

Compressive Sensing on a CMOS Separable-Transform Image Sensor

Ryan Robucci, Jordan Gray, Leung Kin Chiu, Justin Romberg, Paul Hasler

Abstract—This paper demonstrates a computational image sensor capable of implementing compressive sensing operations. Instead of sensing raw pixel data, this image sensor projects the image onto a separable 2-D basis set and measures the corresponding expansion coefficients. The inner products are computed in the analog domain using a computational focal-plane and an analog vector-matrix multiplier. This is more than mere post-processing, as the processing circuitry is integrated as part of the sensing circuitry itself.

We implement compressive imaging on the sensor by using pseudo-random vectors called *noiselets* for the measurement basis. This choice allows us to reconstruct the image from only a small percentage of the transform coefficients. This effectively compresses the image without any digital computation, reduces the throughput of the analog-to-digital converter. The reduction in throughput has the potential to reduce power consumption and increase the frame rate.

The general architecture and a detailed circuit implementation of the image sensor are discussed. We also present experimental results that demonstrate the advantages of using the sensor for compressive imaging rather than more traditional coded imaging strategies.

Index Terms—imaging, image sensors, image sampling, intelligent sensors

I. HARDWARE FOR COMPRESSIVE IMAGE SENSING

Following the standard model for imaging, the amount of data a sensor must capture, quantize, and transfer for processing scales linearly with its resolution. As images are typically structured, they can be compressed by significant ratios with very little information being lost, making storage and transmission less costly. But as this compression is typically implemented on a digital signal processor (or other digital computing device), its benefits are realized only after the entire image has been converted to a digital representation and piped off of the sensor.

In this paper, we present an imaging architecture that compresses the image before it is converted to a digital representation, Fig. 1(a), using analog processing to remap the signal, Fig. 1(b). The architecture is based on a computational image sensor, which we call the *transform imager*. While most high-density imager architectures separate the image readout from the computation, the transform imager takes advantage

Ryan Robucci is with the Department of Computer Science and Electrical Engineering at the University of Maryland Baltimore County, Baltimore, MD 21250 USA (e-mail:ryan.robucci@umbc.edu), Jordan Gray, Leung Kin Chiu, Justin Romberg and Paul Hasler are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-250 USA (e-mail:phasler@ece.gatech.edu). Part of the development of IC described was due to DARPA funding through the Integrated Sensing and Processing Program. Submitted to Proc. IEEE April 22, 2009. Revised Dec 31, 2009.

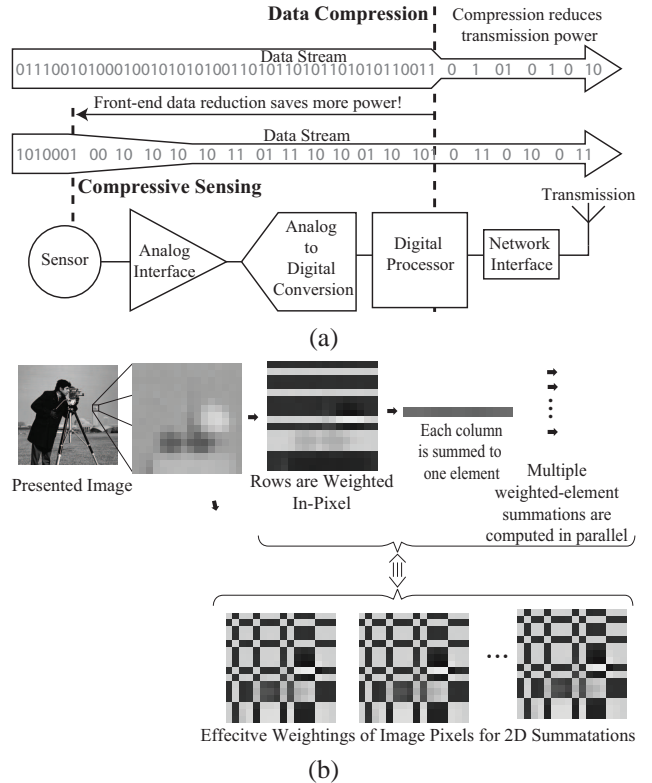


Fig. 1: Compressive Sensing System Design (a) Total data manipulation and power is reduced in the chain from sensor to transmitter by sampling less often instead of just compressing data in the digital domain. (b) Illustration of computation integrated into analog sensor interface circuitry. Time varying row-weightings and parallelized column-weightings are applied according to selected basis functions as summations are performed. The output of the analog system is a transformed version of the image. By utilizing an incomplete subset of basis functions, fewer sensory-computations and analog-to-digital conversions need to be performed.

