CMOS Focal-Plane Spatially-Oversampling Computational Image Sensor

Ashkan Olyaei, *Student Member, IEEE,* and Roman Genov, *Member, IEEE*

CMOS Focal-Plane Spatially-Oversampling Computational Image Sensor

I. Introduction

Many video processing applications employ spatial image transforms such as block-matrix transforms and convolutional transforms. For example, block-matrix transforms such as discrete cosine transform (DCT) or discrete wavelet transform (DWT) are widely used in various image and video compression algorithm standards. Convolutional transforms are often employed in pattern recognition. These transforms require extensive computational resources for their real-time implementations.

A number of techniques for realizing block-matrix and convolutional transforms in sensory systems have been developed. Dedicated digital signal processors (DSPs) rely on high-throughput architectures to compute spatial weighted sums needed in block-matrix transforms, but require significant area and power resources. At high imager resolutions, the input data rate or memory-processor bandwidth of such processors may limit their sustained throughput [2], [3]. An analog-to-digital converter (ADC) to quantize the analog sensory input prior to signal processing is also required for such digital processors.

To overcome these limitations, block-matrix transforms and convolutional transforms have also been implemented in the analog domain directly on the focal plane. Capacitor bank implementations use charge sharing to compute weighted sum and difference [4]–[6] but may have limited scalability. Current-mode weighted averaging implementations [7] use zero-latency current-mode addition but employ multiple matched current mirrors at the expense

of increased pixel area. Charge integration and gain-stage voltage summation [8] utilized in variable resolution imaging do not allow for weighted averaging and require additional column-parallel amplifiers. Current-mode vector-matrix multiplication [9] architectures employ floating-gate arrays for block matrix storage and achieve high power efficiency. Kernel-dependent scan-out imager architectures have been shown to reduce memory requirements in focal-plane spatial image processing [10]. A tree-based partitioning algorithm that implements adaptive compression has also been reported [11]. Circuit implementations based on video compression algorithms utilize in-pixel temporal prediction [12]–[14] and array-based spatial prediction [15] to reduce the amount of transmitted data. All of the aforementioned architectures perform computation in analog VLSI domain and require an extra analog-to-digital converter to provide the digital output.

Analog-digital mixed-domain CMOS imaging and signal processing combine the benefits of the two domains [16]. Analog circuits perform area-efficient and low-power computation directly on the focal plane, eliminating the need for an external processor [17]. The intrinsic parallelism of imaging architectures yields high computational throughput, often beyond that of modern digital processors, allowing to perform complex video processing operations in real time. Digital components provide the output in a convenient digital format, and sustain the accuracy and configurability of such systems.

We present a mixed-signal VLSI implementation of a digital CMOS imager computing block-matrix transforms and convolutional transforms on the focal plane for real-time image processing. The computational image sensor presented here combines image acquisition, signal processing and quantization in a single compact low-power architecture as shown in Figure 1. Our approach combines weighted spatial averaging and oversampling quantization

in a single $\Delta\Sigma$-modulated analog-to-digital conversion cycle, making focal-plane computing an intrinsic part of the quantization process. The approach yields power dissipation below that of a conventional digital imager while the need for a peripheral DSP is eliminated.

II. Block-Matrix and Convolutional Transforms

Block-matrix and convolutional transforms correlate a segment of an image with a spatial kernel, or block matrix, in order to identify statistical redundancies or distinguish particular features depending on the transform type. In the example of image compression, the redundancies are eliminated to reduce data rate. In the case of pattern recognition, the features are employed to form a more precise object description and enhance the classifier performance.

To transform an image I into transformed image T, the kernel or block matrix C is tiled vertically and horizontally across the image as illustrated in Figure **??**. The block matrix is tiled in overlapping or non-overlapping fashion, corresponding to convolutional and block-matrix transforms respectively. For the case of the block-matrix transform coefficients of T are obtained by computing the two-dimensional dot product of C and I at each tile location:

$$T_{ij} = \sum_{h=1}^{H} \sum_{v=1}^{V} C_{hv}\ I_{xy}, \tag{1}$$

$$x = h + (i-1)H, \quad i = 1, 2, \ldots, \frac{L}{H}, \tag{2}$$

$$y = v + (j-1)V, \quad j = 1, 2, \ldots, \frac{K}{V}, \tag{3}$$

where $C_{hv} \in \mathbb{Z}$ are the block matrix coefficients comprising a spatial kernel; L and K are the image horizontal and vertical sizes, assumed for simplicity to be multiples of the kernel dimensions H and V; h and v are the horizontal and vertical block matrix indices, and i and j are the indices of the block-transformed image.

The block-matrix and convolutional transforms are generally computationally expensive. For example, consider an HDTV 1080i imager operating at 30 frames per second (fps). To handle the computational throughput of an 8×8 convolutional transform of the video, an equivalent computational throughput of one 3.5 GHz Pentium processor is needed.

In video compression algorithms, to achieve selective compression of the image, redundant and localized gradient values are filtered out according to a threshold bias, which is based on the required compression ratio and the reconstructed image quality specifications. Another thresholding technique, mainly employed in JPEG image compression, is non-uniform quantization of the block-matrix-transformed image, where the more significant low-frequency spatial information components are quantized with a higher resolution compared to the less important high-frequency ones.

In pattern recognition systems, to enhance the performance of the classifier, based on a particular feature extraction algorithm, a set of features are extracted from the input images by computing their block-matrix transforms. Both training and classification are then performed on these extracted features.

A. Discrete Wavelet Transform (DWT): Haar Wavelet Example

Two-dimensional Haar wavelet transform is a simple example of a block-matrix transform commonly used in image compression and pattern recognition systems. By extracting horizontal, vertical and diagonal edges, Haar wavelets register the relationship between intensities among neighboring pixels in different orientations and hence form a ”ratio template” [18]. The ratio template is independent of illumination conditions. It truly captures the ratio between various features of an object, which within a class exhibit greater correlation than the absolute intensity values. As a result, a more precise object description is generated at a cost of lower

spatial resolution.

The one-dimensional Haar wavelet is composed of the scaling function ϕ(t) or the father wavelet, and the wavelet prototype function ψ(t) also known as the the mother wavelet:

$$\phi(t) = \begin{cases} 1 & \text{if -1}\leq \text{t} \leq 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$\psi(t) = \begin{cases} 1 & \text{if } 0 < \text{t} \leq 1, \\ -1 & \text{if -1} \leq \text{t} \leq 0, \\ 0 & \text{otherwise.} \end{cases}$$

The combination of scaling and wavelet prototype functions in two-dimensional space yields the following scalar, horizontal, vertical and diagonal two-dimensional Haar wavelet functions:

$$\phi(x,y) = \frac{1}{4}\phi(x)\phi(y), \tag{4}$$

$$\psi^H(x,y) = \frac{1}{4}\psi(x)\phi(y), \tag{5}$$

$$\psi^V(x,y) = \frac{1}{4}\phi(x)\psi(y), \tag{6}$$

$$\psi^D(x,y) = \frac{1}{4}\psi(x)\psi(y), \tag{7}$$

where the $\frac{1}{4}$ coefficient is applied so that the maximum value of the wavelet transform stays within the image intensity range. The equivalent spatial kernels of the scalar and wavelet

functions for the first-level Haar wavelet are:

$$\mathbf{\Phi_1} = \frac{1}{4}\begin{pmatrix} +1 & +1 \\ \\ +1 & +1 \end{pmatrix}, \tag{8}$$

$$\mathbf{\Psi_1^H} = \frac{1}{4}\begin{pmatrix} +1 & -1 \\ \\ +1 & -1 \end{pmatrix}, \tag{9}$$

$$\mathbf{\Psi_1^V} = \frac{1}{4}\begin{pmatrix} +1 & +1 \\ \\ -1 & -1 \end{pmatrix}, \tag{10}$$

$$\mathbf{\Psi_1^D} = \frac{1}{4}\begin{pmatrix} +1 & -1 \\ \\ -1 & +1 \end{pmatrix}. \tag{11}$$

Following the block matrix transform notation in (1)–(3), the transformation of the image I into the Haar-wavelet-transformed image T is of the linear form:

$$T_{ij} = \sum_{h=1}^{H}\sum_{v=1}^{V} C_{hv}\ I_{xy}, \tag{12}$$

$$x = h + (i-1)H, \quad i = 1, 2, \ldots, \frac{L}{H}, \tag{13}$$

$$y = v + (j-1)V, \quad j = 1, 2, \ldots, \frac{K}{V}, \tag{14}$$

$$H = V = 2^b, \quad b = 1, \ldots, B, \tag{15}$$

where $C_{hv} \in \{-1, +1\}$ are the Haar wavelet coefficients comprising a square spatial kernel

in (8)–(11):

$$\mathbf{C} \in \{\mathbf{\Phi_l}, \mathbf{\Psi_l^H}, \mathbf{\Psi_l^V}, \mathbf{\Psi_l^D}\}, \tag{16}$$

and B is the number of levels of Haar transform. B-level Haar wavelet features for $B > 1$ can be obtained by repetitive use of level-one transform, or directly using $3B + 1$ spatially averaging Haar wavelet coefficient kernels of size $H = V = 2^b$, with $b = 1, \ldots, B$.

