the negative or refractory phase, while the nonlinear circuit propagates the signal between BCs, arranged in a hexagonal grid. In the second Stage, a signal amplitude divider distributes the incoming signal, produced by convolving the spike with the BPF in a given ratio as, shown in figure 1(d). The peak value is passed onto the first RGC. Finally, the third Stage adds the lower signal value (from the divider) after passing it through the non-linear circuit and combining it with the peak value of the second RGC. This design enables RGCs to generate amplified signals when a moving object activates a series of BCs along its path. The BCs in the motion path receive input from subsequent cells, amplifying their signal compared to those outside the path.

In the following sections, we first implement the neuroscience model of MP as an algorithm. This approach facilitates the identification of key design parameters essential for hardware realization, while also offering a framework to assess the feasibility and efficiency of biological retinal circuits in extracting MP behaviors. Unlike simplistic stimuli commonly employed in neuroscience experiments on surgically extracted retinas [31], our approach focuses on validating complex, real-world stimuli to better understand and replicate MP behavior for technological use-cases [27]. Based on the neuroscience understanding of the MP circuit and algorithmic model, we then develop a mixed-signal CMOS-based MP hardware wherein spatio-temporal computations for MP are distributed in two 3D integrated chips via Cu–Cu hybrid bonds.

3. Algorithmic implementation

In this section, we present the algorithmic model of MP and its correspondence with the neuroscience model discussed in section 2 and shown in figure 2. The implementation follows a structured multi-stage approach, where Stage 1 is captured in algorithms 1 and 2, handling the generation of BC signals and the application of a biphasic filter to process BC signals. Stage 2, implemented in algorithm 3, models the interaction of gap junctions, facilitating lateral signal propagation and enhancing motion-related processing. Stage 3 is implemented in algorithms 4 and 5, describing the accumulation of higher amplitude signals during spike post-processing to refine the final representation. This structured approach ensures a clear transition from the neuroscience model to an algorithmic framework, laying the foundation for efficient hardware realization.

We implement a software defined implementation of our 3D MP circuit with PyTorch [32], receiving BC signals (*dvsSpikes*) as input and returning RGC activations *MP*, representing the likelihood of future motion. Building on the neurological model, we design an algorithm that enables fine-tuning of key MP parameters, such as the type of spatial filters, and convolutional characteristics. This algorithmic formulation is instrumental in demonstrating the applicability and scalability of our approach on complex visual tasks, as well as guiding hardware circuit design. A detailed description of all parameters is shown in table 1.

Bipolar Cell (BC) Activations: *dvsSpikes* is a 4D matrix with the shape (T, P, H, W) where T is the number of discrete time bins, $P = 2$ is the polarity of each spike, H is the height in pixels, and W is the width in pixels. The MP circuit assumes that each DVS pixel is within a BC RF with an activation pattern approaching a

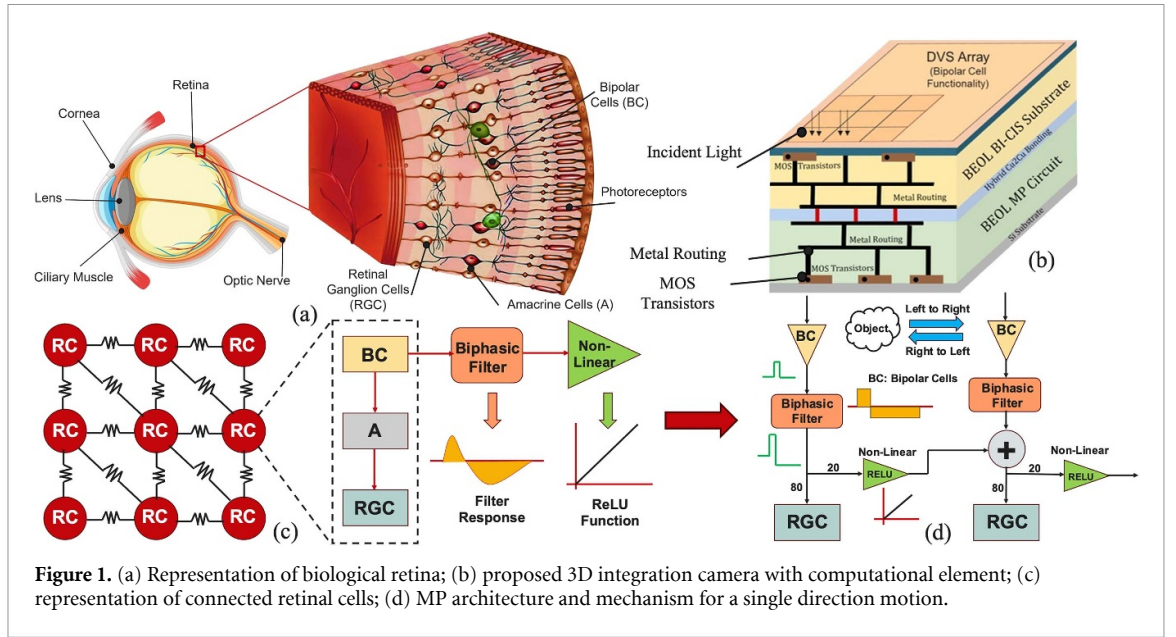


Figure 1. (a) Representation of biological retina; (b) proposed 3D integration camera with computational element; (c) representation of connected retinal cells; (d) MP architecture and mechanism for a single direction motion.

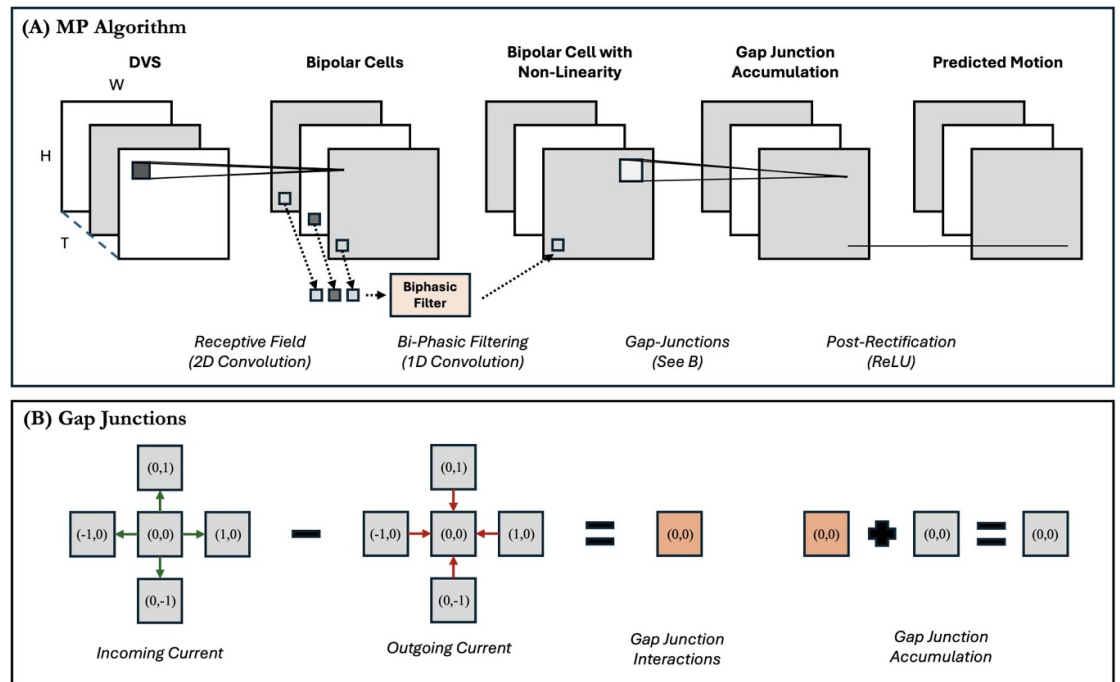


Figure 2. Overview of the motion predictive (MP) algorithm applied to DVS data. (A) End-to-end processing pipeline of the MP algorithm, conceptually similar to convolutional neural networks. Each stage of the algorithm is labeled with its corresponding computational operation, illustrating how DVS frames are transformed into motion-predictive cell activations. (B) Detailed visualization of the gap-junction integration phase, where cumulative incoming and outgoing currents between connected cells are computed and incorporated into the activation dynamics of each target cell.

normal distribution centered in the RF. We simulate this with a 2D convolution using a single input channel representing the incoming positive spikes and a single output channel with multiple features corresponding to the Gaussian responses from individual BCs. This process is shown in algorithm 1 utilizing three parameters and returning a matrix BC representing the BC activations with shape (T, k_x, k_y) . The size of the RFs is parameterized by the kernel size k_s along with the stride that is calculated as $s \leftarrow \text{round}(k_s * k_0)$ where $k_0 = 0.5$ is the percent overlap between subsequent RFs. More information the specifics of these symbols and their derivation can be found in table 1.

Bipolar Cell Non-linearities: Next, the non-linear polarized BPFs are convolved along the time dimension of each BC. This process is shown in algorithm 2 accepting two arguments (BC and f_i) and returning the

Algorithm 1. Bipolar Cell Activations.

Require: $dvsSpikes, k_s, k_o$
 $bcKernel \leftarrow buildGaussianKernel(k_s)$
 $s \leftarrow round(k_s * k_o)$
for $i = 0$ to $len(dvsSpikes)$ **do**
 $spikes \leftarrow dvsSpikes[i]$
 $BC[i] \leftarrow applyKernel(bcKernel, s, spikes)$
end for
return BC

Algorithm 2. Bipolar Cell Non-Linearities.

Require: BC, f_t
 $filter \leftarrow buildBiphasicFilter(f_t)$
for $t = 0$ to $len(BC)$ **do**
 $BC_NL_POS[t] \leftarrow applyKernel(BC[t], filter)$
 $BC_NL_NEG[t] \leftarrow applyKernel(BC[t], -1 * filter)$
end for
return BC_NL_POS, BC_NL_NEG

Table 1. The mathematical symbols used to describe our simulated motion prediction framework along with their default values. (-) The symbol is dependent upon the data sequence. All parameters were specified based on neuroscience experiments and simulations described in [4].

