An Accuracy/Energy-Flexible Configurable Gabor-Filter Chip Based on Stochastic Computation With Dynamic Voltage–Frequency–Length Scaling

Naoya Onizawa^(D), *Member, IEEE*, Daisaku Katagiri, Kazumichi Matsumiya, Warren J. Gross, *Senior Member, IEEE*, and Takahiro Hanyu^(D), *Senior Member, IEEE*

Abstract—This paper introduces an accuracy/energy-flexible configurable 2-D Gabor filter based on stochastic computation, where stochastic bit stream representing information is used. The Gabor filters show a powerful feature extraction capability, but the calculation based on binary computation is complicated. As opposed to traditional memory-based methods that use fixed Gabor coefficients calculated by software in advance, the proposed circuit dynamically generates the coefficients with small hardware, leading to the power-gating capability. For energy-efficient circuits, dynamic voltage-frequency-length scaling (DVFLS) is proposed to match the performance demands depending on situations. DVFLS controls the lengths of the stochastic bit streams with voltage and frequency, which can lower the energy dissipation and/or increase the throughput with a little accuracy loss. The proposed 64 parallel stochastic Gaborfilter chip is fabricated using TSMC 65-nm CMOS technology with a size of 1.79 mm × 1.79 mm. The measurement result shows $4 \times$ higher throughput and $4 \times$ lower energy than that using a conventional DVFS technique with a 0.391% accuracy loss. Compared with a conventional configurable Gabor filter, the proposed chip achieves a higher throughput/area with more flexibility of the Gabor coefficients.

Index Terms—Stochastic logic, digital circuit implementation, power gating, image processing, image classification.

I. INTRODUCTION

G ABOR filters [1] are powerful feature-extraction tools that extract oriented bars and edges of an image. They have been used for various image processing and computer vision applications, such as face recognition [2] and vehicle

Manuscript received December 11, 2017; revised May 7, 2018; accepted June 1, 2018. Date of publication June 6, 2018; date of current version September 11, 2018. This work was supported in part by the MEXT Brainware LSI Project, Japan, in part by JSPS KAKENHI under Grant JP16K12494, and in part by the VLSI Design and Education Center, The University of Tokyo in collaboration with Synopsys Corporation and Cadence Corporation. This paper was recommended by Guest Editor A. Marongiu. (*Corresponding author: Naoya Onizawa.*)

N. Onizawa is with the Frontier Research Institute for Interdisciplinary Sciences, Tohoku University, Sendai 980-8578, Japan (e-mail: nonizawa@m.tohoku.ac.jp).

D. Katagiri, K. Matsumiya, and T. Hanyu are with the Research Institute of Electrical Communication, Tohoku University, Sendai 980-8577, Japan (e-mail: kmat@riec.tohoku.ac.jp; hanyu@riec.tohoku.ac.jp).

W. J. Gross is with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0E9, Canada (e-mail: warren.gross@mcgill.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2018.2844329

verification [3], [4]. It is also known that, the filtering process is similar to that in the primary visual cortex (V1) of brains [5].

The hardware implementation of the Gabor filters is challenging, due to the computation complexity, where the Gabor function is defined by a multiplication of sin/cos function by a Gaussian function. Several hardware using digital and analog circuit techniques have been presented [6]–[11]. Most of them use memory-based techniques that store Gabor coefficients calculated by software in advance in order to mitigate the computation complexity. However, they lose the flexibility of the Gabor function and the power-gating capability as the fixed coefficients with the fixed kernel size are stored in memory. In [7], the coefficients are dynamically generated using CORDIC (COordinate Rotation DIgital Computer) for configurable Gabor filters with sacrificing the throughput.

In order to achieve high throughput with the flexibility, a configurable Gabor filter based on stochastic computation is presented in this paper. Stochastic computation [12], [13] is a purely-digital implementation technique that represents data as streams of random bits, unlike binary computation. It can perform complicated functions like analog circuits, with area-efficient hardware, however it enjoys the scalability of digital circuits. Using the proposed algorithm, the Gabor coefficients are dynamically generated with the memory-less small hardware, leading to the flexibility and the powergating capability. For energy-efficient computation, dynamic voltage-frequency-length scaling (DVFLS) is introduced for stochastic computation. Dynamic voltage frequency scaling (DVFS) decreases the supply voltage and the frequency to lower the power dissipation with sacrificing the throughput. In contrast, the DVFLS technique controls the lengths of the stochastic bit streams depending on situations in addition to the supply voltage and frequency. With a little accuracy loss due to smaller lengths than usual, the energy dissipation is significantly reduced while maintaining the throughput. The performance of the proposed stochastic Gabor filter is evaluated and compared using the fabricated chip with TSMC 65 nm CMOS technology.

The short version of this work presented in [14] showed limited results, but the extended version presented in this paper includes: (a) fully descriptions of the stochastic Gabor algorithm, (b) error analyses of the stochastic Gabor function in comparison with the floating-point Gabor function,

2156-3357 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.





Fig. 1. Stochastic computation: (a) multiplier in unipolar coding, (b) in bipolar coding, (c) scaled adder, (d) hyperbolic tangent function, and (e) exponential function in unipolar coding.

(c) control mechanisms of a stochastic Gabor coefficient generator and (d) energy and accuracy evaluations for an application of HMAX-model based image classifications with DVFLS and power gating techniques. The rest of the paper is organized as follows. Section II reviews stochastic computation. Section III introduces the hardware algorithm of stochastic Gabor function. Section IV illustrates the hardware architecture of the proposed chip and introduces the DVFLS technique. Section V shows the measurement results of the chip and evaluates the classification accuracy of HMAX model using the stochastic Gabor filters. Section VI concludes this paper.