of the required pixel read out to perform computations *in analog*. The imager is very flexible: by integrating focal-plane computation with peripheral analog computation circuitry, a variety of linear transformations can be performed, Fig. 2. Here, we configure the sensor to compute a separable transform of the pixelized data, and the computed coefficients are then quantized with an analog-to-digital converter (ADC). The compression comes from simply limiting the read-out to a subset of these coefficients.

The imager is fabricated using a 0.35- μm CMOS process. The current design has a 256×256 sensor array using a pixel size of $8 \times 8 \mu\text{m}^2$ implemented in 22.75 mm^2 area. Since the data compression is integrated into the sensor array at the front end of the system, all subsequent stages can receive the benefits of compression. Fewer numbers have to be transferred

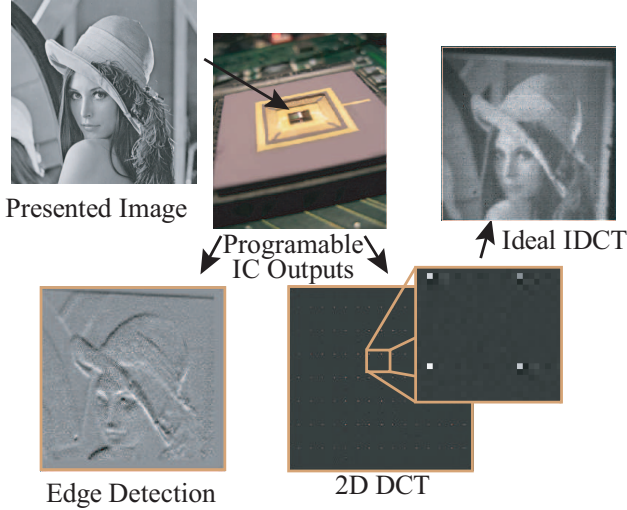


Fig. 2: Separable-Transform Imager Output. This imager can implement a variety of functions depending on how it is programmed. Shown here is edge-detection via a convolution with vector $[1, 2, 3, 0, -1, -2, -3]$ in the horizontal and vertical direction and block DCT. While the original image can be reconstructed, often the transformed result is more useful for processing, analysis, and transmission.

off of the sensor array and passed through the ADC, saving time and power. The digital image computations can also be moved away from the sensor, a feature that is particularly attractive if the camera is part of a distributed wireless sensor network that utilizes a central processing node, as the sensor itself will often have strict power and form-factor constraints.

The question, then, is what type of transform the imager should take to make the compression as efficient as possible. We compare two different sensing strategies. In the first, the imager measures a certain number of discrete cosine transform (DCT) coefficients, starting from low spatial frequencies and working outwards. The image is reconstructed — digitally and away from the sensor — by taking an inverse DCT. This choice of transform (and the ordering of the coefficients) is inspired by the JPEG compression algorithm: it is often the case that we can build a reasonable approximation to an image simply by keeping its low (spatial) frequency components and discarding its high frequency components.

The second sensing strategy is inspired by the recently developed theory of compressive sensing (CS) [1]–[4]. CS, which is reviewed in Section II, is based on a more refined image model than JPEG: rather than approximating an image using a small number of low frequency DCT coefficients, we use a small number of arbitrarily located *wavelet* (or other sparsifying transform) coefficients. From correlations of the image against pseudo-random “incoherent” basis functions, convex programming — again implemented digitally away from the sensor — can be used to simultaneously locate the important transform coefficients and reconstruct their values. The theory of CS suggests that from m of these correlations, we can compute an approximation to the image which is

almost as good as if we had observed the m most important transform coefficients directly. From a broad perspective, CS theory tells us that the amount of data the sensor must capture scales with the *information content* of the image (i.e. its compressibility) rather than its resolution.

