Stochastic Circuits for Real-Time Image-Processing Applications

Armin Alaghi, Cheng Li and John P. Hayes
Advanced Computer Architecture Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI, 48109, USA
{alaghi, elfchris, jhayes}@ umich.edu

ABSTRACT
Real-time image-processing applications impose severe design constraints in terms of area and power. Examples of interest include retinal implants for vision restoration and on-the-fly feature extraction. This work addresses the design of image-processing circuits using stochastic computing techniques. We show how stochastic circuits can be integrated at the pixel level with image sensors, thus supporting efficient real-time (pre)processing of images. We present the design of several representative circuits, which demonstrate that stochastic designs can be significantly smaller, faster, more power-efficient, and more noise-tolerant than conventional ones. Furthermore, the stochastic designs naturally produce images with progressive quality improvement.

Categories and Subject Descriptors
B.2 Arithmetic and Logic Structures, B.6 Logic Design, C.3 Special-Purpose and Application-Based Systems.

General Terms
Design.

Keywords
Emerging Technologies, Image Processing, Real-Time Computing, Stochastic Computing, Vision Chips.

1. INTRODUCTION
Advances in semiconductor technology have enabled many exciting new applications of embedded computers. They have also exposed problems and opportunities that cannot be easily addressed using conventional design approaches. An example that motivates our work is the provision of retinal implants for the visually impaired [15]. This involves designing an integrated circuit (IC) chip that can be surgically placed on a dysfunctional retina to sense images (or process images sent wirelessly from an external camera) and convert an array of pixel streams to streams of neural-style electrical signals that stimulate useful visual sensations. The implanted IC is linked to an external power supply and must not dissipate more than a few mW/mm^2 to avoid heat damage to the eye [17].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DAC'13, May 29–June 07, 2013, Austin, TX, USA.
Copyright 2013 ACM 978-1-4503-2071-9/13/05 …$15.00.

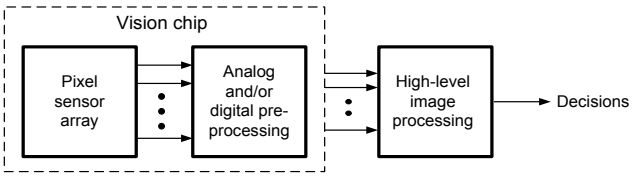

Figure 1. Image-processing system employing a vision chip.

Due to the huge amounts of data in pixel streams, real-time image processing usually requires extensive hardware and/or software resources [8]. If the hardware support is sufficiently small, some of it can be integrated with the imaging-sensing circuits to form a so-called "vision chip," as indicated in Figure 1. Such chips serve as the preprocessing front end of an image-processing system [4] [14].

Vision chips are loosely classified as analog or digital (pulse domain), depending on the type of circuitry used in the preprocessing stage to convert the sensed analog input signals to digital form for final processing. Typical preprocessing circuits are analog-to-digital converters (ADCs), noise filters, and edge detectors [3] [8] [11]. These steps may require many operations per pixel, and consume most of the power of the system [21]. The design of vision chips is very challenging since it involves complex trade-offs among chip area, power, speed and accuracy. It also requires some degree of parallel processing, which can be at the level of individual pixels, groups of pixels, or the overall system [21].

We propose to use stochastic computing (SC) for real-time image preprocessing. This is a method of computing with bit-streams at very low hardware cost [2][7]. A stream of N bits containing N_1 1s and $N - N_1$ 0s denotes the stochastic number (SN) $x = N_1/N$, which is treated as a probability. For example, 0111, 1101, and 10110111 all denote $x = 3/4 = 0.75$. Stochastic circuits are small and so have very low power requirements. For example, multiplication of two N-bit SNs x_1 and x_2 to form the arithmetic product $x_1 \times x_2$ can be done in N clock cycles by means of a single AND gate. Besides low area and power, SC has the advantages of high error tolerance (bit flips have little impact on signal probability) and support for massive, low-level parallelism. Its main disadvantage is the need for very long bit-streams to achieve high precision. However, as we will show, this problem can be greatly mitigated by exploiting a special property of SC we call *progressive precision*, where result quality gets better as the computation proceeds.