II. REVIEW OF STOCHASTIC COMPUTATION

Stochastic computation represents information by sequences of random bits [12], [13]. Stochastic computation has been recently used for several applications, such as image processing [15]–[17], digital filters [18], low-density parity-check (LDPC) decoders [19]–[21], MIMO decoders [22], and braininspired computing [23]–[25]. There are two mappings commonly used: *unipolar* and *bipolar* coding. For a sequence of bits a(t), denote the probability of observing a '1' to be $P_a = \Pr(a(t) = 1)$. In *unipolar* coding, the represented value A is $A = P_a$, $(0 \le A \le 1)$. In *bipolar* coding, the represented value A is $A = (2 \cdot P_a - 1), (-1 \le A \le 1)$.

Fig. 1 (a) shows a stochastic two-input multiplier in unipolar coding. The multiplier is simply realized using a two-input AND gate. The input and output probabilities are represented using N_{sto} -bit length streams. N_{sto} clock cycles are required

Fig. 2. Data converter: (a) binary-to-stochastic converter (B2S) in unipolar and bipolar coding, (b) B2S in unipolar coding with sign bit, and (c) stochastic-to-binary converter (S2B) in unipolar coding with sign bit.

to complete a multiplication, where the computation accuracy depends on N_{sto} . A stochastic multiplier in bipolar coding is realized using a two-input XNOR gate as shown in Fig. 1 (b). A stochastic scaled adder is realized using a two-input multiplier as shown in Fig. 1 (c).

Using finite state machines (FSMs) in stochastic computation, hyperbolic tangent and exponential functions are simply realized as shown in Figs. 1 (d) and (e), respectively. In the FSM-based functions, the state transits to the right, if the input stochastic bit, x(t), is "1" and the state transits to the left, otherwise. The output stochastic bit, y(t), is determined by the current state.

The stochastic tanh function, Stanh, in bipolar coding shown in Fig. 1 (d) is defined as follows:

$$\operatorname{ranh}((N_T/2)x) \approx \operatorname{Stanh}(N_T, x),$$
 (1)

where N_T is the total number of states.

The stochastic exponential function, Sexp, shown in Fig. 1 (e) is designed, where the input is encoded in bipolar coding and the output is encoded in unipolar coding [13]. In order to use both input and ouput in bipolar coding, the output coding needs to be converted to bipolar coding. Hence, the exponential function is approximated in bipolar coding as follows:

$$\exp(-2Gx) \approx (\operatorname{Sexp}(N_E, G, x) + 1)/2,$$
(2)

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

In order to use stochastic circuits with traditional binary circuits, data converters are required. Fig. 2 (a) shows a binary-to-stochastic converter (B2S) in unipolar and bipolar coding. In B2S, *n*-bit binary signals are compared with *n*-bit random



Fig. 3. Example of timing diagrams of data converters in signed unipolar coding: (a) B2S and (b) S2B.

signals generated using linear-feedback shift registers (LFSRs) to generate stochastic bit streams. To increase computation accuracy, unipolar coding with sign bit ("signed unipolar coding") can also be selected [25]. Fig. 2 (b) shows the B2S in signed unipolar coding. Binary data are converted to stochastic bit streams and a sign bit using the sign inverter (SI) circuit. The timing diagram of the B2S is shown in Fig. 3 (a). Fig. 2 (c) shows a stochastic-to-binary converter (S2B) in signed unipolar coding. In S2B, the number of "1" of stochastic bit streams is counted and the values in the counters are binary signals. Depending on the sign bit, the sign of the output data is changed. The timing diagram of the S2B is shown in Fig. 3 (b).

III. STOCHASTIC IMPLEMENTATION OF 2D GABOR FUNCTION

A. 2D Gabor Filter

2D Gabor function (odd phase) [1] is defined as follows:

$$g_{\omega,\sigma,\gamma,\theta}(x,y) = \exp\left(-\frac{{x'}^2 + \gamma^2 {y'}^2}{2\sigma^2}\right)\sin(2\omega x'),\qquad(3)$$

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 of the Gaussian envelope and γ is the spatial aspect ratio of the Gabor function. The filter function is the convolution of input images, i(x, y), and the Gabor function as follows:

$$o(x, y) = i(x, y) * g(x, y),$$
 (4)

where o(x, y) are output images.

The 2D Gabor filters exhibit similar responses of simple cells in primary visual cortex (V1) of brains as shown in Fig. 4. Many different simple cells activated with specific spatial frequencies and angles of images are placed as the hypercolumn structure. Based on the hypercolumn structures, 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 one of the brain-inspired object recognition models using Gabor filters [26].



Fig. 4. Hypercolumn structure of primary visual cortex (V1) of brains including many simple cells activated with specific spatial frequencies and angles of images. 2D Gabor filters show similar responses of the simple cells.



Fig. 5. Graphical representation of Ssin function using five Stahh functions, where $\omega' = 2\pi$ and $\omega = \pi$ are used.

