NOLTA, IEICE

Invited Paper

Application of stochastic computing in brainware

Warren Gross [1a)], *Naoya Onizawa* [2b)], *Kazumichi Matsumiya* [3], *and Takahiro Hanyu* [4]

[1] *Department of Electrical and Computer Engineering, McGill University 3480 University Street, Montreal, Quebec H3A 0E9, Canada*

[2] *Frontier Research Institute for Interdisciplinary Sciences, Tohoku University 6-3 Aramaki Aza Aoba, Aoba-ku, Sendai 980-8578, Japan*

[3] *Graduate School of Information Sciences, Tohoku University 6-3-09 Aoba, Aramaki-aza, Aoba-ku, Sendai 980-8579, Japan*

[4] *Research Institute of Electrical Communication, Tohoku University 2-1-1 Katahira Aoba-ku, Sendai 980-8577, Japan*

[a)] *warren.gross@mcgill.ca*
[b)] *nonizawa@m.tohoku.ac.jp*

Received February 13, 2018; Revised June 11, 2018; Published October 1, 2018

Abstract: This paper reviews applications of stochastic computing in brainware LSI (BLSI) for visual information processing. Stochastic computing exploits random bit streams, realizing the area-efficient hardware of complicated functions, such as multiplication and tanh functions in comparison with binary computation. Using stochastic computing, we implement the hardware of several physiological models of the primary visual cortex of brains, where these models require such the complicated functions. Our vision BLSIs are implemented using Taiwan Semiconductor Manufacturing Company (TSMC) 65 nm CMOS process and discussed with traditional fixed-point implementations in terms of hardware performance and computation accuracy. In addition, an analog-to-stochastic converter is designed using CMOS and magnetic tunnel junctions that exhibit probabilistic switching behaviors for area/energy-efficient signal conversions to stochastic bit streams.

Key Words: probabilistic computation, visual information processing, signal processing, CMOS digital circuits, magnetic tunnel junctions

1. Introduction

Stochastic computing [1] represents information by a random sequence of bits, called a *Bernoulli* sequence and has been exploited for area-efficient hardware implementation. It was first introduced by von Neumann in 1950s [2] and had been fully developed in 1960s [3]. However, since then, it had not been well used, unlike traditional binary computation. In 2000s, stochastic computing has been applied to low-density parity-check (LDPC) decoders [4], where LDPC codes are known as one

Nonlinear Theory and Its Applications, IEICE, vol. 9, no. 4, pp. 406–422 DOI: 10.1587/nolta.9.406

of powerful error-correcting codes. The stochastic LDPC decoders exhibit powerful error-correcting capabilities with high area efficiencies [5–7]. Recently, it has been exploited for many applications, such as image processers [8–10], digital filters [11–13] and MIMO decoders [14].

In this paper, we review the applications of stochastic computing in brainware (brain-like) LSI. In our BLSI project, several hardware of brainware computing has been designed and implemented, including physiological models and deep neural networks. In Section 2, stochastic computing is briefly explained and the overview of brainware LSI (BLSI) is explained. Among several topics of our BLSI design, three hardware implementations are selected: the analog-to-stochastic converter (Section 3) [15], the simple cell model of primary visual cortex in brains (Section 4) [16–18], and the disparity energy model (Section 5) [19]. Section 6 concludes this paper.

2. Overview of Brainware LSI (BLSI)

2.1 Review of stochastic computing

Stochastic computing performs in probabilistic domain that the probabilities are represented by random sequences of bits. The probabilities are calculated by the frequency of ones or zeros in the sequence that can be represented by many different sequences of bits. For example, different sequences of bits (1011) and (1011) mean the same probability. There are two mappings for stochastic bit sequences: unipolar and bipolar coding. For a sequence of bits, $a(t)$, denote the probability of observing a '1' to be $P_a = P_r(a(t) = 1)$. In unipolar coding, the represented value, A, is $A = P_a$, $(0 \leq A \leq 1)$, while, in bipolar coding, the represented value, A, is $A = (2 \cdot P_a - 1)$, $(-1 \leq A \leq 1)$.

Stochastic circuit components are summarized in Fig. 1. Figure 1(a) shows a two-input multiplier in unipolar coding realized using a two-input AND gate. The input and output probabilities are represented using N_{sto}-bit length streams, where N_{sto} is 10 in this example. N_{sto} clock cycles are required to complete a multiplication of binary computation, where the computation accuracy depends on N_{sto}. Figure 1(b) shows a stochastic multiplier in bipolar coding realized using a two-input XNOR gate. A two-input scaled adder is realized using a two-input multiplexer shown in Fig. 1(c). P_s is a probability of selecting one of two inputs.

In stochastic computing, hyperbolic tangent and exponential functions are simply realized using finite state machines (FSMs), as shown in Figs. 1(d) and (e), respectively. In the FSM-based functions, the states transit to the right, if the input stochastic bits, $x(t)$, are "1" and the states transit to the left, otherwise. The output stochastic bit, $y(t)$, is determined by the current state. The stochastic tanh function, Stanh, in bipolar coding is defined as follows:

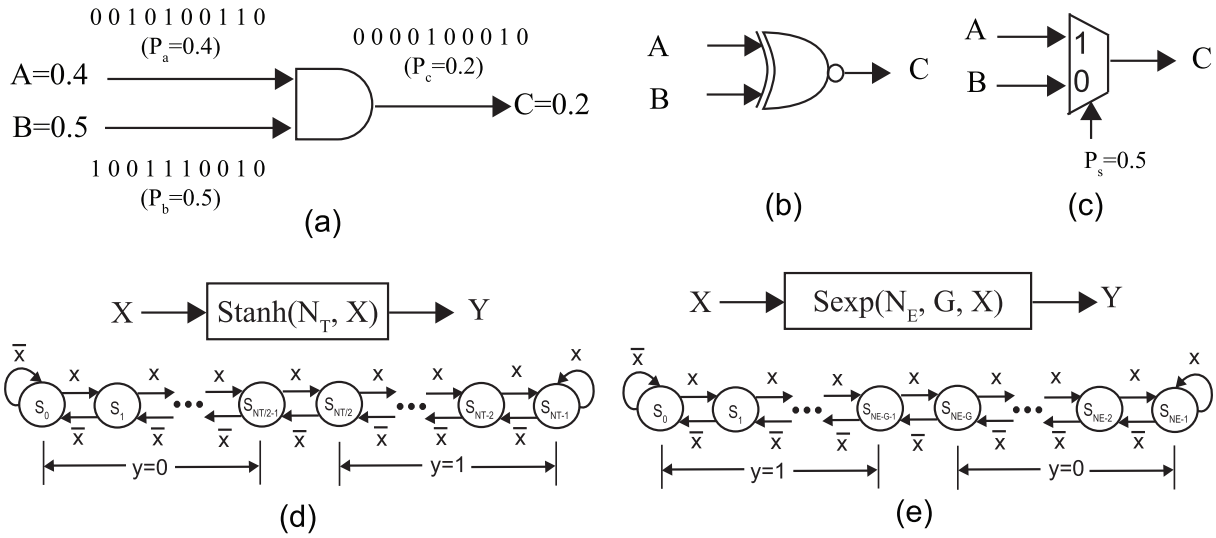

Fig. 1. Stochastic circuit components (a) multiplier (unipolar coding), (b) multiplier (bipolar coding), (c) scaled adder, (d) tanh function, and (e) exponential function.

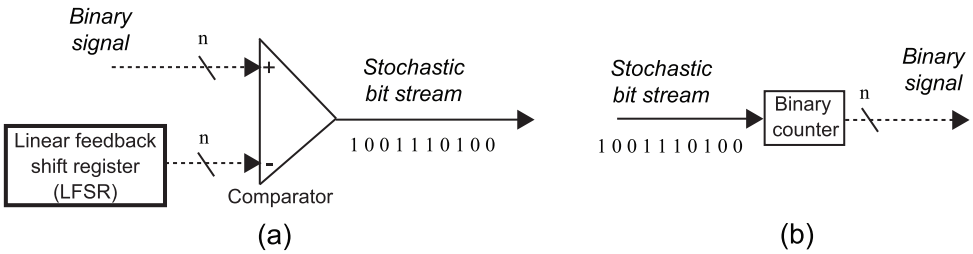

Fig. 2. Signal converter: (a) binary-to-stochastic converter (B2S) and (b) stochastic-to-binary converter (S2B).

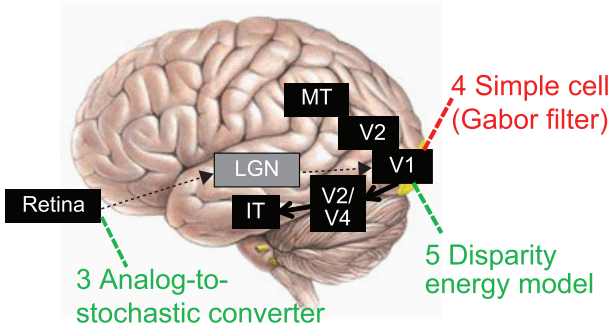

Fig. 3. Brainware LSI (BLSI) design based on stochastic computing that includes analog-to-stochastic converter in Section 3, simple cell model of primary visual cortex (Gabor filter) in Section 4, and disparity energy model in Section 5.

$$\tanh((N_T/2)x) \approx \mathrm{Stanh}(N_T, x), \tag{1}$$

where N_T is the total number of states. The average values of the output bit streams are approximated to the outputs of the tanh function. The stochastic exponential function, Sexp, is defined in unipolar coding as follows:

$$\exp(-2Gx) \approx \mathrm{Sexp}(N_E, G, x), \tag{2}$$

where N_E is the total number of states and G determines the number of states generating outputs of "1".