This paper is organized as follows: An overview of compressive sensing is given in Section II. Several imaging systems based on compressive sensing are surveyed in Section III. Section IV presents the structure of the transform image sensor, the computation of our transform imager, and integrated test setup. Section V discusses using the transform image sensor for compressive sensing, including showing experimental results for this compressed sensing front end, as well as the resulting reconstruction. Integrating non-volatile analog memory and a versatile random access approach enables a variety of operations like multi-resolution and selective sensing; some of these extensions are discussed in Section VI. Final remarks are made in Section VII.

II. COMPRESSIVE SENSING

The image sensor gives us the freedom to measure the pixelized image using inner products with any sequence of (separable) basis functions. We would like to choose these basis functions so that: 1) the image can be reconstructed from the smallest number of measurements, and 2) the reconstruction can be computed in an efficient manner. The theory of *compressed sensing* [1]–[4] tells us that if we have a *sparse representation* for images of interest, the measurement functions should be chosen from a complementary *incoherent* basis.

Mathematically, the measurements the sensor makes can be written as a series of inner products:

$$y_1 = \langle \phi_1, P \rangle, \quad y_2 = \langle \phi_2, P \rangle, \dots, y_m = \langle \phi_m, P \rangle. \quad (1)$$

Above, P is the image we are acquiring and the ϕ_k are the measurement basis functions. As the computations in (1) happen after the image is captured (but before it is digitized), P and the ϕ_k can be taken as vectors. Even though the image P is naturally arranged as a 2-D $n \times n$ array, we will often treat it as a “rasterized” $n^2 \times 1$ vector for notational convenience. The entire measurement process can be written compactly as $Y = \Phi P$, where Φ is a $m \times n^2$ matrix, Y is a vector in \mathbb{R}^m and P is a vector in \mathbb{R}^{n^2} . Reconstructing P from the measurements Y is then a linear inverse problem. Since our goal is to take many fewer measurements than there are pixels in the image, $m \ll n^2$, the inverse problem is severely underdetermined. To counteract this, we incorporate a priori information about the structure of the image into the reconstruction.

The y_1, \dots, y_m can be used to calculate a projection onto the subspace of \mathbb{R}^{n^2} spanned by the ϕ_1, \dots, ϕ_m (this is trivial if the ϕ_k are orthogonal to one another). This points to one possible acquisition strategy: find a subspace in which we believe the energy of the image we are trying to sample is concentrated, and then choose the ϕ_k as a basis for this subspace. A good choice for this subspace, one motivated by the fact that most of the energy in a typical photograph-like image tends to be at low spatial frequencies, is the span of

the first m 2D DCT coefficients taken in the same “zig-zag” order as in the JPEG compression standard [5].

Projecting the image onto the subspace of low-frequency sinusoids is an effective way to get a very low-dimensional, smoothed-out (“blurry”) approximation to the image. However, details in the image (sharp edges and localized features) resolve slowly as m increases — these local features are diffused across all spatial frequencies, making them hard to represent using the DCT. This type of approximation also often suffers from “ringing” around the edges (Gibbs phenomena).

We can construct better low-dimensional approximations to images using the wavelet transform [6], [7]. Wavelets give us a representation for images that is sparse and automatically adapts to local structure; only a small percentage of the wavelet coefficients are significant, and they tend to cluster around edge contours and other singularities. This sparseness allows us to construct an accurate and sharp approximation of medium-size images (one megapixel, say) simply by retaining around 2–5% of their most significant wavelet coefficients and throwing away the rest. That such accurate approximations can be formed so easily is the reason that the wavelet transform lies at the heart of nearly every competitive image compression algorithm, including the JPEG2000 image compression standard [8].

There is a subtle difference between the two types of approximation described above; in the first, we are projecting the image on to a fixed subspace (spanned by low frequency sinusoids), while in the second we are *adapting* the subspace (spanned by the wavelet basis functions corresponding to the largest wavelet coefficients) to the image. It is this adaptation that accounts for almost the entire advantage of wavelet-based approximation over DCT-based approximation. It is also less straightforward to exploit these advantages in our acquisition system: we would like to measure only the significant wavelet coefficients, but we have no idea which coefficients will be significant beforehand.