SC and real-time image processing share some key properties. They both handle streaming analog data (image intensities or probabilities), process the data digitally, and have good noise tolerance. Several proposed vision chips encode the sensed light signals using pulse-frequency modulation (PFM) [9] [11] [20].

This means that pixel information is conveyed by the frequency of a pulse train, as in biological neural networks and SC circuits. SC thus has the potential to meet most of the challenging requirements of the retinal implant application mentioned earlier: streaming neural-style data, very small circuit size, extremely low power, and insensitivity to noise.

Although known for years [7], SC has only recently gained attention with the emergence of applications that can take full advantage of its unique features, such as support for massive parallelism. A notable example is decoding the low-density parity check (LDPC) codes employed in the IEEE WiFi and other communication standards [10]; this is an application that requires massive amounts of fast, but relatively simple, parallel processing. Naderi et al. [16] have used SC to implement an LDPC decoder chip that has performance comparable to conventional (weighted-binary) designs, but is significantly smaller. Li and Lilja [12] and Qian et al. [18] have shown that SC can outperform binary computing in some processing tasks involving stored images. Ma et al. [13] show that SC is useful in fault-tolerant image processing. To apply SC to stored images, data conversion between the weighted-binary and stochastic domains is necessary. This conversion is costly; in some cases, it can consume up to 80 percent of SC circuit area [18]. As we will show, real-time image processing avoids much of this cost. The use of SC in real-time vision chips was briefly discussed by Hammadou et al. in 2003 [9], but otherwise has received very little attention. SC application has also been hindered by lack of a general design methodology, and by limited understanding of its underlying theory.

This paper presents designs for various image-processing tasks like real-time edge detection and gamma correction. The new designs are evaluated by emulation on FPGAs under normal and noisy conditions. The results show that SC can provide high performance and huge savings in area and power. They also illustrate SC's progressive precision and noise-tolerance properties. The main contributions of the paper are:

1. Demonstration of the suitability of SC for real-time image processing with demanding design constraints.
2. Novel stochastic circuits that are far smaller and more efficient than both their conventional counterparts and (where they exist) previous SC designs.
3. Two orthogonal methods of producing images with progressive quality at minimal cost.

2. STOCHASTIC COMPUTING

We begin by reviewing stochastic computing and its relevant properties. As noted already, SC circuits tend to be very small. Figure 2a shows a stochastic adder [7] that implements the scaled addition $z = (x_1 + x_2)/2$. Scaling is needed to ensure that the sum lies in the interval [0,1]. This SC adder is just a 2-way multiplexer with a random bit-stream r of probability value 1/2 applied to its control (select) input.

A typical SN x has many different representations, and the ones chosen for the various inputs of an SC circuit usually need to be uncorrelated and derived from independent stochastic number generators (SNGs). Correlated inputs are generally believed to produce inaccuracies in the computation. A simple example is seen in the use of an AND gate to multiply SN x by itself to obtain x^2. If identical bit-streams are used for the two copies of x (implying maximum correlation), the AND gate produces x instead of x^2, a large error. So circuits with correlated inputs are rarely used in the

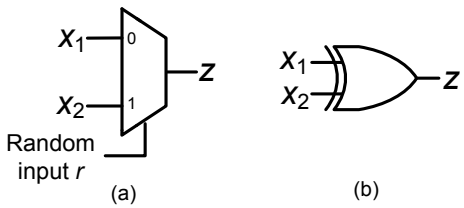

Figure 2. (a) Stochastic adder and (b) subtracter.

SC literature. However, we observe that correlation does *not* necessarily lead to inaccuracy, if properly managed. In particular, in circuits like the SC adder, inputs can be correlated without affecting the accuracy of the result. This can greatly mitigate the complications of random number generation.

In some cases, correlated inputs can change the functionality of a circuit to another, and perhaps more desirable, operation. Figure 2b shows an XOR gate that, assuming independent inputs, performs the function

$$z = x_1 \times (1 - x_2) + x_2 \times (1 - x_1)$$

However, when fed with correlated inputs where x_1 and x_2 have maximum overlap of 1s, we can show that the circuit computes $z = |x_1 - x_2|$, i.e., it acts as a type of subtracter. As we will see, this operation is very useful in SC image processing.