In order to design stochastic circuits with traditional binary circuits, signal converters are required between stochastic bit streams and binary data. Figure 2(a) shows a binary-to-stochastic converter (B2S) including a digital comparator and a linear-feedback shift register (LFSR) [20]. In B2S, n-bit binary signals are compared with n-bit random signals generated using the LFSR to generate stochastic bit streams. Figure 2(b) shows a stochastic-to-binary converter (S2B) in unipolar coding designed using a binary counter. In S2B, the number of "1" of stochastic bit streams is counted in the counter and the stored values are binary data converted. In bipolar coding, absolute values in the counters need to convert to two's complement values in order to deal with the sign bit.

2.2 Brainware LSI based on stochastic computing

Recently, brain-inspired computing, such as TrueNorth [21] and deep learning [22], has been actively studied for highly accurate recognition and classification capabilities, like human brains. Several hardware implementations of brain-inspired computing have been presented in [23, 24], but the energy efficiencies of the current hardware are significantly lower than that of human brains. Since 2014, in our BLSI project, we exploit stochastic computing to design the energy-efficient brainware hardware based on physiological models of brains. The reason to choose stochastic computing for BLSI is that human brains can work well under severe noises and errors. Although stochastic computing generally causes errors due to randomness, BLSIs based on stochastic computing would work well, like human brains. Actually, a large-scale neuromorphic chip based on stochastic computing has been reported and works well under noises [25].

Out stochastic BLSIs are summarized in Fig. 3. This figure shows flows of visual information in human brains. First, electrical signals (information) from retinas are sent to the primary visual cortex

(V1) through the lateral geniculate nucleus (LGN). Then, in V1, information are extracted and the extracted information are distributed to two pathways: dorsal pathway to the middle temporal (MT) and ventral pathway to the inferior temporal (IT).

In this paper, the hardware of several physiological models designed using stochastic computing are reviewed. First, analog-to-stochastic converters are designed to convert external analog signals to stochastic bit streams [15] in Section 3. Second, a 2D Gabor filter that shows similar responses of simple cells of V1 is designed and fabricated using TSMC 65 nm CMOS technology [16–18] in Section 4. Third, a disparity energy model in V1 is implemented, exhibiting the relative depth estimations using two cameras, like human brains [19] in Section 5. Other than the three topics that are not reviewed in this paper, the BLSI applications of stochastic computing have been studied for deep neural networks [26] and auditory signal processing [27].

3. Analog-to-stochastic converter using magnetic tunnel junction (MTJ)

3.1 Vision chip using analog-to-stochastic converter

In this section, an analog-to-stochastic converter using a magnetic tunnel junction (MTJ) device is explained for massively parallel vision chips [15]. The vision chips are front-end image processors for feature extractions in cognitive computing as shown in Fig. 4(a), where the analog-to-stochastic con-

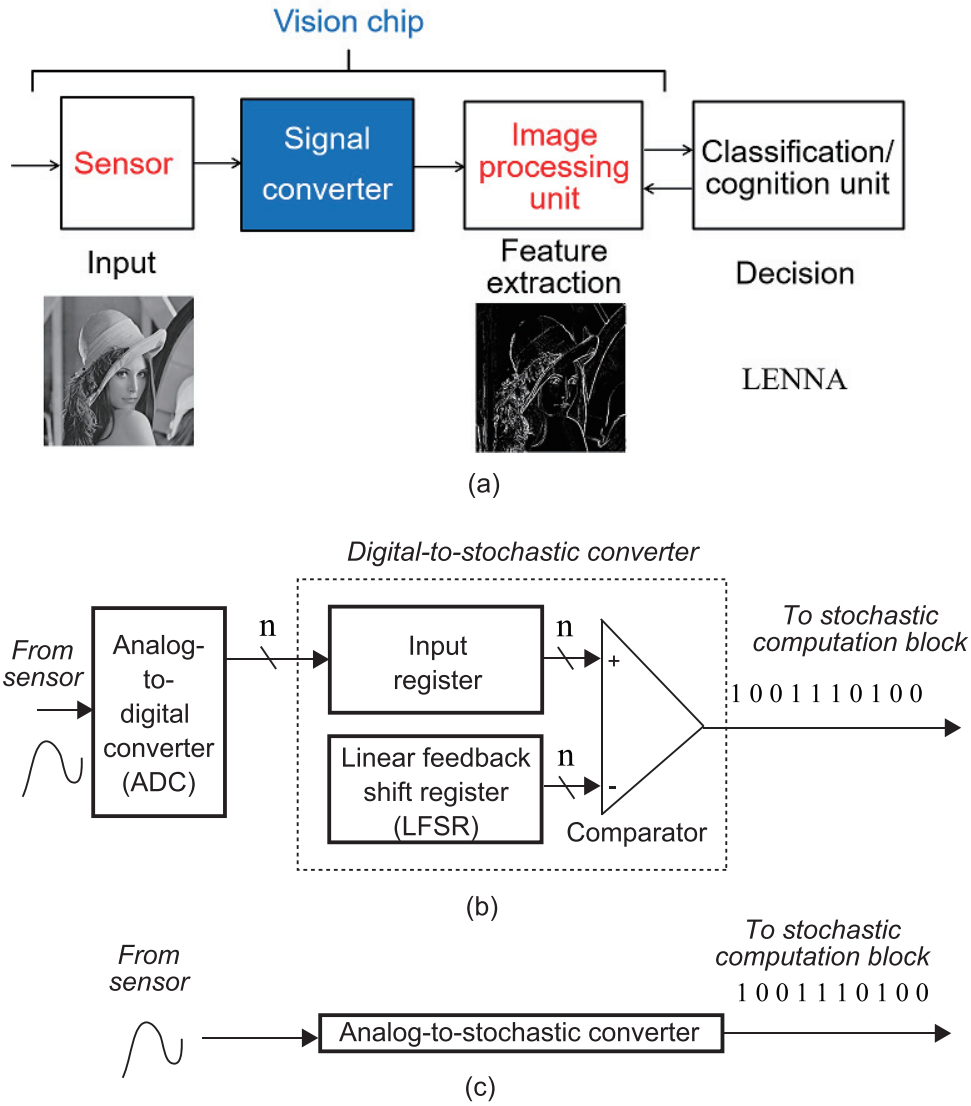

Fig. 4. Example of a vision chip: (a) vision chip using analog-to-stochastic converters, (b) conventional design, and (c) proposed design.

verter is used in the signal-conversion block. The MTJ devices [28] are often exploited as a non-volatile memory that stores one-bit information as a resistance and are often exploited for MRAMs [29]. In addition, as the switching behaviors between the two different resistances of MTJ devices are probabilistic [30, 31], the probabilistic behaviors can be exploited for random number generators [32, 33] and analog-to-stochastic converters.

Figure 4(b) shows a conventional circuit structure of the analog-to-stochastic converter. Using only CMOS transistors, first, an analog-to-digital converter is used to convert from analog to digital signals that are then converted to stochastic bit streams using a digital-to-stochastic converter (binary-to-stochastic converter). In the conventional circuit, the power dissipation of the ADC can be a large portion of the total power dissipation in an image sensor (e.g. 65% in [34]) and the digital-to-stochastic converter tends to be large in the stochastic circuits. To reduce the overhead of the signal conversion block, an analog-to-stochastic converter is designed that the analog signals are directly converted to the stochastic bit streams as shown in Fig. 4(c).

3.2 Circuit design using hybrid MTJ/CMOS devices

Figure 5(a) illustrates the proposed analog-to-stochastic converter using the hybrid MTJ/CMOS devices. It consists of a pulse-signal generator, a random bit generator, a counter, and a probability