B. Stochastic 2D Gabor Function

Sin and cos functions are approximated using several tanh functions [27]. The stochastic sin function, $Ssin(\omega, \lambda, x)$ ($\approx sin(\omega x)$), in bipolar coding is defined as follows:

$$\operatorname{Ssin}(\omega,\lambda,x) = \sum_{k=\lceil -\frac{\omega'}{\pi}\rceil}^{\lfloor \frac{\omega'}{\pi}\rfloor} (-1)^k \operatorname{Stanh}\left(4\omega', \frac{1}{2}\left(\lambda x + \frac{\pi k}{\omega'}\right)\right), \quad (5)$$

where ω' is a constant angular frequency and λ is ω/ω' . ω' determines the maximum angular frequency supported. Fig. 5 shows a graphical representation of Ssin function using five Stanh functions, where $\omega' = 2\pi$ and $\omega = \pi$ are used. In this example, five different Stanh functions are geometrically connected to design the approximated sin function because of $\omega' = 2\pi$. λ can be tuned to set ω desired.

In addition, the stochastic cos function, $S\cos(\omega, \lambda, x)$ ($\approx \cos(\omega x)$), is defined as follows:

 $Scos(\omega, \lambda, x)$

$$=\sum_{k=\lceil-\frac{\omega'}{\pi}-\frac{1}{2}\rceil}^{\lfloor\frac{\omega'}{\pi}-\frac{1}{2}\rfloor}(-1)^{k}\operatorname{Stanh}\left(4\omega',\frac{1}{2}\left(\lambda x+\frac{\pi\left(k+\frac{1}{2}\right)}{\omega'}\right)\right),\quad(6)$$



Fig. 6. Simulated SGabor for a kernel size of 51×51 with different configurations with $T_{cgen} = 2^{18}$.



Fig. 7. Floating-point Gabor and SGabor results for a kernel size of 51×51 .

TABLE I Parameters for SGabor

Using Eqs. (2), (5), and (6), the stochastic 2D Gabor function is defined as follows:

$$\begin{aligned} \text{SGabor}(\omega, \gamma, \lambda, G, \theta, x, y) \\ &= \frac{\text{Sexp}\Big(N_E, G, \frac{1}{2}(x'^2 + \gamma^2 y'^2)\Big) + 1}{2} \text{Ssin}(\omega, \lambda, x'), \quad (7) \\ x' &= x \text{Scos}(\pi, \lambda_{\pi}, \theta/\pi) + y \text{Ssin}(\pi, \lambda_{\pi}, \theta/\pi), \\ y' &= -x \text{Ssin}(\pi, \lambda_{\pi}, \theta/\pi) + y \text{Scos}(\pi, \lambda_{\pi}, \theta/\pi), \end{aligned}$$

where λ_{π} is constant. The original Gabor function on Eq. (3) is approximated as follows:

$$\alpha g_{\omega,\sigma,\gamma,\theta}(x,y) \approx \text{SGabor}(\omega,\gamma,\lambda,G,\theta,x,y), \quad (8)$$

where α is a constant value for fitting SGabor with the original Gabor function.

C. Error Analyses of SGabor Coefficients

Fig. 6 shows simulated Gabor functions using SGabor for a kernel size of 51×51 with different configurations in MATLAB. The length (cycle) of stochastic bit streams for SGabor is defined as T_{cgen} . In this simulation, ω and θ are changed with $T_{cgen} = 2^{18}$. Using SGabor, any ω and θ can



Fig. 8. Hardware architecture of Gabor filter: (a) conventional method and (b) proposed method.

be configured depending on requirements. Fig. 7 shows both floating-point Gabor and SGabor results for a kernel size of 51×51 . The detailed parameters are summarized in Table I. $\omega' = 14$ is selected that supports the maximum angular frequency of 4π . $\gamma = 1$ is selected based on [7] that uses the same Gaussian envelope along with x and y.

The simulated SGabor results are compared with the floating-point Gabor results by changing T_{cgen} as shown in Tables II-III. The kernel sizes are selected as 5×5 , 7×7 , 11×11 , 16×16 , and 51×51 . ω is fixed to 2π and θ are selected from $\theta = 0$ and 30° . For the error evaluation, a normalized RMS (NRMS) error per pixel is defined as follows:

$$NRMS = \frac{|g(\text{floating point}) - g(\text{stochastic})|}{g_{max} - g_{min}}, \qquad (9)$$

where g_{max} is 1 and g_{min} is -1. Then, the averaged NRMS errors are obtained depending on kernel sizes.

In most cases, the NRMS errors are decreased when T_{cgen} is increased because the computation accuracy is increased in stochastic computation. In contrast, the NRMS errors at larger T_{cgen} are not significantly changed in different kernel sizes. Tables IV and V show NRMS errors in different ω and θ with $T_{cgen} = 2^{18}$. In most cases, the NRMS errors are smaller than 5%. Using SGabor, the proposed 2D Gabor-filter chip dynamically generates Gabor coefficients described in the next section.

IV. HARDWARE ARCHITECTURE

A. Design Concept

Fig. 8 shows the hardware architectures of the Gabor filters. In the conventional method [6], [11], the Gabor coefficients are calculated by software in advance and are then stored in memory. Hence, the Gabor coefficients and the kernel sizes are fixed. In addition, static power dissipations remain at idle state as power gating cannot be applied to the memory block. In contrast, the proposed method dynamically generates the Gabor coefficients with different configurations, leading to the flexibility of the Gabor filter and the power-gating capability.

