J Sign Process Syst (2018) 90:709–725
DOI 10.1007/s11265-016-1197-3

A Generalized Stochastic Implementation of the Disparity Energy Model for Depth Perception

Kaushik Boga[1] · François Leduc-Primeau[1] · Naoya Onizawa[2] · Kazumichi Matsumiya[2] · Takahiro Hanyu[2] · Warren J. Gross[1]

Received: 4 April 2016 / Revised: 28 September 2016 / Accepted: 18 October 2016 / Published online: 7 December 2016

Abstract Implementing neuromorphic algorithms is increasingly interesting as the error resilience and low-area, low-energy nature of biological systems becomes the potential solution for problems in robotics and artificial intelligence. While conventional digital methods are inefficient in implementing massively parallel systems, analog solutions are hard to design and program. Stochastic Computing (SC) is a natural bridge that allows pseudo-analog computations in the digital domain using low complexity hardware. However, large scale SC systems traditionally suffered from long latencies, hence higher energy consumption. This work develops a VLSI architecture for an SC based binocular vision system based on a disparity-energy model that emulates the hierarchical multi-layered neural structure in the primary visual cortex. The 3-layer neural network architecture is biologically plausible and is tuned to detecting 5 different disparities. The architecture is compact, adder-free, and achieves better disparity detection compared to a floating-point version by using a modified disparity-energy model. A generalized 1x100 pixel processing system is synthesized using TSMC 65nm CMOS technology and it achieves 71 % reduction in area-delay product and 48 % in energy savings compared to a fixed-point implementation at equivalent precision.

Keywords Stochastic computing · Neuromorphic computing · Approximate computing · Gabor filters · Disparity-energy model · Computer vision · Biomedical electronics · Neural networks

1 Introduction

Measuring the relative depth of objects efficiently in real-time is a crucial issue as advances in robotics and artificial intelligence lead to ever smaller and more versatile devices with constrained resource budgets. Compared to conventional real-time image processing hardware implementations, biological vision achieves considerable power efficiency with high error resilience. Neuromorphic engineering aims to build hardware that exhibits these properties by taking inspiration from the structures and algorithms of biological systems. Ohzawa et al. proposed a disparity-energy model to express the disparity-selective properties of binocular complex cells in the primary visual cortex (V1) that are responsible for depth perception in the brain [19]. 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 [22]. This model has been shown to be

✉ Kaushik Boga
kaushik.boga@mail.mcgill.ca

François Leduc-Primeau
francois.leduc-primeau@mail.mcgill.ca

Naoya Onizawa
nonizawa@m.tohoku.ac.jp

Kazumichi Matsumiya
kmat@riec.tohoku.ac.jp

Takahiro Hanyu
hanyu@riec.tohoku.ac.jp

Warren J. Gross
warren.gross@mcgill.ca

[1] Department of Electrical and Computer Engineering, McGill University, Montréal, Québec, Canada

[2] Tohoku University, Sendai, Japan

valid in monkeys [10] and to describe well the response of binocular complex cells in V1 [23].

Several analog and mixed-signal VLSI vision chip implementations exhibiting low power and high area efficiency have been proposed [9, 15, 23]. These chips rely on analog transistor behavior to emulate the receptive field responses of neurons in V1. However, it is difficult to design highly-programmable analog circuits to describe higher-order neural behaviors compared to the digital counterparts [23]. Also, they do not scale easily to newer technology nodes and are sensitive to manufacturing variations. This motivates us to find digital implementations that exhibit similar characteristics to that of analog circuits. Stochastic computing (SC) bridges this gap by performing pseudo-analog operations in the digital domain.

This paper describes the VLSI implementation of a multi-layered neural network for the disparity-energy model using SC. Layer 1 of the network performs difference-of-Gaussian filtering that mimicks the center-surround receptive fields (RF) in the retina, layer 2 performs Gabor filtering mimicking the orientation selective filtering performed by the receiptive fields of the simple cells and layer 3 has simple and complex cells tuned to detecting 5 different disparities. We use *exponential compression* [11] to transform stochastic streams into the exponential domain where less precise stochastic additions are converted to precise stochastic multiplications, resulting in an adder-free architecture.

Section 2 first reviews SC and the disparity-energy model, then Section 3 presents a modification to the SC exponential compression technique [11]. Section 4 presents the proposed system architecture. Implementation results are given in Section 5, Section 6 presents details of a configurable architecture and Section 7 concludes the paper.

2 Background Material

2.1 Disparity-energy Model

When an object is perceived from the left and right eyes, its position will be horizontally displaced in each of the corresponding images, as illustrated in Fig. 1. The brain uses this horizontal disparity, d, to estimate the relative depths of objects in three dimensions [19]. Zero disparity corresponds to those objects whose position is the same from both perspectives and excites corresponding retinal cells in each eye. Positive and negative disparities (corresponding to farther and closer objects) consequently will excite different retinal cells in each eye [23].

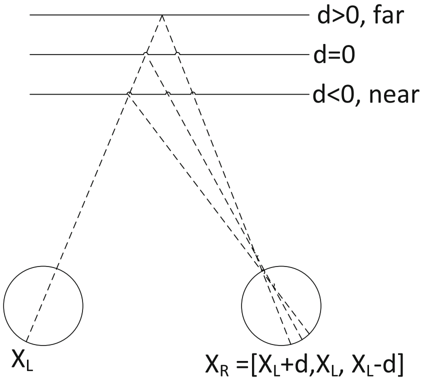


Figure 1 Relation between disparity and depth [4].

The disparity-energy model shown in Fig. 2 explains how the neural hierarchy in the brain processes this information to detect disparity.

In this work, we implement a pre-processing difference-of-gaussian (DOG) filtering stage (not shown in Fig. 2) along with the disparity-energy model. Layer 1 has 2xN two-dimensional (2D) DOG filters that pre-filter the left and right images, where N is the number of pixels in one input image. Layer 2 has 2xN 2D Gabor even and 2xN 2D Gabor odd filters that filter the left and right images. A layer of Gabor even and Gabor odd filters are tuned to detect a single orientation in an image. In this work, only one layer is used, tuned to detect vertical edges only. Layer 3 is composed of 5 disparity groups, where each disparity group is composed

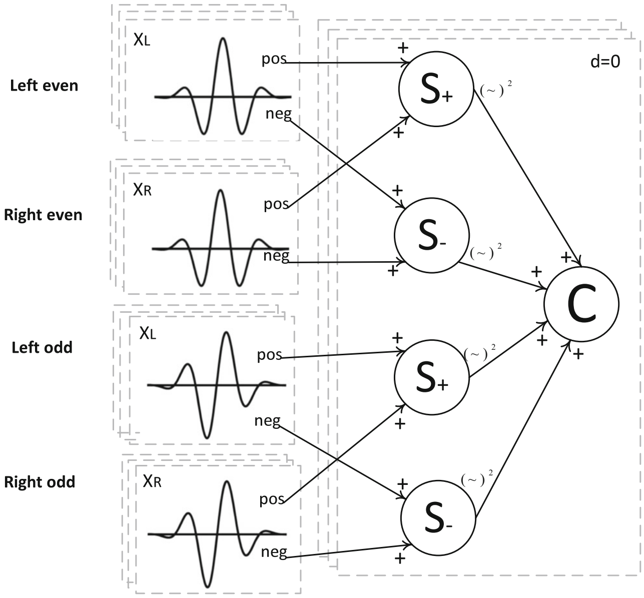


Figure 2 Disparity-energy model [19] showing simple (S) and complex (C) cells.

of 4xN simple cells and 1xN complex cells that combine the Gabor responses from the left and right images to detect one disparity level. A disparity group tuned to detect disparity level d merges Gabor responses from pixel positions from left and right images, having a relative horizontal distance of d. The following sections explain each of the filters in more detail.

2.1.1 DOG Filtering

Before being fed into the disparity-energy model, the image is first pre-processed by the retinal cells that have a concentric center-surround receptive field (RF), i.e. a positive center and negative surround (ON-center) or a negative center and positive surround (OFF-center). Figure 3a presents an ON-center DOG RF. A DOG cell fires a positive response when a stimulus falls in the center region and fires a negative response when a stimulus falls in the surround region. This behavior allows for edge enhancement to ease orientation selectivity in the simple cells stage.

The Difference of Gaussian (DOG) filter [17] approximates the 2D *Mexican hat wavelet* given by

$$\psi(x, y) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}. \qquad (1)$$

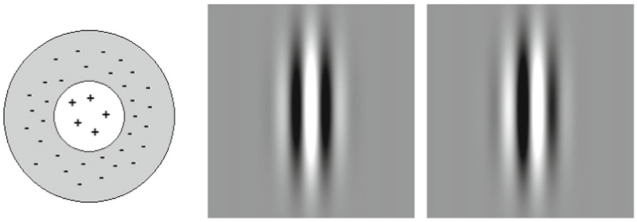
2.1.2 Receptive Fields (RFs) of Simple Cells

Receptive fields of simple cells are located in the left and right eyes. The pre-processed images are fed to these orientation selective spatial RFs, which can be approximated using Gabor filters. 1D Gabor filters are shown below:

$$G_{\text{even}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \cos\left(2\pi w_0 x\right), \qquad (2)$$

$$G_{\text{odd}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \sin(2\pi w_0 x). \qquad (3)$$

Phase of the even (2) and odd (3) Gabor filters differs by 90 degrees as shown in Fig. 3b and c. These receptive fields are tuned to respond to edges of a specific orientation and filter out the rest.