Figure 2 illustrates the spatial kernels of two-dimensional one-level, two-level, and three-level Haar wavelet transforms. The Haar wavelet transformed images are obtained by computing the block-matrix transforms of the original image with these Haar wavelet kernels. Figure 3 depicts the correspondingly computed two-dimensional Haar wavelet transforms of *Audrey*.

III. Architecture

The block-matrix transform of the form (1) can be decomposed as follows:

$$T_{ij} = \sum_{h=1}^{H} \sum_{v=1}^{V} C_{hv}\ I_{xy} = \sum_{h=1}^{H} T_{ij,h}, \tag{17}$$

with partial sums

$$T_{ij,h} = \sum_{v=1}^{V} C_{hv}\ I_{xy} = \sum_{v=1}^{V} |C_{hv}|\ S_{xy}, \tag{18}$$

and the sign of C_{hv} factored into the sign-transformed pixel outputs

$$S_{xy} = sign(C_{hv})\ I_{xy}, \tag{19}$$

where the notation is consistent with that of (2)–(3), $C_{hv} = sign(C_{hv})\ |C_{hv}|$, and I_{xy} is the output of a pixel at location (x,y).

The proposed mixed-signal VLSI architecture efficiently implements computations (19), (18), and (17), in that order, as depicted in Figure 4. Image acquisition and correlated double sampling (CDS) yield offset-compensated pixel output I_{xy} as described in Section IV. A switched-capacitor sign unit circuit multiplies pixel output I_{xy} by the sign of a respective kernel coefficient via selecting the sampling sequence order yielding a sign-transformed pixel output S_{xy} in (19). Section V-A presents the sign unit circuit implementation. Weighted average of V adjacent pixel outputs in an image column is computed by combining oversampling quantization and selective distributed sampling of the sign-transformed pixel outputs to yield $\hat{T}_{ij,h}$, as discussed in Sections V-B and V-C. $\hat{T}_{ij,h}$ is the digital representation of $T_{ij,h}$ in (18). The switch matrix routes the block matrix coefficients and their corresponding sign values bit-serially from a ring shift register, *SR*, with a sequence period of V values and spatial period of H columns, synchronously with image read-out clock *RowScan* to the oversampling quantizers and sign units, respectively. The operation of the switch matrix is discussed in Section V-D. A simple digital delay and adder loop performs spatial accumulation over H adjacent ADC outputs in the digital domain as they are read out to yield $\hat{T}_{ij}$, which is the digital representation of T_{ij} in (17). Section V-E presents an implementation of this digital accumulation.

IV. Image Acquisition

Image acquisition is performed by the active pixel array, the row control and the correlated double sampling (CDS) units. The active pixel comprises a resetable n^+-diffusion–p-substrate photodiode, a selectable analog memory, C_{Mem}, and a selectable source follower with shared

column-parallel current source biased with *IbiasCol* current as shown in Figure 5. The analog memory is implemented as a MOS capacitor for higher density of integration inside the pixel and consequently a larger fill factor.

The row control unit generates the digital signals *Reset*, *Snapshot*, *Sample*, and *RowSelect* controlling the integration and readout phases. There are four modes of operation: snapshot mode, rolling mode, correlated double sampling (CDS) mode and frame-differencing mode. A switched-capacitor CDS unit circuit suppresses fixed pattern noise (FPN) by subtracting the reset pixel output from the sensed pixel output. More information on the design of the image acquisition circuits can be found in [19].

V. Computational Quantization

This section presents a mixed-signal VLSI implementation of (19), (18), and (17). To simplify notation, in this section we consider equations (18) and (19) for a single image column segment (*i.e.,* for given i, *h*, and x) and the first row of the transformed image (*i.e.,* for $j = 1$ and $y = v$). This simplifies the partial sum in equation (18) to

$$T = \sum_{v=1}^{V} C_v \; I_v = \sum_{v=1}^{V} |C_v| \; S_v. \tag{20}$$

For a particular image row (*i.e.,* for given v), the sign-transformed pixel output in equation (19) further simplifies to:

$$S \;\; = \;\; sign(C) \; I, \tag{21}$$

where I is the raw output of a pixel.

A. Sign Unit

The sign unit shown in Figure 6*(a)* is implemented as a switched-capacitor difference circuit. It applies the sign of the coefficient C to the input by selecting a switched-capacitor sampling sequence order as illustrated in the timing diagram in Figure 6*(b)*. This directly implements equation (21). The amplifier in the sign unit is the same as in the CDS circuit. For the sake of simplicity the feedback capacitor reference voltage is shown as ground.

B. $\Delta\Sigma$-Modulated Multiplying ADC

The spatially-compressing image quantizer is implemented as a first-order incremental $\Delta\Sigma$-modulated ADC extended to an oversampling multiplying ADC [20] as described in this section. The first-order incremental oversampling ADC is depicted in Figure 7*(a)*. It converts a sequence of analog samples into a digital word representing a quantized version of the average of all samples. It is comprised of a sample-and-hold (S/H) circuit, an integrator, a comparator and a decimating counter. The rectangular decimation window and initial reset of the accumulator avoid tones in the quantization noise spectrum at DC input that are characteristic of a conventional first-order DS modulator with a low-pass decimation filter. As shown in Figure 7*(b)*, this architecture can be combined with the sign unit and extended to perform both quantization and signed multiplication of the analog input with a digital word

$$C = sign(C)\ |C|, \tag{22}$$

with

$$|C| = \sum_{i=0}^{N-1} c[i], \tag{23}$$

where $c[i]$ are unsigned unary coefficients of C.

Selective sampling of the sign-transformed pixel output S, controlled by the bit-serial unary sequence $c[i]$, yields an analog sequence $u[i] = Sc[i]$. The first-order modulator converts the sequence $Sc[i]$ into a bit stream $y[i]$ in N cycles, using a 'resetable' (RST) analog integrator:

$$w[0] = 0, \tag{24}$$

$$w[i+1] = w[i] + \alpha\ (Sc[i] - y[i]),$$

$$i = 0, \ldots N-1, \tag{25}$$

$$w[N+1] = w[N] - \alpha\ y[N], \tag{26}$$

and a single-bit quantizer:

$$y[0] = -1, \tag{27}$$

$$y[i] = \text{sign}(w[i]), \quad i = 1, \ldots N, \tag{28}$$

where α is the intrinsic gain of the integrator.

A binary counter accumulates the bits $y[i]$ to produce a decimated output. The rectangular decimation window, and initial reset of the integrator, avoid tones in the quantization noise spectrum at DC input that are characteristic of a conventional first-order $\Delta\Sigma$ modulator with low-pass decimation filter [21]. The quantization error (conversion residue) is directly given by the final integrator value $\frac{1}{\alpha}w[N+1]$, as verified by summing (25) and (26) over i:

$$\sum_{i=0}^{N} y[i] = \sum_{i=0}^{N-1} Sc[i] - \frac{1}{\alpha}w[N+1], \tag{29}$$

where

$$\sum_{i=0}^{N} y[i] = \hat{T}' \tag{30}$$

is the digital output.