Fig. 9 (a) shows a timing diagram of the proposed Gaborfilter hardware. In the proposed hardware, the Gabor coefficients are generated before the convolution of images and the

TABLE II	
NRMS Errors at $\omega = 2\pi$ and $\theta =$	= 0

Kernel size		NRMS error at T_{cgen} =								
	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	218	2^{19}
5x5	$\sim 0\%$	$\sim 0\%$	$\sim 0\%$	$\sim 0\%$	$\sim 0\%$	$\sim 0\%$	$\sim 0\%$	$\sim 0\%$	$\sim 0\%$	$\sim 0\%$
7x7	20.52%	14.04%	8.001%	6.604%	4.121%	5.493%	3.973%	3.071%	3.251%	2.852%
11x11	26.17%	20.59%	11.75%	7.844%	6.157%	6.575%	4.628%	4.287%	3.863%	4.105%
16x16	25.64%	17.74%	13.19%	8.081%	5.974%	5.197%	4.606%	4.551%	4.145%	4.097%
51x51	24.97%	18.11%	11.97%	10.17%	6.861%	5.91%	4.798%	4.974%	4.6%	4.275%

TABLE III NRMS Errors at $\omega = 2\pi$ and $\theta = 30^{\circ}$

Kernel size		NRMS error at $T_{cqen} =$								
	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}
5x5	8.789%	9.551%	6.636%	6.664%	5.681%	5.943%	3.975%	2.671%	3.007%	3.900%
7x7	22.09%	16.12%	10.89%	5.178%	4.418%	4.582%	4.147%	2.806%	3.481%	2.824%
11X11	24.75%	17.45%	9.378%	7.02%	5.594%	3.935%	4.608%	3.468%	3.206%	3.35%
16X16	24.07%	17.61%	11.31%	7.583%	5.885%	4.34%	4.319%	3.029%	3.4%	2.745%
51X51	27.50%	19.34%	12.22%	8.006%	6.291%	5.846%	4.215%	3.748%	3.147%	2.807%

TABLE IV

NRMS ERRORS AT KERNEL SIZE OF 5 \times 5 AND $T_{cgen} = 2^{18}$

ω	NRMS error at θ =							
	-180°	-135°	-90°	-45°	0°	45°	90°	135°
π	1.011%	1.205%	1.268%	1.093%	1.241%	0.6699%	1.288%	1.192%
2π	$\sim 0\%$	2.032%	$\sim 0\%$	1.426%	$\sim 0\%$	1.734%	$\sim 0\%$	2.703%
3π	2.801%	1.896%	3.153%	2.696%	3.447%	1.822%	3.294%	2.427%

TABLE V		
NRMS Errors at Kernel Size of 11×11 and T_{cgen}	=	2^{18}

ω			NRM	AS error at	$\theta =$			
	-180°	-135°	-90°	-45°	00	45°	90°	135°
π	2.193%	1.154%	1.898%	1.249%	1.898%	1.156%	2.541%	1.438%
2π	4.279%	2.28%	4.714%	2.146%	3.774%	1.916%	4.57%	2.308%
3π	5.447%	3.783%	6.105%	3.197%	5.154%	2.942%	5.928%	4.187%

coefficients, if new coefficients are required. T_{conv} is defined as the number of cycles for filtering an image as follows:

$$T_{conv} = H \cdot W \cdot N_{sto}/N_P, \qquad (10)$$

where H and W are height and width of input images. N_P is the number of parallel convolution units. In order to generate new coefficients before filtering, the following condition needs to be satisfied:

$$T_{cgen} < T_{conv}.$$
 (11)

As the NRMS errors of SGabor are decreased with a large T_{cgen} , the maximum T_{cgen} is selected while satisfying Eq. (11).

Fig. 9 (b) shows the timing diagram after the power supply is recovered. When the power supply is off at the idle state, all the coefficients of SGabor are disapear. Hence, after the power supply is recovered, all the coefficients need to be generated before filtering. Hence, in the first operation, the coefficients are generated and then the filtering starts in the second operation.

B. Overall Structure

Fig. 10 (a) shows a block diagram of the proposed 64 parallel stochastic configurable 2D Gabor-filter chip. The 64 parallel Gabor-filtering block is designed in signed unipolar coding while the coefficient generator is designed in bipolar coding. The signed unipolar coding is selected in order to reduce N_{sto} with high computation accuracy. Each filtering block is used for filtering a different 5×5 subwindow. As stochastic computation takes N_{sto} clock cycles to complete the computation, 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. Sliding the 5×5 subwindow performs in the line buffer.

Suppose that the image sizes are VGA (640 x 480). In this chip, there are three cases of N_{sto} : 64, 128, and 256. The selector signal, *sel*, for MUX and DMUX changes every clock cycle in case of $N_{sto} = 64$. In case of $N_{sto} = 128$ and 256, *sel* changes every two and four clock cycles, respectively. N_{sto} is controlled to match the energy and the computation accuracy required for applications. For example, if a small N_{sto} (e.g. 64) is selected, the energy dissipation is smaller and the computation accuracy is lower than that in case of a large N_{sto} (e.g. 256).

Each convolution unit with a 5×5 kernel size is designed using hybrid stochastic/binary computation. The multiplications are realized in signed unipolar coding based on stochastic computation and the additions are realized based on binary



Fig. 9. Example of timing diagrams of the proposed Gabor-filter hardware: (a) normal operation and (b) initial operation after the power supply is recovered.

TABLE VI T_{conv} vs Kernel Sizes for VGA Images

Kernel size		5x5	11x11	16x16	51x51
T_{max}	N_{sto} =64 N_{sto} =128	307,200 614 400	1,536,000	3,379,200 6,758,400	32,256,000 64 512 000
1 conv	N_{sto} =256	1,228,800	6,144,000	13,516,800	129,024,000

computation based on [25]. The timing of the stochastic convolution is similar to Fig. 3. For the first $(N_{sto} - 1)$ cycles, multiplications perform in stochastic domain using a two-input AND gate and a two-input XNOR gate as shown in Fig. 10 (b). In the last cycle, the stochastic bit streams stored in the counter are converted to binary data, which are then added in binary domain.