The theory of compressive sensing shows us how we can choose a set of ϕ_k to take advantage of the fact that the image is sparse. One of the central results of CS says that if the image is sparse in an orthonormal basis Ψ (e.g. the wavelet domain), then we should select the ϕ_k from an alternative basis Φ' that is very different than (or *incoherent with*) Ψ . Formally, the coherence between two image representations is simply the maximum correlation of a basis vector from Ψ with a basis vector from Φ' :

$$\mu(\Psi, \Phi') = \max_{\substack{\psi \in \Psi \\ \phi \in \Phi'}} |\langle \psi, \phi \rangle|.$$

We will always have $\mu \geq 1/n$, and so we will call Ψ and Φ' “incoherent” if $\mu(\Psi, \Phi') \approx 1/n$. Fortunately, there is a known orthogonal system, called the *noiselet transform* [9], that is incoherent with the wavelet transform. Examples of noiselets are shown in Fig. 7; they are binary, 2-D separable basis vectors which look like pseudo-random noise. They are also spread out in the wavelet domain. If Ψ is a 2-D Daubechies-8 wavelet transform for $n = 256$ and Φ' is the noiselet transform, then $\mu(\Psi, \Phi') \approx 4.72/n$.

Given the measurements Y , there are a variety of ways we can reconstruct the image. One popular way, which is provably effective, is to set up an optimization program that encourages the image to be sparse in the Ψ domain while simultaneously explaining the measurements we have made:

$$\min_X \|\Psi(X)\|_1 \quad \text{subject to} \quad \Phi X = Y. \quad (2)$$

This program sorts through all the of images consistent with what we have observed and returns the one with smallest ℓ_1 norm in the Ψ domain. The ℓ_1 norm promotes sparsity — sparse vectors have smaller ℓ_1 norm than non-sparse vectors with the same energy. When there is noise or other uncertainties in the measurements, (2) can be relaxed to

$$\min_X \|\Psi(X)\|_1 \quad \text{subject to} \quad \|\Phi X - Y\|_2 \leq \epsilon, \quad (3)$$

where ϵ is chosen based on the expected noise level.

When Φ is chosen from an incoherent system and the image P we are trying to recover is sparse, the recovery programs (2), (3) come with certain theoretical performance guarantees [1], [2], [4]. From m incoherent measurements, (2) will produce an approximation \hat{P} to P that is as good as a wavelet approximation using on the order of $\sim m/\log^4 n$ terms. Numerical experiments suggest even more favorable behavior; \hat{P} is often as good as a $\approx m/4$ term wavelet approximation [10], [11]. The essential message here is that the number of measurements we need to faithfully acquire P depends mainly on how well we can compress P (i.e. its inherent complexity) and not on the number of pixels it has. If the vast majority of the information about the image is contained in 2-5% of the wavelet coefficients, then we can acquire it using a number of measurements which is $\approx 8 - 20\%$ of the total number of pixels.

There are two extensions to the CS methodology that help produce better image reconstructions in practice. The first is to make Ψ a redundant (or *undecimated*) wavelet transform [12]; this choice tends to produce images with fewer local artifacts. Another way to boost image reconstruction quality is by using an *iterative reweighting* scheme [13]. After obtaining an initial solution \hat{P}_0 to (3) (or (2)), we re-solve the program, but with a weighted ℓ_1 norm:

$$\min_X \|W \cdot \Psi(X)\|_1 \quad \text{subject to} \quad \|\Phi X - Y\|_2 \leq \epsilon$$

The weighting matrix W is diagonal, and its entries are inversely proportional to the magnitudes of the corresponding entries in $\Psi(\hat{P}_0)$

$$W_{ii} = \frac{1}{|\Psi(\hat{P}_0)_i| + \gamma},$$

for small $\gamma > 0$. This reweighting can be applied multiple times in succession, with the weights adjusted according to the transform coefficients of the previous solution.