This operation yields multiplication of the sign-transformed analog pixel output S with the unsigned digital coefficient $|C|$ defined in (23), while a digital output resolution of $\log_2(N)$ bits is warranted:

$$\hat{T}' = |C| \; S + q', \tag{31}$$

which in combination with (21) and (22) yields:

$$\hat{T}' = C \; I + q', \tag{32}$$

where

$$|q'| = |\frac{1}{\alpha} w[N+1]| \tag{33}$$

is the multiplication quantization noise. Higher resolution at lower oversampling ratio N can be obtained using higher-order incremental conversion [22].

As the input is amplitude modulated with unary signed coefficients, an error in the amplitude of these coefficients can contribute to the noise. A noise analysis due to this non-ideality is given in [23] where amplitude modulation of an analog sequence with a Hadamard sequence is utilized in the design of a Nyquist-rate $\Delta\Sigma$-Modulated ADC. This noise is deemed negligible in this design as a simple analog multiplexer is employed to modulate the input.

C. $\Delta\Sigma$-Modulated Weighted Averaging ADC

The multiplying ADC architecture in Section V-B can be extended to perform weighted averaging. Accumulation is achieved by sampling V adjacent pixels in one column, $I_1, I_2 \ldots I_V$, without resetting the integrator or the binary counter. A discrete-time index v is thus introduced.

The architecture of the oversampling weighted averaging ADC is depicted in Figure 8. The bit stream $y[vi]$ is now generated for V inputs each sampled N times:

$$w[0] = 0, \tag{34}$$

$$w[v(i+1)] = w[vi] + \alpha\ (S[v]c[i,v] - y[vi]), \quad i = 0, \ldots N-1, \quad v = 1, \ldots V, \tag{35}$$

$$w[V(N+1)] = w[VN] - \alpha\ y[VN], \tag{36}$$

and a single-bit quantizer:

$$y[0] = -1, \tag{37}$$

$$y[vi] = \mathrm{sign}(w[vi]), \quad i = 1, \ldots N, \quad v = 1, \ldots V. \tag{38}$$

The quantization error, $\frac{1}{\alpha}w[V(N+1)]$, is obtained similarly:

$$\sum_{v=1}^{V}\sum_{i=0}^{N} y[vi] = \sum_{v=1}^{V}\sum_{i=0}^{N-1} S[v]c[i,v] - \frac{1}{\alpha}w[V(N+1)], \tag{39}$$

where

$$\sum_{v=1}^{V}\sum_{i=0}^{N} y[vi] = \hat{T}, \tag{40}$$

and the notation is consistent with that of (24)–(29).

This realizes the computation of a weighted sum of sign-transformed pixel outputs $S[v]$ with the unsigned digital coefficients $|C[v]|$, defined in (23), with an output resolution of $\log_2(VN)$ bits:

$$\hat{T} = \sum_{v=1}^{V} |C[v]|\ S[v] + q, \tag{41}$$

which in combination with (21) and (22) yields:

$$\hat{T} = \sum_{v=1}^{V} C[v]\ I[v] + \ q, \tag{42}$$

where

$$|q| = |\frac{1}{\alpha} w[V(N+1)]| \tag{43}$$

is the weighted averaging quantization error. We arrive at expression (41) for the digital weighted sum, $\hat{T}$, which is a discrete-time equivalent of (20).

Optimization of the number of oversampling cycles based on particular block matrix coefficients can enhance the computational throughput of the architecture. When the maximum absolute value of coefficients in a single row of a block matrix (scaled to all integers), N_V, is less than N, the oversampling computational cycle is stopped on the N_V-th sample and continued on the next row.

Further improvements of the computational throughput of this architecture are achieved by employing an algorithmic $\Delta\Sigma$-modulated ADC [24]. When the maximum of sums of absolute values of coefficients in all columns of a block matrix (scaled to all integers) is less than VN, the oversampling computational cycle is stopped once all of the coefficients have been fed in. Higher resolution bits are then obtained by subsequent algorithmic residue resampling and extended counting on the residue [24].

D. Switch Matrix

The switch matrix routes the H different time-dependent block matrix coefficients and sign signals to $\frac{L}{H}$ groups of adjacent column-parallel ADCs and sign unit circuits, respectively. The block diagram of the switch matrix is shown in Figure 9. A total of V pixel rows are sampled while V coefficients are being shifted out from the shift register. The coefficients are looped back to the shift register input to maintain V-row time period. Each kernel coefficient is stored in a binary format of length $log_2(N)$-bits and is digitally oversampled to yield its unary representation of length N bits, to match the sampling mechanism of an oversampling ADC and correspondingly weight each pixel output.

E. Digital Accumulation

Re-introducing the spatial indices i, j and h back into equation (42) yields the general expression for the column-wise weighted average:

$$\hat{T}_{ij,h} = \sum_{v=1}^{V} C_{hv}\ I_{xy} + q_{ij,h},$$

where $q_{ij,h}$ is the column-wise weighted averaging quantization noise with standard deviation $\sigma_{ij,h}$.

A simple digital delay and adder loop performs spatial accumulation over H adjacent ADC outputs in the digital domain as they are read out :

$$\hat{T}_{ij} = \sum_{h=1}^{H} \hat{T}_{ij,h} = \sum_{h=1}^{H} \sum_{v=1}^{V} C_{hv}\ I_{xy} + q_{ij}, \tag{44}$$

where q_{ij} is the transformed image quantization noise with standard deviation

$$\sigma_{ij} \approx \sqrt{\sum_{h=1}^{H} \sigma_{ij,h}^2},$$

for *i.i.d* noise and large H, yielding an additional improvement in SNR. This mixed-signal VLSI computation realizes a block-matrix transform in (1) with $H \leq 8$. The switch matrix size scales linearly with H. Maximum of H equal to 8 is chosen here to strike a balance between the switch matrix implementation complexity and the overall functionality. The area overhead of sign unit circuits, switch matrix and digital accumulator scales linearly with the imager size and becomes small for large K and L. As computing is interleaved with quantization, the extra computational time and thus power dissipation are small compared to those of raw image quantization in a conventional CMOS digital imager.

VI. Comparative Example

This section compares the presented architecture with a conventional approach where column-parallel algorithmic ADCs performing no computation are employed and an additional peripheral serial digital multiplier and accumulator performs video compression. It is assumed that the kernel is a square matrix of size V with M-bit coefficients and the frame rate is the same in both cases. The comparison is performed for one column only as the two-dimensional computation can be partitioned such that multiplication is performed

in the vertical dimension, and only V additions per kernel are performed in the horizontal dimension.

The first order incremental $\Delta\Sigma$-modulated ADC requires a number of clock cycles exponential with the number of bits of resolution, M. This is a disadvantage compared to the algorithmic ADC which requires a number of clock cycles proportional to M. On the other hand, the SNR of the spatially-oversampling ADC is much higher than that of the algorithmic ADC for the same resolution and the same energy-per-cycle due to in-pixel and inter-pixel oversampling and subsequent noise shaping. In thermal noise limited circuits, power dissipation is linear with SNR. Thus, for the same SNR power dissipation of the oversampling ADC can be reduced below that of the algorithmic ADC.

The numeric comparison depends on the degree of vertical overlap of kernels in subsequent computations. In the worst case, corresponding to the highest number of computations, the subsequent kernels overlap by $V - 1$ pixels in the vertical dimension. Assuming $V = 8$ and $M = 8$, in the worst case, the power dissipation of the $\Delta\Sigma$ modulated spatially-oversampling ADC is 63 percent of the power dissipation of the algorithmic ADC for the same SNR. In the nominal case, when the kernels do not overlap, the power dissipation of the spatially-oversampling ADC is only eight percent of the power dissipation of the algorithmic ADC. This assumes multiplication and addition accuracy of 8 bits as necessary for many image compression tasks. In addition, the conventional approach requires a serial digital multiplier and an adder. At HDTV 1020i imager resolution, a computational throughput of several billions of operations per second is required and would need to be delivered by the digital multiplier and adder at the cost of significant additional power dissipation and integration area. In the proposed approach, besides savings in the ADC power dissipation, the need for

such a high-throughput DSP is eliminated.