When larger kernel sizes (e.g. 11×11 , 16×16 , and etc) are used for filtering, images are firstly divided into sub images of less than or equal to 5×5 , which are then filtered several times. Table VI summarizes T_{conv} depending on kernel sizes in case of VGA images. In order to satisfy Eq. (11), $T_{cgen} = 2^{18}$ can be selected as the maximum value of T_{cgen} .

Fig. 11 (a) shows the stochastic Gabor coefficient generator. The coefficient generator is designed in bipolar stochastic computation based on Eq. (8) with $\gamma = 1$. The desired ω is controlled by λ . The stochastic sin function, Ssin, is designed using five Stanh functions corresponding to Fig. 5 as shown in Fig. 11 (b). When the summation is realized using the



Fig. 10. Overall structure: (a) 64 parallel stochastic configurable 2D Gabor filter and (b) stochastic multiplication in signed unipolar coding.

TABLE VII Normalized Parameters of DVFS and DVFLS in Stochastic Computation

	Normal	DVFS	DVFLS
Voltage	1	δ	δ
Frequency	1	κ	κ
N_{sto}	1	1	ρ
Dynamic power	1	$\delta^2 \kappa$	$\delta^2 \kappa$
Throughput	1	κ	κ/ ho
Dynamic energy	1	δ^2	$\delta^2 \rho$
Accuracy	_	_	depends on ρ

mulltiplexor (scaled adder), the output is scaled down by 5. However, at a specific x, one of five Stanh is active while the other four Stanh functions generate a fixed value of +1 or -1. It means that the other four Stanh determines the sign of the output of Ssin. Therefore, the summation in Eq. (5) can be simply designed using the XOR gate in this case, removing the scaling factor of 1/5 in the output. Note that the summation is realized by an XOR gate when $\lfloor \frac{\omega'}{\pi} \rfloor$ is odd and is realized by an XNOR gate when $\lfloor \frac{\omega'}{\pi} \rfloor$ is even (See details in [27]). In this work, the simplified summation is used. In addition, Scos is designed as well as Ssin.

C. Dynamic Voltage-Frequency-Length Scaling (DVFLS)

Dynamic voltage frequency scaling (DVFS) is often used to reduce the power dissipation depending on situations. In this chip, dynamic voltage-frequency-length scaling (DVFLS) for stochastic computation is proposed to control the lengths of



Fig. 11. Stochastic 2D Gabor coefficient generator: (a) SGabor function and (b) Ssin function using the scaled adder.

the stochastic bit streams in addition to the supply voltage and the clock frequency. DVFLS can provide more flexible performance (e.g. high throughput and/or low energy) depending on situations.

Table VII summarizes the differences between DVFS and DVFLS. In both techniques, the normalized supply voltage can be reduced to δ ($0 \le \delta \le 1$) and the normalized clock frequency can be reduced to κ (0 $\leq \kappa \leq$ 1). In DVFS, the dynamic power can be reduced to $\delta^2 \kappa$ while the throughput can be also lower to κ . In contrast, in DVFLS, the length of the stochastic bit stream, N_{sto} , is controlled by ρ ($\rho > 0$), providing the flexible performance required in situation. When ρ is equal to κ , the throughput is maintained as the normal case while the dynamic power is reduced to $\delta^2 \kappa$ with a little computation accuracy loss. Higher computation accuracy than the normal case is realized when ρ is larger than 1. In addition, when ρ is less than κ , higher throughput is also realized. The DVFLS technique is applied to the stochastic configurable 2D Gabor filter, which can be fit in different performance demands.

V. EVALUATION

A. Measurement Results With DVFLS

Fig. 12 (a) shows the photomicrograph of the proposed stochastic Gabor filter chip using TSMC 65 nm CMOS technology. The supply voltage is 1.0 V and the area is 1.79 mm \times 1.79 mm including I/Os. The proposed circuit is designed using Verilog HDL and the test chip is realized using Synopsys Design Compiler and Cadence SoC Encounter.



Fig. 12. Fabricated chip using TSMC 65 nm CMOS technology: (a) photomicrograph of the stochastic Gabor filter circuit, (b) system overview of the test environment, and (c) test environment with FPGA.

Fig. 12 (b) shows the system overview of the test environment. N_{sto} , the clock frequency, and the supply voltage of the chip are controlled for DVFLS. Fig. 12 (c) shows the test environment with Keysight N6705B that controls the supply voltage. N_{sto} , the clock frequency and I/O signals are controlled by the FPGA. Images are captured by a camera and the input pixels in grayscale are transferred to the chip through the FPGA (Digilent Genesys 2). The output pixels of the test chip are sent back to the FPGA and are displayed with VGA.

Fig. 13 shows measured power dissipations of the test chip at 50 MP(pixel)/s with DVFLS. The normal case is that the clock frequency is 200 MHz and N_{sto} is 256. Using DVFLS, N_{sto} with both the supply voltage and the clock frequency can be reduced to lower the power dissipations. In case of $N_{sto} = 128$, the supply voltage is lower to the minimum voltage of 0.7 with a clock frequency of 100 MHz. In the same way, the supply voltage is lower to the minimum voltage of 0.55 with a clock frequency of 50 MHz. Compared with the normal case, the power dissipation is reduced by 70.0% and 88.6% in case of N_{sto} of 128 and 64, respectively, while



Fig. 13. Measured power dissipations at 50 MP(pixel)/s with DVFLS.