(a) DOG RF (b) Gabor even RF (c) Gabor odd RF

Figure 3 Receptive fields.

2.1.3 Simple and Complex Cells

The even and odd Gabor RF responses from each eye converge onto binocular simple cells. The complex cell C^d then takes the even and odd binocular cell responses and squares and adds them [23]:

$$\begin{aligned} C^d(x_L, x_R) &= \left(G_{\text{even}+}\left(x_L + \frac{d}{2}\right) + G_{\text{even}+}\left(x_R - \frac{d}{2}\right)\right)^2 \\ &+ \left(G_{\text{even}-}\left(x_L + \frac{d}{2}\right) + G_{\text{even}-}\left(x_R - \frac{d}{2}\right)\right)^2 \\ &+ \left(G_{\text{odd}+}\left(x_L + \frac{d}{2}\right) + G_{\text{odd}+}\left(x_R - \frac{d}{2}\right)\right)^2 \\ &+ \left(G_{\text{odd}-}\left(x_L + \frac{d}{2}\right) + G_{\text{odd}-}\left(x_R - \frac{d}{2}\right)\right)^2. \qquad (4) \end{aligned}$$

In Eq. 4, x_L and x_R are the horizontal pixel positions for the left and right eye respectively. $G_{even+}(x) = G_{even}(x)$ when $x > 0$ and 0 otherwise. $G_{even-}(x) = G_{even}(x)$ when $x < 0$ and 0 otherwise. Gabor odd functions are defined similarly. There are two ways of encoding disparity in the disparity-energy model: *position* shift and *phase* shift [22]. In this work, we only use position shift, where d is defined by the difference in position of the receptive field.

In our implementation, we use the original disparity-energy model introduced in [19], see Fig. 2, which uses simple cells that cannot fire negative responses. Hence, this model has four simple cells where each pair handles the positive and negative streams separately. Some other work such as [22, 23] used an equivalent model where the simple cell handles both positive and negative numbers, reducing the number of simple cells to two.

2.2 Stochastic Computing

In SC, numbers are represented using random bit streams. We denote by P_x the probability that a bit in the stream is a 1. A *unipolar* SC encoding can represent a value $x \in [0, 1]$ by choosing $P_x = x$ and a *bipolar* encoding can represent a value $x \in [-1, 1]$ by choosing $P_x = (x + 1)/2$. For example, a stream of 8 bits in unipolar encoding will represent $x = 0.5$ when 50 % of the bit stream is composed of ones and 50 % of zeros. SC allows for very simple circuit implementations of complex functions [6, 13]. As an example, Fig. 4 shows the multiplication of two independent stochastic streams using an AND gate and scaled addition using a multiplexer. Since SC cannot represent numbers greater than 1, a multiplexer scales the output by 2 to represent the sum as a stochastic stream.

Typically, numbers in conventional binary encoding are converted to stochastic streams by comparing the number

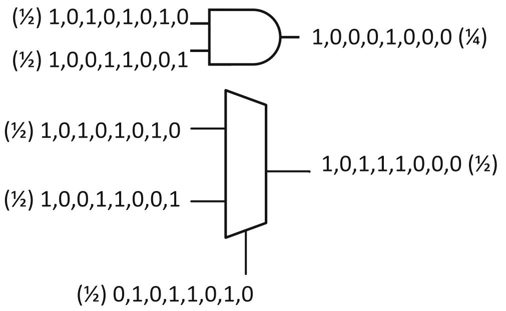


Figure 4 SC multiplication and scaled addition.

with a sequence of uniformly-distributed random numbers, see Fig. 5. Linear feedback shift-registers (LFSRs) are usually used to generate the sequence of random numbers due to their ease of design, but any other source can be used. To convert 0.3 into a unipolar stochastic stream with 5-bit equivalent unsigned fixed-point precision, we use a 5-bit LFSR to generate the $2^5 - 1$ uniform random sequence. 0.3 in unsigned number representation is “01001”. At each clock cycle, a comparator compares “01001” with the LFSR output and generates a stream where $P_x = 0.3$. Such streams are then processed using stochastic circuits (e.g. as in Fig. 5) and converted back to binary numbers using counters that accumulate the bit stream for $2^5 - 1$ cycles [6].

Since SC uses bit streams where the position of a bit in the stream is inconsequential, it allows for graceful degradation in the presence of errors, in contrast to binary circuits [13]. This ability can be exploited by allowing errors in transistor operation, for example by voltage overscaling, to potentially further minimize power consumption.

In recent years, the field of SC has progressed rapidly and has demonstrated several practical implementations for applications ranging from LDPC decoders [12, 24], to image processing [2, 13] and neural network implementations [6]. These applications generally achieve good performance with relatively low-precision at corresponding stochastic stream lengths. However, a wider range of applications, especially ones requiring multiple stages with large number of parallel computations suffer from long latencies due to high precision requirements. Stochastic multiplication is very area efficient and precise whereas stochastic scaled addition is not. For every two terms added, due to the scaling involved in multiplexer based addition, precision is severely reduced; leading to rapid precision loss when many numbers need to be added in the stochastic domain. This is usually compensated by very long stream lengths thereby increasing latency and energy consumption. For example, [16] implements a 5x5 Gaussian convolution on an image using standard multiplexer based SC addition which we estimate required stream lengths of at least $2^{16} - 1$ as per the 16-bit precision requirement stated for the binary digital system.

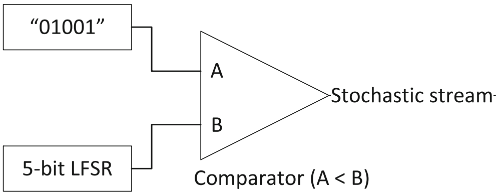


Figure 5 Stochastic number generator (SNG) for $x = 0.3$.

Several approaches have been explored in literature to improve the precision of SC addition. Chang and Parhi [7] explore a novel multiplexer based SC addition technique that relies on scaling the output with the total sum of input coefficients instead of a standard scaling factor of 0.5. Reference [25] presents a hard-wired weighted average (HWA) adder for FIR filters, that duplicates a specific input to several ports of the multiplexer depending on the weight of the input coefficient. The bundleplexer method from [8] uses a novel wire selection scheme similar to [25], but using only wires leading to significant energy savings. The modified signed stochastic adder designed in [14] defines the desired scaled adder behaviour in a truth table and derives the circuit. This method achieves exact results for a scaled adder by adding the bit streams deterministically.

However, a majority of these multiplexer-based designs inherently scale, reducing precision as large number of inputs are added. Further, in contrast to fixed-point arithmetic, multiplexer-based stochastic adders are much larger than stochastic multipliers, causing designs built using stochastic adders to have a large area-delay product. Rather than using a multiplexer-based design, we chose the SC exponential compression method presented in [11]. This method allows an arbitrary number of terms to be added with less precision loss, significantly reducing the required length of the stochastic stream while supporting a low area-delay product. We expand upon the method in Section 3 to improve its accuracy and thereby enable the design of low latency multi-stage stochastic circuits. One important tool that enables this result is the use of correlation in SC computations.

2.2.1 Correlation

Traditionally, stochastic circuits rely on uncorrelated bit streams to perform computations. Two streams are uncorrelated when two independent random number sequences are used to convert two inputs into stochastic streams. Similarly, two streams are maximally (positively) correlated when the same random number sequence is used to convert two different inputs into stochastic streams.

As correlation increases between streams, the accuracy of most SC circuits decreases and hence they are usually carefully designed to avoid it. But, [1] discovers that

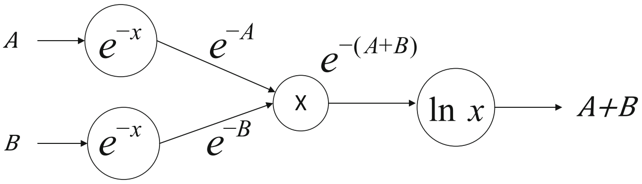


Figure 6 Exponential compression illustration.

changing correlation actually changes the stochastic behaviour of logic circuits and introduces a novel framework to analyze and quantify this correlation to design SC circuits that are small and accurate. As an example, the same AND gate which performs SC multiplication with uncorrelated streams, behaves as a minimum function when streams are positively correlated. Of interest is the XOR gate which implements an absolute-valued subtraction with positively correlated inputs [1]. This behaviour is exploited in Section 3 to improve accuracy of the SC exponential convolution architecture.

3 Stochastic Computing Convolution Architecture

The exponential compression method illustrated in Fig. 6 transforms the stochastic streams of interest using an exponential function, such that additions become multiplications [11]. The $\exp(x)$ and $\ln(x)$ functions are approximated using Taylor series expansions.