The major drawback of SC is that very long bit-streams are required for high-precision calculations. Precision is defined as the number of bits needed to represent a given number x in conventional weighted-binary format. An n-bit binary number maps to an SN of length of $N = 2^n$. Hence, the SN has precision of roughly $\log_2 N$, and takes $N = 2^n$ clock cycles to generate or process. This overhead makes SC impractical for most high-precision digital computations. About 8 bits of precision suffice for most image-processing tasks, implying a maximum SN length of $2^8 = 256$ bits—a reasonable size.

The speed of stochastic circuits can be increased by exploiting the *progressive precision* properties of SC: if properly chosen, the first few bits of an SN can yield a rough approximation to the final number. For instance, consider the following bit-stream representing the number 9/16:

0 1 1 0 1 1 0 0 1 0 1 1 0 1 0 1

The first two, four and eight bits, i.e., 01, 0110 and 01101100, all represent the number 1/2, which is a good approximation of the final number 9/16. This progressive precision property of SC can be exploited if a decision can be made quickly from a particular image. For example, in the edge-detection application discussed later, many sharp edges are detected as early as four clock cycles into the computation.

Stochastic image-processing circuits have been proposed [12] [18] that are smaller than conventional designs, but are not particularly efficient. As noted above, the use of many conversion units degrades the performance of those designs. Our approach integrates format conversion into stochastic image processors in a way that minimizes the overhead of conversion circuits. Even if conversion costs are ignored, our designs are smaller and more efficient than those of [12] and [18].

3. SYSTEM OVERVIEW

Vision chips vary widely in how their sensors and processing circuits are laid out. In the simplest form, one processor handles all the pixels in series and no parallel processing occurs. At the other extreme, each pixel has a processing element (PE) of its own,

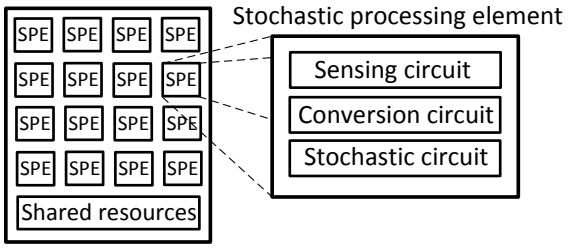

Figure 3. Top-level view of an SC-based vision chip and its stochastic processing elements (SPEs).

providing maximum parallelism. Since conventional digital PEs can be large, this approach does not scale well [21]. For real-time applications, one processor per pixel is desirable and, as we show, is achievable using stochastic computing techniques.

We propose a vision chip with maximum parallelism using stochastic processing elements (SPEs) that are very small and scale well. Our designs are also applicable to cases where processing circuits are shared among pixels. Figure 3 shows a high-level view of the proposed chip and its SPEs. For clarity, a 4×4 pixel array is shown, but it is possible to have many more pixels on chip. In addition to the SPEs, the chip has shared resources that manage random number generation and include a few counters based on LFSRs (linear feedback shift-registers). The area cost of these resources is minor since they are small and their cost does not change with the pixel count.

As Figure 3 shows, vision chips have image sensors that convert the perceived light intensity to an analog electrical voltage. To enable digital processing, this analog signal must be converted to digital form using a conventional ADC or, in the SC case, an analog-to-stochastic converter. As noted earlier, the cost of an analog-to-stochastic converter is very similar to that of a conventional ADC, which is depicted in Figure 4. In the conventional case, the analog voltage from the sensor is converted to a digital number using a ramp-compare technique. This requires an analog comparator fed by the sensor voltage, and a ramp voltage generated by a counter and a digital-to-analog converter (DAC). The comparator directly triggers a second counter which produces the desired digital output. In the SC case, the sensor voltage is converted to a stochastic number by comparing it to a random voltage generated by an LFSR-based counter and a DAC. A stochastic number appears at the output of the comparator and can then be processed by an application-specific stochastic circuit, such as an edge detector. The second counter is used to convert the final result to weighted binary form. It should be clear from Figure 4 that analog-to-stochastic conversion imposes little overhead as it employs essentially the same ADC circuits found in any digital vision chip.