III. COMPRESSIVE SENSING-INSPIRED IMAGING SYSTEMS

In this section, we briefly describe other compressive imaging architectures that have been proposed in recent years. Generally speaking, in these alternative architectures the

compressive measurements (inner products with measurement basis functions as in (1)) are taken by manipulating the light field using some type of high-resolution, spatial light modulation or optical dispersion, and then measured using a low-resolution sensor array, the output of which is immediately converted into the digital domain. In contrast, the optics in our transform imager are completely standard, and the image is measured using a high-resolution sensor array. Before the pixel values are converted to digital, we use low-power analog electronic processing to take the compressive measurements, and digitize the result. Another imager architecture that uses random electronic modulation is also discussed.

In [14], the concept of a high-resolution SLM followed by a low-resolution sensor array is taken to the extreme with a *single pixel camera*. As light enters the system, it is focused with a lens onto a digital micromirror device (DMD). The DMD is a large array (1024×768 in this case) of tiny mirrors, each of which can be individually controlled to point in one of two directions. In one of these directions lies another lens which focuses the “output” of the DMD onto a single photodetector. The net effect is that the photodetector measures an inner product of the image incident on the DMD with a single binary (0-1 valued) basis function. The measurements are taken serially, and reconstructed using the same techniques discussed in Section II. Because it uses only a single sensor, the single pixel camera is very well-suited for imaging modes where detectors are expensive; see for example its application to low-light imaging with photomultiplier tubes [15] and terahertz imaging [16].

The Compressive Optical MONTAGE Photography Initiative (COMP-I) at Duke University has developed a number of compressive imaging architectures [17]–[19] with the goal of creating very thin cameras. One such architecture, described in detail in [17], compressive imaging is implemented using a camera with multiple apertures. There, a block transform is implemented on the focal plane as follows. The incoming light meets an array of lenses (or “lenslets”), each of which focuses the entire image onto a different section of the sensor array. Before each image hits its part of the array, it passes through a high-resolution aperture mask. Each section of the pixel array then measures the correlation of the image with the code determined by the aperture mask; if there are multiple pixel sensors per aperture, the inner product is broken up locally.

In [20], a camera with a “random lens” is detailed. In this work, the lens of a camera is replaced with a random reflective surface. The effect is that a single “pixel” in the input image contributes to several different pixels in the measurement array, the locations of which may be completely non-local. The random reflective surface is created in an uncontrolled way, so the camera must be calibrated to discover the sensing matrix Φ . There is a trade-off here, as the sparser the sensing matrix Φ is, the easier it is to discover (by training the camera on a relatively small set of randomly generated but known input images) and compute with, but the less like a compressive sampling matrix it is—sparse matrices have high coherence with the wavelet transform (which is also sparse).

The approaches discussed thus far utilize moving parts and/or specialized optics to perform optical computations,

while our transform imager utilizes analog electronic computations. Another approach using electronic computations is a CMOS convolution imager architecture introduced in [21]. Like the transform imager, this convolution imager is based on a CMOS pixel array which does local, per-pixel computations. As the analog image data sits on the sensor array, it is convolved with a random, *binary* pulse sequence of 1’s and -1 ’s, and the result is randomly subsampled (convolution with a random pulse sequence followed by subsampling comes with the same compressive sampling guarantees as sampling in an incoherent basis, see [22], [23]). The *binary* weights for the random pulse sequence are passed through the array using a circular shift register initialized with a random sequence. This imager does not rely on a separable basis as our transform imager does, but it lacks the ability to use adaptable or analog weights, perform other computations such as DCT, or tailor measurements to extract details from portions of the image plane.

IV. TRANSFORM IMAGE SENSOR

The separable transform image sensor uses a combination of focal-plane processing performed directly in the pixels, and an on-die, analog, computational block to perform computation before the analog-to-digital conversion occurs. Unlike traditional imagers, this imager performs computation on-chip and in-pixel. The primary computation performed is a separable matrix transformation. This sensor includes a novel overlapping block scheme allowing 8×8 general separable 2-D convolutions and 16×16 block transforms.