To implement the filters presented in Section 4, we must evaluate

$$z = \sum_i a_i x_i, \tag{5}$$

where a_i are the coefficients and x_i the system inputs. The corresponding stochastic circuit is shown in Fig. 7.

We assume 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. Equation 5 can be rewritten as

$$z = \sum_i a_i^+ x_i - \sum_i a_i^- x_i = \sum_i y_i^+ - \sum_i y_i^- = z^+ - z^-. \tag{6}$$

We also assume that $\sum a_i^- x_i \leq 2$ and $\sum a_i^+ x_i \leq 2$. Positive and negative summations have to be constrained to ensure the terms are within range in the stochastic domain. These summations can be greater than 1 because the first order Taylor approximation of $\ln(x)$ scales the result down by 2; so we can double the input coefficients and still stay within [0,1] range. Since these summations are calculated in the exponential domain, the compressed sum stays in the [0,1] range.

We first perform the multiplications in Eq. 6 using AND gates to obtain $y_i^+ = a_i^+ x_i$ and $y_i^- = a_i^- x_i$. To evaluate $\sum_i y_i^+$, we compress the individual streams $\{y_i^+\}$ as $\{e^{-y_i^+}\}$. The exponential compression can be approximated by a Taylor order 1, 2 or 3 approximation and the circuits are presented in [11].

The resulting set of compressed streams are multiplied together:

$$\prod e^{-y_i^+} = e^{-\sum y_i^+} = e^{-z^+}. \tag{7}$$

The $\{y_i^-\}$ streams are treated similarly. We obtain $z^+/2$ from e^{-z^+}, when we invert the exponential transformation by using a first order Taylor approximation of the natural logarithm:

$$z^+/2 \approx 1 - e^{-z^+}, \tag{8}$$

and similarly for $z^-/2$.

Now, instead of accumulating for $z/2$ with an up-down counter fed by Z^+ and Z^- (where $\mathbb{E}[Z^+] = z^+/2$ and $\mathbb{E}[Z^-] = z^-/2$), to allow further stochastic processing, we transform the streams into Z_{pos} and Z_{neg}:

$$Z_{\text{pos}} = Z^+(1 - Z^-),\ Z_{\text{neg}} = Z^-(1 - Z^+). \tag{9}$$

Note that $Z_{\text{pos}} + Z_{\text{neg}}$ is the exclusive-or of Z^+ and Z^-. When the streams Z^+ and Z^- are maximally correlated, the circuit with outputs Z_{pos} and Z_{neg} implements absolute valued subtraction. $\mathbb{E}[Z_{\text{pos}}]$ and $\mathbb{E}[Z_{\text{neg}}]$ become

$$z_{\text{pos}} \approx \begin{cases} \frac{z^+ - z^-}{2} & \text{if } z^+ > z^-, \\ 0 & \text{otherwise} \end{cases} \quad z_{\text{neg}} \approx \begin{cases} \frac{z^- - z^+}{2} & \text{if } z^- > z^+, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Finally, $\mathbb{E}[Z] = z/2 = z_{\text{pos}} - z_{\text{neg}}$. Note that z_{pos} and z_{neg} represent the positive and negative axis values respectively of $z/2$. The positive and negative streams never simultaneously take a non-zero value; a property that allows re-use of stochastic streams for further processing.

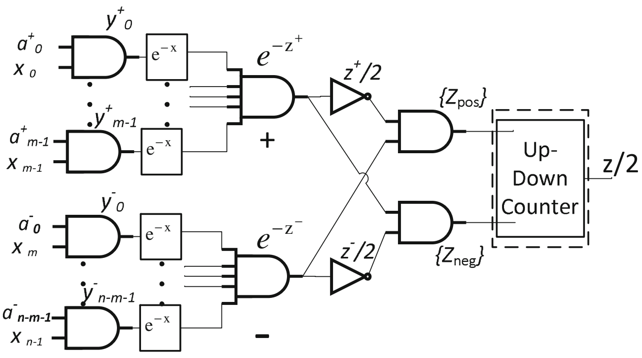
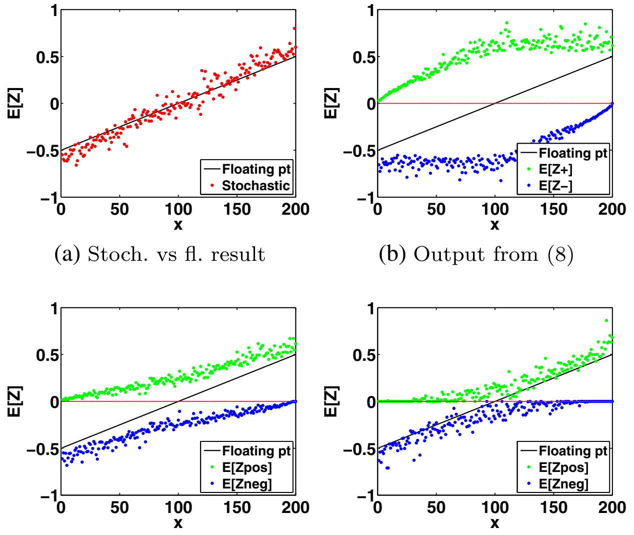


Figure 7 Exponential convolution architecture.

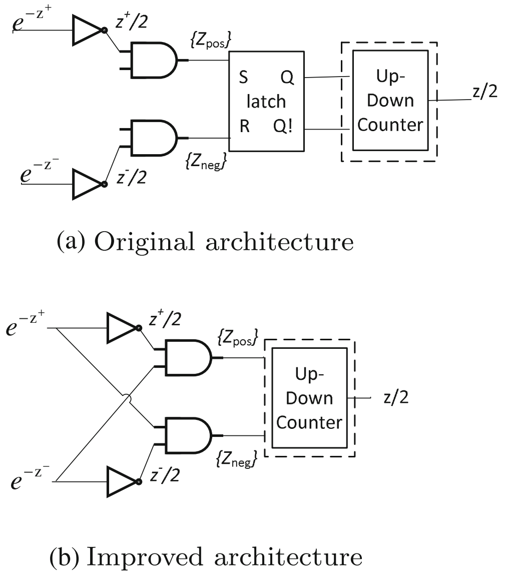
Figure 8 shows SC addition using exponential compression order 1 with stream length of $2^8 - 1$. The 8 inputs are



(a) Stoch. vs fl. result (b) Output from (8)

(c) Output from (10) without correlation (d) Output from (10) with correlation

Figure 8 8-input SC addition with stream length $2^8 - 1$.



(a) Original architecture

(b) Improved architecture

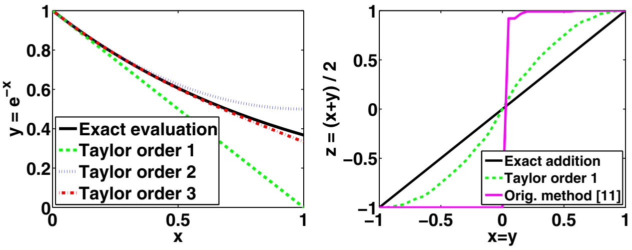
Figure 9 Differences in decompression architecture.

split into 4 positive and 4 negative pairs randomly generated and sum to an increasing value of Z. Figure 8a shows overlayed plots comparing the stochastic and floating-point results. Figure 8b shows the output of $\mathbb{E}[Z+]$ and $\mathbb{E}[Z-]$ from Eq. 8 before subtracting the streams from each other. As can be seen, E[Z] cannot be interpreted individually by either stream. After Eq. 10 with maximum correlation between Z+ and Z−, Fig. 8d shows E[Z_{pos}] and E[Z_{neg}] becoming mutually exclusive values, representing the positive and negative axis values of z/2. Comparing Fig. 8d with Fig. 8c, we see that without employing correlation, values in Fig. 8c are not yet mutually exclusive. Maximal correlation can be arranged by using a single noise source to generate all the x_i streams, and the same noise source for each pair of (a_i^+, a_i^-).

Figure 10a plots the Taylor series expansions of order 1, 2, and 3 to the exponential function. Taylor order 1 is a poor approximation for large numbers, but is reasonably accurate for $|x| < 0.3$. As the expected value of the streams increases, higher order approximations can be used [11]. Figure 10b shows the result of two-number addition with values ranging from -0.5 to 0.5 for the proposed exponential compression method and the original method from [11], compared with exact addition. In this work, a first order expansion was sufficient to detect disparity. Accuracy of a 2-input adder can be improved by increasing the order of the $\exp(x)$ transformation only [5, 11]. Please refer to [5] for a discussion on the area overhead incurred as different Taylor order approximations are used for $\exp(x)$ while implementing an SC system using the exponential compression technique. Note that a first-order Taylor approximation simply corresponds to using an OR gate as an SC adder.

Note that our proposed implementation of exponential compression is different from [11], as we do not include an S-R latch at the output (see Fig. 9). It can be seen from the plots in Fig. 10 that using the S-R latch leads to less precise addition. The hold state of the S-R latch quickly saturates the stream towards 0 or 1, disallowing representation of intermediate numbers. As a result, we have reformulated the mathematical description from [11] to yield a better approximation.