Although analog comparators are well understood, they still present some circuit design challenges; for instance, low-area comparators are susceptible to noise. It is feasible to place comparators of suitable quality and size at every pixel [6]. In conventional digital image processors, a noise reduction step such as median filtering is needed [12]. In the proposed SC vision chip, however, the impact of noise is minimal thanks to the error tolerance of stochastic numbers, and a separate noise suppression step is unnecessary.

4. IMAGE PROCESSING OPERATIONS

This section discusses two basic image preprocessing categories, namely pixel-wise operations and windowing operations, examples

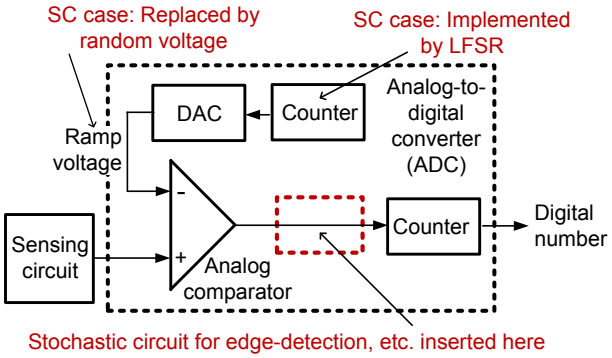

Figure 4. Conventional ADC circuit for a vision chip with the changes needed for analog-to-stochastic conversion.

of which are implemented later. We then present two ways to produce images with progressive quality improvement, which greatly speed up stochastic processing.

4.1 Basic Operations

Pixel-wise operations modify a pixel's intensity value x independent of the values at other pixels. They typically implement a real-valued function $f(x)$ that is used to adjust intensity values. A well-known example is gamma correction, which is used to compensate for non-linearities in recording or display devices, or to increase pixel contrast [8]. One of the simpler gamma-correction functions is $f(x) = x^{0.45}$. To synthesize stochastic circuits that implement functions like $f(x)$, we use the synthesis method of [1]. This approach produces efficient circuits for a broad class of arithmetic functions.

A second category of image-processing operations of concern are windowing operations, where a weighted moving-average operation is performed on a small window of pixels, either to extract features of the image or to enhance its quality. Examples of such operations are edge detection, sharpening and blurring [8]. The pixel windows are typically of size 2 × 2, 3 × 3, or 5 × 5. In order to design operations of this type, we mainly use the components of Figure 2. An m-to-1 multiplexer with a random select input performs averaging operations of the form $z = \frac{1}{m}(x_1 + x_2 + \cdots + x_m)$; if negative weights are present, subtraction can be implemented by XOR gates.

4.2 Spatial Progressive Quality Improvement

Generating images that progressively improve is an important task in image processing because it enables a trade-off between accuracy and computation that can be exploited in several ways. Image standards such as JPEG2000 [5] encode images of various qualities simultaneously. A conventional method of reducing the quality of an image is to reduce its number of pixels, i.e., its resolution. Figure 5 shows an image with several resolution levels; clearly, the quality diminishes as the resolution decreases.

Processing images with multiple resolutions imposes some computational overhead. However, we show by an example, that in the SC case, this overhead is minimal. Assume that a given image is to be processed at its original resolution, and at a lower-quality version with 16 times less resolution. In the latter case, intensity signals from 16 neighboring pixels of the original image are averaged to produce a super-pixel.

Figure 6 shows this averaging process implemented by a 16-to-1 multiplexer. This circuit processes each input individually, and records its results in the corresponding counter. Meanwhile, it

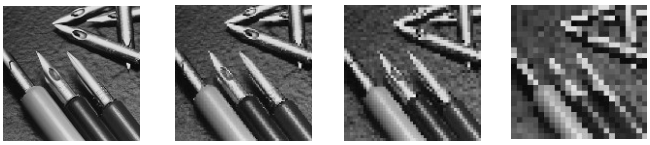

Figure 5. An image at four different resolution levels: (a) 400×400, (b) 100×100, (c) 50×50, and (d) 25×25 pixels.

performs the same computation on the low-resolution super-pixel and records that result in a separate counter. As seen in the figure, the overhead of a super-pixel computation is the additional counter, implying a very low cost.