VII. Results

Experimental results are obtained from a 0.35 micron CMOS prototype containing a 128×128-pixel array and a bank of 128 column-parallel algorithmic $\Delta\Sigma$-modulated ADCs. Figure 10 shows the die micrograph of the image compression sensor. Table I summarizes its electrical and optical characteristics. The value of parameters V and H are programmable in the range of 2 to 8. Any transform in this size range with signed digital coefficients can be computed. Two-dimensional Haar wavelet transform, a block-matrix transform commonly used in image compression ([25], [26]), is chosen here as a simple example to illustrate the functionality of the presented computational imager implementation. The test setup is shown in Figure 11.

Figure 12*(a)* shows an image acquired by the pixel array with 25 ms integration time. The algorithmic $\Delta\Sigma$-modulated ADC performs distributed image sampling and concurrent signed weighted average quantization, realizing a one-dimensional spatial Haar wavelet transform. Two oversampling phases each of length N=32 clock cycles are interleaved with a single algorithmic residue resampling cycle. Image read-out and computational quantization are characterized off-line in two sequential steps. A digital delay and adder loop implemented off-chip in digital domain performs spatial accumulation over multiple ADC outputs. This amounts to computing a two-dimensional Haar wavelet transform. Figure 12*(b)* depicts experimentally measured two-dimensional one-, two-, and three-level Haar wavelet transforms of the original image. Figure 12*(c)* shows the reconstructed images of the corresponding Haar wavelet transforms. The reconstructed images of one-level Haar transform are compared in Figure 13 for various peak signal-to-noise and compression ratios.

The horizontal resolution of the imager is limited only by maximum scan-out clock frequency for a given frame rate as is the case in conventional imagers. Area and power dissipation scale linearly with the horizontal imager size. In the vertical dimension, all pixels have to be sampled within the a given frame period as set by the programmable spatial kernel with parameters H, V, and coefficients C as well as the imager resolution with parameters L, K, in equations (1)-(3). When computing the discrete cosine transform using 64 8 $\times$ 8 blocks at 30 frames per second, the sensory processor is projected to yield a computational throughput of 4 GMACS when scaled to HDTV 1080i resolution. The throughput is based on a conservative quantizer sampling rate of 40 ksps and a pixel integration time of 5 ms. If a higher resolution in the vertical dimension is required, either the integration time has to be reduced, or the ADC sampling rate has to increase.

VIII. Conclusions

We present a mixed-signal VLSI implementation of a digital CMOS imager computing block-matrix transforms on the focal plane for real-time video compression. The approach combines weighted spatial averaging and oversampling quantization in a single algorithmic $\Delta\Sigma$-modulated analog-to-digital conversion cycle, making focal-plane computing an intrinsic part of the quantization process. The approach yields power dissipation lower than that of a conventional digital imager while the need for a peripheral DSP is eliminated. The experimental results obtained from a 0.35 micron 128$\times$128-pixel CMOS prototype validate the utility of the design for large-scale focal-plane signal processing.

REFERENCES

[1] H. Yamauchi, S. Okada, K. Taketa, T. Ohyama, Y. Matsuda, T. Mori, T. Watanabe, Y. Matsuo, Y. Yamada, T. Ichikawa, and Y. Matsushita, "Image processor capable of block-noise-free JPEG2000 compression with 30 frames/s for digital camera applications," *IEEE International Solid-State Circuits Conference (ISSCC'03)*, vol. **1**, 2003.

[2] C. Chen, Z. Yang, T. Wang, and L. Chen, "A programmable VLSI architecture for 2-D discrete wavelet transform," *IEEE Int. Symp. on Circuits and Systems (ISCAS'00)*, vol. **1**, pp 619-622, 2000.

[3] U. Sjostrom, I. Defilippis, M. Ansorge, and F. Pellandini, "Discrete cosine transform chip for real-time video applications," *IEEE Int. Symp. on Circuits and Systems (ISCAS'90)*, vol. **2**, pp 1620-1623, 1990.

[4] S.E. Kemeny, R. Panicacci, B. Pain, L. Matthies, and E.R. Fossum, "Multiresolution image sensor," *IEEE Trans. Circuits and Systems for Video Technology*, vol. **7** (4), pp 575-583, 1997.

[5] Q. Luo, and J.G. Harris, "A novel integration of on-sensor wavelet compression for a CMOS imager," *IEEE Int. Symp. on Circuits and Systems (ISCAS'02)*, Scottsdale, AZ, May 26-29, 2002. ISCAS'2002.

[6] S. Kawahito, M. Yoshida, M. Sasaki, K. Umehara, D. Miyazaki, Y. Tadokoro, K. Murata, S. Doushou, and A. Matsuzawa, "A CMOS image sensor with analog two-dimensional DCT-based compression circuits for one-chip cameras," *IEEE J. Solid-State Circuits*, vol **32** (12), pp 2030-2041, 1997.

[7] V. Gruev, and R. Etienne-Cummings, "Implementation of steerable spatiotemporal image filters on the focal plane, " *IEEE T. Circuits and Systems II*, vol. **49** (4), 2002.

[8] Z. Zhou, B. Pain, and E.R. Fossum, "Frame-transfer CMOS active pixel sensor with pixel binning," *IEEE Trans. Electron Devices*, vol. **44** (10), pp 1764-1768, 1997.

[9] A. Bandyopadhyay, J. Lee, R. Robucci, and P. Hasler, "A 80μW/frame 104x128 CMOS imager front end for JPEG Compression," *IEEE Int. Symp. on Circuits and Systems (ISCAS'05)*, pp 5318-5321, 2005.

[10] E. Artyomov, Y. Rivenson, G. Levi, and O. Yadid-Pecht, "Morton (Z) scan based real-time variable resolution CMOS image sensor," *IEEE Trans. Circuits and Systems for Video Technology*, vol. **15** (7), pp 947-952, 2005.

[11] E. Artyomov, and O. Yadid-Pecht, "Adaptive multiple resolution CMOS active pixel sensor," *IEEE Int. Symp. on Circuits and Systems (ISCAS'04)*, vol. **4**, pp 836-839, 2004.

[12] T. Hamamoto, K. Aizawa, Y. Egi, T. Hamamoto, M. Hatori, H. Maruyama, and J. Yamazaki, "On sensor image compression," *IEEE Signal Processing Society Workshop on VLSI Signal Processing*, pp 61-69, 1995.

[13] K. Aizawa, H. Ohno, Y. Egi, M. Hatori, and J. Yamazaki, "Image sensor for compression and enhancement," *IEEE Trans. Circuits and Systems for Video Technology*, vol. **7** (3), pp 543-548, 1997.

[14] U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings, “Temporal change threshold detection imager,” *IEEE International Solid-State Circuits Conference (ISSCC’05)*, vol. **1**, 2005.

[15] D. Leon, S. Balkir, K. Sayood, and M.W. Hoffman, “A CMOS imager with pixel prediction for image compression,” *IEEE Int. Symp. on Circuits and Systems (ISCAS’03)*, vol. **4**, pp 776-779, 2003.

[16] A. Graupner, J. Schreiter, S. Getzlaff, and R. Schuffny, “CMOS image sensor with mixed-signal processor array,” *IEEE J. Solid-State Circuits*, vol **38** (6), pp 948-957, 2003.

[17] E.R. Fossum, “CMOS image sensor: Electronic camera-on-a-chip,” *IEEE Trans. Electron Devices*, vol. **44**, pp 1689-1698, 1997.

[18] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, “Pedestrian Detection Using Wavelet Templates,” in *Computer Vision and Pattern Recognition*, pp 193-199, 1997.

[19] A. Olyaei and R. Genov, “Focal-Plane spatially oversampling CMOS image compression sensor,” *IEEE T. Circuits and Systems I*, vol. **49** (1), pp.26-34, 2007.

[20] A. Olyaei, and R. Genov, “Algorithmic Delta-Sigma Modulated FIR Filter,” *IEEE Int. Symp. on Circuits and Systems (ISCAS’06)*.

[21] J. Robert, G.C. Temes, V. Valencic, R. Dessoulavy, and P. Deval, “A 16-Bit Low-Voltage CMOS A/D Converter,” *IEEE J. Solid-State Circuits*, vol. **SC-22**, pp 157-163, 1987.

[22] O.J.A.P. Nys and E. Dijkstra, “On Configurable Oversampled A/D Converters,” *IEEE J. Solid-State Circuits*, vol. **28** (7), pp 736-742, 1993.