TABL	E VIII
MEASUREMENT RESULTS	WITH DVFS AND DVFLS

	Normal	DVFS		DVFLS	
			Case 1	Case 2	Case 3
Supply voltage [V]	1	0.55	0.7	0.55	1.0
Frequency [MHz]	200	50	100	50	200
N_{sto}	256	256	128	64	64
Throughput [MP/s]	50	12.5	50	50	200
Total power [mW]	102	11.7	30.9	11.7	102
Static power [mW]	13.1	5.6	6.8	5.6	13.1
Energy $[\mu J/P]$	2.05	0.515	0.433	0.129	0.510
Average error [%]	-	-	0.195	0.391	0.391

TABLE IX Throughput vs. Kernel Size in Case 3

Kernel size	5x5	7x7	11x11	16x16	51x51
Throughput [MP/s]	200	100	40	18.2	1.91

maintaining the same throughput of 50 MP/s with a little accuracy loss.

Table VIII summarizes the measured performance with DVFS and DVFLS. There are five cases: normal, DVFS and three DVFLS. In the DVFS case, the clock frequency is 50 MHz and N_{sto} is 64 with the supply voltage of 0.55 V. Using the conventional DVFS technique, the power dissipation is significantly reduced in comparison with the normal case while the throughput is $4 \times$ lower.

In DVFLS, there are three cases depending on different supply voltages, clock frequencies and N_{sto} . The proposed DVFLS technique in case 1 provides a 78.8% energy reduction and the same throughput as the normal case with a 0.195% accuracy loss on average. In case 2, the smaller energy dissipation is achieved with a 0.391% accuracy loss on average. In addition, in case 3, 4× higher throughput than the normal case is obtained while maintaining the same power dissipation with a 0.391% accuracy loss on average. Using DVFLS, the throughput, energy, and accuracy are dynamically controlled depending on demands.

The throughput also depends on the kernel size as summarized in Table IX.

B. Comparisons With Related Works

Table X shows performance comparisons with related works. It is hard to compare the performance directly because they are designed with different functionalities and configurations. The memory-based methods [6], [11] use fixed



Fig. 14. HMAX model for image classifications. The first layer, S1, is 2D Gabor filtering.

coefficients with fixed kernel sizes, lacking the flexibility. In the conventional configurable Gabor filter [7], CORDIC is exploited to dynamically generate the coefficients related to sinusoidal function for flexible Gabor filtering. However, other coefficients are stored in memory, losing the powergating capability. In contrast, the proposed circuits achieve a higher throughput/area and a more flexible filtering than the conventional configurable Gabor filter with the power-gating capability, leading to zero standby power.

C. Application to HMAX-Model Based Image Classifications

The stochastic Gabor filter is applied to HMAX-model based image recognitions [26] in order to evaluate the classification accuracy and the power dissipation. Fig. 14 shows a block diagram of HMAX model, consisting of four layers: S1, C1, S2, and C2. The first layer, S1, is 2D Gabor filtering to extract features of images, such as lines and edges. C1 and C2 are max-pooling layers for the extracted features. In S2, the extracted features are compared with features previously trained in order to find the nearest features.

The kernel sizes of S1 are from 7×7 to 39×39 with four angles of 0°, 45°, 90°, and 135°. In this paper, there are two scenarios of image recognitions tested: car and face. In the training phase, 50 car (face) images and 50 natural images are trained in floating point. In the classification phase, 50 car (face) images and 50 natural images are used to classify if they are car (face) or others in floating point or stochasitc computation.

Table XI shows the classification accuracy in the HMAX application between the proposed stochastic method and the floating point. In the proposed method, the first layer, S1, performs in the chip to obtain extracted features of images in cases 1 and 2. In addition, the MATLAB code of [26] is modified in order to remove the S1 function. Then, the extracted features are used as the inputs of the modified MATLAB code to calculate the classification accuracy. As a result, the classification accuracies in normal case are the same as the floating-point results. In case 1, the classification accuracies are slightly dropped in comparison with the floating-point results because of smaller N_{sto} than that of the normal case.

Fig. 15 shows the average power dissipations of the stochastic Gabor-filter chip in case of S1 of HMAX model with DVFLS and power gating (PG). In this scenario, the ratio of active mode to the total operation time is defined. When the ratio is 50%, power gating can be applied at the idle time of

TABLE X
COMPARISONS WITH RELATED WORKS

	[11]	[6]	[7]	This work
Computation	Analog/Digital	Digital	Digital	Stochastic
Technology	0.35 µm CMOS	(FPGA: Altera Stratix IV 230))	$0.13 \ \mu m \ CMOS$	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)	200 (5x5)
			2.1 (11x11)	40 (11x11)
Frequency [MHz]	1	148.5	250	200
Power dissipation [mW]	800	-	-	102.3
Area (equivalent gate count)	-	-	63.8k	644k



Fig. 15. Average power dissipations of the stochastic Gabor-filter chip in case of S1 of HMAX model with DVFLS and power gating (PG).

TABLE XI CLASSIFICATION ACCURACY IN HMAX APPLICATIONS

	Stochastic			Floating point
	Normal	DVFLS		
		Case 1	Case 2	
Accuracy (face) [%]	91	86	83	91
Accuracy (car) [%]	97	97	62	97

the Gabor filter. In case 1, the average power dissipations are reduced by 84.9% and 97.0% at the ratio of 50% and 10%, respectively, while achieving the similar classification accuracy to the normal case. As a result, the proposed memory-less structure can significantly reduce the power dissipation when applications are often at the idle state.