4.3 Temporal Progressive Quality Improvement

As noted earlier, stochastic numbers have the progressive precision property, meaning that short sub-sequences of an SN can provide low-precision estimates of its value. This property can also be used to obtain images of different qualities because we can have SN-encoded pixels with different precisions. This approach is orthogonal to the previous spatial-resolution method, and, since it is an inherent property of SC, it comes at essentially no cost. One simply uses the values appearing at the output counters of Figure 6 at successive points of time.

Figure 7 shows how this property can be exploited in image processing. An edge-detection operation is being performed on an image. The input image has 8 bits of precision (the precision of an image corresponds to its gray-scale resolution [8]), and hence requires SN bit-streams of length 256. However, if the output image is checked at different points of time, it can be seen that as early as 4 clock cycles into the computation, many edges of the input image are detected, and after 32 clock cycles, almost all the edges are detected.

5. STOCHASTIC EDGE DETECTION

Edge detection is useful in image processing and computer vision because it allows objects to be extracted from an image by highlighting their edges. In the retinal implant application, a real-time edge-detecting circuit generates high-contrast images of the environment that greatly help a vision-impaired person to navigate correctly and avoid obstacles. We now consider the design of high-efficiency SC circuits for this task.

5.1 Circuit Design

Many edge-detection algorithms are known [8], and a few have been implemented with (non-real-time) SC [12]. Here we use the Roberts cross algorithm [8]. It computes a moving average on a window of size 2 × 2 for each pixel $x_{i,j}$ at row i and j of the image, and generates an output value $z_{i,j}$ according to the following formula.

$$z_{i,j} = 0.5 \times (|x_{i,j} - x_{i+1,j+1}| + |x_{i,j+1} - x_{i+1,j}|) \qquad (1)$$

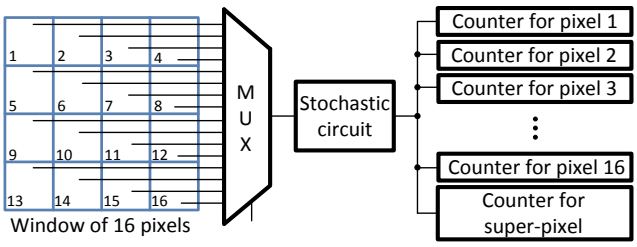

Figure 6. Stochastic processing of 16 pixels individually and as a super-pixel.

Figure 7. Progressive precision results for edge detection: (a) input image; output image after (b) 4, (c) 32, and (d) 256 clock cycles.

A stochastic implementation of this operation has been proposed by Li and Lilja [12], but it uses relatively large sequential circuits to approximate the absolute value function. Instead, we use the simple combinational SC components of Figure 2, which lead to a design that is more than 20 times smaller than that of [12], but has similar (or even better) performance. Figure 8a shows the proposed stochastic circuit for edge detection. It uses just two XOR gates to implement the subtractions in (1) and a multiplexer to perform the addition. As discussed, the inputs must be correlated, which is assured by assigning them a common random number source. The multiplexer's select input is fed with a random input r, which is produced at minimal cost since it is shared among the SPEs. In contrast, the corresponding binary design (Figure 8b), assuming 8-bit precision, contains several big arithmetic units such as adders and subtracters.

As proof-of-concept, we implemented and validated the SC edge-detection circuit (and the other examples in this paper) on a Xilinx Virtex-5 FPGA chip. Figure 9 illustrates the FPGA board (XUPV5-LX110T) used for our experiments, along with a representative image-processing task. It is important to note that the FPGA implementation was only used to verify the functionality of the circuits, and not for performance comparison. The output generated by the circuit is validated by comparing it to an expected output generated via conventional approaches.

In order to compare the proposed edge-detection design with previous work, we used the SIS synthesis CAD tools [19]. Table 1 summarizes the results. All the numbers are estimated values based on a generic library of cells using 0.35μm CMOS technology. The delay of each circuit is obtained by multiplying the clock period by the number of cycles required to perform the operation. The conventional binary design shown in Figure 8b implements (1) in a single clock cycle. The SC designs, on the other hand, require 256 cycles for the full 8 bits of precision. The dynamic power consumption of the circuits are also estimated using the SIS tools [19].