[23] I. Galton, H.T. Jensen,“Delta-Sigma Modulator Based A?D Conversion without Oversampling,” *IEEE T. Circuits and Systems II*, vol. **42** (12), 1995.

[24] G. Mulliken, F. Adil, G. Cauwenberghs, and R. Genov, “Delta-Sigma Algorithmic Analog-to-Digital Conversion,” *IEEE Int. Symp. on Circuits and Systems (ISCAS’02)*, Scottsdale, AZ, May 26-29, 2002. ISCAS’2002.

[25] A. Olyaei, and R. Genov, “CMOS Wavelet Compression Imager Architecture,” *IEEE CAS Emerging Technologies Workshop*, St. Petersburg, Russia, June 23-24, 2005.

[26] A. Olyaei, and R. Genov, “Mixed-Signal CMOS Haar Wavelet Compression Imager Architecture,” *Midwest Symposium on Circuits and Systems (MWSCAS05)*, Cincinnati, Ohio, Aug. 7-10, 2005.

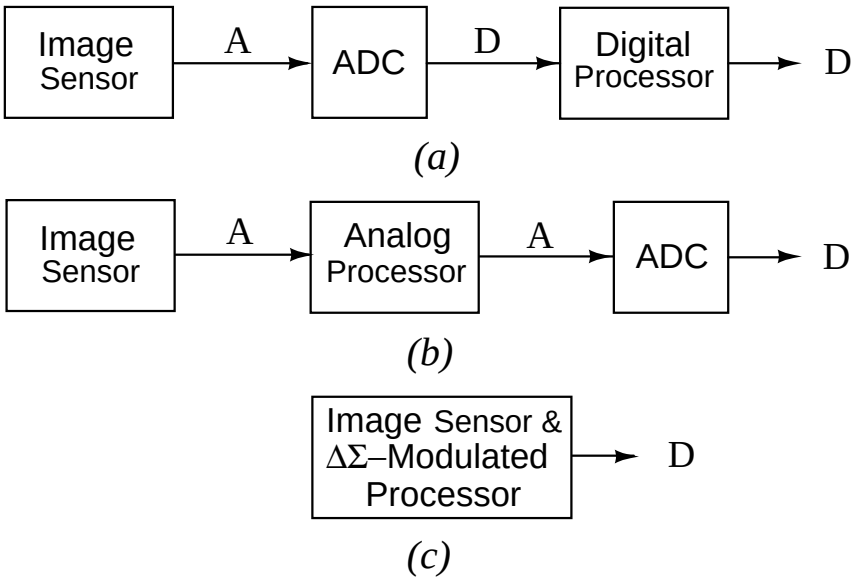

Fig. 1. Video block-matrix and convolutional transform computing architectures with *(a)* a digital processor, *(b)* an analog processor, and *(c)* the computational image sensor.

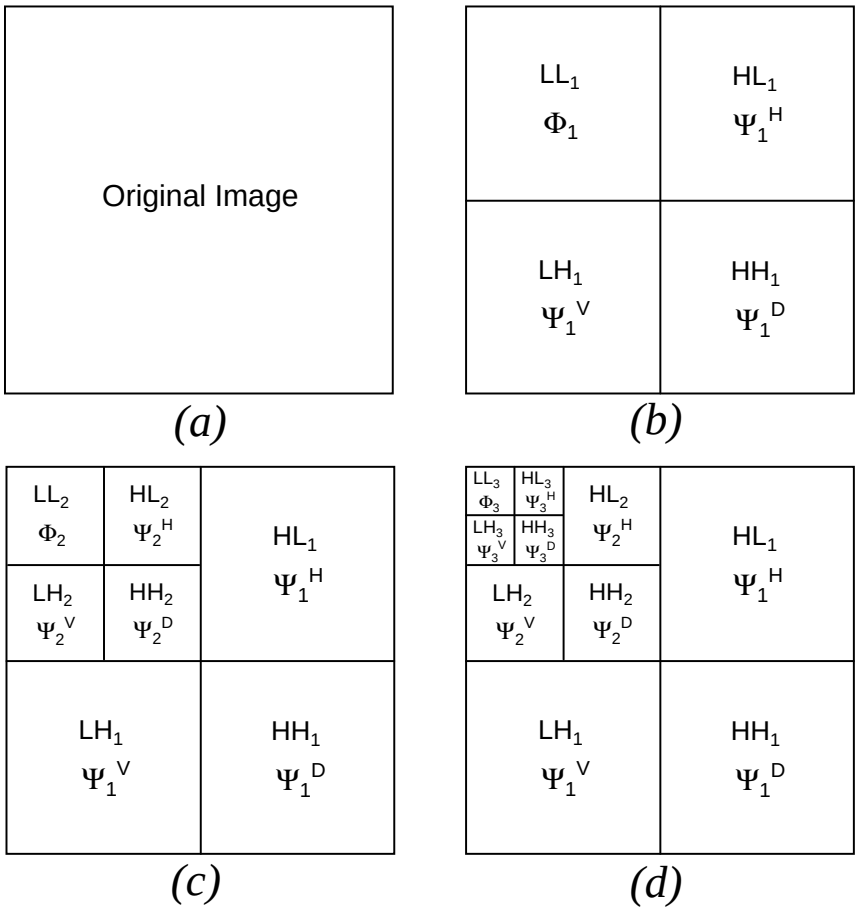

Fig. 2. An illustration of spatial kernels of two-dimensional *(b)* one-level, *(c)* two-level, and *(d)* three-level Haar wavelet transforms.

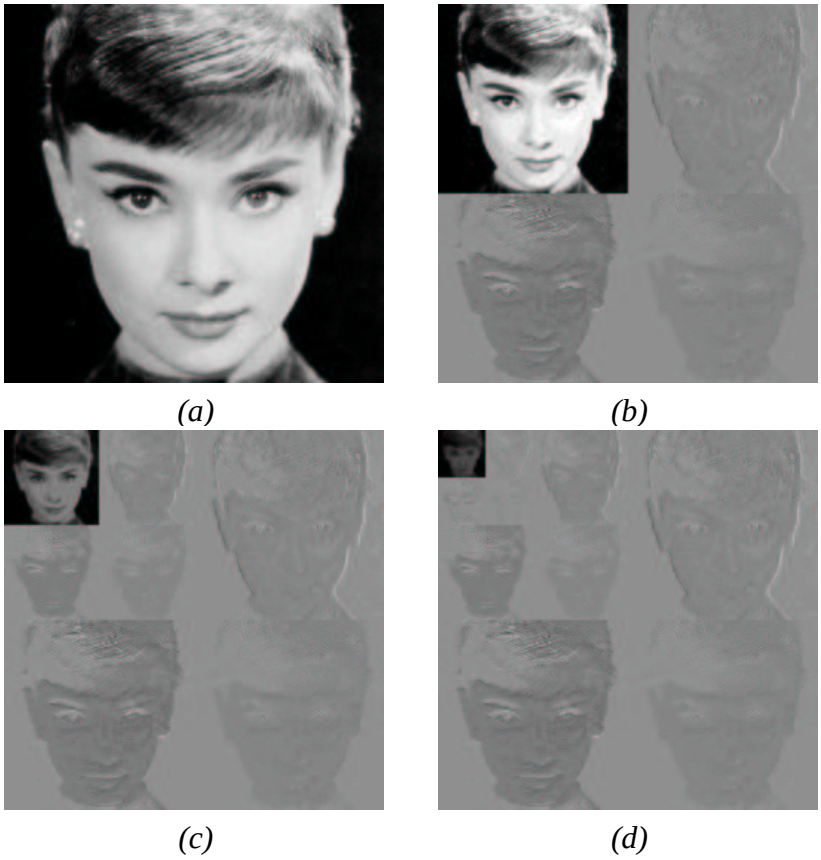

Fig. 3. *(a) Audrey* and computed *Audrey*'s Haar transforms: *(b)* one-level, *(c)* two-level, and *(d)* three-level transforms.