The fundamental capability of this imager can be described as a matrix transform: $Y_\sigma = A^T P_\sigma B$, where A and B are transformation matrices, Y is the output, P is the image, and the subscript σ denotes the selected sub-region of the image under transform. The region σ is a 16×16 pixel block starting at an offset $(8m, 8n)$ where m and n are positive integers. The offset increments are smaller than the support region to allow transforms that can reduce or eliminate blocking artifacts, such as separable convolutions up to 8×8 . These separable transform capabilities are demonstrated in hardware to be able to perform compressive sensing.

The first computation is performed at the focal plane, in the pixels, using a computational sensor element as depicted in Fig. 3(a). It uses a differential transistor pair to create a differential current output that is proportional to a multiplication of the amount of light falling on the photodiode and the differential voltage input.¹ This operation is represented in Fig. 4 as the element for the P_σ block. The electrical current outputs from pixels in a column add together, obeying Kirchhoff’s current law. This aggregation results in a weighted summation of the pixels in a column, with the weights being set by the voltages entered into the left of the array. With a given set of voltage inputs from a selected row of A , every column of the computational pixel array computes its weighted summation in parallel. This parallel computation is of key importance, reducing the speed requirements of the individual

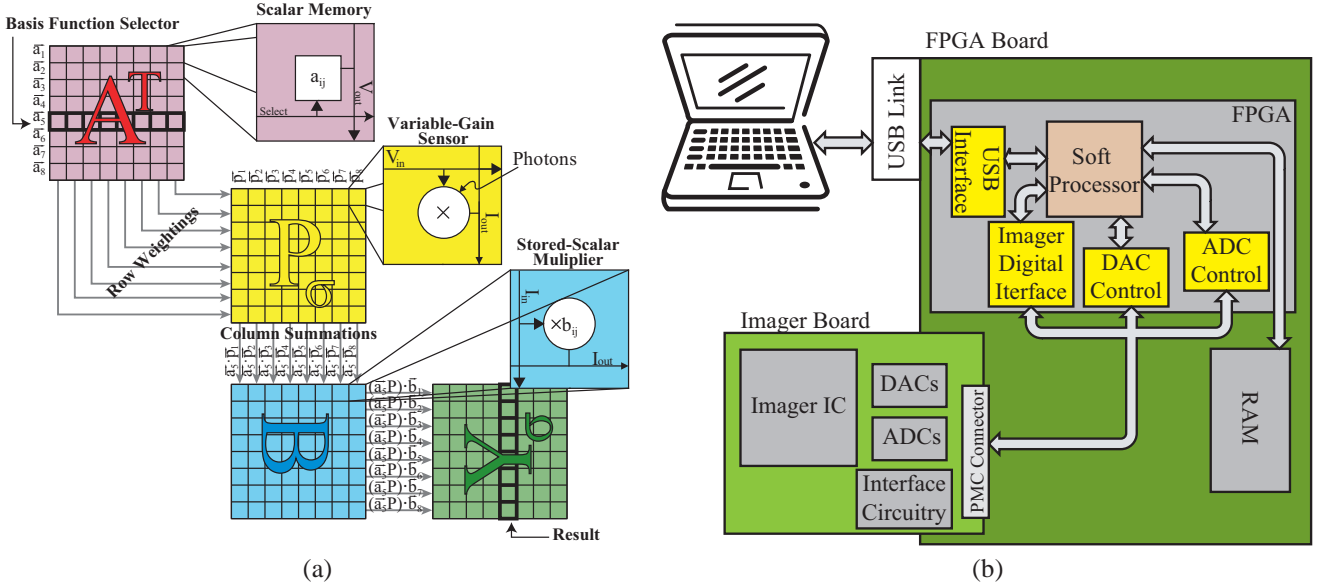


Fig. 3: Basics of the Transform Imager Approach. (a) Block matrix computation performed in the analog domain. Illustrated here as an 8×8 block transform, both a computational pixel array and an analog vector-matrix multiplier are used to perform signal projection before data is converted into the digital domain. (b) A daughter card houses the computational image sensor IC along with supporting components. This card mounts on an FPGA board through a standard PCI mezzanine connector. Communication with the PC is done through USB.