VI. CONCLUSION

In this paper, the 65-nm stochastic configurable 2D Gaborfilter chip with DVFLS has been presented. The proposed stochastic Gabor circuit dynamically generates the coefficients with different kernel sizes using simple hardware, leading to the flexibility of the Gabor filter, as opposed to the conventional memory-based methods. The memory-less structure provides the power-gating capability for zero standby power dissipation at an idle case. Compared with the conventional DVFS technique, the proposed chip with DVFLS achieves a $4 \times$ higher throughput and a $4 \times$ lower energy dissipation with a 0.391% accuracy loss. In addition, the proposed stochastic configurable Gabor filter shows a higher throughput/area and a more flexible filtering than the conventional configurable Gabor filter based on binary computation. In the application of the HMAX-model based image classifications, the average power dissipations with DVFLS are reduced by 84.9% and 97.0% at the activation ratio of 50% and 10%, respectively, while achieving the similar classification accuracy without DVFLS. In the future prospect, the proposed DVFLS technique can be used for different stochastic circuits, such as deep neural networks and image processing.

REFERENCES

- D. Gabor, "Theory of communication. Part 1: The analysis of information," J. Inst. Elect. Eng. III, Radio Commun. Eng., vol. 93, no. 26, pp. 429–441, Nov. 1946.
- [2] J. Wu, G. An, and Q. Ruan, "Independent Gabor analysis of discriminant features fusion for face recognition," *IEEE Signal Process. Lett.*, vol. 16, no. 2, pp. 97–100, Feb. 2009.
- [3] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time precrash vehicle detection system," in *Proc. 6th IEEE Workshop Appl. Comput. Vis. (WACV)*, Dec. 2002, pp. 171–176.
- [4] J.-M. Guo, H. Prasetyo, and K. Wong, "Vehicle verification using Gabor filter magnitude with gamma distribution modeling," *IEEE Signal Process. Lett.*, vol. 21, no. 5, pp. 600–604, May 2014.
- [5] S. Marcelja, "Mathematical description of the responses of simple cortical cells," *J. Opt. Soc. Amer.*, vol. 70, no. 11, pp. 1297–1300, Nov. 1980.
- [6] E. Cesur, N. Yildiz, and V. Tavsanoglu, "On an improved FPGA implementation of CNN-based Gabor-type filters," *IEEE Trans. Circuits Syst.*, *II, Exp. Briefs*, vol. 59, no. 11, pp. 815–819, Nov. 2012.
- [7] J.-B. Liu, S. Wang, Y. Li, J. Han, and X.-Y. Zeng, "Configurable pipelined Gabor filter implementation for fingerprint image enhancement," in *Proc. 10th IEEE Int. Conf. Solid-State Integr. Circuit Technol. (ICSICT)*, Nov. 2010, pp. 584–586.
- [8] T. Y. W. Choi, B. E. Shi, and K. A. Boahen, "An ON-OFF orientation selective address event representation image transceiver chip," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 2, pp. 342–353, Feb. 2004.
- [9] T. Y. W. Choi, P. A. Merolla, J. V. Arthur, K. A. Boahen, and B. E. Shi, "Neuromorphic implementation of orientation hypercolumns," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 6, pp. 1049–1060, Jun. 2005.
- [10] F. Borghetti, P. Malcovati, and F. Maloberti, "A current-mode 64 × 1 programmable Gabor filter for early vision systems," in *Proc. 27th Eur. Solid-State Circuits Conf. (ESSCIRC)*, Sep. 2001, pp. 205–208.
- [11] T. Morie, J. Umezawa, and A. Iwata, "A pixel-parallel image processor for Gabor filtering based on merged analog/digital architecture," in *Dig. Tech. Papers Symp. VLSI Circuits*, Jun. 2004, pp. 212–213.
- [12] B. R. Gaines, "Stochastic computing systems," in Advances in Information Systems Science. Springer, 1969, pp. 37–172.
- [13] B. D. Brown and H. C. Card, "Stochastic neural computation. I. Computational elements," *IEEE Trans. Comput.*, vol. 50, no. 9, pp. 891–905, Sep. 2001.
- [14] N. Onizawa, K. Matsumiya, W. J. Gross, and T. Hanyu, "Accuracy/energy-flexible stochastic configurable 2D Gabor filter with instant-on capability," in *Proc. 43rd IEEE Eur. Solid State Circuits Conf. (ESSCIRC)*, Sep. 2017, pp. 43–46.

- [15] P. Li and D. J. Lilja, "Using stochastic computing to implement digital image processing algorithms," in *Proc. 29th IEEE Int. Conf. Comput. Design (ICCD)*, Oct. 2011, pp. 154–161.
- [16] A. Alaghi, C. Li, and J. Hayes, "Stochastic circuits for real-time imageprocessing applications," in *Proc. 50th ACM EDAC IEEE Design Autom. Conf. (DAC)*, May 2013, pp. 1–6.
- [17] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel, "Computation on stochastic bit streams digital image processing case studies," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 3, pp. 449–462, Mar. 2014.
- [18] Y. Liu and K. K. Parhi, "Architectures for recursive digital filters using stochastic computing," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3705–3718, Jul. 2016.
- [19] V. C. Gaudet and A. C. Rapley, "Iterative decoding using stochastic computation," *IET Electron. Lett.*, vol. 39, no. 3, pp. 299–301, Apr. 2003.
- [20] S. S. Tehrani, W. J. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 10, pp. 716–718, Oct. 2006.
- [21] S. S. Tehrani, S. Mannor, and W. J. Gross, "Fully parallel stochastic LDPC decoders," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5692–5703, Nov. 2008.
- [22] J. Chen, J. Hu, and J. Zhou, "Hardware and energy-efficient stochastic LU decomposition scheme for MIMO receivers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 4, pp. 1391–1401, Apr. 2016.
- [23] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. J. Gross, "VLSI implementation of deep neural network using integral stochastic computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2688–2699, Oct. 2017.
- [24] 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," *J. Signal Process. Syst.*, vol. 90, no. 5, pp. 709–725, May 2018.
- [25] N. Onizawa, S. Koshita, S. Sakamoto, M. Abe, M. Kawamata, and T. Hanyu, "Area/energy-efficient gammatone filters based on stochastic computation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2724–2735, Oct. 2017.
- [26] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neurosci.*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999.
- [27] N. Onizawa, D. Katagiri, K. Matsumiya, W. J. Gross, and T. Hanyu, "Gabor filter based on stochastic computation," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1224–1228, Sep. 2015.