Symbol	Meaning	Value
H	number of y -axis elements	—
W	number of x -axis elements	—
k_s	the receptive field of each RGC	50
k_o	the percent overlap of each RGC	0.5
k_x	the number of RGCs along the x -axis	$\lceil (W - k_s) / (k_s - \text{int}(k_s * k_o)) \rceil + 1$
k_y	the number of RGCs along the y -axis	$\lceil (H - k_s) / (k_s - \text{int}(k_s * k_o)) \rceil + 1$
s	kernel stride in pixels	$round(k_s * k_o)$
T	the number of discrete time bins	—
f_t	time window of biphasic filter	250 ms
GJ_{nf}	gap junction normalization factor	0.5
γ	number of neighbors per RGC	4
GJ_s	gap junction scaler	GJ_{nf} / γ

positive and negative filtered outputs (BC_NL_POS and BC_NL_NEG). These non-linear filters are the critical element that allows the MP circuit to predict motion by exciting or inhibiting future activations.

Gap Junction (GJ) Interactions: Our MP circuit also includes support for localized interactions between the individual Bipolar cells, with another procedure shown in algorithm 3 that accepts 4 arguments (BC_NL, k_x, k_y , and GJ_s) and returns a single 5D matrix GJ representing the incoming and outgoing interactions across all GJ. BC_NL represents the activations of the Bipolar cells following the integration of the Bi-phasic non-linearity. k_x and k_y represent the numbers of RGCs along the x and y axes. Lastly, the GJ_s represents the scaling factor used to modulate the amount of current transmitted through the gap junctions. The GJ normalizer GJ_s inhibits the influence of neighbor activity for each individual RGC. This process is responsible for limiting the amount of current an individual cell can share with its neighbors. It is calculated as $GJ_s = GJ_{nf} / \gamma$ where GJ_{nf} is a real number between 0 and 1 and γ is the number of neighbors. For all experiments, we specify $GJ_{nf} = 0.5$ and $\gamma = 4$. We apply this function across both positive and negative non-linear activations (BC_NL_POS and BC_NL_NEG) resulting in two 5D matrices: $BC_NL_POS_GJ$ and $BC_NL_NEG_GJ$.

Gap junction interaction accumulation: With the GJ interactions calculated for both the positive and negative BC non-linearities, we accumulate the difference of the sum of incoming and outgoing interactions as shown in algorithm 4. We apply this process over both GJ matrices ($BC_NL_POS_GJ$ and $BC_NL_NEG_GJ$) and both activation matrices (BC_NL_POS, BC_NL_NEG) resulting in two BC activation matrices with GJ interactions (POS_GJ, NEG_GJ).

Algorithm 3. Gap Junction Interactions.

Require: BC_NL, k_x, k_y, GJ_s

```

1:  $GJ \in \mathbb{R}^{k_x \times k_y \times k_x \times k_y \times t}$ 
2:  $value \leftarrow BC\_NL[x, y, i] * GJ_s$ 
3:  $\mathcal{T} = \{0, 1, \dots, T-1\}$ 
4:  $X = \{0, 1, \dots, k_x-1\}$ 
5:  $Y = \{0, 1, \dots, k_y-1\}$ 
6:  $D = \{-1, 0, 1\}$ 
7: for  $t, x, y, dx, dy \in \text{combos}(\mathcal{T}, X, Y, D, D)$  do
8:    $nx \leftarrow x + dx; ny \leftarrow y + dy$ 
9:   if not  $(0 \leq nx < k_x \text{ and } 0 \leq ny < k_y)$  then
10:    skip iteration
11:   end if
12:    $incoming \leftarrow \text{False}; outgoing \leftarrow \text{False}$ 
13:   if  $(ny < y \text{ and } (nx \leq x)) \text{ or } (ny == y \text{ and } nx < x)$  then
14:      $incoming \leftarrow \text{True}$ 
15:   else if  $(ny > y \text{ and } nx \geq x) \text{ or } (ny == y \text{ and } nx > x)$  then
16:      $outgoing \leftarrow \text{True}$ 
17:   end if
18:   if  $incoming$  then
19:      $GJ[nx, ny, x, y, t] += value$ 
20:   else if  $outgoing$  then
21:      $GJ[x, y, nx, ny, t] -= value$ 
22:   end if
23: end for
24: return  $GJ$ 

```

Algorithm 4. Accumulate Gap Junction Interactions.

Require: BC, GJ, T, k_x, k_y

```

 $\mathcal{T} = \{0, 1, \dots, T-1\}$ 
 $X = \{0, 1, \dots, k_x-1\}$ 
 $Y = \{0, 1, \dots, k_y-1\}$ 
for  $t, x, y \in \text{combos}(\mathcal{T}, X, Y)$  do
   $in \leftarrow \sum GJ[:, :, x, y, t]$ 
   $out \leftarrow \sum GJ[x, y, :, :, t]$ 
   $BC[x, y, t] += in - out$ 
end for
return  $BC$ 

```

Algorithm 5. Post-Rectification.

Require: POS_GJ, NEG_GJ

```

 $POS\_GJ[POS\_GJ < 0] = 0$ 
 $NEG\_GJ[NEG\_GJ < 0] = 0$ 
 $RGC = POS\_GJ + NEG\_GJ$ 
return  $MP$ 

```

Post Rectification: Aggregating everything together, we remove all elements less than zero from both activation matrices and accumulate everything into the final RGC neural activity MP . This process is shown procedurally in algorithm 5.

4. Hardware implementation

This section presents the hardware architecture of MP and its correspondence with the neuroscience and algorithmic models discussed in sections 2 and 3. The circuit implementation follows a structured multi-stage approach, mirroring the algorithmic framework. Stage 1, described in section 2, is implemented using a dynamic vision sensor (DVS) to generate BC spikes, corresponding to algorithm 1. Algorithm 1 describes the use of a Gaussian filter; however, this component is omitted from our hardware design, as it

does not significantly affect the effectiveness of the MP circuit. It is important to note that precise implementation of a Gaussian filter is not strictly necessary, its effect can be reasonably approximated using a simple square filter, with minimal impact on overall performance. In our hardware prototype, we chose not to implement the Gaussian filter because doing so would introduce considerable area and power overhead. Instead, we employ a unit step response, which offers a more power- and area-efficient alternative while preserving the essential functional behavior of the original neuroscience-inspired model. Similar, approximation neglecting the Gaussian filtering has been used for other retinal features like OMS as shown in [25]. Algorithm 2, which defines the biphasic filter, is realized through a combination of an MP block and a motion-selective (MS) filter, detailed later in this section. The 1D Motion Prediction subsection implements Stages 2 and 3, corresponding to algorithms 3 and 4, where amplified spikes are generated based on object motion. Additionally, the hardware also generates predictive spikes before the object reaches the next BC, enabling the implementation of real-time MP on-chip. Finally, the 2D MP subsection extends the 1D architecture, scaling it to a hexagonal grid while incorporating a readout circuit for spike detection. This structured approach ensures that the hardware accurately replicates both the neuroscience and algorithmic models, maintaining fidelity to the underlying biological processes.

The proposed hardware operates on a predictive principle, incorporating a dedicated predictive node (V_P) and an amplified voltage node (V_D). The predictive node forecasts object motion in all four cardinal directions based on a defined prediction level, which is explained in detail later. The amplified voltage node generates an amplified output when the actual object motion aligns with the predicted motion. This mechanism emulates the neurological process of MP observed in the retina, enabling real-world object prediction in our hardware architecture.

4.1. DVS - BCs Activation

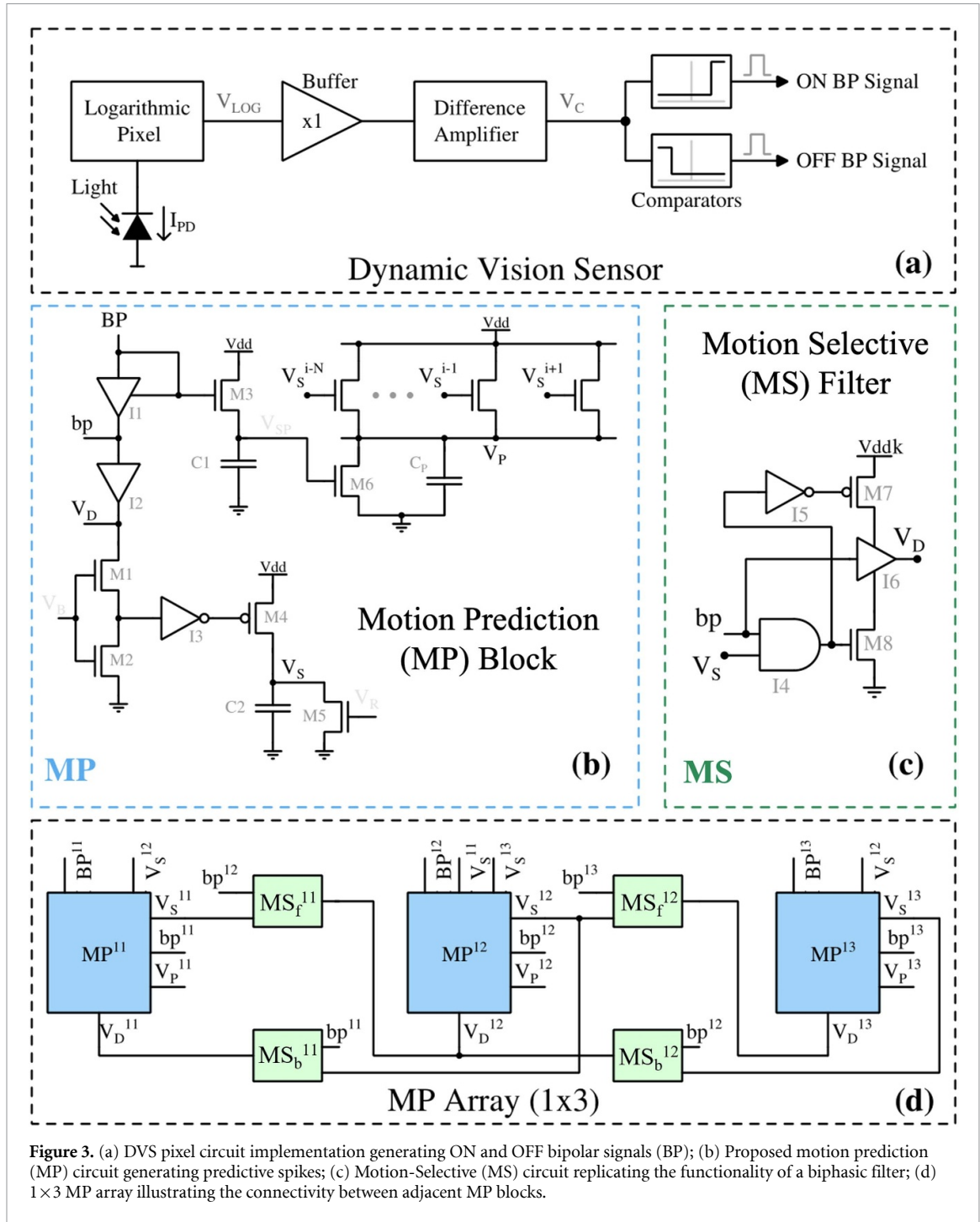
The DVS camera pixel emulates retinal BCs by employing a contrast-sensitive mechanism that generates ON and OFF bipolar signals (BP) in response to scene contrast variations, as illustrated in figure 3(a) [33, 34]. The circuit comprises a logarithmic receptor, voltage buffer, difference amplifier, and thresholding components. The logarithmic photoreceptor converts incoming light into a logarithmic voltage output, which is then isolated by a source follower buffer ($\times 1$) to protect the sensitive pixel node (V_{LOG}) before being fed into the difference amplifier. A capacitive feedback difference amplifier [35] asynchronously computes the voltage gradient corresponding to changes in light intensity. Finally, the output voltage of the difference amplifier (V_C) is processed by two thresholding circuits, one comparing contrast increases with a threshold and the other comparing contrast decreases with a threshold, to generate an ON/OFF bipolar signal. For our MP circuit, detailed in section 4.2, we utilize bipolar signals from DVS pixels as input, regardless of their polarity.