The results reveal that the proposed edge-detection circuit is strictly better than the previous SC design [12]. Our SC design is about 30 times smaller than the conventional designs, and only 3 times slower. From the area-delay numbers, we see that the SC design has a significant cost advantage. The dynamic power consumption of each circuit is also reported in Table 1, and show that for a given clock frequency (in this case 20MHz), SC has much lower power consumption. However, since a stochastic circuit with fixed precision runs for a longer time, its energy

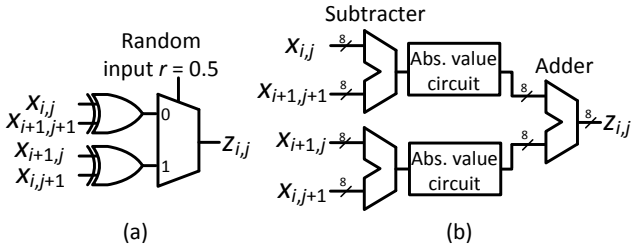

Figure 8. Edge detectors: (a) stochastic and (b) conventional designs.

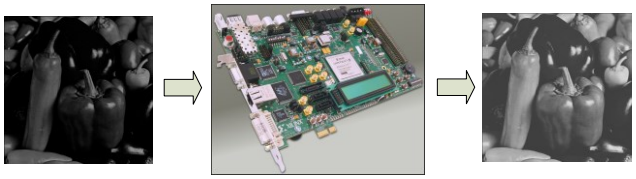

Figure 9. FPGA setup for emulating image-processing tasks, in this case, gamma correction.

consumption eventually becomes higher than that of a conventional design. We do not report the leakage power/ energy of each circuit, but since leakage power is directly proportional to area, we can conclude that the leakage power of SC circuits is lower than the conventional case.

Table 1. Synthesis results for the edge detection circuits.

Implementation	Area (μm^2)	Delay (ns)	Power @20MHz (μW)	Energy (nJ)	Area × Delay (μm^2×ps)
Conventional weighted-binary	6928	19.49	7767.9	0.39	135.03
Previous SC design [12]	4312	1300	2213.7	28.34	5607.67
SC design proposed here	200	58.88	88.7	1.14	11.78

5.2 Error-Tolerant Behavior

Stochastic circuits are inherently noise tolerant, and their performance is not significantly degraded if the inputs contain noise, or even if the circuit components are noisy and unreliable [12] [18]. This is potentially a huge advantage, since pixel-level circuits need to be extremely small, with electrical characteristics that are difficult to control precisely. In this section, we demonstrate this benefit by a qualitative comparison between our SC-based edge detector and a conventional binary implementation when a noisy image is perceived by the sensors.

We model input noise as a Gaussian random variable with mean value 0 and standard deviation σ, which is added to the voltage signal generated by the sensor, i.e., right before the analog-to-digital or analog-to-stochastic conversion stage. We define the noise level as the ratio of σ to the full voltage swing of the sensor, so a 10% noise level means that the noise is 1/10 of the voltage range of the sensor.

Figure 10 illustrates the simulated performance of the three design methods at various noise levels. The performance of the conventional weighted-binary approach significantly degrades as the noise level reaches 5%; at noise levels of 10% and 20%, the output images become useless. The second implementation again employs a conventional binary approach, but with noise reduction implemented by a median filter [8]. This costly noise-reduction step improves the results, but significant quality degradation is still seen at higher noise levels. On the other hand, the SC implementation is almost unaffected by noise and is able to detect the edges even in the 20% noise case. The impact of noise on the SC circuit appears as a gray background, a result also seen in [18].

5.3 Progressive Precision

Also of interest is the performance of the edge-detection circuit when producing images with progressive quality improvement. The examples in Figure 7 suggest that the runtime of the edge-detection circuit can be further reduced (by a factor of 8), without compromising accuracy. This implies that edge detection requires