To summarize, we improved the exponential compression technique by removing the S-R latch, increased accuracy by exploiting correlation and used the separated positive and negative streams to allow for multi-stage stochastic architectures.



(a) Taylor approx. for e^{-x} (b) Approx for $z = \frac{x+y}{2}$

Figure 10 Taylor approximations.

4 System Architectures

A previously proposed system architecture presented in [5], referred herein as the 1D system architecture, used 1D Gabor filters to mimic the receptive fields for simple cells. These are weakly orientation selective compared to 2D Gabor filters. The previous hardware implementation could detect vertical edges only, and it is difficult to change the orientation of the 1D convolver to detect arbitrary angles.

To address these issues, we improved the architecture in several ways. Firstly, 1D 1x9 Gabor filters are upgraded to 2D 7x7 Gabor filters to enhance orientation selectivity and make the architecture reusable. Secondly, Gabor odd filters are removed and its response is approximated using a method of pixel subtraction [23] to save area.

4.1 Generalized Fully Parallel System Architecture

The proposed architecture implements a three-layer fully parallel 1x100 pixel processing circuit to detect 5 different disparity levels: $[-8, -4, 0, 4, 8]$. This architecture was inspired by [23] but we instead implement the system using stochastic computing. The 2D stochastic processing architecture is shown in Fig. 11.

The first stage, concentric center-surround RF, is implemented with a 2D DOG filter using a 5x5 kernel, with coefficients generated using Eq. 1 with $\sigma = 0.75$, $x = [-2, 2]$ and $y = [-2, 2]$. The second stage, orientation-selective Gabor 2D even filter coefficients are generated using:

$$G_{\text{even}}(x, y) = \exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right) \cos\left(X\frac{2\pi}{\lambda}\right), \qquad (11)$$

where $X = x\cos\theta - y\sin\theta$, $Y = x\sin\theta + y\cos\theta$, $\lambda = 3.5$, $\sigma = 2.8$, $\gamma = 0.3$, $\theta = 0$, $x = [-3, 3]$ and $y = [-3, 3]$ [18] and the coefficients are scaled as per Section 3.

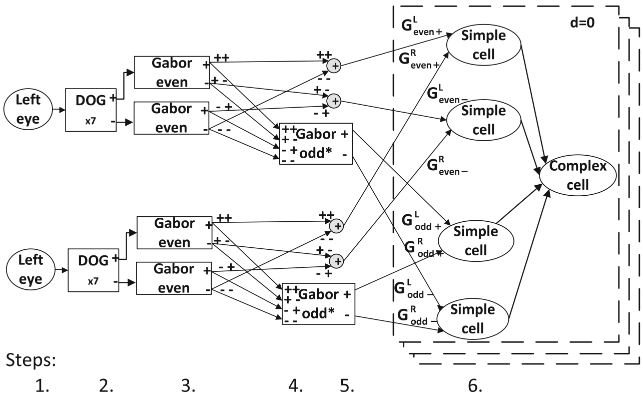


Figure 11 System diagram for generalized stochastic implementation.

The filtering stages are implemented using the stochastic exponential compression technique. As a result, each filtering stage takes a single stochastic stream as input and outputs a positive and a negative stream. The processing can be decomposed into the following steps (step numbers are also shown in Fig. 11):

1. The 1x100 pixel processing system takes 11 consecutive rows of 110 pixels shifted by 1 every clock cycle.
2. There are 7 2D 5x5 DOG filters processing 11 rows in parallel. The DOG filters then output a 7x106 pixel stream to 2D 7x7 Gabor even filters.
3. There are 2 Gabor even 1x100 filters implemented in parallel for each eye to process the positive and negative streams from the DOG filters separately.
4. An approximating circuit then takes the Gabor even processed streams to calculate the Gabor odd response without needing another 2D convolution.
5. The resulting streams are merged to have single positive and negative streams for all 1x100 pixels.
6. The streams finally feed into the last stage of the circuit shown inside the dashed box, that contains binocular simple and complex cells tuned to detect one disparity level. Overall, the single pixel processing stochastic circuit has 5 disparity groups (not shown) to detect the 5 disparity levels in parallel.

4.1.1 Gabor 2D Even Filters

The positive and negative 7x106 stochastic streams from the stochastic DOG filters are processed separately using unipolar stochastic Gabor even filters. After this second stage of filtering, each processed pixel is separated into four separate streams that we denote $++$, $--$, $+-$ and $-+$. We combine $++$ with $--$ into a first stream called even+, and $+-$ with $-+$ into a second stream called even−. The even+ and the even− stream represent respectively the positive and negative axis response for the Gabor even filtered pixel.

Gabor even and odd filters tuned to 90 degrees respond maximally to vertical edges and show weakened response to increased angular orientation. To test the system, bars oriented at angles varying from 0 to 180 degrees were presented. We summed the absolute value of the pixels forming the processed bar as a measure of the filter's response to various orientations. These results are normalized over all responses and presented in Fig. 12 for stochastic implementation with stream lengths of $2^5 - 1$ bits. The stochastic results are compared with a 2D floating-point, 6-bit fixed-point and the 1D floating-point [5] implementations. 6 bits is the minimum possible bit length where most of the filter coefficients can still be represented.

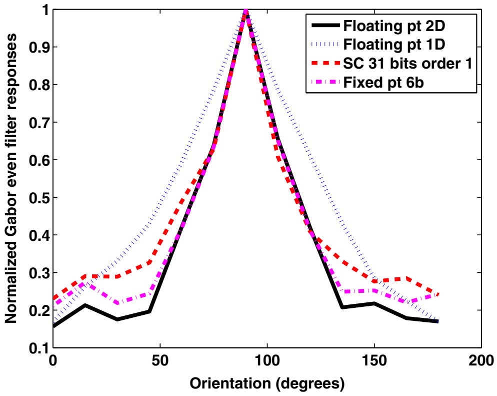


Figure 12 Orientation selectivity curves for 2D Gabor even responses.

All implementations can be seen to show similar and sharper orientation selectivity compared to the 1D counterpart.

Using 2D Gabor filters brings the implementation a step closer to a configurable system that can detect arbitrary orientations by simply swapping different Gabor kernels without any hardware changes.

4.1.2 Gabor 2D Odd Filters

A Gabor odd response can be derived from a Gabor even response by subtracting alternate pixels row-wise, processed by Gabor even filters [23].

$$G_{\text{odd}}(i) = \frac{1}{2}(G_{\text{even}}(i+1) - G_{\text{even}}(i-1)) = \frac{1}{2}(b-a) \quad (12)$$

This avoids the parallel instantiation of another 2D convolver, saving the extra area and energy required.

Figure 13 compares the cross-section of a 7x7 Gabor odd filter with an approximated version achieved using pixel subtraction. The approximated response is very close to the true response.

Expanding Eq. 12 using the 4 stream pixel representation (++,–, +- and -+) for a and b:

$$\begin{aligned} G_{odd}(i) &\\ &= \frac{1}{2}((b^{++} + b^{--} - b^{+-} - b^{-+}) - (a^{++} + a^{--} - a^{+-} - a^{-+})) \\ &= \frac{1}{2}((b^{++} + b^{--} + a^{+-} + a^{-+}) - (a^{++} + a^{--} + b^{+-} + b^{-+})) \end{aligned} \quad (13)$$

When the exponential compression technique is used to add the 8 stochastic streams, we get

$$G_{odd}(i) \approx \frac{1}{2}G^{+}_{odd}(i) - \frac{1}{2}G^{-}_{odd}(i). \quad (14)$$

Note that the output is naturally scaled by half due to the nature of exponential compression addition.

To test the approximation's effectiveness in orientation selectivity, the system was similarly presented with bars at varying angles and the volume of the processed bar measured. Figure 14 presents orientation selectivity of all the approximated Gabor odd filters. The filtering performance of the stochastic design is better than the 1D floating-point performance.

Note that the stochastic response is weaker than the 2D fixed-point implementation unlike the Gabor even response in Fig. 12. Since we approximate the Gabor odd response from the Gabor even response, this becomes the third stage of stochastic processing where correlation between streams becomes a strong factor in determining the sharpness of orientation selectivity. However, the approximated 2D response is still stronger than the 1D floating-point counterpart. It is also useful to note that the pixel subtraction technique can be used for any orientation of a Gabor filter.

4.1.3 Simple and Complex Cells

The pixel streams finally converge onto simple cells. Whereas simple cells in Eq. 4 add filtered responses

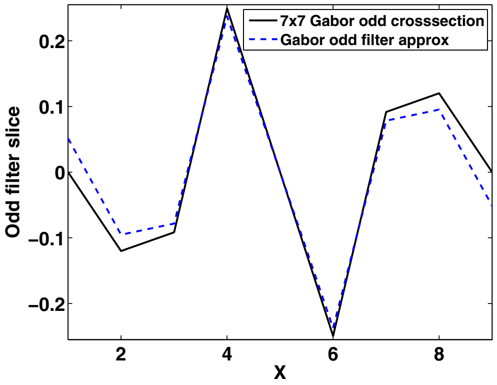


Figure 13 Gabor odd filter using Pixel subtraction.