4.2. 1D MP—gap junction interaction

Figure 3(b) shows the schematic of the proposed 1D MP block. The input bipolar signals (BP) from each DVS pass through a power-gated buffer (I1) and are sent to the bipolar node (bp) of the MP block, serving as the input for the MP circuit. This design minimizes power consumption by enabling signal propagation only when a bipolar signal is generated. Additionally, these buffers can operate at a lower supply voltage than the DVS sensor's, allowing power and speed optimization of the compute circuit in advanced process nodes. With lower supply voltage, the power consumption in CMOS circuits reduces [36]. This is the reason behind implementing the buffers, which converts to a lower voltage. Another buffer (I2) connects the bp node to V_D , propagating the bipolar signal when the object with predictive motion first activates the DVS pixels.

The node V_D , which emulates the output functionality of MP RGC cells, is connected to a network of resistive dividers modeled using NFET transistors (M1 and M2) operating in the linear region. Next, an inverter-based comparator circuit (I3) with a fixed threshold processes the signal. The voltage at the V_D node is scaled by the divider network, mimicking the gap junction fraction, before being fed into the comparator (I3). When the scaled V_D voltage exceeds the comparator threshold, this information is stored by precharging the MOM capacitor (C2) via the M4 transistor, as shown in figure 3(b). This capacitor manages the time lag between the bipolar signals caused by changes in the object's speed by temporarily storing the signal value until the next bipolar signals arrive, ensuring reliable MP.

The BPF, as discussed in section 2, operates in two distinct phases: a positive phase, where signals are allowed to pass, and a negative (or refractory) phase, where incoming signals are blocked. This filtering mechanism is crucial for detecting motion in a specific direction. For instance, when a stimulus moves from left to right, adjacent BCs generate signals in response to the change in light intensity. The BPF ensures that only forward-moving signals (left-to-right) are transmitted, while suppressing any backward-moving signals (right-to-left). The BPF is implemented using the MS circuit as shown in figure 3(c), which determines motion direction by analyzing the present and past locations of spikes in surrounding cells. The storage node



V_S holds information about past spikes of the adjacent cells, while the bp node represents the bipolar signal information of the current cell. If both spikes occur simultaneously, the MS circuit generates an amplified spike (similar to a spike adder), which is then sent to the V_D node of current cells. The amplified spike is created by using a supply voltage of V_{ddk} , which is higher than the V_{dd} used in the MP circuit and through buffer I6.

The predictive node (V_P) is designed to identify pixels where an object is expected to move next while excluding locations it has already passed. It consists of a spike storage node (V_{SP}) that records spike occurrences at a given location and integrates predictive logic to anticipate motion. When adjacent pixels have previously generated spikes, their corresponding MP blocks store this information at node V_S , leading to the charging of capacitor (C_P). However, if a new BP signal is detected at the current pixel, indicating the object's presence, the V_{SP} node discharges the capacitor, dynamically clearing the predicted movement path. Let's analyze the process when an object moves MP¹¹ to MP¹² in a left-to-right direction, illustrating the step-by-step operation of the MP.

- When the object reaches MP^{11} , the spike storage node V_{sp}^{11} becomes charged, indicating its presence. Since MP^{11} is the first MP block encountered, it does not receive any predictive signal V_s^{12} from its neighboring MP block MP^{12} . As a result:
 - The predictive node V_p^{11} remains at zero because the charged V_{sp}^{11} establishes a discharge path through the M6 transistor.
 - Meanwhile, MP^{12} receives the predictive signal V_s^{11} from MP^{11} , causing V_p^{12} to charge. This indicates the anticipated movement of the object toward MP^{12} .
- As the object moves to MP^{12} , the generation of BP^{12} triggers the charging of the spike storage node V_{sp}^{12} via the M3 transistor. As a result:
 - The M6 transistor establishes a discharge path for the capacitor C_p within MP^{12} , resetting the predictive node V_p^{12} to zero.
 - MP^{12} transmits the predictive signal V_s^{12} to MP^{11} , but since V_{sp}^{11} retains its previous charge, it prevents the charging of V_p^{11} . This ensures that previously visited locations are excluded from anticipation.
- Initially, when the object was at MP^{11} , the circuit anticipated MP^{12} as the next likely location. However, upon reaching MP^{12} , it does not predict MP^{11} again, as that location has already been visited.
- The number of transistors connected to the predictive node V_p depends on the number of future locations to be anticipated. For example, if the system is designed to predict four positions ahead in each direction, a total of 16 transistors would be required.

4.3. 2D MP array

The 1D array has two directional paths: a forward path for left-to-right motion and a backward path for right-to-left motion. During left-to-right movement, the forward path serves as the primary route for MP, while the backward path is crucial for right-to-left motion. Between two MP blocks, as shown in figure 4(a), there are two MS blocks. In the case of a 2D MP array, a similar strategy is applied across columns to enable top-to-bottom and bottom-to-top MP, with each MP block connected to four MS blocks to facilitate MP in all directions. When motion occurs from top to bottom or bottom to top, the RGC nodes (V_D) and the predictive voltage nodes (V_P) are charged and discharged according to the method described in section 4.2. Note, diagonal connections are excluded from the circuit design because, given the practical size difference between the object and the pixels, they would contribute to increased power consumption without enhancing prediction performance. The object is adequately large to be detected through horizontal and vertical connections with reasonably good performance. Moreover, the predictive level—defined by the number of surrounding cells the object may move to—can be extended by adding more NFETs in parallel at the (V_P) node circuit, enabling greater anticipation of potential movement. The predicted spikes can be transmitted using the standard address event representation (AER) method, following a conventional timing diagram [37, 38]. The handshaking signals, including row request (RR), row acknowledge (RA), column request (CR), and column acknowledge (CA), are illustrated in figure 4.

Figure 5 demonstrates a one-level MP mechanism, utilizing four NFETs to enable movement detection by one pixel in four possible directions. To illustrate the circuit's functionality step by step, consider an object moving from left to right, as shown in figure 5. The amplified spike (AM) at the RGC node corresponds to the V_D node, while the predicted spike (P) corresponds to the V_P node. As the object transitions from frame 1 to frame 2, the DVS pixels generate bipolar signals due to contrast variations caused by the object's movement. In this example, $bp(6,7)$, $bp(7,7)$, and $bp(8,7)$ produce bipolar signals shown in red in frame 2 of figure 5(a) (an example bipolar signal for the pixel location (6,7) is shown in figure 5(c) at $1 \mu s$), which are then transmitted to their respective RGC nodes (V_D) and stored in the V_S nodes. Within this 2D grid representation, the first number in parentheses indicates the row index, while the second represents the column index. Since spikes are present at the V_S nodes of these cells, the surrounding cells in all four directions generate predictive spikes at their respective V_P nodes, as depicted with yellow in frame 2 (top) of figure 5(b). For instance, the predictive node voltage at pixel location (6,8) goes high, as seen in figure 5(e) at $1 \mu s$. A similar behavior is observed in the predictive nodes of the surrounding cells. Notably, since the object appears in these pixels for the first time, the corresponding RGC nodes output bipolar-like signals (without amplification from the MS block), as observed in $AM(6,7)$ shown in green in figure 5(d) at $1 \mu s$. A similar behavior will be observed at the RGC nodes of these cells, as shown in frame 2 (bottom) of figure 5(b).