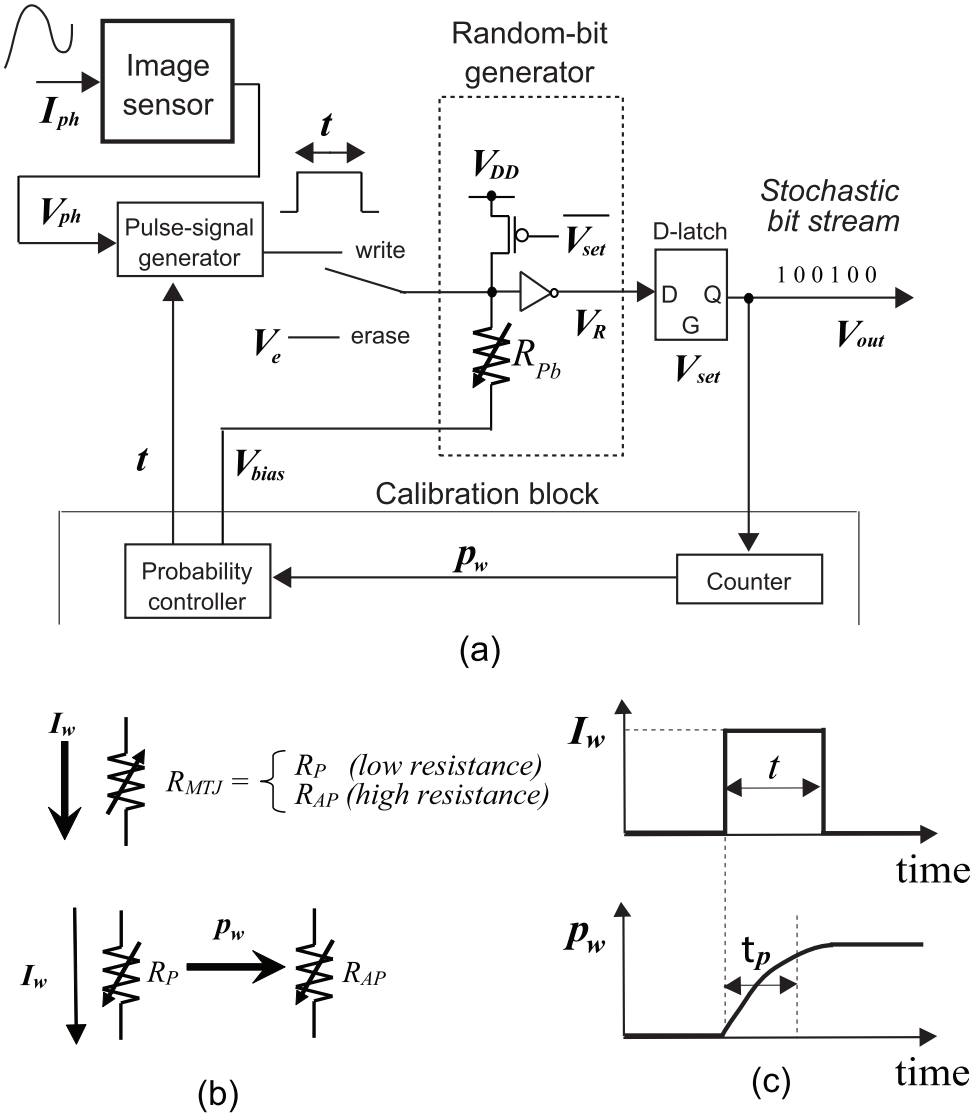

Fig. 5. Analog-to-stochastic converter using CMOS and MTJ devices: (a) circuit diagram, (b) MTJ switching by current and (c) probabilistic switching behavior.

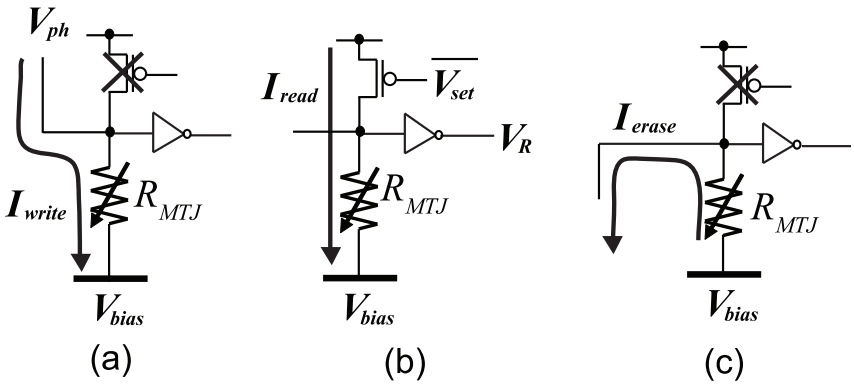

Fig. 6. Circuit operations of the proposed analog-to-stochastic converter at, (a) write phase, (b) set phase, and (c) erase phase.

controller. The two parameters, t (pulse width in time) and V_{bias}, are set in the calibration step before using the converter in order to compensate variabilities of MTJ and CMOS devices. Suppose that an analog current signal is received in a logarithm image sensor that realizes a high dynamic range [35, 36]. The random bit generator is designed using three transistors and one MTJ device.

The switching behavior of the MTJ device between low resistance (R_P (parallel)) and high resistance (R_{AP} (anti parallel)) is probabilistic [30, 31] as illustrated in Figs. 5(b) and (c). Suppose that the initial state of the MTJ device is R_P. When the analog voltage signal, V_{ph}, is generated from the sensor, a write current signal, I_W, is applied during t. In this case, the switching probability of the MTJ device, p_w, is approximated [30, 31] as follows :

$$p_w \approx 1 - \exp(-t/\tau_p), \tag{3}$$

where τ_p is the switching time constant. The detailed switching behavior is described and modelled in the SPICE model [37] used in this paper.

Figure 6 shows the circuit operations of the proposed analog-to-stochastic converter. The converter iteratively operates at one of three phases: write, set, and erase. First, in the write phase, I_{write} is generated to probabilistically switch the MTJ device at a probability depending on V_{ph}. Second, in the set phase, the read current, I_{read}, is generated to read the MTJ resistance, and the output voltage, V_R, is determined as follows:

$$V_R = \begin{cases} \text{High} \ (\text{“1”}) & \text{if } R_{MTJ} = R_P \\ \text{Low} \ (\text{“0”}) & \text{otherwise,} \end{cases} \tag{4}$$

where R_{MTJ} is the resistance of the MTJ device. V_R is stored in the latch next to the converter as shown in Fig. 5(a). Finally, in the erase phase, the erase current, I_{erase}, is generated to switch the resistance back to R_P. After the erase phase, the phase is back to the write phase.

3.3 Simulation results

Figure 7(a) shows simulated waveforms of the proposed analog-to-stochastic converter using NS-SPICE in 90 nm CMOS and the MTJ model [37]. The hybrid 90 nm CMOS and MTJ process is the same as that used in a fabricated chip of [38]. NS-SPICE is a transistor-level simulator that can handle both the transistors and the MTJ models. The cycle time of the converter is set to 10 ns for generating a random bit, where the write phase is 5 ns, and the set phase is 1 ns, and the erase phase is 4 ns. In the write phase, there is a write current, I_{write}, during 4.73 ns and no current during 0.27 ns. I_{write}, is 236 μA corresponding to the switching probability, p_w, of 50% at room temperature. In this simulation, the proposed converter generates three random bits. At the first and the second trials, the resistance of the MTJ device is changed from R_P to R_{AP} in the write phase. Hence, the output of the converter, V_{OUT} is “0”. In contrast, at the third trial, the resistance of the MTJ device is not changed even if I_W is applied to the MTJ device, leading to V_{OUT} of “1”.

Figure 7(b) shows a monte-carlo simulation result of the proposed analog-to-stochastic converter in the write phase. The number of trials is 100 and I_{write} is 236 μA corresponding to p_w of 50%. The

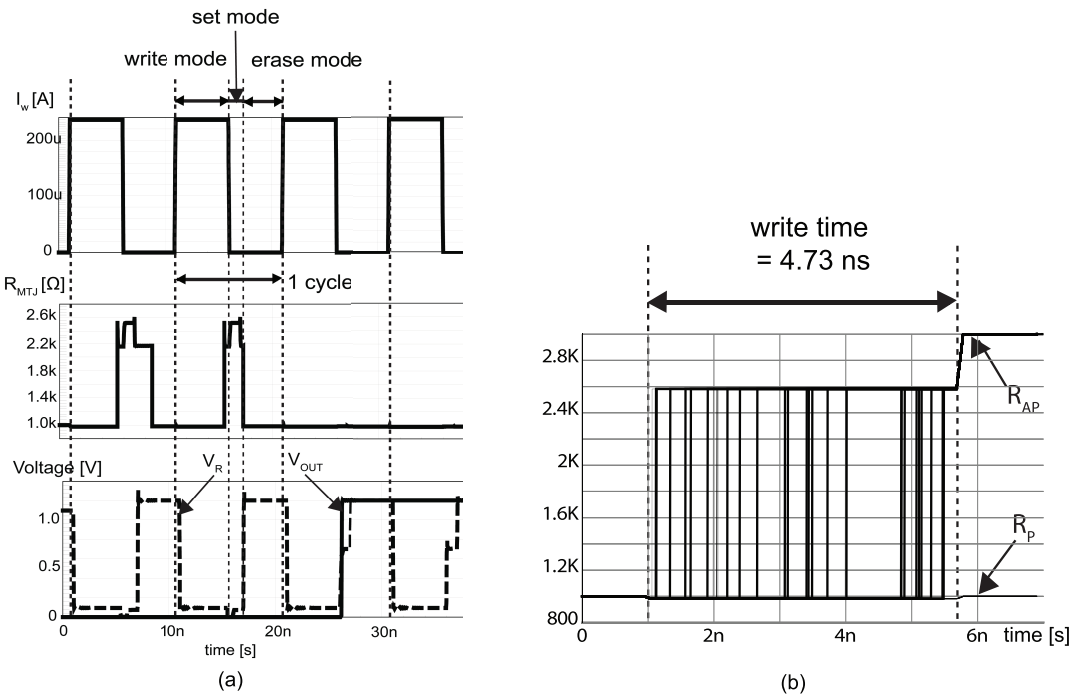

Fig. 7. Simulated waveforms of the analog-to-stochastic converter using 90 nm CMOs and MTJ model in NS-SPICE: (a) circuit behavior with a write current of 236 μA corresponding to the switching probability of 50% and (b) monte-carlo simulation with the number of trials of 100.