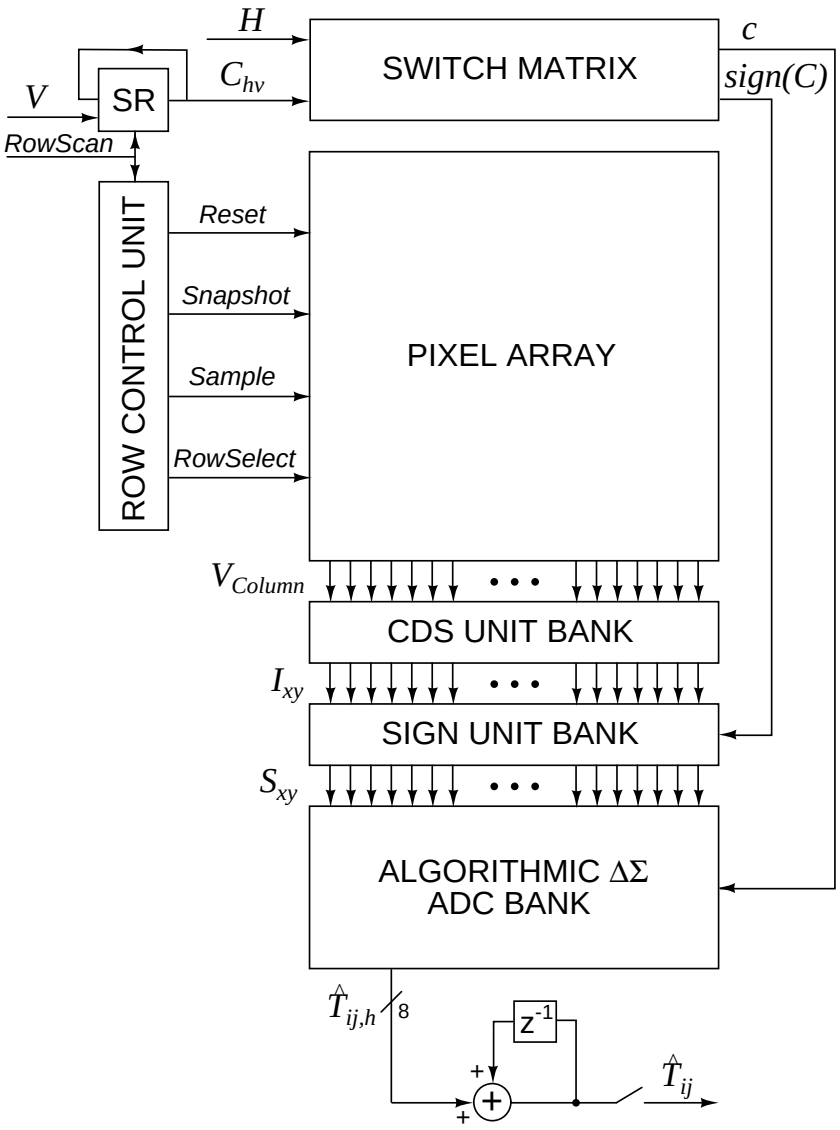

Fig. 4. Top-level architecture of the focal-plane spatially-oversampling CMOS image compression sensor. Digital accumulation and thresholding (not shown) blocks are implemented off-chip.

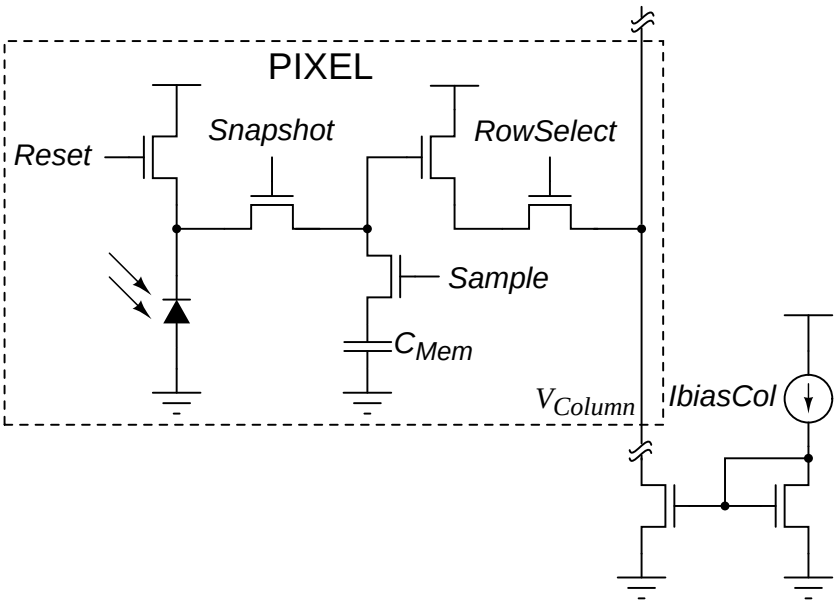

Fig. 5. Photodiode-based active pixel sensor circuit.

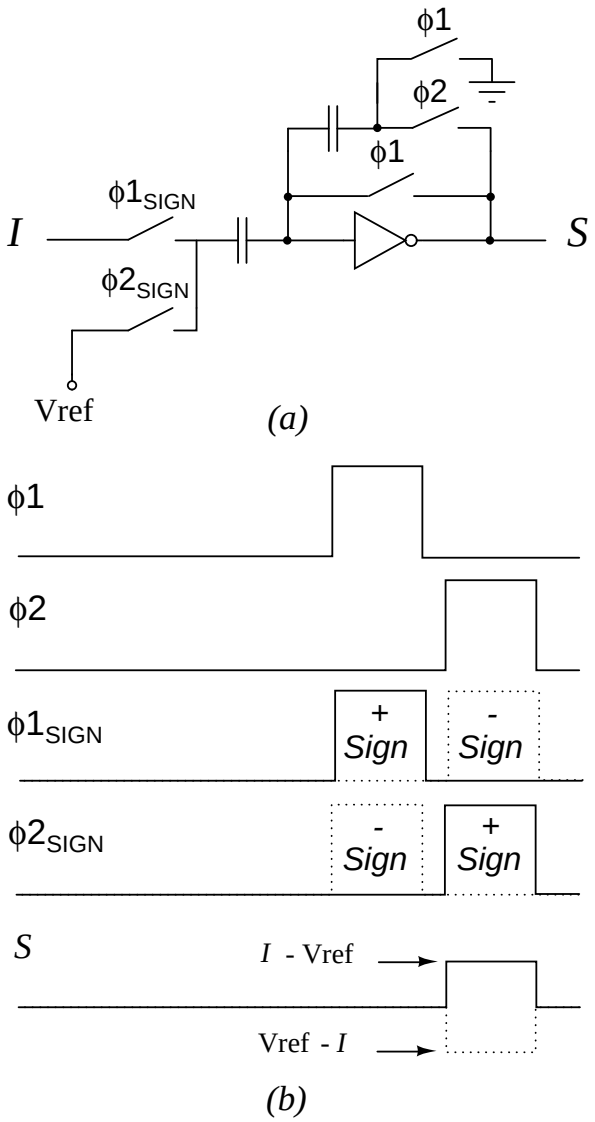

Fig. 6. *(a)* The sign unit circuit in the sensory processor. *(b)* Sign circuit timing diagram; for a continuous range of signed outputs, $Vref = min\{I\}$.

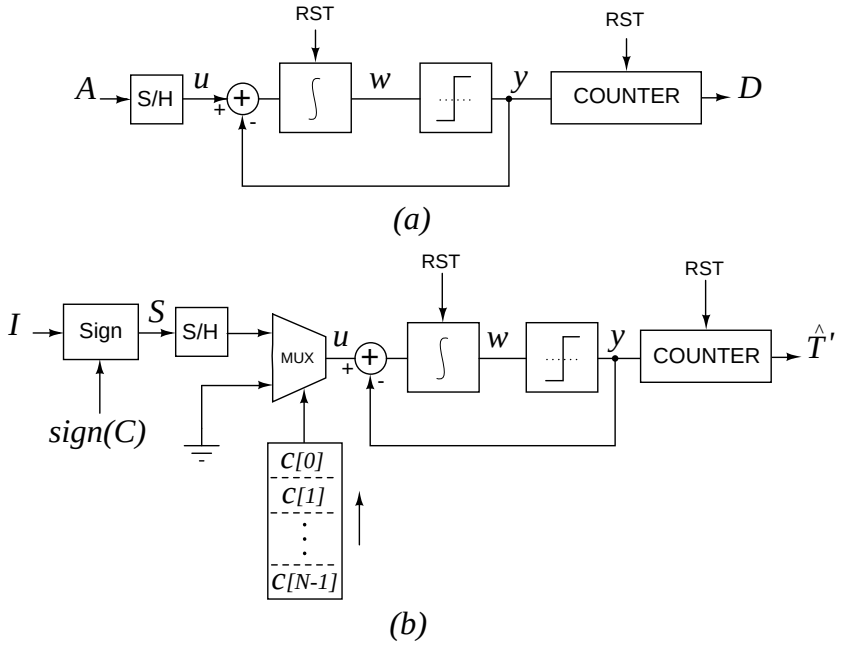

Fig. 7. *(a)* First-order $\Delta\Sigma$ incremental A/D converter. *(b)* $\Delta\Sigma$-modulated multiplying ADC. The sign unit circuit shown in Figure 6*(a)* also performs the sample-and-hold operation. Here the S/H cell is explicitly shown for clarity.