As the object transitions from frame 2 to frame 3, the DVS generates bipolar signals at pixel locations (6,8), (7,8), and (8,8), resulting in $bp(6,8)$, $bp(7,8)$, and $bp(8,8)$ receiving these signals, as illustrated in frame 3 of figure 5(a). Consequently, the predictive nodes of the surrounding pixels generate spikes due to updates in the V_S nodes at these locations. The spikes in the predictive nodes are depicted in frame 3 (top) of figure 5(b), with an example spike at $P(6,9)$ shown in figure 5(e) at $2.5 \mu s$. For example, when the object is at position (6,7), the predictive node at $P(6,8)$ activates simultaneously, demonstrating its anticipatory response. Notably, the predictive nodes of pixels previously occupied by the object are reset, as exemplified by

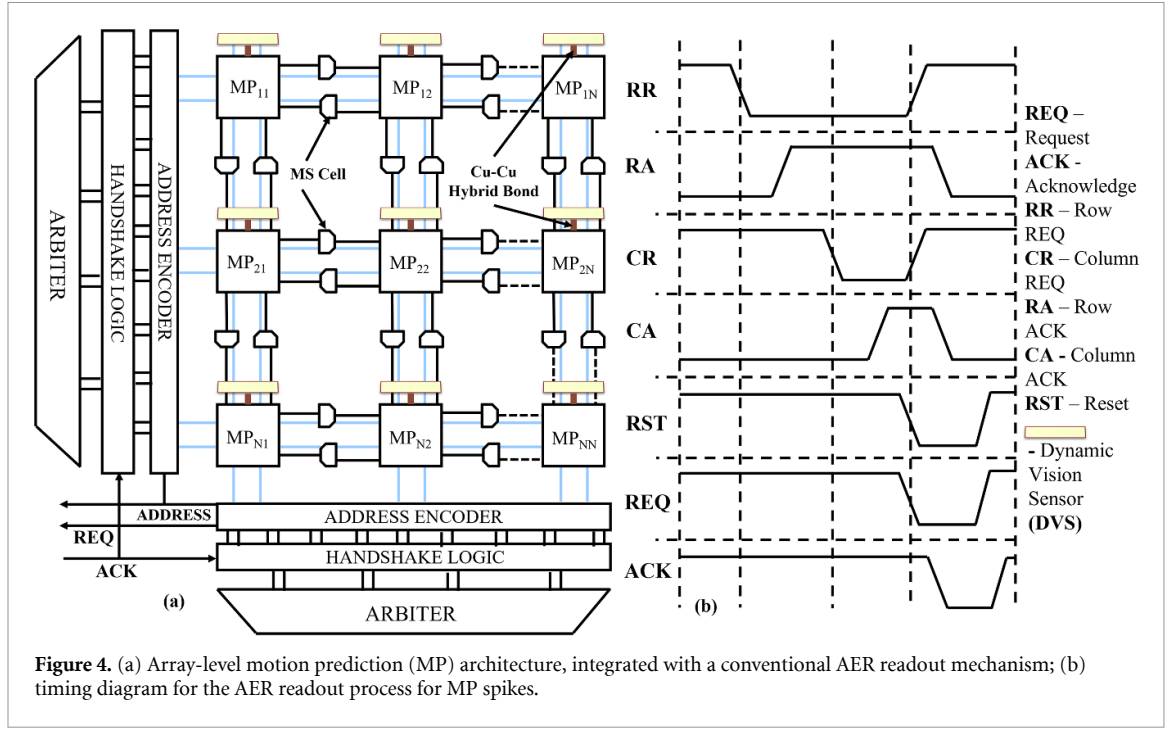


Figure 4. (a) Array-level motion prediction (MP) architecture, integrated with a conventional AER readout mechanism; (b) timing diagram for the AER readout process for MP spikes.

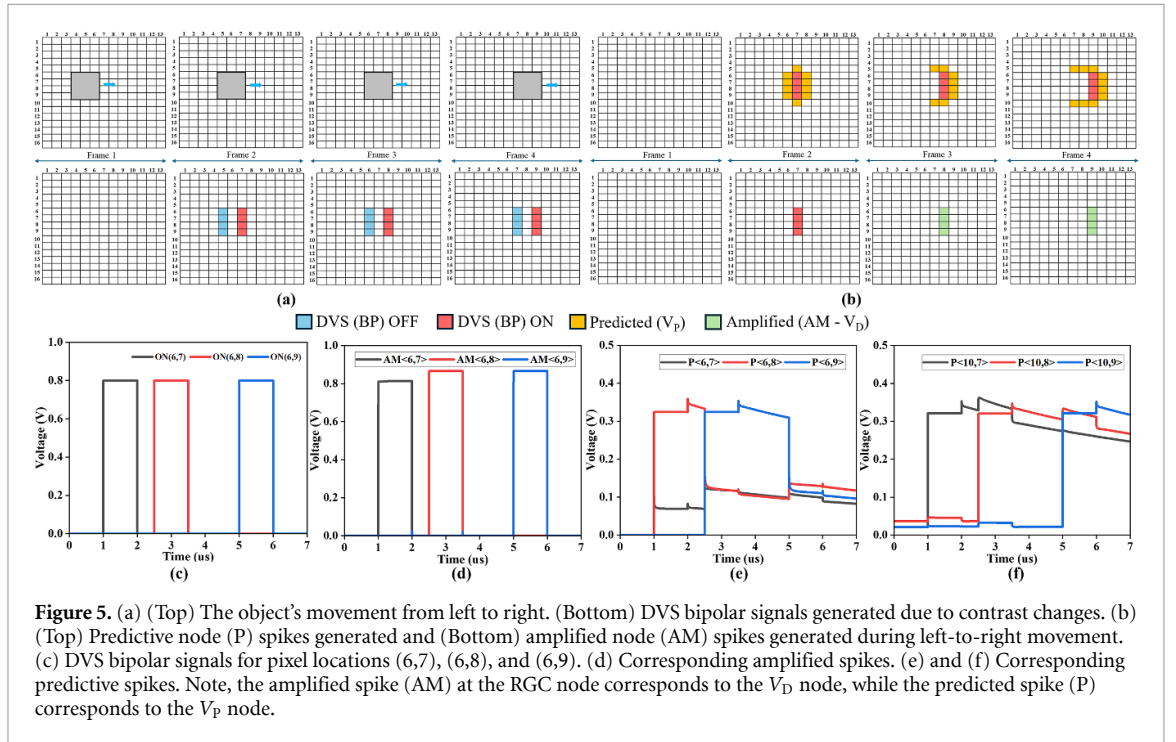


Figure 5. (a) (Top) The object's movement from left to right. (Bottom) DVS bipolar signals generated due to contrast changes. (b) (Top) Predictive node (P) spikes generated and (Bottom) amplified node (AM) spikes generated during left-to-right movement. (c) DVS bipolar signals for pixel locations (6,7), (6,8), and (6,9). (d) Corresponding amplified spikes. (e) and (f) Corresponding predictive spikes. Note, the amplified spike (AM) at the RGC node corresponds to the V_D node, while the predicted spike (P) corresponds to the V_P node.

the P(6,8) node decreasing, as seen in figure 5(e) at $2.5 \mu s$. Additionally, the RGC nodes corresponding to pixel locations (6,8), (7,8), and (8,8) receive amplified signals from the MS circuits, as observed in AM(6,8) in figure 5(d) at $2.5 \mu s$. As the object continues to move, the surrounding predictive nodes generate spikes, while all the pixels the object has already passed are reset. The paths that the object did not cross, but were predicted, show a slower decay, as seen in figure 5(f), where pixel locations (10,7), (10,8), and (10,9) generated predicted spikes during the object's movement. Meanwhile, the RGC nodes along the object's path receive amplified spikes, confirming that the object is following one of the predicted paths.

4.4. 3D integration

Figure 1(b) presents a 3D heterogeneously integrated design of our proposed retina-inspired camera for MP [39, 40]. The system is composed of two distinct dies: (1) a backside-illuminated CMOS image sensor (BI-CIS) that accommodates the DVS pixels and bias circuitry, and (2) a bottom die incorporating the 2D

Algorithm 6. Digital Flow of Motion Prediction.

Require: $DVS[N][N]$
Ensure: $amplified_spike[N][N]$, $prediction[N][N]$

```

1: Initialize parameters:
2:    $fraction \leftarrow 0.5$ ,  $threshold \leftarrow 0.5$ 
3:    $amplified\_fraction \leftarrow 1.5$ 
4: for  $i, j \in \{0, 1, \dots, N-1\} \times \{0, 1, \dots, N-1\}$  do
5:    $(reduced\_spike[i][j], Vstore[i][j]) \leftarrow \text{SpikeDivision}(DVS[i][j], fraction, threshold)$ 
6: end for
7: for  $i, j \in \{0, 1, \dots, N-1\} \times \{0, 1, \dots, N-1\}$  do
8:   Compute neighborhood states:
9:    $top \leftarrow (i_1 \geq 1) ? Vstore[i_1 - 1][j_1] : 0$ 
10:   $bottom \leftarrow (i_1 + 1 < N) ? Vstore[i_1 + 1][j_1] : 0$ 
11:   $left \leftarrow (j_1 \geq 1) ? Vstore[i_1][j_1 - 1] : 0$ 
12:   $right \leftarrow (j_1 + 1 < N) ? Vstore[i_1][j_1 + 1] : 0$ 
13:   $amplified\_spike[i][j] \leftarrow$ 
14:     $\text{DigitalMS}(DVS[i][j], top, bottom, left, right, amplified\_fraction)$ 
15: end for
16: for  $i, j \in \{0, 1, \dots, N-1\} \times \{0, 1, \dots, N-1\}$  do
17:   $spike\_occurred[i][j] \leftarrow \mathbb{I}(DVS[i][j] > 0)$ 
18: end for
19: for  $i, j \in \{0, 1, \dots, N-1\} \times \{0, 1, \dots, N-1\}$  do
20:   Compute 4-neighbor states:
21:    $top\_neighbors \leftarrow [(i \geq k + 1) ? Vstore[i - k][j] : 0 \mid k \in \{0, 1, 2, 3\}]$ 
22:    $bottom\_neighbors \leftarrow [(i + k < N) ? Vstore[i + k][j] : 0 \mid k \in \{1, 2, 3, 4\}]$ 
23:    $left\_neighbors \leftarrow [(j \geq k + 1) ? Vstore[i][j - k] : 0 \mid k \in \{1, 2, 3, 4\}]$ 
24:    $right\_neighbors \leftarrow [(j + k < N) ? Vstore[i][j + k] : 0 \mid k \in \{1, 2, 3, 4\}]$ 
25:    $prediction[i][j] \leftarrow \text{PredictNode}(top\_neighbors, bottom\_neighbors,$ 
26:      $left\_neighbors, right\_neighbors, spike\_occurred[i][j])$ 
27: end for
28: return  $amplified\_spike, prediction$ 

```

MP compute array, MS circuit, AER readout circuit, and other peripheral components. A major advantage of this heterogeneous integration lies in the ability to fabricate the bottom die using an advanced technology node while keeping the DVS pixel array in a separate BI-CIS die, typically designed in a lagging process node. The adoption of 3D stacking significantly reduces routing complexity and enhances processing efficiency (due to short routing), while the integration of MOM capacitors above transistors optimizes area utilization without compromising pixel density. Moreover, the design employs fine-pitched hybrid Cu–Cu bonding [41], enabling compact and efficient interconnects. Each DVS pixel transmits its bipolar signal to the MP compute circuit via a single Cu–Cu hybrid bond. Given that DVS pixels are relatively large [33], the bonding pitch and MP circuit can be precisely aligned with the top-layer DVS pixels. Furthermore, 3D integration enhances transmission energy efficiency over traditional 2D integration by minimizing interconnect length [42].