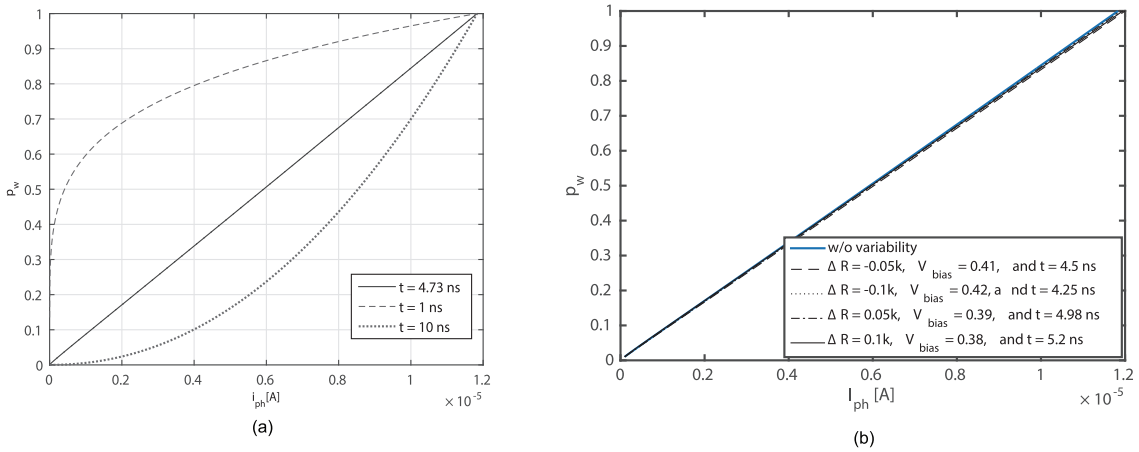

Fig. 8. Relationship between the switching probability and the analog input current, I_{ph} of the analog-to-stochastic converter: (a) the relationship depending on t when V_{bias} is $0.4V$ and I_{write} is 236 μA, and (b) the relationship with MTJ variabilities.

simulation waveforms show that the switching behavior of the MTJ device is probabilistic and the switching timing is random. In this simulation, the resistance of the MTJ device is changed from R_P of 1 kΩ to R_{AP} of 3 kΩ at 50% after writing a bit to the MTJ device.

Figure 8(a) shows a relationship between the switching probability, p_w and the input current, I_{ph}, when V_{bias} is 0.4 V and I_{write} is 236 μA. The attempt time, t, varies from 1 to 10 ns. When t is 4.73 ns, the relationship between p_w and I_{ph} is almost linear, realizing the linear analog-to-stochastic conversion. In addition, the MTJ variabilities are considered as shown in Fig. 8(b). In order to compensate the MTJ variability, two parameters, V_{bias} and t, are set in the calibration step. The resistance variability is defined by ΔR. To control both V_{bias} and t, the relationships between the switching probabilities and the I_{ph} are almost linear under the MTJ variability.

4. Stochastic configurable 2D Gabor-filter chip

4.1 Review of Gabor filter

Gabor filters [39] are powerful feature-extraction tools that extract oriented bars and edges of images. They have been applied for various image processing and computer vision applications, such as face recognition [40] and vehicle verification [41, 42]. The 2D Gabor function (odd phase) is defined as follows:

$$g_{\omega,\sigma,\gamma,\theta}(x,y) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\sin(2\omega x'), \tag{5}$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$. ω represents the spatial angular frequency of the sinusoidal factor. θ represents the orientation of the normal to the parallel stripes of a Gabor function. σ is the sigma and γ is the spatial aspect ratio of the Gaussian envelope.

The 2D Gabor filters exhibit similar responses of simple cells in primary visual cortex (V1) of human brains as shown in Fig. 9. In V1, many different simple cells activated with specific spatial frequencies and angles of images are placed as the hypercolumn structure. Based on the hypercolumn structures, human brains can extract many different features, such as edges and lines of images for object recognitions and classifications in the latter part of brains. HMAX model is known as one of the brain-inspired object recognition models using Gabor filters [43].

4.2 Hardware architecture

Figure 10 shows a hardware architecture of the proposed 64 parallel stochastic configurable 2D Gabor-filter chip. The input image sizes are VGA (640 x 480) with grayscale. As stochastic computation

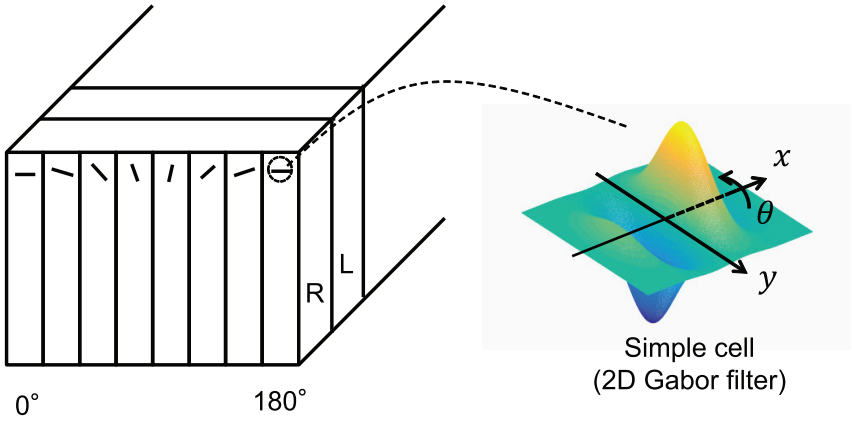

Fig. 9. Hypercolumn structure in V1 including many simple cells, which are activated at specific angles and spatial frequencies. Gabor filters exhibit similar responses of the simple cells.

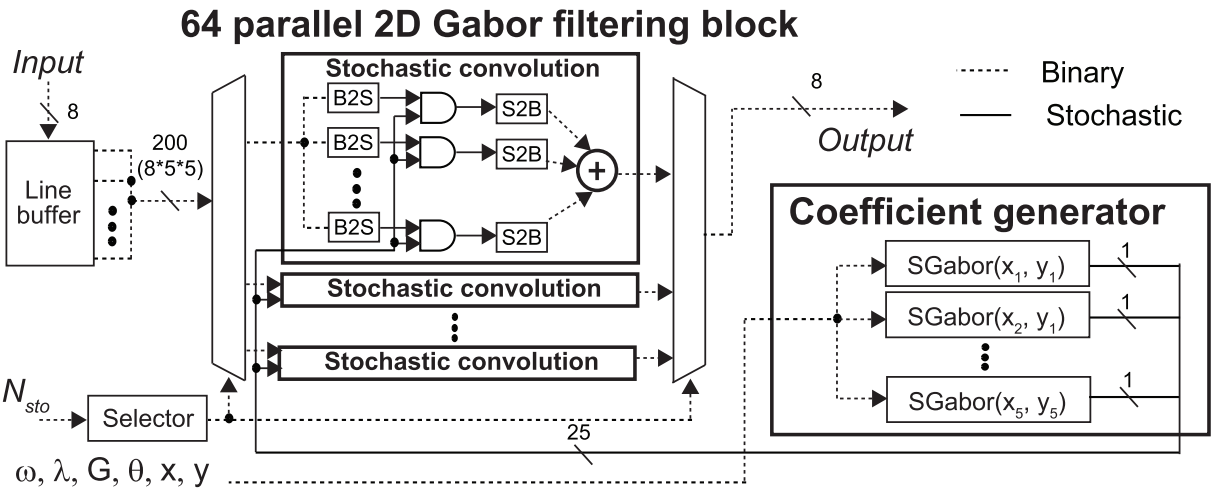

Fig. 10. Architecture of 64-parallel stochastic 2D Gabor-filter chip.