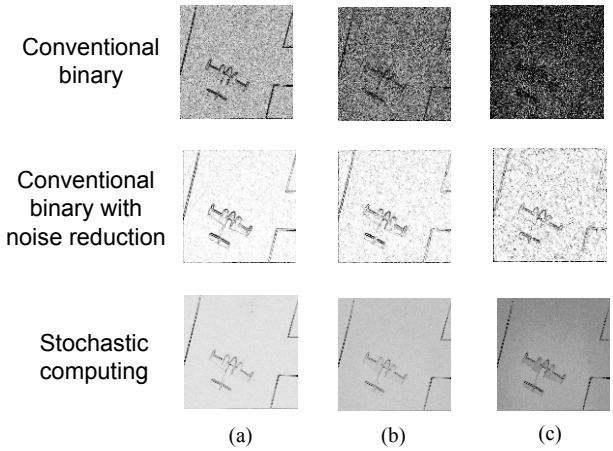

Figure 10. Edge-detection performance for three implementation methods with noise levels of (a) 5%, (b) 10% and (c) 20%.

less precision (than the original 8 bits), so for a fair comparison, we also implemented a low-precision version of the conventional edge detection circuit. As can be seen from Table 2, the stochastic design is strictly better than the conventional one. Also, the stochastic edge-detection is so efficient that can operate in real-time (15 frames per second) at 1nW power consumption. This number might be further reduced by switching to sub-threshold technologies.

Table 2. Synthesis results for low-precision edge-detection circuits.

Implementation	Area (μm^2)	Delay (ns)	Power @20MHz (μW)	Energy (nJ)	Area × Delay (μm^2×ps)
Conventional weighted-binary	4344	7.66	5156.8	0.26	33.28
SC design proposed here	200	7.36	88.7	0.14	1.47

6. OTHER IMAGE-PROCESSING OPERATIONS

Besides edge detectors, we designed several other stochastic image-processing circuits and evaluated their performance compared to alternative designs. We used the synthesis method of [1] to obtain the gamma-correction circuit in Figure 11a. A flip-flop is used in this design in order to produce a second uncorrelated copy of the input bit-stream. The function implemented approximates the target function $z_{i,j} = x_{i,j}^{0.45}$ but produces acceptable results, as seen in Figure 9. Figure 11b shows a conventional binary implementation of the same function. Like our edge-detection circuit, the stochastic gamma-correction circuit has a random input of probability 0.25 produced by a vision chip's shared resources.

Table 3 shows the synthesis results of our gamma-correction circuit, along with a conventional design of the same precision, and a previous SC design [18]. The design of [18] has better accuracy,

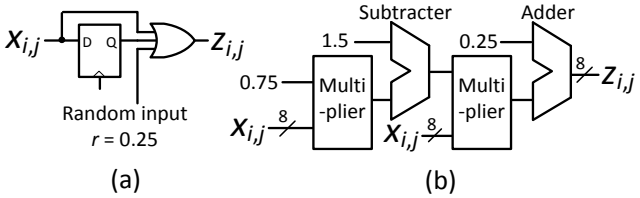

Figure 11. Gamma correctors: (a) stochastic and (b) conventional.

Table 3. Synthesis results for gamma-correction, blurring, and gradient calculation circuits.

Task	Design method	Area (μm^2)	Delay (ns)	Power 20MHz (μW)	Energy (nJ)	Area × Delay (μm^2×ps)
Gamma correction	Conventional binary	10576	27.2	21486	1.07	287.77
	Previous SC method [18]	1416	5365	970.3	49.68	7597.9
	Our SC method	168	15.4	55.8	0.71	2.58
Blurring	Conventional binary	19464	32.6	13196	0.66	634.92
	Our SC method	664	589	433.4	5.55	390.96
Gradient calculation	Conventional binary	1520	3.4	716.6	0.04	5.20
	Our SC method	72	51.2	26.9	0.34	3.69

but is much costlier. The table also includes synthesis results for two other image-processing tasks, namely, blurring and gradient calculation. These results are consistent with those we obtained for edge detection.

7. CONCLUSIONS

We have shown that stochastic computing (SC) is practical for high-performance vision chips. Complex image-processing tasks can be implemented with only a few gates, thus enabling massively parallel processing at the pixel level and real-time operation. We presented designs for stochastic image-processing circuits that outperform existing designs (both conventional and SC) in most aspects. In particular, we have designed an edge-detection circuit that is strictly better than equivalent conventional designs. This highly efficient circuit seems ideal for vision chips and retinal implants. We also designed other representative image-processing circuits and made detailed comparisons with alternative implementations. We conclude that stochastic circuits are, in general, much smaller than conventional designs, and are much more efficient in terms of power consumption and area-delay product.