4.5. Complete digital implementation of MP

We also introduce an alternative fully digital design approach through an automated digital flow, employing Questa Modelsim [43] for simulation and Synopsys DC Compiler for [44] synthesis using the GF 22 nm FDSOI node. This approach is particularly suited for scenarios where power limitations are not the primary concern. Algorithm 6 details the implementation of a retina-inspired MP circuit using the digital blocks. The design includes key MP computational components, such as the voltage divider, predictive node, and amplified spike, all realized through digital blocks. Lines 4 to 6 implement a resistive divider equivalent using a floating-point multiplier, storing the value at $Vstore$ (V_S) once it surpasses the threshold. Lines 7 to 15 correspond to the MS circuit mentioned earlier, responsible for generating the amplified spike for predicting object motion. Line 17 represents the V_{SP} node, which holds the value if a bipolar signal indicates that the object has passed that node. Lines 19 to 27 define our predictive node (V_P), which offers a four-level prediction, similar to having 16 transistors connected to the V_P node as in the mixed-signal approach. The mixed-signal readout circuit is implemented using an asynchronous AER scheme, while the digital flow necessitates the use of additional flip-flops to store intermediate results and outputs, as well as a threshold circuit.

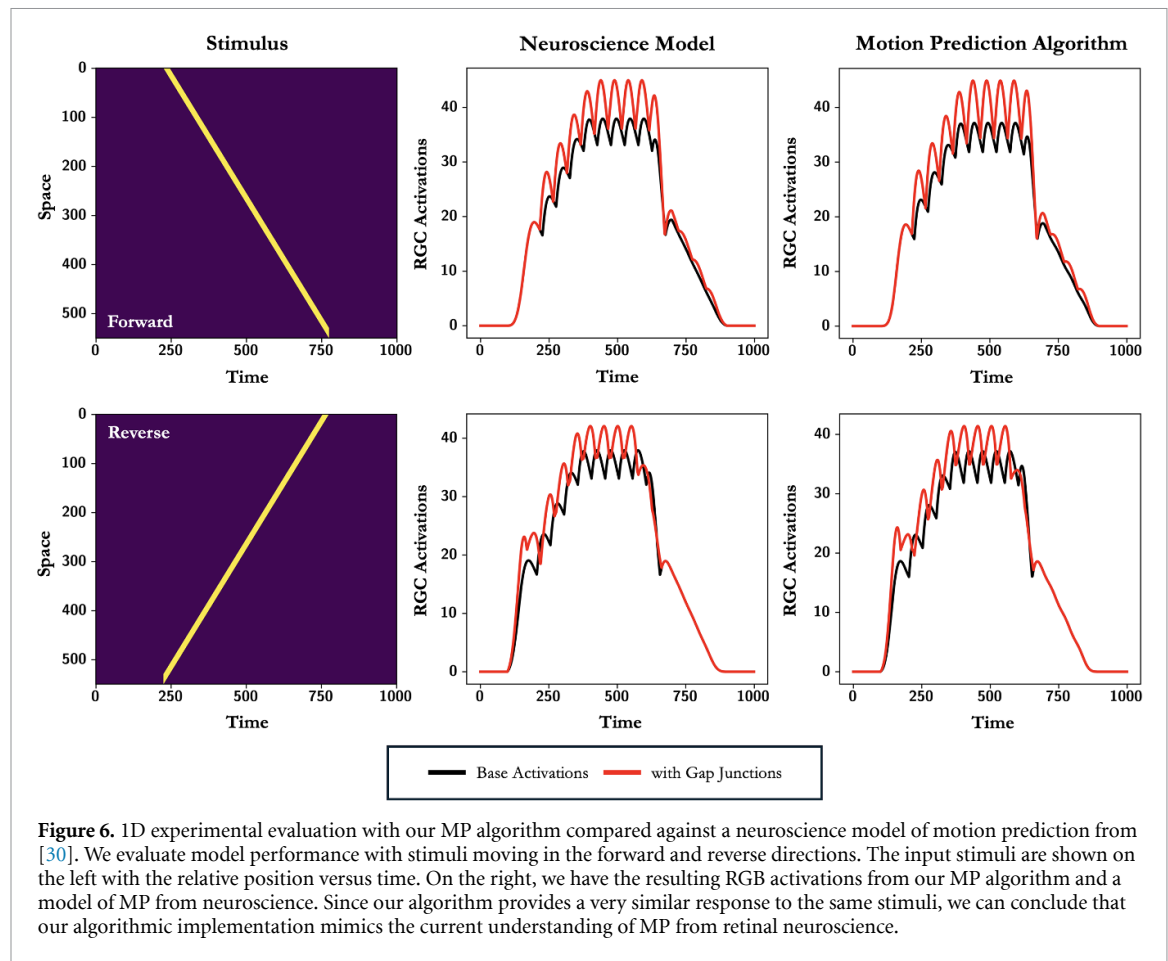


Figure 6. 1D experimental evaluation with our MP algorithm compared against a neuroscience model of motion prediction from [30]. We evaluate model performance with stimuli moving in the forward and reverse directions. The input stimuli are shown on the left with the relative position versus time. On the right, we have the resulting RGB activations from our MP algorithm and a model of MP from neuroscience. Since our algorithm provides a very similar response to the same stimuli, we can conclude that our algorithmic implementation mimics the current understanding of MP from retinal neuroscience.

5. Results & discussion

This section presents the algorithmic and hardware verification results of our proposed retina-inspired MP framework.

5.1. Algorithmic implementation

We evaluate the software implementation of our MP circuit using a simulated motion stimulus from [30] and a real-world data sequence from [27], which we refer to as S1 and S2, respectively. S1 is a simulated sequence depicting one-dimensional object motion in both left and right (forward and backward) directions. These results are shown in figure 6, where we compare our MP algorithm against the neuroscience model from [30] in a one-dimensional setting to emphasize the increased RGC activity in the direction of predicted motion. These results indicate that our algorithm closely matches the neuroscience model in MP while also generalizing to multiple directions. Furthermore, our implementation of gap junctions to connect multiple RGC cells achieves performance comparable to the neuroscience model, indicating that our approach closely replicates its biological counterpart by enhancing current transferability during MP. We pre-process this sequence in two steps: (1) cropping the images horizontally to ensure a square final resolution and (2) converting the images to DVS format through grayscale frame differencing, where pixels increasing in luminance generate a positive bipolar signal (green), while those decreasing in luminance generate a negative bipolar signal (red). We chose this approach of cropping the images based on hardware constraints to simplify the design. This decision did not negatively impact the downstream performance of the MP algorithm, similar to how convolutional neural networks can generalise across input sizes using fixed-size kernels. To reduce noise in this process, we apply a spiking threshold of 50 to the individual pixel deltas before generating a spike.

As shown in figure 7(A), we visualize the output of our MP algorithm for time-deltas of 100 ms using S2. This value was chosen based on recent findings in experimental neuroscience, which identify 100 ms as the forecast limit of the biological MP circuit [30]. Our results demonstrate that the MP algorithm successfully anticipates future motion, as indicated by increased spiking activity in regions where motion is expected (Brighter regions—amplified spikes). The trail behind the bright regions signifies the suppression of

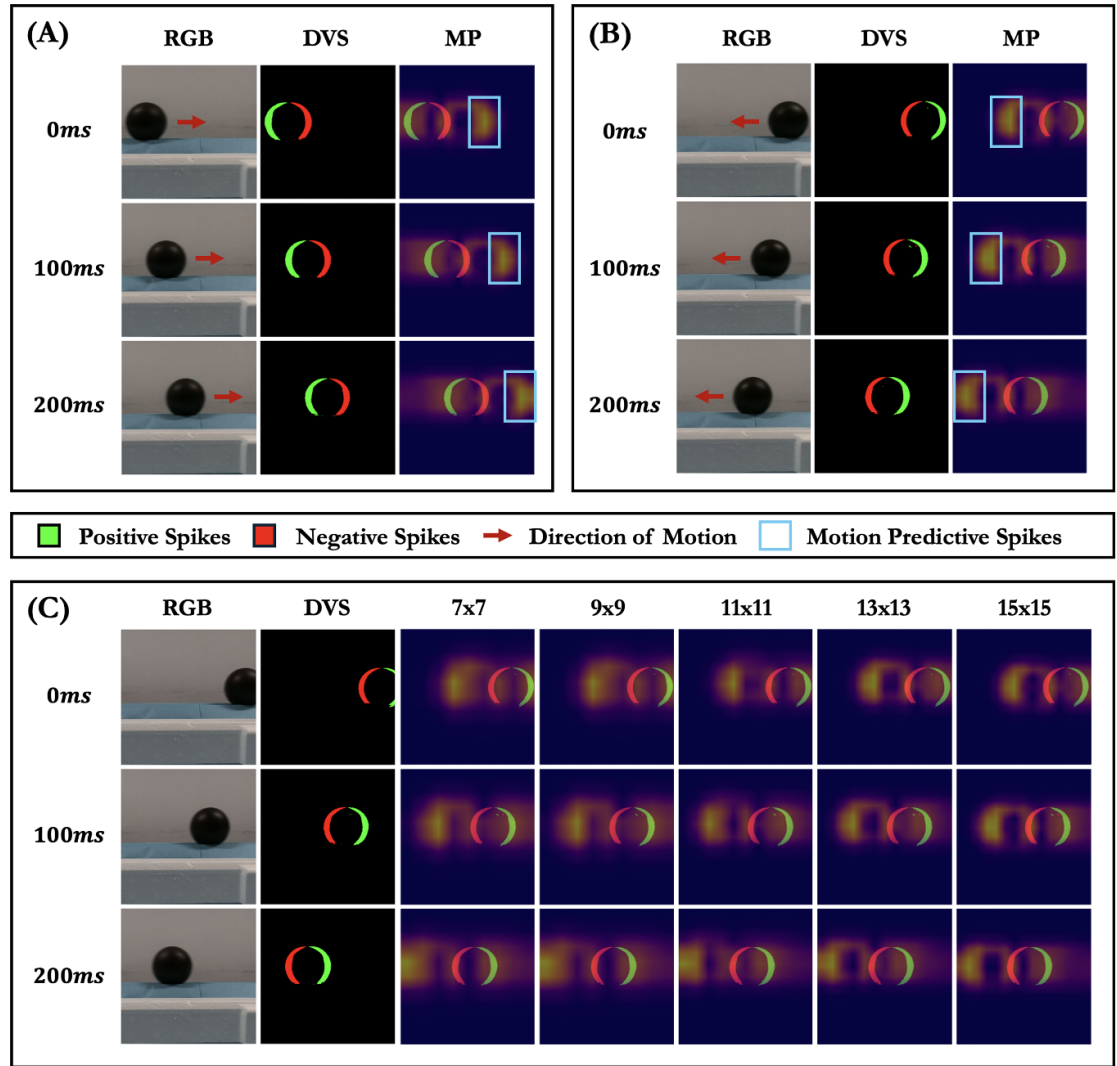


Figure 7. Experimental evaluation with our MP algorithm. (A) Our MP algorithm at three intervals: 0 ms, 100 ms, and 200 ms with a sequence from [27]. (B) Our MP algorithm with motion in the opposite direction of (A). (C) Evaluation of five motion prediction array sizes: 7×7 , 9×9 , 11×11 , 13×13 , and 15×15 .

predictive spikes due to the presence of a biphasic filter. To further validate the model, we reverse the direction of motion in figure 7(B), confirming the circuit's ability to predict motion in both directions. Additionally, figure 7(C) illustrates how varying the MP array size affects prediction characteristics: smaller arrays yield coarser predictions extending further into the future, while larger arrays provide higher-resolution predictions with a shorter temporal range. This trade-off between spatial resolution and temporal prediction will be crucial for downstream applications, and we plan to explore its implications in future work.