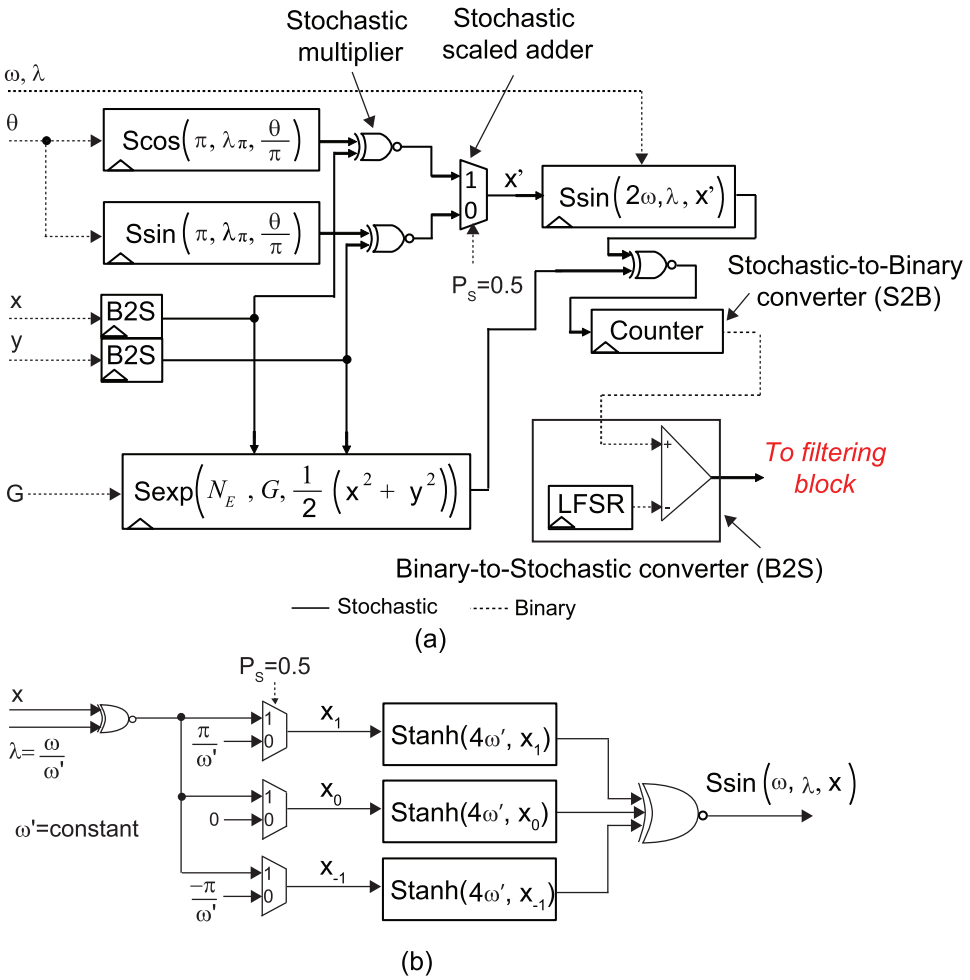

Fig. 11. Block diagram of coefficient generator: (a) SGabor function and (b) Ssin function with $\omega' = \pi$.

takes N_{sto} clock cycles to complete one computation based on traditional binary implementation, the parallel structure is exploited to hide long computation cycles. 8-bit input signals (pixels) from grayscale images are stored in the line buffer and are then transferred to one of the 64 parallel stochastic convolution units. In this chip, there are three cases of N_{sto}: 64, 128, and 256. In the convolution block for Gabor filtering, the multipliers are realized based on stochastic computing and the adders are designed based on traditional binary computation. The hybrid circuit achieves a better computation accuracy than the purely stochastic circuit with an acceptable area overhead [44].

Figure 11(a) shows the block diagram of the stochastic Gabor coefficient generator. The coefficient generator is designed based on the stochastic Gabor (SGabor) function defined as follows:

$$\mathrm{SGabor}(\omega,\gamma,\lambda,G,\theta,x,y) = \frac{\mathrm{Sexp}\Big(N_E, G, \frac{1}{2}(x'^2 + +\gamma^2 y'^2)\Big) + 1}{2}\,\mathrm{Ssin}(\omega,\lambda,x'), \tag{6}$$

$$\begin{aligned} x' &= x\,\mathrm{Scos}(\pi,\lambda_\pi,\theta/\pi) + y\,\mathrm{Ssin}(\pi,\lambda_\pi,\theta/\pi),\\ y' &= -x\,\mathrm{Ssin}(\pi,\lambda_\pi,\theta/\pi) + y\,\mathrm{Scos}(\pi,\lambda_\pi,\theta/\pi), \end{aligned}$$

$$\mathrm{Ssin}(\omega,\lambda,x) = \sum_{k=\lceil -\frac{\omega'}{\pi}\rceil}^{\lfloor \frac{\omega'}{\pi}\rfloor} (-1)^k\,\mathrm{Stanh}\Big(4\omega', \frac{1}{2}\big(\lambda x + \frac{\pi k}{\omega'}\big)\Big), \tag{7}$$

$$\mathrm{Scos}(\omega,\lambda,x) = \sum_{k=\lceil -\frac{\omega'}{\pi}-\frac{1}{2}\rceil}^{\lfloor \frac{\omega'}{\pi}-\frac{1}{2}\rfloor} (-1)^k\,\mathrm{Stanh}\Big(4\omega', \frac{1}{2}\big(\lambda x + \frac{\pi(k+\frac{1}{2})}{\omega'}\big)\Big), \tag{8}$$

where ω' is a constant angular frequency and λ is ω/ω' and λ_π is constant. The original Gabor function on Eq. (5) is approximated as follows:

$$\alpha g_{\omega,\sigma,\gamma,\theta}(x,y) \approx \mathrm{SGabor}(\omega,\gamma,\lambda,G,\theta,x,y), \tag{9}$$

where α is a constant value for fitting SGabor with the original Gabor function. The stochastic sin function, Ssin, is designed using five Stanh functions based on [16] as shown in Fig. 11(b). Figure 11(b) shows the example with $\omega' = \pi$. ω required is controlled by λ. The stochastic cos function, Scos, is designed as well as Ssin.

4.3 Simulation and measurement results

Figure 12 shows simulated Gabor functions using SGabor for a kernel size of 51x51 with different configurations using MATLAB. The length (cycle) of stochastic bit streams for SGabor is defined as N_{sto}. In this simulation, ω and θ are changed with $N_{sto} = 2^{18}$. Using SGabor, any ω and θ can be configured depending on requirements.

Figure 13 shows the test environment of of the proposed stochastic 2D configurable Gabor-filter chip using TSMC 65 nm CMOS process. The proposed circuit is designed using Verilog HDL and the chip layout is obtained using Synopsys Design Compiler and Cadence SoC Encounter. The supply voltage is 1.0 V and the area is 1.79 mm × 1.79 mm. The fabricated chip is tested with an FPGA (Digilent Genesys 2) board. Images are captured by a camera (VGA) and the input pixels in grayscale are transferred to the chip through the FPGA. The output pixels of the test chip are sent back to the FPGA and are displayed using the FPGA.

Table I shows performance comparisons of the proposed stochastic Gabor filter with related works. It is hard to directly compare the performance because they are designed with different functionalities and configurations. The memory-based methods [45, 46] use fixed coefficients with fixed kernel sizes that are calculated in software in advance, causing the lack of flexibility. As opposed to the memory-based circuits, in the conventional configurable Gabor filter [47], CORDIC is exploited to dynamically generate the coefficients related to sinusoidal function for flexible Gabor filtering. However, this

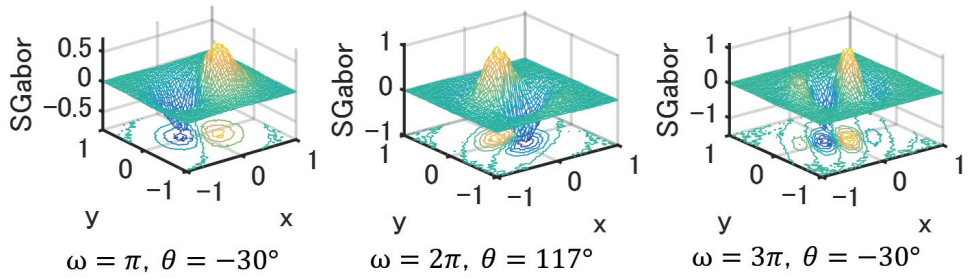

Fig. 12. Simulated SGabor function with different configurations.

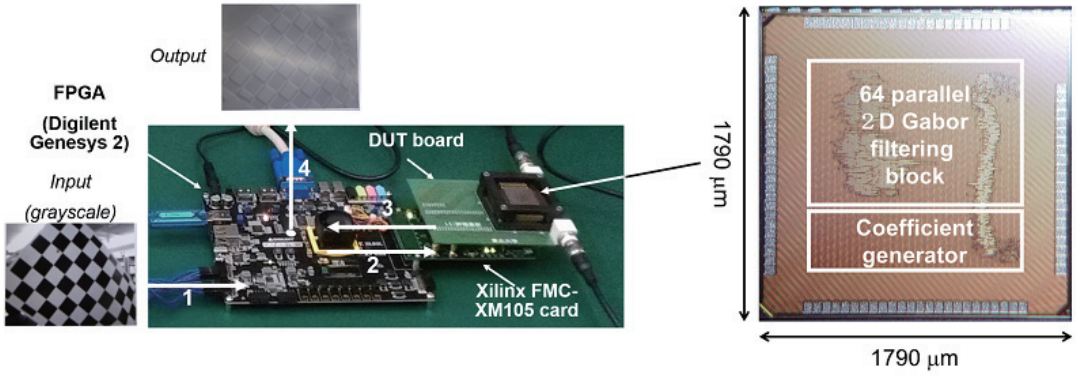

Fig. 13. Test environment of the fabricated chip of the 64-parallel stochastic 2D Gabor filter using TSMC 65 nm CMOS.

Table I. Performance comparisons of Gabor filters.