Naoya Onizawa (M'09) received the B.E., M.E., and D.E. degrees in electrical and communication engineering from Tohoku University, Japan, in 2004, 2006, and 2009, respectively. In 2011, he joined the University of Waterloo, Canada, as a Post-Doctoral Fellow. From 2011 to 2013, he was a Post-Doctoral Fellow with McGill University, Canada. In 2015, he joined the University of Southern Brittany, Lorient, France, as a Visiting Associate Professor. He is currently an Assistant Professor with the Frontier Research Institute for Interdisciplinary Sciences,

Tohoku University. His main interests and activities are in the energy-efficient very large-scale integration (VLSI) design based on asynchronous circuits and probabilistic computation and their applications, such as associative memories and brain-like computers.

He received the Best Paper Award in the 2010 IEEE ISVLSI, the Best Paper Finalist in the 2014 IEEE ASYNC, the 20th Research Promotion Award from the Aoba Foundation for the Promotion of Engineering in 2014, and the Kenneth C. Smith Early Career Award for Microelectronics Research in the 2016 IEEE ISMVL.



Daisaku Katagiri received the B.E. and M.E. degrees from Tohoku University, Sendai, Japan, in 2014 and 2016, respectively. His main interests and activities are in design of area efficient hardware based on stochastic computation.



Kazumichi Matsumiya received the Ph.D. degree from the Tokyo Institute of Technology in 2000. From 2000 to 2001, he was a Post-Doctoral Fellow with the Centre for Vision Research, York University, Canada. From 2002 to 2003, he was a Post-Doctoral Fellow with the Imaging Science and Engineering Laboratory, Tokyo Institute of Technology. From 2004 to 2005, he was a Research Fellow with ATR. In 2005, he joined the Research Institute of Electrical Communication, Tohoku University, as an Assistant Professor, where he was an

Associate Professor from 2014 to 2018. Since 2016, he has been a JST PRESTO Researcher for the Design of Information Infrastructure Technologies Harmonized with Societies concurrently. Since 2018, he has been a Professor of cognitive psychology with the Graduate School of Information Sciences, Tohoku University. His research interests are in human cognitive and perceptual processing, such as body awareness, visual-haptic integration, interactions between vision and action, and eye movements.

He was a recipient of the Minoru Ishida Foundation Research Encouragement Award in 2016 and the 13th Japan Society for the Promotion of Science Prize in 2016. He received the Distinguished Contributed Paper Award of the 2010 Society for Information Display International Symposium, the Ukai Paper Award of the Vision Society of Japan in 2012, and the Best Paper Award of FAN 2015.



Warren J. Gross (S'92–M'04–SM'10) received the B.A.Sc. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, in 1999 and 2003, respectively. He is currently a Professor and the Chair of the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada. His research interests are in the design and implementation of signal processing systems and custom computer architectures.

Dr. Gross served as the Chair for the IEEE Signal Processing Society Technical Committee on Design and Implementation of Signal Processing Systems. He served as the General Co-Chair for the IEEE GlobalSIP 2017 and the IEEE SiPS 2017 and the Technical Program Co-Chair for SiPS 2012. He also served as an Organizer for the Workshop on Polar Coding in Wireless Communications at WCNC 2017, the Symposium on Data Flow Algorithms and Architecture for Signal Processing Systems (GlobalSIP 2014), and the IEEE ICC 2012 Workshop on Emerging Data Storage Technologies. He served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and as a Senior Area Editor. He is a Licensed Professional Engineer in the Province of Ontario.



Takahiro Hanyu (SM'12) received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1984, 1986, and 1989, respectively. He is currently a Professor with the Research Institute of Electrical Communication, Tohoku University. His general research interests include nonvolatile logic circuits and their applications to ultra-low-power and/or highly dependable very large-scale integration (VLSI) processors and post-binary computing and its application to brain-inspired VLSI systems.

He was a recipient of the Ichimura Academic Award in 2010. He received the Sakai Memorial Award from the Information Processing Society of Japan in 2000, the Judge's Special Award at the 9th LSI Design of the Year from the Semiconductor Industry News of Japan in 2002, the Special Feature Award at the University LSI Design Contest from ASP-DAC in 2007, the APEX Paper Award of Japan Society of Applied Physics in 2009, the Excellent Paper Award of IEICE, Japan, in 2010, the Best Paper Award of IEEE ISVLSI 2010, the Paper Award of SSDM 2012, the Best Paper Finalist of IEEE ASYNC 2014, and the Commendation for Science and Technology by MEXT, Japan in 2015.