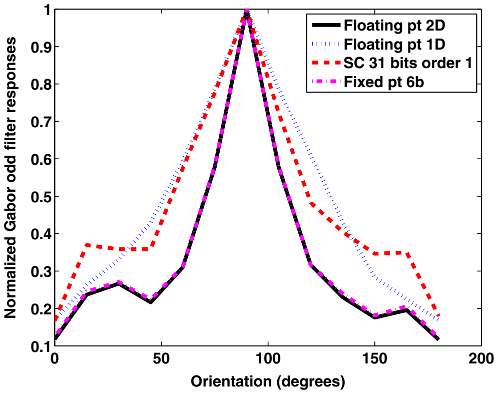


Figure 14 Orientation selectivity curves for 2D Gabor odd responses.

from each eye together before squaring them, the stochastic implementation presented here multiplies them before squaring:

$$C^d(x_L, x_R) = \left(G_{\text{even}+}\left(x_L + \frac{d}{2}\right) \cdot G_{\text{even}+}\left(x_R - \frac{d}{2}\right)\right)^2 + \left(G_{\text{even}-}\left(x_L + \frac{d}{2}\right) \cdot G_{\text{even}-}\left(x_R - \frac{d}{2}\right)\right)^2 + \left(G_{\text{odd}+}\left(x_L + \frac{d}{2}\right) \cdot G_{\text{odd}+}\left(x_R - \frac{d}{2}\right)\right)^2 + \left(G_{\text{odd}-}\left(x_L + \frac{d}{2}\right) \cdot G \quad _{\text{odd}-}\left(x_R - \frac{d}{2}\right)\right)^2. \tag{15}$$

This modification allows for better disparity detection than the original model. The functionality of the binocular simple cell is to respond maximally when two converged streams have a positive response, and show an inhibitory response when those streams are mismatched, as modeled by the sum of squares in Eq. 4. We propose to replace the sum with a multiplication as in Eq. 15 because when there are mismatched responses, one of the streams will be 0 and multiplication by 0 eliminates this failed response. This acts as boolean checking of bit streams and helps to provide a positive response only for the disparity the simple and complex cells are tuned for. While the original disparity-energy models (i.e. based on Eq. 4) with 2 and 4 simple cells are equivalent [22, 23], we found empirically that the modified disparity-energy model (based on Eq. 15) must use the 4 simple cell model for stricter disparity detection.

To evaluate the effectiveness of disparity selectivity, the system was presented with impulse inputs. The responses measured from the complex cell tuned to d=0 are presented in Fig. 15. The impulse input response between the left and right eyes had a varying disparity difference from -12 to 12 pixels. Disparity selectivity using Eq. 15 is better for the stochastic implementation for bit stream lengths of 31 than the floating-point and equivalent fixed-point implementations using Eq. 4. Compared to the stochastic implementation, truncation and quantization noise highly weakens disparity selectivity for the fixed-point implementation. In post-processing, though all implementations use a thresholding function to distinguish strong responses, the digital implementations require a variable threshold value to distinguish from the failed responses, which varies based on the stimuli. Due to the nature of ANDing in the stochastic implementation, a low fixed threshold value is usually sufficient to filter out the failed responses. Furthermore, we simplify the implementation of additions after Gabor filters and within the complex cells using OR gates because the stream densities are low enough for addition.

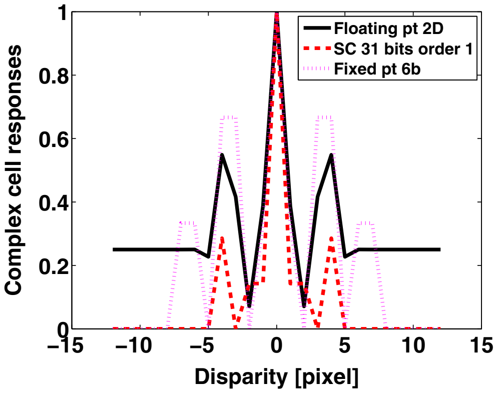


Figure 15 Disparity selectivity curves for complex cell, d=0.

4.2 Implementation Details

In the stochastic system, a simple cell is comprised of 2 AND gates and a 1-bit register. One AND gate is for multiplication of the converged responses, and the other for squaring; by ANDing the output and its delayed response from the 1-bit register together. The complex cell is implemented using an OR gate because stream density is low.

We tried implementing both a fully parallel stochastic architecture (which uses 7 DOG filters) and a partially parallel system (which uses 1 DOG filter , SIPO (Serial-In-Parallel-Out) buffer to feed data serially and conversion circuits (LFSRs, comparators, counters) to switch between stochastic and fixed-point representations). These results have confirmed that the fully-parallel implementation is more energy efficient for an architecture using taylor order 1 approximation in the SC exponential compression convolvers.

4.2.1 Noise Generation

Since the 2D SC design employs larger convolutions compared to the 1D architecture and passes through 5 stochastic stages instead of 4, to achieve good performance with a stream length of 31, a more extensive noise analysis was required. Seed analysis was important to ensure the computations were not suffering due to poor stochastic conversion and unintended correlation between streams.

Not all configurations of an n-bit LFSR generate good, unique random sequences [3]. We used 2 XNOR positions on the 5-bit LFSR which generated a sequence with numbers 0 to 30 represented at least once. A different starting seed can also be used for the same LFSR to generate a time-shifted sequence, allowing us to use the same LFSR configuration to convert two numbers into two stochastic streams with less correlation between them. We generated the needed noise using the 2 LFSR configurations and different starting seeds.

Images were converted using 1 LFSR and parallel instantiations of comparators. Even though individual pixels will be maximally correlated between each other, once multiplied with the filter coefficients, the streams were found to be sufficiently random to perform the computations. Furthermore, using multiple LFSRs to generate the pixel streams hurt the system by adding salt and pepper noise in the processed image due to the random biases added from different stochastic streams. DOG and Gabor even filter coefficients are generated as per Section 3 to maximize correlation.

The outputs from the two Gabor even filtering blocks are correlated relative to each other. We reduce correlation by delaying the Gabor even coefficient stream by one cycle for one of the Gabor even fiters. This trick helped sharpen the orientation selectivity curve for the Gabor odd filter stage.

We used one pair of left and right training images to optimize the seeds used in the stochastic system. The optimization is performed by simulating the system with random seed values, and selecting the values that minimize the error on the training image pair.

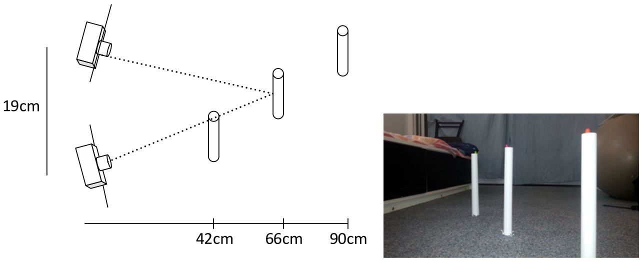
4.3 Fixed-point Architecture

For comparison, a conventional digital system using the original disparity-energy model given by Eq. 4 was implemented using 6-bit signed fixed-point number representation. This precision corresponds to a $2^5 - 1$-bit unipolar stochastic implementation. We have chosen not to use the modified model for fixed-point implementation because the multiplications in Eq. 15 would yield a larger circuit. The input coefficients are identical to the stochastic system except that they have been scaled to avoid overflow.

The architecture is partially parallel using 1 DOG filter instead of 7 and some memory to buffer the past 6 rows necessary for the Gabor filter layer; to reduce area. The memory unit will be a shifting Serial-In-Parallel-Out buffer, implemented using registers. The memory required will be 6 rows x 106 pixels x 6 bits.

DOG and Gabor even filtering stages use 6-bit multipliers and the 12-bit output truncated to 9 and 10 bits for addition respectively. The output from these stages is truncated to 6 bits and passed to the next stage. Gabor odd approximating circuit uses 7-bit adders and scales the output back to 6 bits. Simple cells use 7-bit adders and 7-bit multipliers. Complex cells use 8-bit adders providing an 8-bit unsigned output for the entire system.

When signed number representation is used, the model in Fig. 2 is simplified to use two simple cells only since we use the original disparity energy model here [22]. The adder and multiplier sizes have been carefully chosen to ensure it is the smallest possible design for the system to work and that the dynamic range of the system is preserved.



(a) Experimental setup (b) Right camera view

Figure 16 Experiment to detect disparity.

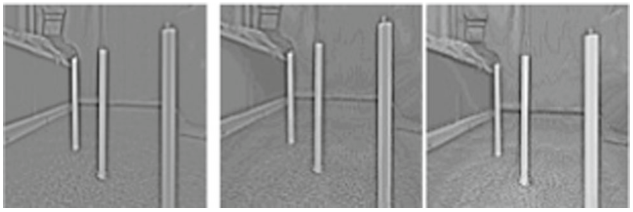
5 Results

5.1 Experiment Setup

To detect the depth of objects, an experiment was setup similar to [23] as shown in Fig. 16a. The cameras were setup 19 cm apart with an 8 degree angle from the vertical. As a result, the fixation point (the point at the intersection of the line of sight of each camera) is 66 cm away. At this range, disparity corresponds to around 3 cm per pixel. We place one white pole on the fixation point. To detect disparities of -8 and +8, two white poles were placed at a distance of 42 and 90 cm respectively from the cameras center. The poles were then adjusted appropriately to derive images with the right disparity to account for the error in the setup. These poles sit on a circle with a 60 cm diameter.