	[45]	[46]	[47]	This work
Computation	Analog/Digital	Digital	Digital	Stochastic
Technology	0.35 μm CMOS	(FPGA)	0.13 μm CMOS	65 nm CMOS
Kernel size	(Only nearest neighbor)	3x3	3x3, 5x5, 7x7, 9x9, 11x11	**NxN (flexible)**
Kernel parameter	Fixed	Fixed	**Flexible**	**Flexible**
Power-gating capability	No	No	No	**Yes**
# of processing elements	61x72	1	1	64
Throughput (MP/s)	-	124.4 (3x3)	10 (5x5) 2.1 (11x11)	**200 (5x5)** **40 (11x11)**
Frequency [MHz]	1	148.5	250	200
Power dissipation [mW]	800	-	-	102.3

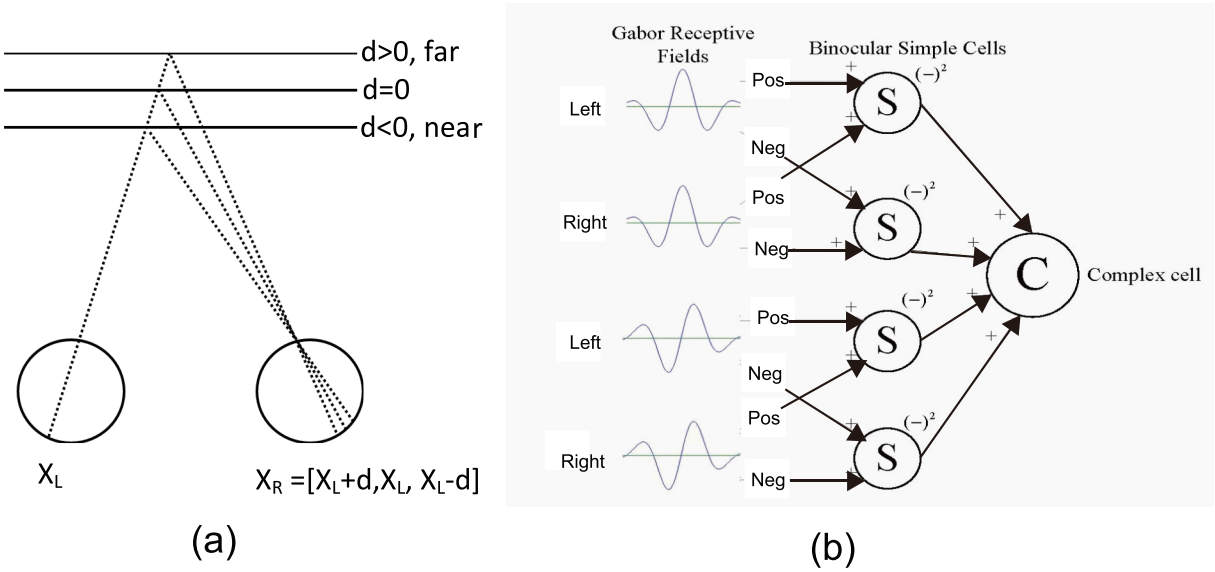

Fig. 14. Estimation of the relative depths of objects: (a) relation between disparity and depth, and (b) disparity energy model using simple cells (S) and complex cells (C).

method is low throughput due to the hardware complexity and several parameters need to be stored in memory, losing the power-gating capability. In contrast, the proposed memory-less circuit achieves an order-of-magnitude higher throughput than the conventional configurable Gabor filter with the power-gating capability, leading to zero standby power.

5. Stochastic disparity energy model

5.1 Review of disparity energy model

Measuring the relative depth of objects efficiently in real-time is a crucial issue as advances in robotics. A disparity-energy model was presented to express the disparity-selective properties of binocular complex cells in V1 that are responsible for depth perception in brains [48]. In the disparity-energy model, binocular disparity measures the depth of objects using two images taken from different vantage points, and is defined as the difference in horizontal positioning of the same object in these two images. This model was used to be valid in monkeys [49] and to describe well the response of binocular complex cells in V1 [50].

When an object is perceived from the left and right eyes, its position is horizontally displaced in each of the corresponding images, as illustrated in Fig. 14(a). The brains use this horizontal disparity, d, to estimate the relative depths of objects in three dimensions. Positive and negative disparities (corresponding to farther and closer objects) consequently excite different retinal cells in each eye.

Zero disparity corresponds to those objects whose positions are the same from both perspectives and excites corresponding retinal cells in each eye.

Figure 14(b) shows the disparity-energy model that shows how the neural hierarchy in the brain processes this information to detect disparity [48, 51, 52]. The simple cells are approximated using Gabor filters explained in the prevision section. The complex cells C^d then take the even and odd binocular cell responses and squares and adds them:

$$\begin{aligned} C^d(x_L, x_R) = &\left(G_{\text{even}+}\left(x_L + \frac{d}{2}\right) + G_{\text{even}+}\left(x_R - \frac{d}{2}\right)\right)^2 \\ &+ \left(G_{\text{even}-}\left(x_L + \frac{d}{2}\right) + G_{\text{even}-}\left(x_R - \frac{d}{2}\right)\right)^2 \\ &+ \left(G_{\text{odd}+}\left(x_L + \frac{d}{2}\right) + G_{\text{odd}+}\left(x_R - \frac{d}{2}\right)\right)^2 \\ &+ \left(G_{\text{odd}-}\left(x_L + \frac{d}{2}\right) + G_{\text{odd}-}\left(x_R - \frac{d}{2}\right)\right)^2 \end{aligned} \tag{10}$$

x_L and x_R are the horizontal pixel positions for the left and right eye, respectively. $G_{even+}(x) = G_{even}(x)$ if $x > 0$, and 0 otherwise. $G_{even-}(x) = G_{even}(x)$ if $x \leq 0$, and 0 otherwise. There are two ways of encoding disparity in the model: position shift and phase shift [53]. In this paper, we only use position shift, where d is defined by the difference in position of the receptive field.

5.2 Stochastic convolution architecture

Key circuit components for designing the disparity energy model are convolution units used in Gabor filtering. The convolution is defined as follows:

$$z = \sum_i a_i x_i, \tag{11}$$

where a_i is the coefficients and x_i is the system inputs. Figure 15(a) shows a conventional stochastic architecture of convolution units. It consists of AND gates (stochastic multiplier) and a multiplier

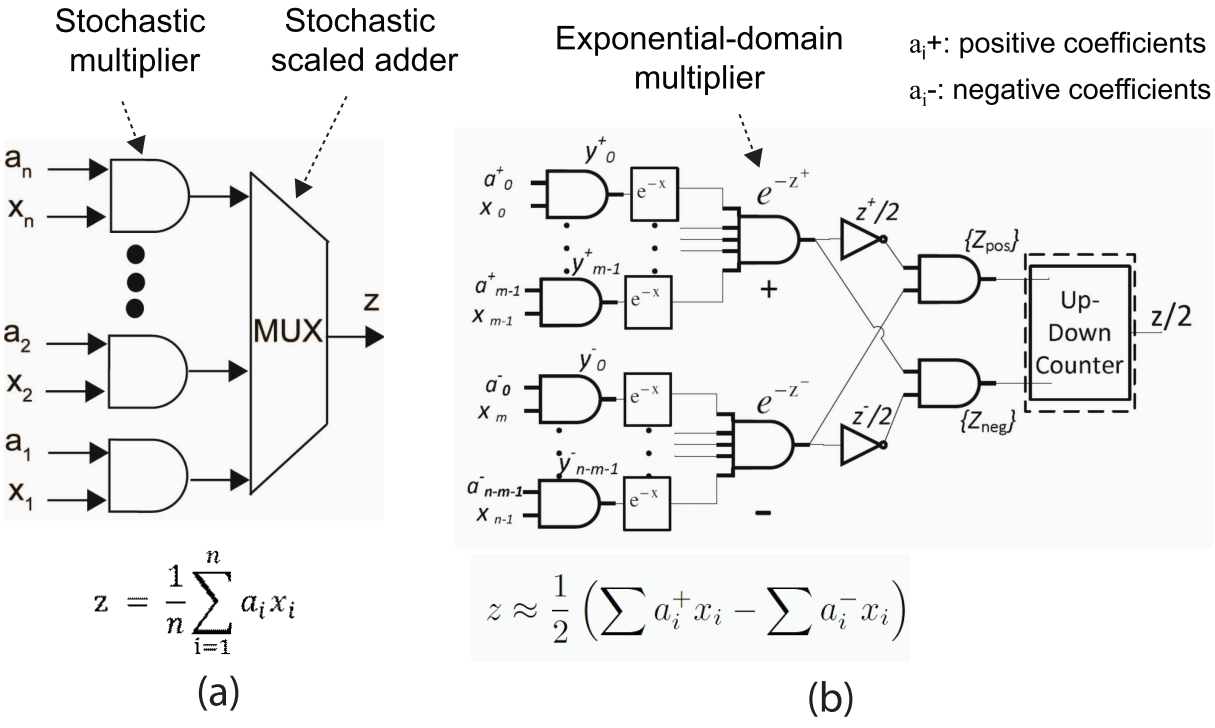

Fig. 15. Stochastic convolution circuits: (a) conventional and (b) proposed.

(stochastic scaled adder). The drawback of this circuit is that the computation accuracy is significantly lower when the number of inputs, n, is increased. In the Gabor filters of the disparity energy model, 7×7 kernel sizes are used to extract features. In this case, n is 49, causing a low computation accuracy.

To achieve a high computation accuracy with a large number of inputs, the exponential based convolution circuit was presented as shown in Fig. 15(b). In the proposed circuit, the exponential compression method transforms the stochastic streams of interest using an exponential function, such that additions become multiplications [54]. The $\exp(x)$ and $\ln(x)$ functions are approximated using Taylor series expansions. Suppose that a_i and x_i have been properly scaled such that $|x_i| \leq 1$ and $|a_i| \leq 1$. The set a_i of coefficients is partitioned into a set a_i+ containing the positive coefficients, and a set a_i- containing the absolute values of the negative coefficients.