Furthermore, we demonstrated two different ways to process images with progressive quality improvement at minimal cost overhead. These properties compensate for SC's longer computation times. We successfully implemented and validated the discussed image-processing tasks on an FPGA development system. Finally, we showed that unlike conventional designs, the SC circuits can process very noisy images with almost no performance degradation.

8. ACKNOWLEDGEMENTS

This work was supported by Grant CCF-1017142 from the U.S. National Science Foundation.

9. REFERENCES

[1] A. Alaghi & J.P. Hayes, "A spectral transform approach to stochastic circuits," *Proc. Intl. Conf. Computer Design*, pp.315-321, 2012.

[2] A. Alaghi & J.P. Hayes, "Survey of stochastic computing," to appear in *ACM Trans. Embedded Computing Systems*, 2012.

[3] F. Andoh *et al*., "A digital pixel image sensor for real-time readout," *IEEE Trans. Electron. Dev,* **47**, pp. 2123-2127, 2000.

[4] Centeye Inc. "Introduction to current Centeye vision chips" http://centeye.com/technology/vision-chips/, Feb. 2011.

[5] C. Christopoulos *et al*., "The JPEG2000 still image coding system: an overview," *IEEE Tran. Consumer Electronics, IEEE Transactions on* , **46**, 4, pp. 1103-1127, 2000.

[6] P. Dudek and P.J. Hicks, "A general-purpose processor-per-pixel analog SIMD vision chip," *IEEE Trans. Ccts. & Sys. I*, **52**, pp.13-20, 2005.

[7] B.R. Gaines, "Stochastic computing systems," *Advances in Information Systems Science*, **2**, pp. 37-172, 1969.

[8] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 2nd ed., Prentice Hall, 2002.

[9] T. Hammadou *et al*., "A 96 × 64 intelligent digital pixel array with extended binary stochastic arithmetic," *Proc. Intl. Symp. Ccts. & Sys.* (ISCAS), pp. IV-772–IV-775, 2003.

[10] IEEE, *Standard 802.11n for Info. Technology Telecommunications & Info. Exchange between Systems Local & Metropolitan Area Networks.* http://standards.ieee.org, 2009.

[11] K. Kagawa *et al*., "Pulse-domain digital image processing for vision chips employing low-voltage operation in deep-submicrometer technologies," *IEEE Jour. Sel. Topics in Quantum Electronics*, **10**, pp. 816-828, 2004.

[12] P. Li and D.J. Lilja, "Using stochastic computing to implement digital image processing algorithms," *Proc. Intl. Conf. Computer Design*, pp. 154-161, 2011.

[13] C. Ma *et al.*, "High fault tolerant image processing system based on stochastic computing," *Proc. Intl. Conf. Computer Science & Service System*, pp. 1587-1590, 2012.

[14] A. Moini, *Vision Chips,* Kluwer, 2000.

[15] W. Mokwa, "Retinal implants to restore vision in blind people," *Proc. Transducers,* pp. 2825-2830, Beijing, 2011.

[16] A. Naderi *et al*., "Delayed stochastic decoding of LDPC codes," *IEEE Tran. Signal Proc.*, **59**, pp. 5617-5626, 2011.

[17] N.L. Opie *et al*., "Heating of the eye by a retinal prosthesis: modeling, cadaver and *in vivo* studies," *IEEE Trans. Biomed. Engin.,* **59**, pp. 339-345, 2012.

[18] W. Qian *et al*., "An architecture for fault-tolerant computation with stochastic logic," *IEEE Trans. Comp.*, **60**, pp.93-105, 2011.

[19] E.M. Sentovich, *et al*., "SIS: A system for sequential circuit synthesis," Univ. of California, Berkeley, Tech. Report UCB/ERL M92/41, Electronics Research Lab, 1992.

[20] F. Taherian & D. Asemani, "Design and implementation of digital image processing techniques in pulse-domain," *Proc. Asia Pacific Conf. Ccts. & Sys.* (APCCAS), pp. 895-898, 2010.

[21] H. Yamashita & C.G. Sodini, "A CMOS imager with a programmable bit-serial column-parallel SIMD/MIMD processor," *IEEE Trans. Electron. Dev.,* **56**, pp. 2534-2545, 2009.