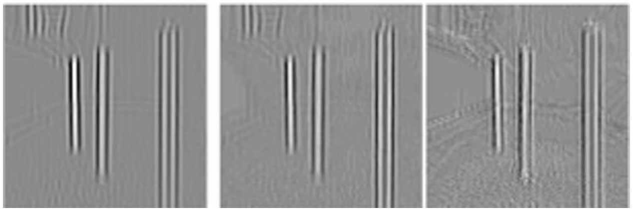
5.2 Filter Outputs

Figures 17, 18 and 19 show the Difference-of-Gaussian (DOG), Gabor even and Gabor odd filtered responses from the right eye image for different implementations. Also, the images are normalized and streams accumulated using counters. Visually, all implementations show similar behaviour. Here, the SC order 1 visual performance with stream lengths of 31 is closer to the fixed-point filter output. For the 1D architecture from [5], at least $2^8 - 1$ stream length with order 2 was needed to see similar visual behaviour between SC and fixed-point filter outputs.



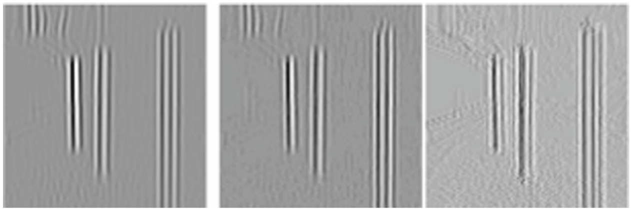
(a) Floating-point (b) Fixed-point 6b (c) SC 31b order 1

Figure 17 DOG filtered outputs.



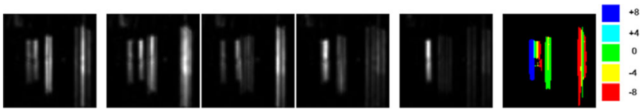
(a) Floating-point (b) Fixed-point 6b (c) SC 31b order 1

Figure 18 Gabor even filtered outputs.



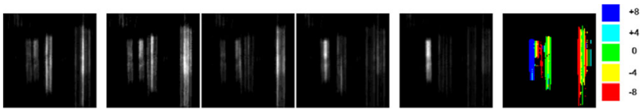
(a) Floating-point (b) Fixed-point 6b (c) SC 31b order 1

Figure 19 Gabor odd filtered outputs.



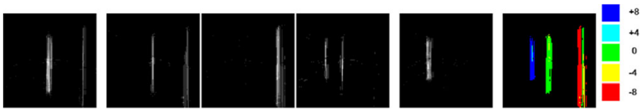
(a) d=0 (b) d=-4 (c) d=-8 (d) d=4 (e) d=8 (f) all-inc (g)

Figure 20 Disparity maps (floating-point).



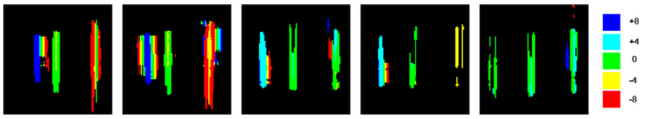
(a) d=0 (b) d=-4 (c) d=-8 (d) d=4 (e) d=8 (f) all-inc (g)

Figure 21 Disparity maps (fixed-point 6 bits).



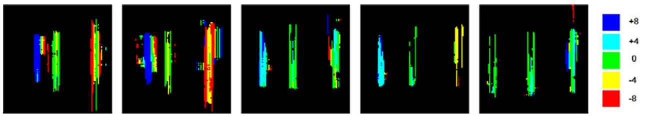
(a) d=0 (b) d=-4 (c) d=-8 (d) d=4 (e) d=8 (f) all-inc (g)

Figure 22 Disparity maps (Improved stochastic system with SC length = 31).



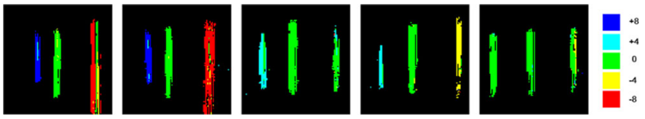
(a) 8,0,-8 (b) 8,0,-8 (c) 4,0,0 (d) 4,0,-4 (e) 0,0,0 (f)

Figure 23 Floating-point disparity results from different image pairs.



(a) 8,0,-8 (b) 8,0,-8 (c) 4,0,0 (d) 4,0,-4 (e) 0,0,0 (f)

Figure 24 Fixed point 6b disparity results from different image pairs.



(a) 8,0,-8 (b) 8,0,-8 (c) 4,0,0 (d) 4,0,-4 (e) 0,0,0 (f)

Figure 25 Improved stochastic system SC 31b results from different image pairs.

Table 1 Average error for different implementations.

Type	Eqn. used	Threshold	Average Error % (min, mean, max)	# Trials
Floating-pt	4	0.27	14.6	1
Fixed-pt 6b	4	7	16.4	1
Mod. floating-pt	15	0.0013	12.7	1
Stoch. 31b	15	1	(8.9, 14.2, 22)	30

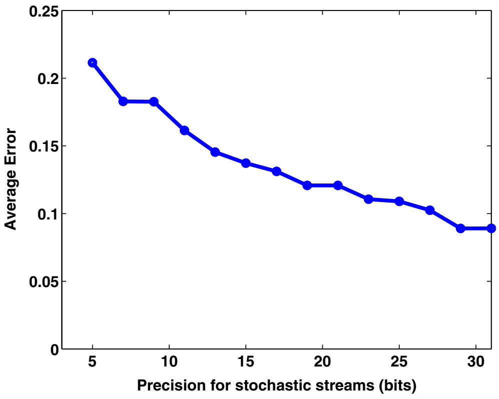


Figure 26 Effect of stream length on average error.

Table 2 Synthesis results without interface circuitry.

Instance (1x100)	Logic cell count	Total area [mm^2]	Delay [ns]	Area ratio	ADP [mm^2 ·ns]	ADP ratio
SC ord 1	50.8k	0.308	5448	0.03	1679	0.285
Fixed pt.	1.37M	9.95	592	1	5886	1

5.3 Disparity Detection

Figures 20, 21 and 22 show the disparity maps for the floating-point, fixed-point and the improved stochastic 2D implementations. MATLAB post-processing merges the disparity maps into 1 image, $D(x, y)$ with color coded disparity regions based on the following equation from [23]:

$$D(x, y) = \arg\max_d C^d(x, y),\ d \in (-8, -4, 0, +4, +8), \tag{16}$$

where C^d is defined by original Eq. 4 or modified Eq. 15 complex cell. We use a threshold, T, to filter out failed responses. If $C^d(x, y) > T$, at each pixel position, the maximum value from the 5 disparity maps is taken and colored according to its disparity. [Blue, cyan, green, yellow, red] represent the disparities of [+8,+4,0,-4,-8].

The stochastic system provides better disparity detection using the modified Eq. 15 than fixed or floating-point systems using the original Eq. 4.

To better quantify the error, we obtained 4 additional image pairs with poles at different disparities using a similar setup and manually created ideal disparity maps depending on the position and dimensions of the poles from the right and left images to estimate the error. The error metric weighs missing/incorrect pixels where a pole is expected by 3x compared to misclassification where a pole is absent. This weighting is necessary to worsen the error of the null system; that is a system that fails to detect any of the predefined detectable disparities. The null system performs relatively well with an un-weighted error metric because of the large empty regions in the images. This weighting allows the error metric to give more importance to the misclassification of an object's disparity than to incorrectly assigned disparities in empty regions.

The floating-point, fixed-point and stochastic results are presented in Figs. 23, 24 and 25. Error reported in this paper is the average error over the 5 image pairs.

Table 1 presents the average error for various implementations. Average error for the 5 merged disparity map outputs from Figs. 23 and 24 are shown as "Floating-pt" and "Fixed-pt 6b". Also, included just in Table 1 are the average results over the 5 image pairs for a floating-point implementation that uses the modified Eq. 15 labelled "Mod. floating-pt". This result provides a point of comparison for the performance of the stochastic system, but note that a conventional system based on Eq. 15 would have a large implementation complexity because of the required multipliers.

For the stochastic result labelled "Stoch. 31b", the system is run on the 5 image pairs for 30 trials, where the start values for the LFSRs are randomized (but the LFSR configurations, i.e. XNOR positions used were fixed). The best, average, and worst result over the 30 trials are presented in Table 1. On average, the 31b stochastic system provides more accurate disparity maps than the floating and fixed-point implementations based on Eq. 4. We can improve the performance of the stochastic system by selecting the best performing seed configuration, as described in Section 4.2.1, yielding an improved 8.9 % error. The outputs of the system for the optimized seed configuration are shown in Fig. 25.