5.3 Experimental results of disparity energy model

To detect the depths of objects, an experiment is setup that is similar to [50] as shown in Fig. 16. The two cameras are setup 19-cm apart, where 8-degree angle from the vertical is realized. The fixation point that is the point at the intersection of the line of sight of each camera is 66 cm away. At this range, disparities correspond to around 3 cm per pixel. One white pole is placed on the fixation point. To detect disparities of -8 and $+8$, two white poles are also placed at a distance of 42 and 90 cm, respectively, from the cameras center.

Figure 17 shows the disparity maps for the floating-point, the conventional stochastic and the proposed stochastic circuits. The lengths of stochastic bit streams are $2^6 - 1$ corresponding to a

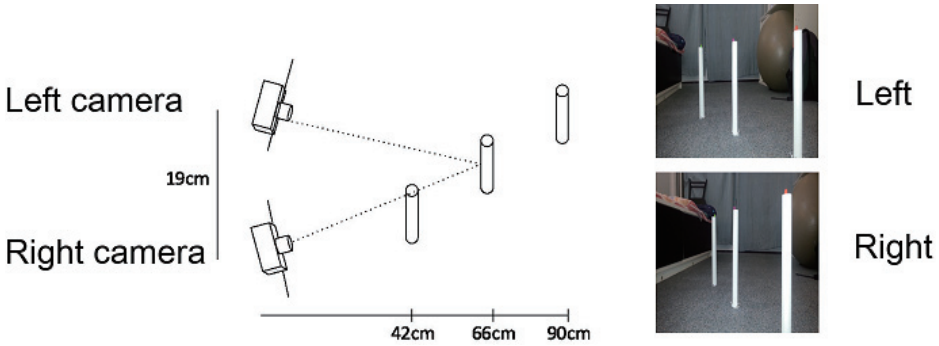

Fig. 16. Experimental setup to detect disparity.

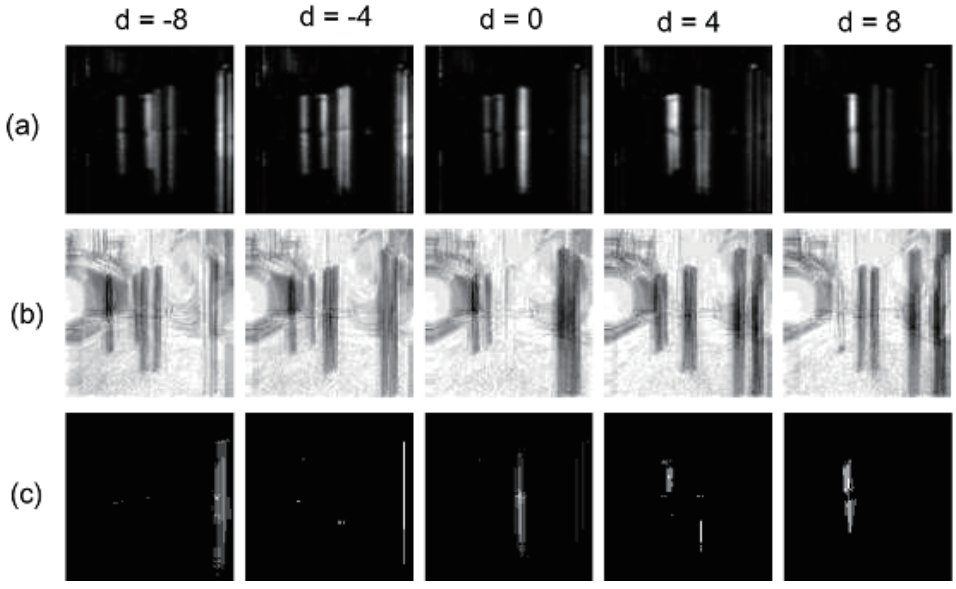

Fig. 17. Disparity maps using: (a) floating point, (b) conventional stochastic circuit, and (c) proposed stochastic circuit.

Table II. Performance comparisons of disparity-energy-model hardware using TSMC 65 nm CMOS process.

	Fixed-point (6-bit)	Stochastic	
		(w/o interface)	(w/ interface)
Area [mm^2]	10.22	0.308	0.639
Delay [ns]	592	5,448	5,448
Area × delay product (ADP)	5,886	1,679	3,453
Dynamic power [mW]	776.3	43.4	152.2
Static power [mW]	41.7	0.8	1.8
Energy [nJ]	484	241	839
Average error	16.4	14.2	
(min, max)	-	(8,9, 22)	

6-bit fixed-point precision. To quantify the errors, we obtain 4 additional image pairs with poles at different disparities using a similar setup and manually create ideal disparity maps depending on the position and dimensions of the poles from the left and right images to estimate the error. Using the conventional stochastic circuit, the disparities are not obtained, unlike the floating-point result. The reason is that the computation accuracy of the stochastic convolution unit shown in Fig. 15(a). is significantly lower than the floating point. In contrast, using the proposed stochastic circuit, the similar disparities to the floating-point results is obtained because of the high computation accuracy of the exponential based convolution unit.

5.4 Hardware evaluation

Table II summarizes the performance of disparity-energy-model hardware using TSMC 65 nm CMOS technology. For both fixed-point and stochastic circuits, a 2D 1×100 architecture is synthesized using Cadence RC compiler. The worst-case delay is 5.5 ns and 1.7 ns in the fixed-point and the stochastic circuits, respectively. In the fixed-point design, the interface circuitry includes the input and output registers. In the stochastic design, it includes input registers, linear feedback shift registers (LFSRs) for random number generation, comparators and counters to convert from digital to stochastic domain and back.

To provide a fair comparison, we use the area × delay product (ADP) measure to normalize for latency of the stochastic system. Note that such a stream length allows outperforming the floating-point system even when the performance is averaged over the seed configurations. The stochastic circuit with the interface circuitry achieves a 41.3% reduction in ADP in comparison with the fixed-point circuit.

The dynamic and the static power dissipations of the stochastic design are significantly smaller than that of the fixed-point design because of the small area. However, the energy dissipation with the interface is 73% larger than the fixed-point design. The reason is the stochastic circuits take 2^6-1 cycles for a one-cycle operation of the fixed-point design. The energy overhead also comes from the interface that includes binary-to-stochastic and stochastic-to-binary converters. The overhead can be mitigated using MTJ-based converters explained in Section 3.

The average error of the stochastic design is slightly smaller than that of the fixed-point design. As the stochastic circuits exhibit the variability of computation accuracy depending on random bit streams, the minimum and the maximum computation accuracies are also listed.

6. Conclusion

In this paper, we have reviewed the applications of stochastic computing in brainware for visual signal processing. The two physiological models in V1 of the human brains have been implemented in TSMC 65 nm CMOS process. The hardware performance is compared and discussed with that of the fixed-point design with the computation accuracy. In addition, the area-efficient analog-to-stochastic converter has been designed in order to mitigate the signal-conversion overhead to the stochastic bit

streams from external analog signals.

Future prospect includes the application of stochastic computing for models of higher order visual cortex, such as visual attention models.

Acknowledgments

This work was supported by Brainware LSI Project of MEXT and JSPS KAKENHI Grant Number JP16K12494. This work is supported by VLSI Design and Education Center (VDEC), The University of Tokyo with the collaboration with Synopsys Corporation and Cadence Corporation.

References

[1] B.D. Brown and H.C. Card, "Stochastic neural computation. I. computational elements," *IEEE Transactions on Computers*, vol. 50, no. 9, pp. 891–905, September 2001.

[2] J. von Neumann, " Probabilistic logics and the synthesis of reliable organisms from unreliable components," *lectures delivered at the California Institute of Technology*, January 1952.

[3] B.R. Gaines, "Stochastic computing systems," *Adv. Inf. Syst. Sci. Plenum*, vol. 2, no. 2, pp. 37–172, 1969.

[4] V.C. Gaudet and A.C. Rapley, "Iterative decoding using stochastic computation," *Electronics Letters*, vol. 39, no. 3, pp. 299–301, February 2003.

[5] S. Sharifi Tehrani, W.J. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *IEEE Communications Letters*, vol. 10, no. 10, pp. 716 –718, October 2006.

[6] S. Sharifi Tehrani, S. Mannor, and W.J. Gross, "Fully parallel stochastic LDPC decoders," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5692 –5703, November 2008.

[7] S. Sharifi Tehrani, A. Naderi, G.-A. Kamendje, S. Hemati, S. Mannor, and W.J. Gross, "Majority-based tracking forecast memories for stochastic LDPC decoding," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4883–4896, September 2010.

[8] P. Li and D.J. Lilja, "Using stochastic computing to implement digital image processing algorithms," In *29th IEEE International Conference on Computer Design (ICCD)*, pp.154–161, October 2011.

[9] P. Li, D.J. Lilja, W. Qian, K. Bazargan, and M.D. Riedel, "Computation on stochastic bit streams digital image processing case studies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 3, pp. 449–462, March 2014.

[10] A. Alaghi, C. Li, and J.P. Hayes, "Stochastic circuits for real-time image-processing applications," In *50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, May 2013.

[11] K. Parhi and L. Yin, "Architectures for IIR digital filters using stochastic computing," In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 373–376, June 2014.