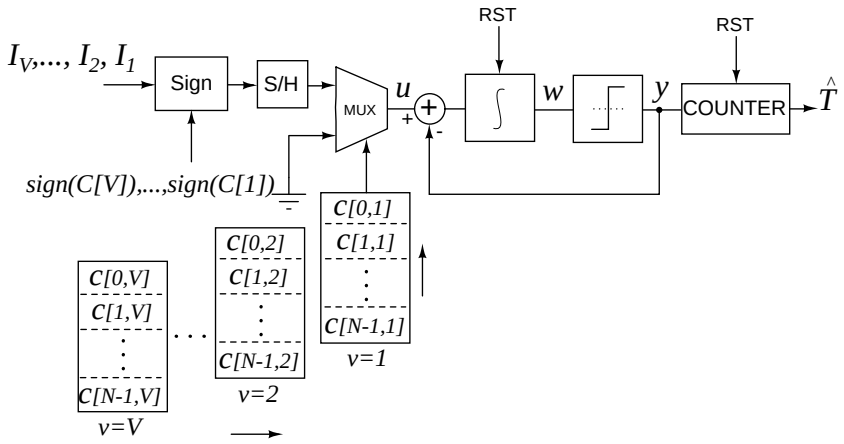

Fig. 8. $\Delta\Sigma$-modulated weighted averaging ADC for $j = 1$ and given i and h. The ADC samples V adjacent pixels in one column, weights each by coefficient C_v, and concurrently quantizes their sum.

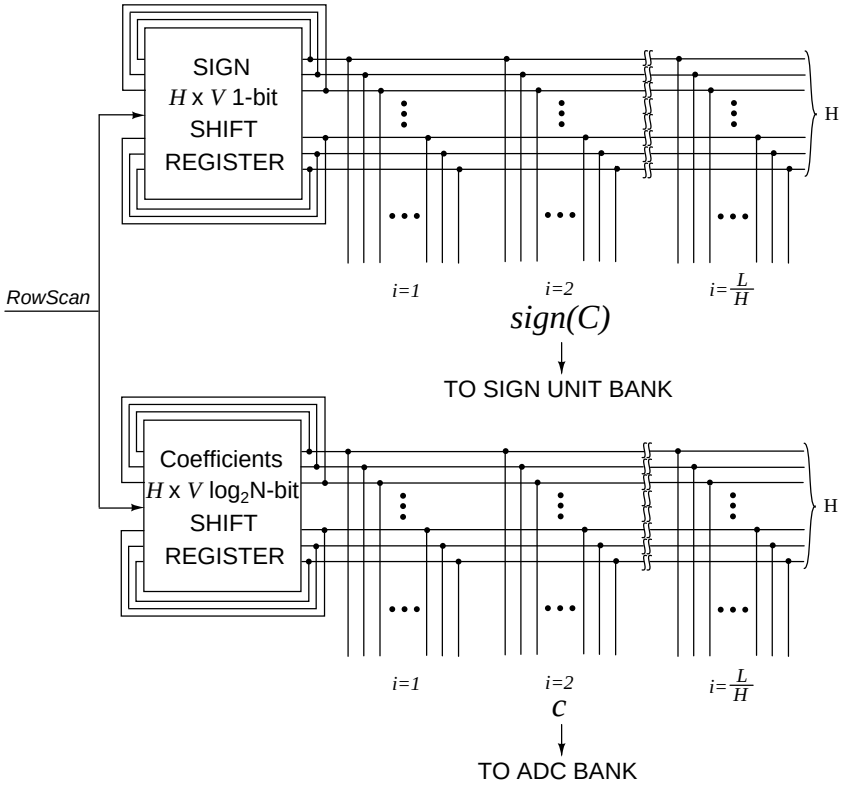

Fig. 9. Block diagram of the switch matrix.

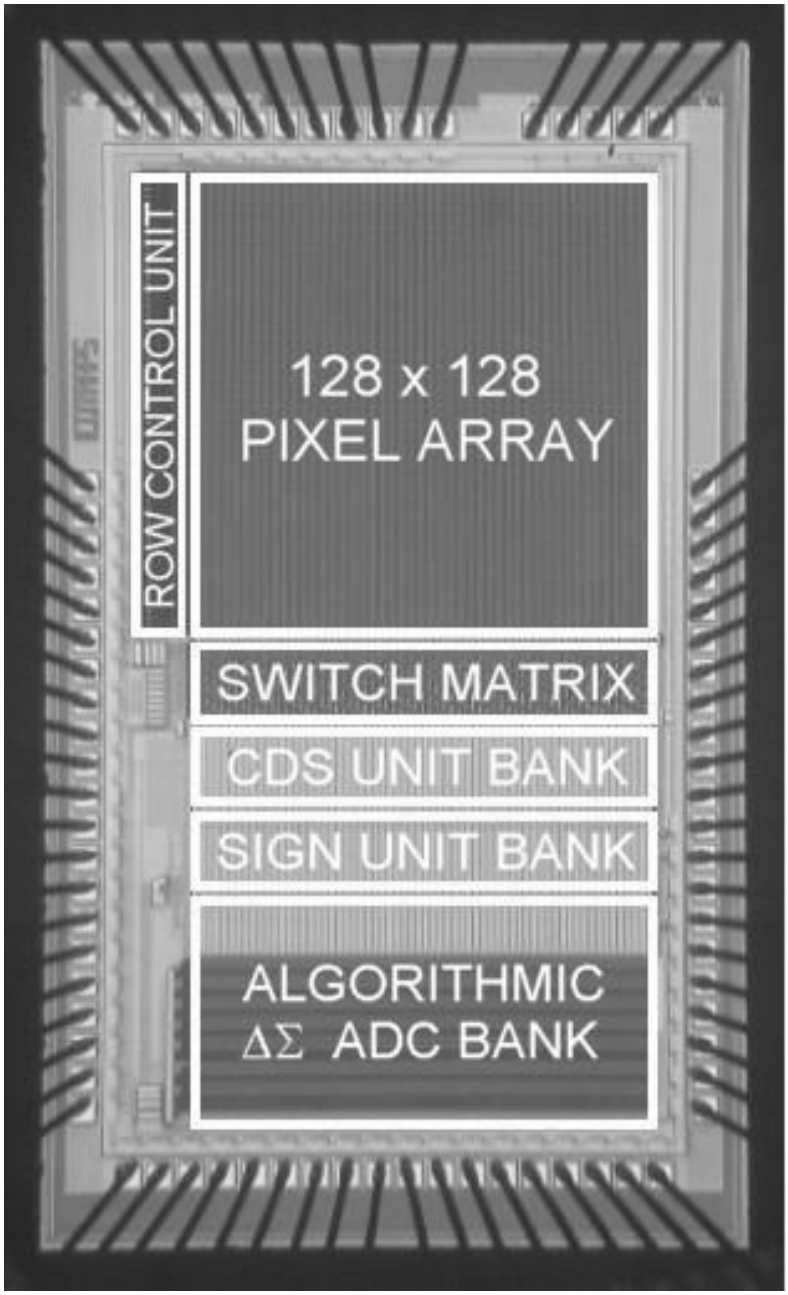

Fig. 10. Die micrograph of the focal-plane spatially-oversampling CMOS image compression sensor. The integrated 3.1mm $\times$ 1.9mm prototype was fabricated in a 0.35 μm CMOS technology.