5.2. Circuit implementation

We validate the MP functionality of our proposed retina-inspired neuromorphic camera using HSPICE simulations with GlobalFoundries' 22 nm FDSOI technology. To evaluate the DVS-based contrast detection circuit, simulations are conducted considering local mismatches and global supply voltage variations, incorporating a 10 mV standard deviation in the nominal supply voltage to account for supply voltage fluctuations. Figure 8 presents Monte Carlo (MC) simulation results based on over 1000 samples for positive (left) and negative (right) contrast change scenarios. In these simulations, I_{PD1} and I_{PD2} represent photodetector currents at two different time instants, with $I_{PD1} > I_{PD2}$, at 30% contrast sensitivity, which refers to the smallest luminance change between frames that triggers ON/OFF bipolar signals. For the purpose of MC analysis, we use a 30% contrast sensitivity as an example simulation setup to illustrate the robustness of the contrast detection behavior. The contrast sensitivity is defined as the minimum relative luminance change $(I_{PD2} - I_{PD1})/I_{PD1}$ required to trigger an ON or OFF event, depending on the direction of the contrast change. In our extensive testing, the output voltage of each photoreceptor shows a standard

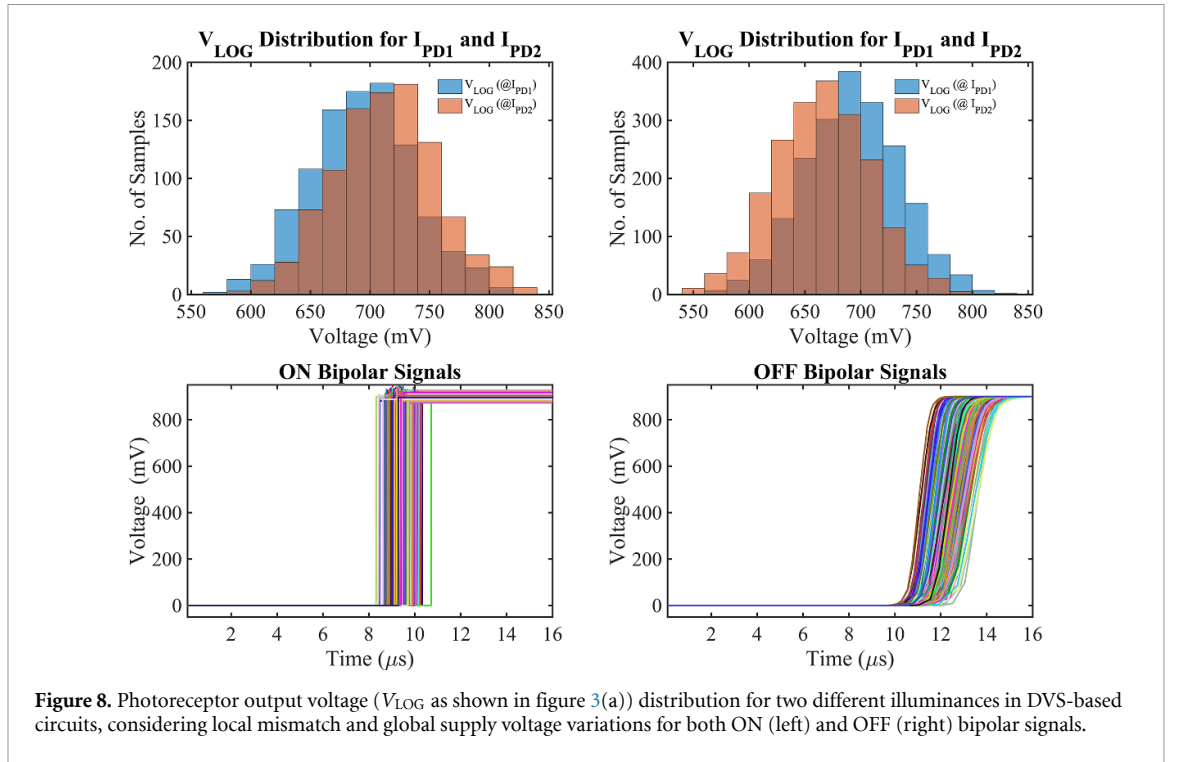


Figure 8. Photoreceptor output voltage (V_{LOG} as shown in figure 3(a)) distribution for two different illuminances in DVS-based circuits, considering local mismatch and global supply voltage variations for both ON (left) and OFF (right) bipolar signals.

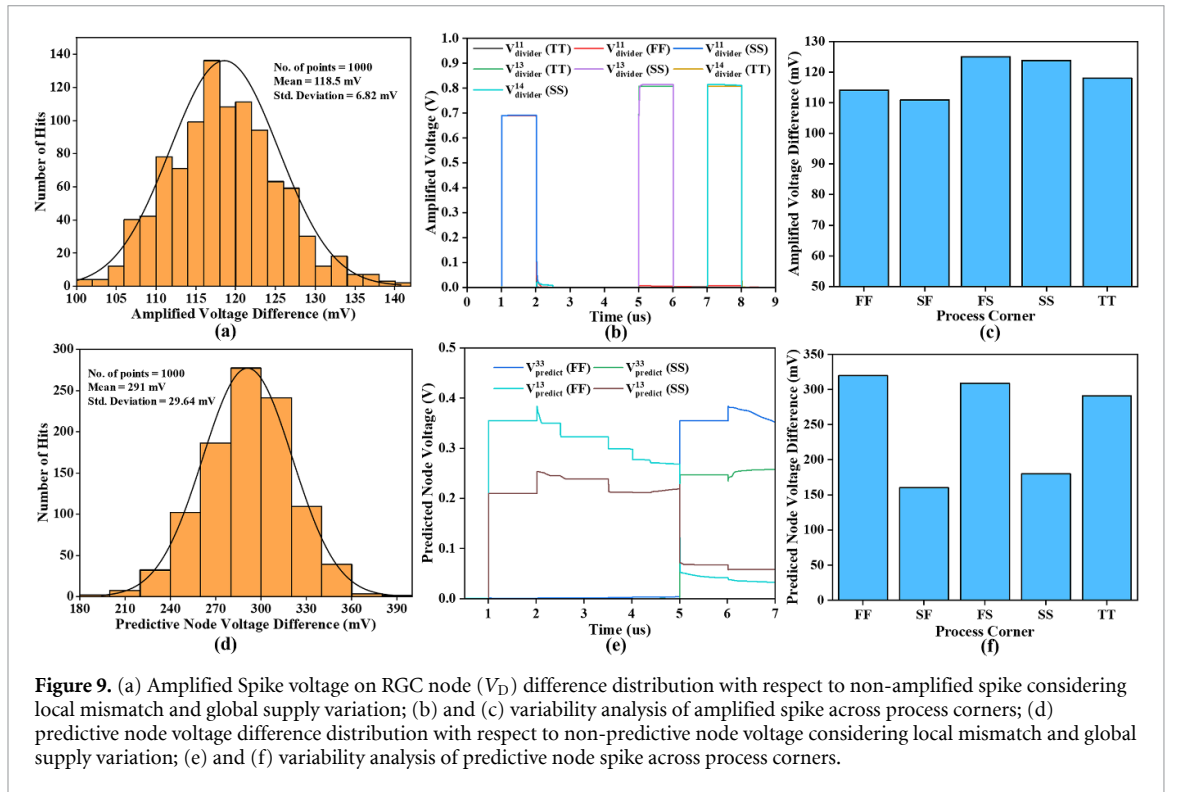
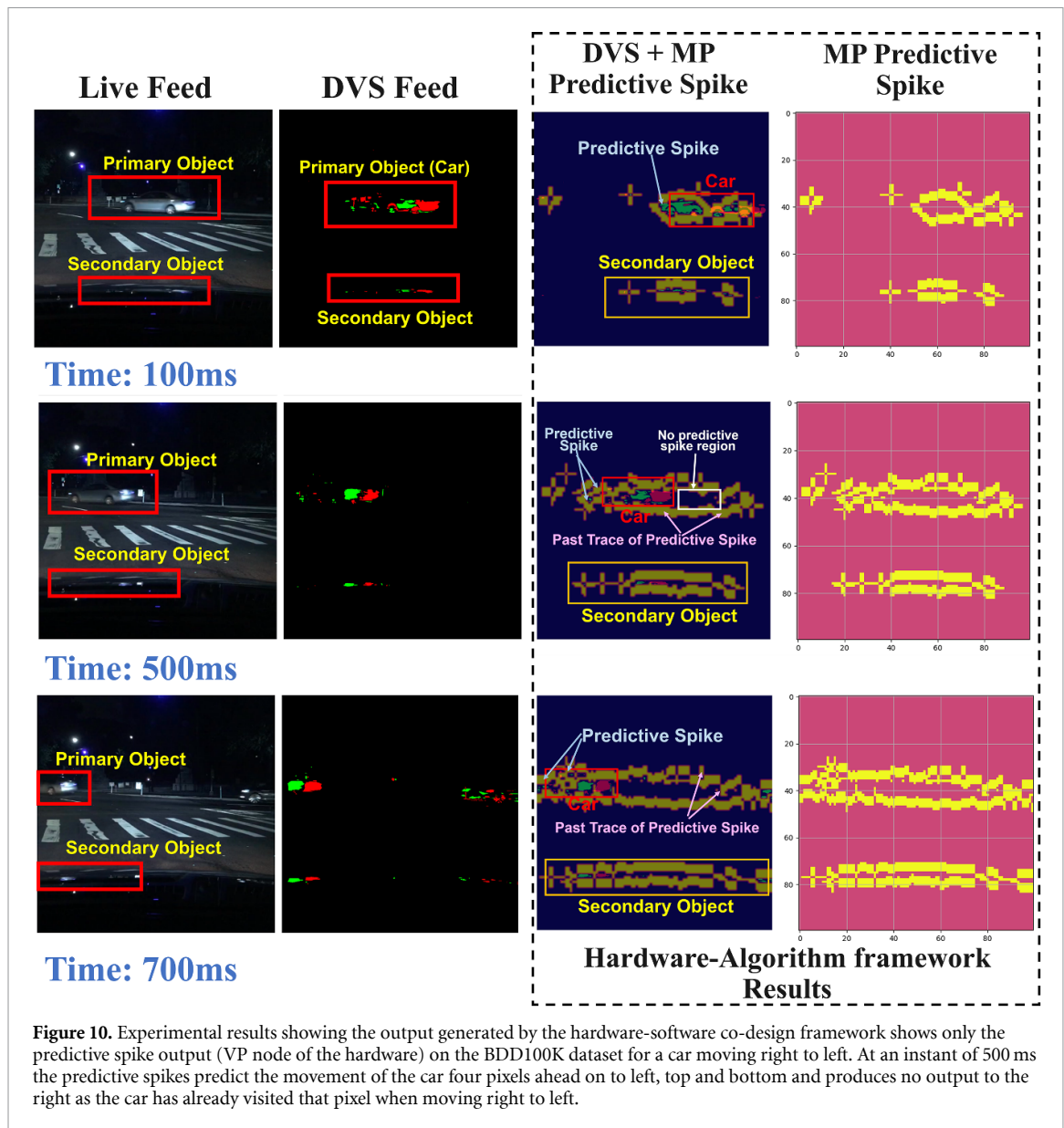