[12] N. Saraf, K. Bazargan, D.J. Lilja, and M.D. Riedel, "IIR filters using stochastic arithmetic," In *2014 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1–6, March 2014.

[13] Y. Liu and K.K. Parhi, "Architectures for recursive digital filters using stochastic computing," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3705–3718, July 2016.

[14] J. Chen, J. Hu, and J. Zhou, "Hardware and energy-efficient stochastic LU decomposition scheme for MIMO receivers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 4, pp. 1391–1401, April 2016.

[15] N. Onizawa, D. Katagiri, W.J. Gross, and T. Hanyu, "Analog-to-stochastic converter using magnetic tunnel junction devices for vision chips," *IEEE Transactions on Nanotechnology*, vol. 15, no. 5, pp. 705–714, 2016.

[16] N. Onizawa, D. Katagiri, K. Matsumiya, W.J. Gross, and T. Hanyu, "Gabor filter based on stochastic computation," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1224–1228, September 2015.

[17] N. Onizawa, K. Matsumiya, W.J. Gross, and T. Hanyu, "Accuracy/energy-flexible stochastic configurable 2D Gabor filter with instant-on capability, " In *43rd IEEE European Solid State*

Circuits Conference (ESSCIRC), pp. 43–46, September 2017.

[18] N. Onizawa, D. Katagiri, K. Matsumiya, W.J. Gross, and T. Hanyu, "An accuracy/energy-flexible configurable Gabor-filter chip based on stochastic computation with dynamic voltage-frequency-length scaling," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS)*, 2018 (to appear).

[19] K. Boga, F. Leduc-Primeau, N. Onizawa, K. Matsumiya, T. Hanyu, and W.J. Gross, "A generalized stochastic implementation of the disparity energy model for depth perception," *Journal of Signal Processing Systems*, vol. 90, no. 5, pp. 709–725, May 2018.

[20] P. Alfke, " Efficient shift registers, lfsr counters, and long pseudo-random sequence generators," http://www.xilinx.com/bvdocs/appnotes/xapp052.pdf, 1998.

[21] P.A. Merolla et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, August 2014.

[22] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Science*, vol. 18, no. 7, pp. 1527–1554, July 1996.

[23] S. Park, K. Bong, D. Shin, J. Lee, S. Choi, and H.-J. Yoo, "1.93TOPS/W scalable deep learning/inference processor with tetra-parallel mimd architecture for big-data applications," In *2015 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 1–3, February 2015.

[24] S. Park, I. Hong, J. Park, and H.J. Yoo, "An energy-efficient embedded deep neural network processor for high speed visual attention in mobile vision recognition soc," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 10, pp. 2380–2388, October 2016.

[25] S. Sato, K. Nemoto, S. Akimoto, M. Kinjo, and K. Nakajima. "Implementation of a new neurochip using stochastic logic," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1122–1127, September 2003.

[26] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W.J. Gross, "VLSI implementation of deep neural network using integral stochastic computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2588–2599, October 2017.

[27] N. Onizawa, S. Koshita, S. Sakamoto, M. Abe, M. Kawamata, and T. Hanyu, "Area/energy-efficient gammatone filters based on stochastic computation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2724–2735, October 2017.

[28] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H.D. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukura, and H. Ohno, "A perpendicular-anisotropy CoFeB-MgO magnetic tunnel junction," *Nature Materials*, vol. 9, pp. 721–724, 2010.

[29] T. Kawahara, R. Takemura, K. Miura, J. Hayakawa, S. Ikeda, Y.M. Lee, R. Sasaki, Y. Goto, K. Ito, T. Meguro, F. Matsukura, H. Takahashi, H. Matsuoka, and H. Ohno, "2 Mb SPRAM (spin-transfer torque RAM) with bit-by-bit bi-directional current write and parallelizing-direction current read," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 109–120, January 2008.

[30] N.D. Rizzo, M. DeHerrera, J. Janesky, B. Engel, J. Slaughter, and S. Tehrani, "Thermally activated magnetization reversal in submicron magnetic tunnel junctions for magnetoresistive random access memory," *Applid Physics Letters*, vol. 80, no. 13, 2002.

[31] K. Yagami, A.A. Tulapurkar, A. Fukushima, and Y. Suzuki, "Inspection of intrinsic critical currents for spin-transfer magnetization switching," *IEEE Transactions on Magnetics*, vol. 41, no. 10, pp. 2615–2617, October 2005.

[32] A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasa, and K. Ando, "Spin dice: A scalable truly random number generator based on spintronics," *Applied Physics Express*, vol. 7, no. 8, p. 083001, 2014.

[33] S. Oosawa, T. Konishi, N. Onizawa, and T. Hanyu, "Design of an STT-MTJ based true random number generator using digitally controlled probability-locked loop," In *13th IEEE New Circuits and Systems (NEWCAS)*, pp. 1–4, June 2015.

[34] Y. Oike and A. El Gamal, "CMOS image sensor with per-column $\Sigma\Delta$ ADC and programmable compressed sensing," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 318–328, January 2013.

[35] D. Stoppa, A. Simoni, L. Gonzo, M. Gottardi, and G.-F. Dalla Betta, "Novel CMOS image

sensor with a 132-dB dynamic range," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1846–1852, December 2002.

[36] W.-F. Chou, S.-F. Yeh, C.-F. Chiu, and C.-C. Hsieh, "A linear-logarithmic CMOS image sensor with pixel-FPN reduction and tunable response curve," *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1625–1632, May 2014.

[37] N. Sakimura, R. Nebashi, Y. Tsuji, H. Honjo, T. Sugibayashi, H. Koike, T. Ohsawa, S. Fukami, T. Hanyu, H. Ohno, and T. Endoh, "High-speed simulator including accurate mtj models for spintronics integrated circuit design," In *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1971–1974, May 2012.

[38] H. Jarollahi, N. Onizawa, V. Gripon, N. Sakimura, T. Sugibayashi, T. Endoh, H. Ohno, T. Hanyu, and W.J. Gross, "A nonvolatile associative memory-based context-driven search engine using 90 nm CMOS/MTJ-hybrid logic-in-memory architecture," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 4, pp. 460–474, December 2014.

[39] D. Gabor, "Theory of communications," *Journal of Inst. Elect. Eng. - Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, November 1946.

[40] J. Wu, G. An, and Q. Ruan, "Independent Gabor analysis of discriminant features fusion for face recognition," *IEEE Signal Processing Letters*, vol. 16, no. 2, pp. 97–100, February 2009.

[41] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, " A real-time precrash vehicle detection system," In *Proc. Sixth IEEE Workshop on Applications of Computer Vision, 2002 (WACV 2002)*, pp. 171–176, 2002.

[42] J.-M. Guo, H. Prasetyo, and K. Wong, "Vehicle verification using Gabor filter magnitude with gamma distribution modeling," *IEEE Signal Processing Letters*, vol. 21, no. 5, pp. 600–604, May 2014.

[43] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, pp. 1019–1025, 1999.

[44] S. Koshita, N. Onizawa, M. Abe, T. Hanyu, and M. Kawamata, "High-accuracy and area-efficient stochastic fir digital filters based on hybrid computation," *IEICE Trans. on Inf. and Syst.*, vol. E100-D, no. 8, pp. 592–1602, August 2017.

[45] T. Morie, J. Umezawa, and A. Iwata, "A pixel-parallel image processor for Gabor filtering based on merged analog/digital architecture," In *Digest of Technical Papers in 2004 Symposium on VLSI Circuits*, pp. 212–213, June 2004.

[46] E. Cesur, N. Yildiz, and V. Tavsanoglu, "On an improved FPGA implementation of CNN-based Gabor-type filters," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 11, pp. 815–819, November 2012.

[47] J.-B. Liu, S. Wang, Y. Li, J. Han, and X.-Y. Zeng, "Configurable pipelined Gabor filter implementation for fingerprint image enhancement," In *2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pp. 584–586, November 2010.

[48] I. Ohzawa, G.C. Deangelis, and R.D. Freeman, "Stereoscopic depth discrimination in the visual cortex: neurons ideally suited as disparity detectors," *Science*, vol. 249, pp. 1037–1041, 1990.

[49] B.G. Cumming and A.J. Parker, "Responses of primary visual cortical neurons to binocular disparity without depth perception," *Nature*, vol. 389, pp. 280–283, 1997.

[50] K. Shimonomura, T. Kushima, and T. Yagi, "Binocular robot vision emulating disparity computation in the primary visual cortex," *Neural Networks*, vol. 21, no. 2, pp. 331–340, 2008.

[51] R.D. Freeman and I. Ohzawa, "On the neurophysiological organization of binocular vision," *Vision Res.*, vol. 30, pp.1661–1676, 1990.

[52] G.C. DeAngelis, I. Ohzawa, and R.D. Freeman, "Depth is encoded in the visual cortex by a specialized receptive field structure," *Nature*, vol. 352, no. 11, pp. 156–159, July 1991.

[53] N. Qian, "Binocular disparity and the perception of depth," *Neuron*, vol. 18, no. 3, pp. 359–368, 1997.

[54] C.L. Janer, J.M. Quero, J.G. Ortega, and L.G. Franquelo, "Fully parallel stochastic computation architecture," *IEEE Transactions on Signal Processing*, vol. 44, no. 8, pp. 2110–2117, August 1996.