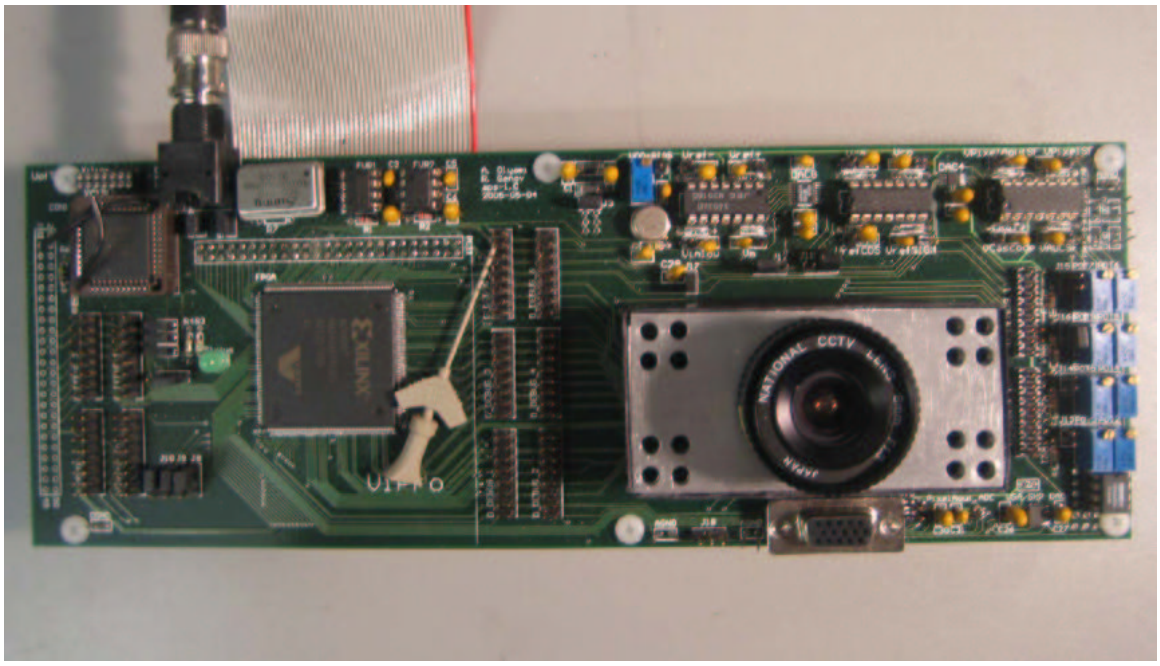

Fig. 11. The printed circuit board for experimental characterization of the 0.35 μm CMOS prototype of the sensory image processor.

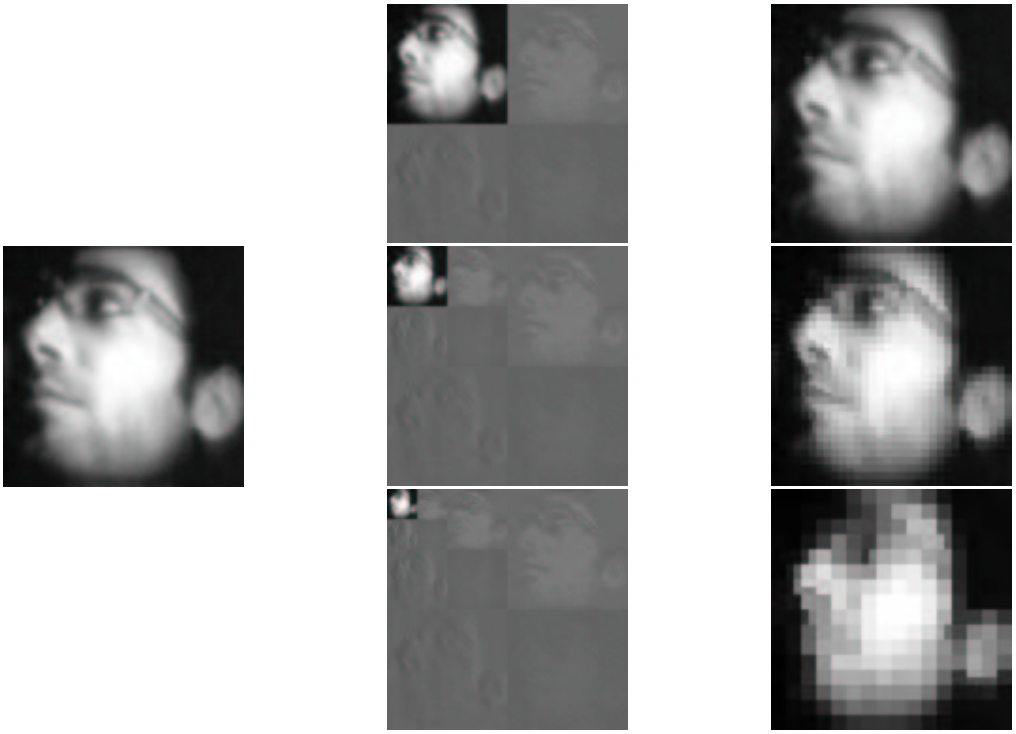

Fig. 12. *(a)* An image captured by the CMOS image compression sensor at 30 fps. *(b)* Experimentally recorded one-level (*top*), two-level (*center*), and three-level (*bottom*) Haar wavelet transforms of the image in (a) computed on the CMOS image compression sensor. *(c)* Reconstructed images for one-level (*top*), two-level (*center*), and three-level (*bottom*) Haar wavelet transforms for the same compression threshold. Compression ratios from top to bottom are: 5.33, 20.27, and 41.53.

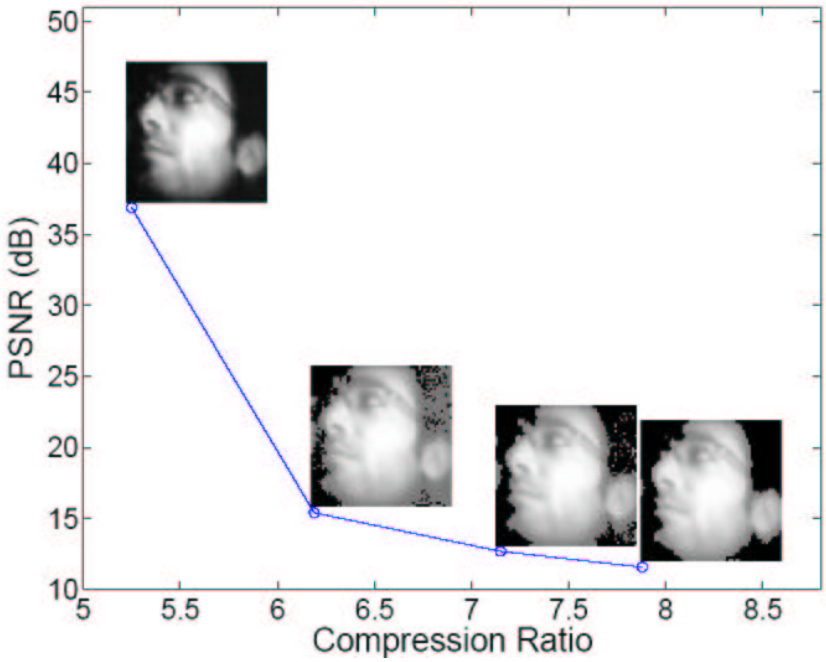

Fig. 13. Reconstructed images obtained by decompression of the experimentally computed one-level transform of the original image (top of Fig. 12(b)) for varying compression thresholds.

TABLE I
SUMMARY OF CHARACTERISTICS

Technology	0.35 μm CMOS
Area	3.1mm $\times$ 1.9mm
Supply Voltage	3.3 V
Array Size	128 $\times$ 128 pixels
Pixel Size	10.45 μm $\times$ 10.45 μm
Fill Factor	42%
Frame Rate	30 fps
Kernel Size	2$\times$2 — 8$\times$8 programmable
Throughput	4 GMACS in HDTV 1080i DCT
Optical Dynamic Range	105 dB
Dark Current	17.5 fA/pixel
ADC Power Consumption	4.3 mW
Output Resolution	8-bit