Figure 9. (a) Amplified Spike voltage on RGC node (V_D) difference distribution with respect to non-amplified spike considering local mismatch and global supply variation; (b) and (c) variability analysis of amplified spike across process corners; (d) predictive node voltage difference distribution with respect to non-predictive node voltage considering local mismatch and global supply variation; (e) and (f) variability analysis of predictive node spike across process corners.

deviation of 43.2 mV, causing overlapping voltage distributions (V_{LOG}) for I_{PD1} and I_{PD2} across various test samples. The worst-case difference in photoreceptor voltages between two consecutive frames among the 1000 samples is 18.4 mV, based on 30% contrast sensitivity. The difference voltage is amplified, and when it exceeds the threshold voltage, the DVS pixels asynchronously generate ON/OFF bipolar signals at different time instants, influenced by local variations and supply voltage fluctuations. The bipolar signal generation energy was 1.34 pJ, and the biasing energy was 0.32 pJ for a 1 microsecond simulation.

The timing waveforms of the amplified spike on the RGC node (V_D) and the predictive node (V_P), obtained from the proposed MP circuit under various process corners, are shown in figures 9(b) and (e). The amplified and predictive spikes of an MP array were evaluated in a controlled test setup, where an object



moves in one of four directions: left to right, right to left, top to bottom, or bottom to top. The signals V_D^{11} , V_D^{13} , and V_D^{14} represent the amplified outputs, highlighting the enhancement achieved through the predictive nature of the gap junction. For simulation, the supply voltage (V_{dd}) was set to 0.7 V, while the amplification voltage for the MS (V_{ddk}) was set to 0.9 V. A 16×16 2D MP array was simulated over $1 \mu s$ per bipolar signal, with energy consumption measured at 18.56 pJ/event for four prediction levels in each direction, resulting in a total of 16 predictive cells. To evaluate the robustness of the MP circuit, we included both local mismatch effects and global supply voltage variations under various process corners. Process corners in CMOS fabrication represent variations in transistor performance, where both PFET and NFET devices can operate at different speeds due to manufacturing differences. Among the standard process corners, two of the most critical are FF (Fast PFET, Fast NFET) and SS (Slow PFET, Slow NFET), which are commonly analyzed to evaluate best- and worst-case performance scenarios. The differences in spike levels between the predictive and amplified spikes, which characterize the MP behavior, were analyzed using MC simulations with 1000 samples, as shown in figures 9(a) and (d). The mean voltage difference for the amplified spikes was 118.5 mV with a standard deviation of 6.82 mV, while for the predictive spikes, the mean voltage difference was 291 mV with a standard deviation of 29.64 mV. As shown in figures 9(c) and (f), the voltage difference remains stable despite process and mismatch variations, demonstrating the robustness of the MP circuit under different operating conditions.

Figure 10 presents the MP results produced by our hardware-algorithm framework on the BDD100K dataset [26]. We begin by processing the input video from the dataset through a DVS filter, which converts the RGB frames into bipolar output responses. These responses are then passed through two stages: the

Table 2. Energy consumption comparison for different configurations.

Energy type	DVS Chip + digital MP	DVS Chip + mixed signal MP (3D integration)	DVS Chip + digital MP (3D integration)
MP compute energy	2.24 nJ	0.519 nJ	2.24 nJ
MP readout energy	345.52 pJ	4.9336 pJ	4.9336 pJ
MP total energy	2.585 nJ	0.5239 nJ	2.2449 nJ
Normalized MP total energy	1	0.202	0.868

Table 3. Comparison with previous state-of-the art work.

	2013 [11]	2021 [47]	2023 [48]	2024 [17]	This work
Material	CMOS	CMOS	CMOS	WS2	22 nm GF FDSOI
Model	Motion tracking	General	General	Motion tracking	Motion prediction
Functionality	temporal	temporal	Spatio-temporal	Spatio-temporal	Spatio-temporal
Mixed signal flow	Yes	No	No	Yes	Yes
Complete digital flow	No	Yes	Yes	No	Yes
Motion tracking	Yes	No	No	Yes	Yes
Motion prediction	No	No	No	No	Yes
Ganglion cell type	NA	No	ON and OFF	ON and OFF	ON and OFF
Speed adaption	No	No	No	Yes	Yes

initial algorithm inspired by the neuroscience model and our complete hardware-algorithm framework, which simulates the hardware behavior. The neuroscience-inspired algorithm highlights amplified spikes that indicate where the object is likely to move next. The MP predictive spike output, corresponding to the node V_P in our hardware, represents the MP with a look-ahead of four pixels (prediction level = 4) in each direction. This predictive capability is confirmed by the visual results and marked as predictive spike. To refine the output, past trace of predictive spike are filtered out by overlapping them with the output of the amplified node V_D , yielding a clean prediction. The past trace of predictive spike refer to MP predictive spikes generated by the hardware in directions where the object ultimately did not move. These spikes represent potential motion paths that were anticipated but not followed, and thus, no corresponding amplified response was observed in those pixels. These traces provide additional information about where the object came from. In our framework, such trace of predictions are filtered out by overlapping them with the output of the amplified node (V_D) $\pm$ prediction level, ensuring that only the actual motion path is retained in the final predicted output.

5.3. Energy comparison

Table 2 presents the energy consumption for three proposed configurations: (a) a DVS sensor chip with a digitally implemented MP logic chip integrated on the same PCB, (b) a DVS sensor chip with a 3D integrated mixed-signal MP circuit, and (c) a DVS sensor chip with a 3D integrated digital MP chip. The first approach, which combines the DVS chip and the digital MP on the same PCB, accounts for energy used in DVS static and event generation, as well as the digital MP circuit and the low-voltage differential signaling (LVDS) utilized for data transfer between the sensor and the MP die. The second and third architectures utilize 3D integration, employing either Cu–Cu hybrid bonding or through-silicon vias, in addition to the DVS static and event generation energy. The second architecture adopts a mixed-signal approach, while the third architecture uses a purely digital design. The table includes a simulation scenario with a 7×7 MP array, which generates 28 spikes, each lasting $1 \mu\text{s}$, over a 1 ms simulation period. The total energy consumption for DVS static and event generation is 15.68 nJ and 37.52 pJ, respectively. The energy consumption for LVDS is 12.34 pJ/bit [45], while 3D integrated die-to-die communication consumes only 176.2 fJ/bit [46]. From the table, it can be observed that the 3D integrated mixed-signal implementation offers significantly better energy efficiency compared to other two configurations.

5.4. Comparison with previous works

Our proposed approach has demonstrated on-chip MP in real time, with potential applications in threat detection and autonomous navigation. Table 3 provides a summary comparing our proposed retina-inspired framework with previous studies. Previous studies [11, 17] have primarily focused on motion tracking or direction prediction, both of which are inherently supported by our proposed architecture. We introduce a novel MP hardware design capable of predicting surrounding pixel motion by leveraging computational principles inspired by the biological retina. The proposed system is implemented using an advanced CMOS foundry-supported process design kit, which requires no specialized materials for fabrication. Furthermore,

we present a digital equivalent model of MP, which can be deployed on an FPGA for specific applications rather than general retinal emulation. This model incorporates the leaky integrate-and-fire neuron model on an FPGA, as previously demonstrated in [47, 48]. Additionally, we propose a 3D-integrated mixed-signal architecture that optimizes both energy efficiency and area utilization, offering a scalable and high-performance solution for real-time MP.