In general, it is important to optimize the seeds used for the stochastic system for good performance. However, because of the limited sets of images we are using as a benchmark, we want to be cautious in reporting the results because we cannot guarantee that the best-seed performance holds over a large class of images. So, Sections 5.4 and 5.5 report conservative area-delay product (ADP) and energy estimates that are based on the full 31b streams. Recall that with 31b streams, the stochastic system outperforms the conventional floating-point system not just for the best choice of seed, but also when the performance is averaged over the choice of seed. By measuring the performance in this way, we remove the possibility that the performance gain is the result of an overfitting of the system to the type of image found in the benchmark set.

Table 3 Synthesis results with interface circuitry.

Instance (1x100)	Logic cell count	Total area [mm^2]	Delay [ns]	Area ratio	ADP [mm^2·ns]	ADP ratio
SC ord 1	91.3k	0.639	5448	0.06	3453	0.57
Fixed pt.	1.38M	10.22	591.6	1	6049	1

Table 4 Power and energy without interface circuitry.

Inst.	Switch. Power (mW)	Leak. Power (mW)	Total Power (mW)	Power ratio	Energy (nJ)	Energy ratio
SC. 1	21.3	0.82	44.2	0.056	241	0.52
Fix-pt.	137.0	41.3	785	1	464	1

5.3.1 Effect of Latency on Average Stochastic Error

One advantage of stochastic systems is the ability to trade-off accuracy for reduced latency by shortening the length of the stochastic sequences.

The stochastic system with optimized seed values was simulated to detect disparity on the 5 image pairs as latency is decreased from 31 bits per computation to 5 bits. Figure 26 plots the error on the benchmark image set at different latencies. We are able to reduce the bits required per computation from 31 bits down to 11 and still match or perform better disparity detection than the fixed-point design. The error gracefully degrades from 8.9 % down to 16.14. %.

5.4 Synthesis and Performance Results

A 2D 1x100 architecture is synthesized using Cadence RC compiler for TSMC 65 nm CMOS technology and area results are reported here. Table 2 presents results for the circuits without interface circuitry included, while Table 3 presents results including them. The stochastic and fixed-point circuits operate with a clock period of 1.7ns and 5.5ns respectively. For the fixed-point design, interface circuitry includes the input and output registers. For the stochastic design, this 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. The area and ADP measure of the stochastic implementation are very small compared to the fixed-point implementation even if interface circuitry is included.

The stochastic results in Tables 2–5 assume a stream length of 31 bits. Recall that such a stream length allows outperforming the floating-point system even when the performance is averaged over the seed configurations. If instead the performance is measured by using the best seed value, the ADP ratio can be improved to 0.107 (without interface circuitry) or 0.214 (with interface circuitry) by reducing the stream length to 11 bits, which yields a performance of 16 %, just slightly better than the fixed-point system. The delay is also reduced to 2046 ns. Note that for low latency applications, we can increase the number of parallel instances of the circuit operating on the same pixel without affecting the ADP metric.

Typical SC systems require longer run-times than 2^n to account for precision loss due to SC arithmetic to match an equivalent fixed-point system. We chose 6 bits for the signed fixed-point system since it is sufficient to get performance similar to the floating-point system. Note that the precision of the stochastic system could be increased by using larger LFSRs and counters in the interface circuitry, without changing the stochastic circuit.

5.5 Energy and Power Estimation

We used Synopsys PrimeTime to estimate power consumption of the circuit by providing realistic gate level switching activities from ModelSim along with parasitics information from Cadence RC Compiler tool. Tables 4 and 5 present the estimates for power and energy for the proposed circuit when the full 31-bit stochastic stream is used for all pixels. The stochastic circuit consumes lower power and energy than the fixed-point design. SC circuit order 1 consumes only 5.6 % of the power required for the fixed-point design without interface circuitry. Due to the longer latency of the stochastic system, energy consumption becomes 52 % of the fixed-point design, but is still much lower than the fixed-point design.

Overall energy ratios are usually higher for stochastic circuits when compared with the ADP measure. This is because switching activity tends to be higher in stochastic circuits than in fixed-point circuits. So, stochastic circuits tend to consume more energy than under fixed-point operation. However, stochastic circuits allow tuning latency for the accuracy required. Also, they are highly error resilient and will be able to function in highly unreliable conditions where greater energy savings might be possible.

The energy usage including interface circuitry is much higher when compared to the energy used by the stochastic

Table 5 Power and energy with interface circuitry.

Inst.	Switch. Power (mW)	Leak. Power (mW)	Total Power (mW)	Power ratio	Energy (nJ)	Energy ratio
SC. 1	36.0	1.8	154	0.19	839	1.73
Fix-pt.	138	41.7	818	1	484	1

Table 6 Stochastic design breakdown.

Instance	Logic cell count	Total area [mm^2]	Switching Power (mW)	Leakage Power (mW)	Total Power (mW)	Area ratio	Energy (nJ)	Energy ratio
Complete circuit	91.3k	0.634	36.0	1.80	154.0	1	839	1
Stochastic circuit	50.8k	0.308	21.3	0.82	44.2	0.49	241	0.29
Input Interface	30.9k	0.226	14.3	0.78	95.8	0.36	522	0.62
Output Interface	9.5k	0.580	0.33	0.20	14.0	0.09	78	0.09

circuit. To illustrate the inefficiency of using LFSRs to convert unsigned numbers to stochastic streams, Table 6 presents the breakdown of the resource usage of the stochastic design. While the Input Interface takes up only 36 % of the total area, it consumes 62 % of the energy. We have found that registers used in stochastic circuits tend to consume far greater energy, compared to their operation in a fixed-point design due to high switching activity. This might be addressed in the future by using sensors that provide the data directly in the form of stochastic streams, for example by building them from MTJ devices [20].

When the performance is measured by using the best seed values instead, the energy ratios can be improved to 0.20 (without interface circuitry) and 0.65 (with interface circuitry) by reducing the stream length from 31b to 11b (see Fig. 26).

6 Configurable Architecture

Using 2D Gabor filters allows us to bring the implementation a step towards reusability since 2D filters can be oriented in any direction. In the stochastic synthesized design, to optimize area, the Gabor filter coefficients are pre-sorted into positive and negative groups before synthesis before being presented to the exponential convolution architecture. This hardwires the implementation to the specific filter used here. However, if the coefficients are to be fully swappable similar to the original exponential convolution architecture [11], we need to use de-multiplexers in the exponential convolution architecture to choose between the positive and negative AND gate as presented in Fig. 27.

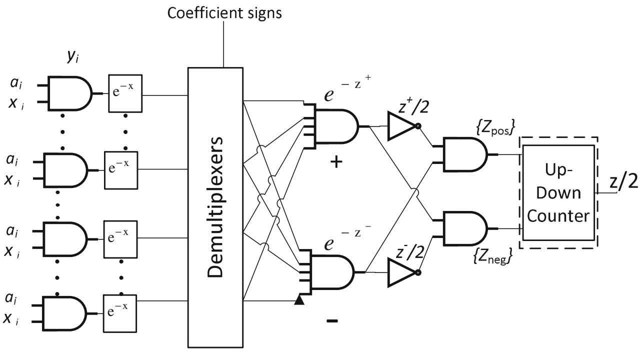


Figure 27 Configurable exponential convolution architecture [11].

The overall area increases by 32 %, but the energy increases only by 6 % as the added circuitry will not be switching during run-time for a specific filter.

7 Discussion and Conclusion

In this paper, we presented a generalized stochastic VLSI architecture and implementation of the disparity-energy model for depth perception. Using 2D filters allowed stronger orientation selectivity than 1D ones. Approximating the Gabor odd response allowed reduction in area. The circuit can be generalized to use swappable Gabor filters to detect edges with arbitrary angles.

The proposed fully parallel five-stage stochastic architecture using the modified disparity-energy model functions better at latencies of 31 bits compared to the floating-point and fixed-point 6-bit implementations using the original disparity-energy model. This was possible due to the improved SC exponential compression technique that performs large number of additions in the stochastic domain with minimal scaling loss. By exploiting correlation between streams, we minimized the number of LFSRs needed and improved accuracy of intermediate number representations. Separating and computing on the positive and negative streams in this unipolar architecture allowed the multi-stage feed-forward architecture with low scaling and accuracy loss. Further, a careful noise analysis allowed choosing good random number generators to generate stochastic numbers accurately with appropriate correlation choices where necessary. Conservatively, we achieve an area-delay product savings of 71 % and energy savings of 48 % compared to a reference fixed-point implementation. At the same error rate between the stochastic and fixed-point systems when the best-case seed values are used for LFSRs, ADP and energy savings increase to 89 % and 81 % respectively.

The inherent robustness of SC circuits can enable further energy savings by using the latest technology nodes or increasing unreliability; for example by reducing the supply voltage and allowing timing violations. Asynchronous design methods can also further reduce energy usage. Lastly, latency can be traded for accuracy, without extra hardware. These benefits serve to highlight the wide flexibility that SC systems offer, especially when considering the implementation of large scale neuromorphic models.