6. Conclusion

Our proposed retina-inspired neuromorphic framework bridges advancements in retinal neuroscience, mixed-signal circuit techniques, and algorithmic modeling to achieve efficient, pixel-level MP for machine vision applications. By identifying and selecting only the most essential retinal features necessary for MP, we reduce design complexity, minimize energy consumption, and lower area overhead, making our solution well-suited for integration into CMOS image sensors. Our hardware-algorithm framework is implemented using GlobalFoundries' 22 nm FDSOI technology and offers two design approaches: a mixed-signal implementation that emulates key biological retinal functions using specialized computational elements—such as a biphasic filter, signal divider, spike adder, and non-linear circuit—and a fully digital implementation that serves as a comparison for our proposed circuit. Furthermore, we introduce a novel 3D integration approach using Cu–Cu hybrid bonding with one hybrid bond per pixel, significantly optimizing area utilization and simplifying routing complexity, enabling a more compact and efficient hardware design. Validated on real-world object stimuli, our model demonstrates its ability to perform accurate MP with low energy consumption, achieving just 18.56 pJ/MP in our mixed-signal implementation. These results highlight the potential of our neuromorphic framework in predictive visual computation for decision-making applications, including robotics, autonomous navigation, and augmented reality.

Data availability statement

The data cannot be made publicly available upon publication because no suitable repository exists for hosting data in this field of study. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

The research was funded in part by National Science Foundation through Awards CCF2319617 and CCF2319619.

References

- [1] Land M F 2005 The optical structures of animal eyes *Curr. Biol.* **15** R319–23
- [2] Eggers E D and Lukasiewicz P D 2011 Multiple pathways of inhibition shape bipolar cell responses in the retina *Vis. Neurosci.* **28** 95–108
- [3] Gamal A E and Eltoukhy H 2005 CMOS image sensors *IEEE Circuits Devices Mag.* **21** 6–20
- [4] Schwartz G 2021 *Retinal Computation* (Academic)
- [5] Kuo S, Schwartz G and Rieke F 2016 Nonlinear spatiotemporal integration by electrical and chemical synapses in the retina *Neuron* **90** 320–32
- [6] Berry M J, Brivanlou I H, Jordan T A and Meister M 1999 Anticipation of moving stimuli by the retina *Nature* **398** 334–8
- [7] Trenholm S, Schwab D J, Balasubramanian V and Awatramani G B 2013 Lag normalization in an electrically coupled neural network *Nat. Neurosci.* **16** 154–6
- [8] Chi Y M, Mallik U, Clapp M A, Choi E, Cauwenberghs G and Etienne-Cummings R 2007 CMOS camera with in-pixel temporal change detection and ADC *IEEE J. Solid-State Circuits* **42** 2187–96
- [9] Andreou A, Strohhahn K and Jenkins R 1991 Silicon retina for motion computation 1991 *IEEE Int. Symp. on Circuits and Systems (ISCAS)* vol 3 pp 1373–6
- [10] Zhao B, Zhang X and Chen S 2011 A CMOS image sensor with on-chip motion detection and object localization 2011 *IEEE Custom Integrated Circuits Conf. (CICC)* pp 1–4
- [11] Liu X, Zhang M and Van der Spiegel J 2013 A low power multi-mode CMOS image sensor with integrated on-chip motion detection 2013 *IEEE Int. Symp. on Circuits and Systems (ISCAS)* pp 2416–9
- [12] Mead C A and Mahowald M 1988 A silicon model of early visual processing *Neural Netw.* **1** 91–97
- [13] Liu S-C and Delbruck T 2010 Neuromorphic sensory systems *Curr. Opin. Neurobiol.* **20** 288–95
- [14] Zhu M, He T and Lee C 2020 Technologies toward next generation human machine interfaces: From machine learning enhanced tactile sensing to neuromorphic sensory systems *Appl. Phys. Rev.* **7** 031305
- [15] Tseng K-C and Parker A C 2012 A neuromorphic circuit that computes differential motion 2012 *IEEE 55th Int. Midwest Symp. on Circuits and Systems (MWSCAS)* pp 89–92
- [16] Li L et al 2024 Adaptive machine vision with microsecond-level accurate perception beyond human retina *Nat. Commun.* **15** 6261
- [17] Peng Z et al 2024 Multifunctional human visual pathway-replicated hardware based on 2d materials *Nat. Commun.* **15** 8650

- [18] Han S, Kim T, Kim C and Lee S 2024 Design and simulation of artificial retinal stimulation ic with switched capacitor using si nanowire optical properties *Sci. Prog.* **107** 00368504241275372
- [19] Milde M B, Bertrand O J, Ramachandran H, Egelhaaf M and Chicca E 2018 Spiking elementary motion detector in neuromorphic systems *Neural Comput.* **30** 2384–417
- [20] Borst A 2000 Models of motion detection *Nat. Neurosci.* **3** 1168–1168
- [21] D'Angelo G, Janotte E, Schoepe T, O'Keeffe J, Milde M B, Chicca E and Bartolozzi C 2020 Event-based eccentric motion detection exploiting time difference encoding *Front. Neurosci.* **14** 2020
- [22] Clerico V, Snyder S, Lohia A, Kaiser M A-A, Schwartz G, Jaiswal A and Parsa M 2024 Retina-inspired object motion segmentation (arXiv:2408.09454)
- [23] Sinaga J, Clerico V, Kaiser M A-A, Snyder S, Lohia A, Schwartz G, Parsa M and Jaiswal A 2024 Hardware-algorithm re-engineering of retinal circuit for intelligent object motion segmentation 2024 *Int. Conf. on Neuromorphic Systems (ICONS)* (IEEE) pp 256–63
- [24] Snyder S, Thompson H, Kaiser M A-A, Schwartz G, Jaiswal A and Parsa M 2023 Object motion sensitivity: A bio-inspired solution to the ego-motion problem for event-based cameras (arXiv:2303.14114)
- [25] Yin Z, Kaiser M A-A, Camara L O, Camarena M, Parsa M, Jacob A, Schwartz G and Jaiswal A 2023 Iris: Integrated retinal functionality in image sensors *Front. Neurosci.* **17** 1241691
- [26] Yu F, Chen H, Wang X, Xian W, Chen Y, Liu F, Madhavan V and Darrell T 2020 Bdd100k: A diverse driving dataset for heterogeneous multitask learning *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp 2636–45
- [27] Qin Z, Fu Q and Peng J 2024 A computationally efficient and robust looming perception model based on dynamic neural field *Neural Netw.* **179** 106502
- [28] Yan W, Laboulaye M A, Tran N M, Whitney I E, Benhar I and Sanes J R 2020 Mouse retinal cell atlas: Molecular identification of over sixty amacrine cell types *J. Neurosci.* **40** 5177–95
- [29] Goetz J et al 2022 Unified classification of mouse retinal ganglion cells using function, morphology and gene expression *Cell Rep.* **40** 111040
- [30] Schwartz G W 2021 Chapter 13 - motion anticipation *Retinal Computation* ed G W Schwartz (Academic) pp 246–58
- [31] Sekirnjak C, Hottowy P, Sher A, Dabrowski W, Litke A and Chichilnisky E 2006 Electrical stimulation of mammalian retinal ganglion cells with multielectrode arrays *J. Neurophysiol.* **95** 3311–27
- [32] Paszke A et al 2019 Pytorch: An imperative style, high-performance deep learning library
- [33] Lichtsteiner P, Posch C and Delbruck T 2008 A 128 × 128 120 db 15 μ s latency asynchronous temporal contrast vision sensor *IEEE J. Solid-State Circuits* **43** 566–76
- [34] Serrano-Gotarredona T and Linares-Barranco B 2013 A 128 × 128 1.5% contrast sensitivity 0.9% fpn 3 μ s latency 4 mw asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers *IEEE J. Solid-State Circuits* **48** 827–38
- [35] Son B et al 2017 4.1 a 640 × 480 dynamic vision sensor with a 9 μ m pixel and 300meps address-event representation 2017 *IEEE Int. Solid-State Circuits Conf. (ISSCC)* (IEEE) pp 66–67
- [36] Ong S et al 2019 22nm fd-soi technology with back-biasing capability offers excellent performance for enabling efficient, ultra-low power analog and rf/millimeter-wave designs 2019 *IEEE Radio Frequency Integrated Circuits Symp. (RFIC)* (IEEE) pp 323–6
- [37] Boahen K A 2000 Point-to-point connectivity between neuromorphic chips using address events *IEEE Trans. Circuits Syst. II* **47** 416–34
- [38] Boahen K 2004 A burst-mode word-serial address-event link-i: transmitter design *IEEE Trans. Circuits Syst. I: Regular Pap.* **51** 1269–80
- [39] Kagawa Y et al 2016 Novel stacked CMOS image sensor with advanced cu2cu hybrid bonding 2016 *IEEE Int. Electron Devices Meeting (IEDM)* (IEEE) pp 8–4
- [40] Seo M-W et al 2021 A 2.6 e-rms low-random-noise, 116.2 mw low-power 2-mp global shutter CMOS image sensor with pixel-level adc and in-pixel memory 2021 *Symp. on VLSI Technology* (IEEE) pp 1–2
- [41] Kagawa Y, Hashiguchi H, Kamibayashi T, Haneda M, Fujii N, Furuse S, Hirano T and Iwamoto H 2020 Impacts of misalignment on 1 μ m pitch Cu-Cu hybrid bonding 2020 *IEEE Int. Interconnect Technology Conf. (IITC)* (IEEE) pp 148–50
- [42] Kaiser M A-A, Datta G, Sarkar S, Kundu S, Yin Z, Garg M, Jacob A P, Beerel P A and Jaiswal A R 2023 Technology-circuit-algorithm tri-design for processing-in-pixel-in-memory (p2m) *Proc. Great Lakes Symp. on VLSI 2023* pp 613–8
- [43] Siemens E D A 2024 Questa onesim: one simulator for your complete verification flow (available at: <https://eda.sw.siemens.com/en-US/ic/questa-one/simulation/questa-one-sim/>)
- [44] Compiler S D 2016 Synopsys design compiler Pages/default.aspx
- [45] Teja K R et al 2021 Design of 1.8 v lvsds transmitter in gf 22nm for associative memory 2021 *Int. Semiconductor Conf. (CAS)* (IEEE) pp 201–4
- [46] Zhang Y, Zhang X and Bakir M S 2018 Benchmarking digital die-to-die channels in 2.5-D and 3-D heterogeneous integration platforms *IEEE Trans. Electron Devices* **65** 5460–7
- [47] Ghanbarpour M, Naderi A, Haghir S and Ahmadi A 2021 An efficient digital realization of retinal light adaptation in cone photoreceptors *IEEE Trans. Circuits Syst. I: Regular Pap.* **68** 5072–80
- [48] Phillip P, Nath P K, Jainwal K, van Schaik A and Thakur C S 2025 Neuromorphic retina: an fpga-based emulator (arXiv:2501.08943)