Since the proposed configurable architecture allows modifying the Gabor filter coefficients, a similar approach could be used to implement the HMAX based neuromorphic algorithms for object detection [18]. Future work will also use [21] to generate the Gabor filter coefficients directly in the stochastic domain for any orientation, without needing to store them and convert them to stochastic streams.

Acknowledgments The authors would like to thank Hasan Mozafari, Arash Ardakani and Xinchi Chen for useful discussions. Warren J. Gross is a member of ReSMiQ (Regroupement Stratégique en Microsystémes du Québec) and SYTACom (Centre de recherche sur les systèmes et les technologies avancés en communications). This work was supported by the Brainware LSI Project of MEXT (Ministry of education, culture, sports, science and technology), Japan.

References

1. Alaghi, A., & Hayes, J.P. (2013). Exploiting correlation in stochastic circuit design. In *IEEE 31st International Conference on Computer Design (ICCD)* (pp. 39–46): IEEE.
2. Alaghi, A., Li, C., & Hayes, J.P. (2013). Stochastic circuits for real-time image-processing applications. In *Design Automation Conference*.
3. Alfke, P. (1998). Efficient shift registers, LFSR counters, and long pseudo-random sequence generators. http://www.xilinx.com/bvdocs/appnotes/xapp052.pdf.
4. Anzai, A., Ohzawa, I., & Freeman, R.D. (1997). Neural mechanisms underlying binocular fusion and stereopsis: position vs. phase. *Proceedings of the National Academy of Sciences*, *94*(10), 5438–5443.
5. Boga, K., Onizawa, N., Leduc-Primeau, F., Matsumiya, K., Hanyu, T., & Gross, W.J. (2015). Stochastic implementation of the disparity energy model for depth perception. In *IEEE Workshop on Signal Processing Systems (SiPS)* (pp. 1–6).
6. Brown, B.D., & Card, H.C. (2001). Stochastic neural computation. I. computational elements. *IEEE Computer*, *50*(9), 891–905.
7. Chang, Y.-N., & Parhi, K.K. (2013). Architectures for digital filters using stochastic computing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2697–2701).
8. Chen, J., & Hu, J. (2013). A novel FIR filter based on stochastic logic. In *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 2050–2053).
9. Choi, T.Y.W., Merolla, P.A., Arthur, J.V., Boahen, K.A., & Shi, B.E. (2005). Neuromorphic implementation of orientation hypercolumns. *IEEE Transactions Circuits Systems I, Reg Papers*, *52*(6), 1049–1060.
10. Cumming, B.G., & Parker, A.J. (1997). Responses of primary visual cortical neurons to binocular disparity without depth perception. *Nature*, *389*(6648), 280–283.
11. Janer, C.L., Quero, J.M., Ortega, J.G., & Franquelo, L.G. (1996). Fully parallel stochastic computation architecture. *IEEE Signal Processing*, *44*(8), 2110–2117.
12. Leduc-Primeau, F., Gaudet, V.C., & Gross, W.J. (2015). Stochastic decoders for LDPC codes. In *Advanced Hardware Design for Error Correcting Codes* (pp. 105–128): Springer.
13. Li, P., Lilja, D.J., Qian, W., Bazargan, K., & Riedel, M.D. (2014). Computation on stochastic bit streams digital image processing case studies. *IEEE Trans Very Large Scale Integration (VLSI) System*, *22*(3), 449–462.
14. Li, Y., & Hu, J. (2013). A novel implementation scheme for high area-efficient dct based on signed stochastic computation. In *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 990?-993).
15. Liu, S.-C., Kramer, J., Indiveri, G., Delbrüeck, T., Burg, T.s., & Douglas, R. (2001). Orientation-selective aVLSI spiking neurons. *Neural Networks*, *14*(6), 629–643.
16. Ma, C., Zhong, S., & Dang, H. (2012). *High fault tolerant image processing system based on stochastic computing*.
17. McMahon, M.J., Packer, O.S., & Dacey, D.M. (2004). The classical receptive field surround of primate parasol ganglion cells is mediated primarily by a non-gabaergic pathway. *The Journal of neuroscience*, *24*(15), 3736–3745.
18. Mutch, J., & Lowe, D.G. (2008). Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision*, *80*(1), 45–57.
19. Ohzawa, I., Deangelis, G.C., & Freeman, R.D. (1990). Stereoscopic depth discrimination in the visual cortex: neurons ideally suited as disparity detectors. *Science*, *249*(4972), 1037–1041.
20. Onizawa, N., Katagiri, D., Gross, W.J., & Hanyu, T. Analog-to-stochastic converter using magnetic tunnel junction devices for vision chips. In *IEEE Transactions on Nanotechnology, 2016(to appear)*.
21. Onizawa, N., Katagiri, D., Matsumiya, K., Gross, W.J., & Hanyu, T. (2015). Gabor filter based on stochastic computation. *IEEE Signal Process Letter*, *22*(9), 1224–1228.
22. Qian, N. (1997). Binocular disparity and the perception of depth. *Neuron*, *18*(3), 359–368.
23. Shimonomura, K., Kushima, T., & Yagi, T. (2008). Binocular robot vision emulating disparity computation in the primary visual cortex. *Neural Networks*, *21*(2), 331–340.
24. Tehrani, S.S., Naderi, A., Kamendje, G.-A., Hemati, S., Mannor, S., & Gross, W.J. (2010). Majority-based tracking forecast memories for stochastic LDPC decoding. *IEEE Transactions Signal Processing*, *58*(9).
25. Wang, R., Han, J., Cockburn, B., & Elliott, D. (2015). Design and evaluation of stochastic FIR filters. In *IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing (PACRIM)* (pp. 407–412).

Kaushik Boga received the B.Eng. degree in Electrical Engineering and M.Eng. degree (thesis) in Computer Engineering from McGill University, Montréal, Canada in 2013 and 2016 respectively. In 2016, he joined Qualcomm, Canada as a Design Engineer working in the Digital Video Processing group. His research interests include approximate/ stochastic computing, neuromorphic computing and vision.

François Leduc-Primeau received the B.Eng. degree in 2007, the M.Eng. degree in 2010, and the Ph.D. degree in 2016, all in Computer Engineering from McGill University, Montréal, Canada. In 2016, he joined Telecom Bretagne (Institut Mines-Telecom), Brest, France, as a postdoctoral researcher. His research interests include error-correction codes, algorithms and systems for telecommunications and signal processing, and novel approaches for improving the energy efficiency of digital systems. Dr. Leduc-Primeau is a member of IEEE.

Naoya Onizawa 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. He is currently an Assistant Professor in Frontier Research Institute for Interdisciplinary Sciences at Tohoku University, Japan. He was a postdoctoral fellow at University of Waterloo, Canada in 2011 and at McGill University, Canada from 2011 to 2013. In 2015, he was a Visiting Associate Professor at University of Southern Brittany, France. His main interests and activities are in the energy-efficient VLSI design based on asynchronous circuits and probabilistic computation, and their applications, such as associative memories and brainlike computers. He received the Best Paper Award in 2010 IEEE ISVLSI, the Best Paper Finalist in 2014 IEEE ASYNC, 20th Research Promotion Award, Aoba Foundation for the Promotion of Engineering in 2014, and Kenneth C. Smith Early Career Award for Microelectronics Research in 2016 IEEE ISMVL. Dr. Onizawa is a Member of the IEEE.

Kazumichi Matsumiya received Ph.D. degree from Tokyo Institute of Technology in 2000. Then, he was a postdoctoral researcher at York University in Canada until December, 2001. From January, 2002 to December 2003, he was a postdoctoral researcher at Imaging Science and Engineering Laboratory, Tokyo Institute of Technology. From January, 2004 to March, 2005, he was a research fellow at Human Information Sciences Laboratory, ATR. He moved to Research Institute of Electrical Communication, Tohoku University as a research associate in April, 2005. He is currently working there as an associate professor. His research interests are in visual psychophysics, visuo-haptic integration, vision for action, and eye movements.

Takahiro Hanyu 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 in 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 VLSI processors, and post-binary computing and its application to brain-inspired VLSI systems. 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, Ichimura Academic Award 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. Dr. Hanyu is a Senior Member of the IEEE.

Warren J. Gross received the B.A.Sc. degree in electrical engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 1996, and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, Ontario, Canada, in 1999 and 2003, respectively. Currently, he is a Professor with the Department of Electrical and Computer Engineering, McGill University, Montréal, Québec, Canada. His research interests are in the design and implementation of signal processing systems and custom computer architectures.

Dr. Gross served as Chair of the IEEE Signal Processing Society Technical Committee on Design and Implementation of Signal Processing Systems. He has served as Technical Program Co-Chair of the IEEE Workshop on Signal Processing Systems (SiPS 2012) and as Chair of the IEEE ICC 2012 Workshop on Emerging Data Storage Technologies. Dr. Gross served as Associate Editor for the IEEE Transactions on Signal Processing and currently is a Senior Area Editor. Dr. Gross is a Senior Member of the IEEE and a licensed Professional Engineer in the Province of Ontario.