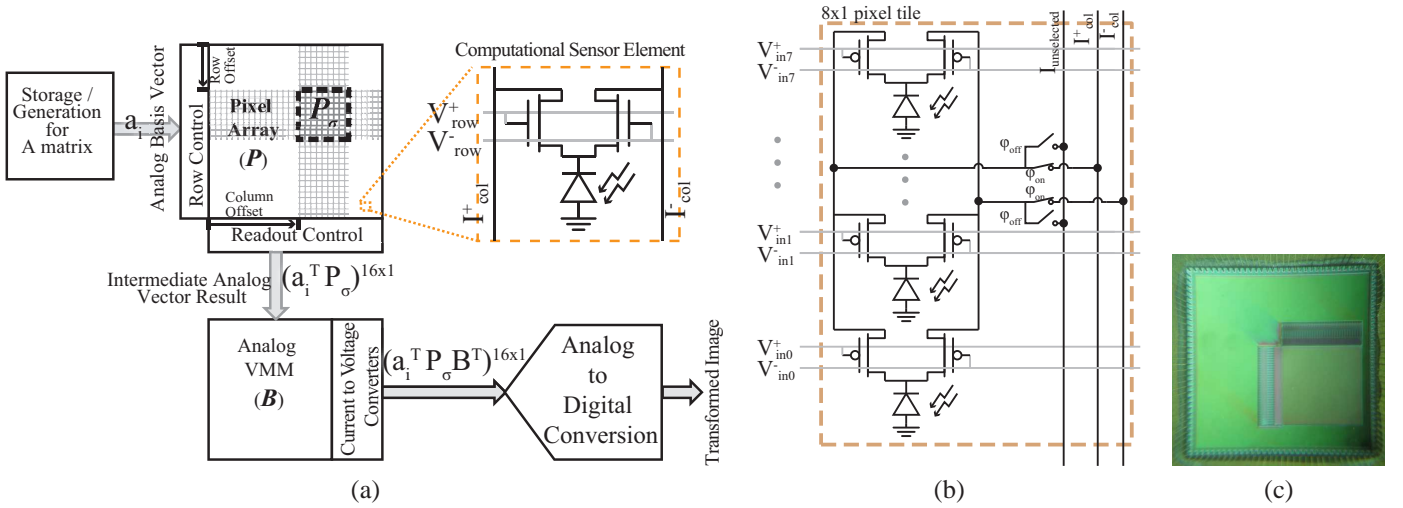


Fig. 4: Separable-Transform Imager IC. (a) Block Diagram of the Chip. The chip enables the computation as well as sensing at the pixel elements. (b) 8×1 Pixel Tile Schematic. We connect the pixels in groups of 8 to minimize the parasitic leakage currents as well as parasitic line capacitance of the non-photodiode p-n junctions due to transistor source-drain junctions. (c) Die Photograph of the Transform Imager IC. The $5 \text{ mm} \times 5 \text{ mm}$ IC was fabricated in a 350 nm CMOS process available through MOSIS.

computational elements.

The second computation is performed in an analog vector-matrix multiplier (VMM) [24]. This VMM may be designed so

¹The term differential refers to the technique of encoding signals in the difference of two complimentary signals. The two signals are called a differential pair and the signal encoded in their difference is called a differential signal. The average of the two signals is called the common-mode signal, and is present for practical reasons even though information is usually not encoded in it. The term differential is also used to denote techniques and circuits that process differential signals, usually involving complementary circuit components. Differential signaling and processing is more robust to system noise that affects the complementary signals equally, and minimizes distortion [24].

that it accepts input from all of the columns of the pixel array, or it can be designed with multiplexing circuitry to only accept a time-multiplexed subset of the columns. This decision sets the support region for the computation. The implementation used for these experiments uses the time-multiplexed column option. The elements of the VMM use analog floating-gate transistors to perform multiplication in the analog domain. Each element takes the input from its column and multiplies it by a unique, reprogrammable coefficient. The result is an electrical current that is contributed to a shared column row. Using the same automatic current summation as the image sensing P matrix, a parallel set of weighted summations occur,

resulting in the second matrix operation.

The setup is shown in Fig. 3(b). The image sensor IC is mounted on a custom PC board, which includes ADCs, DACs, current measurement, and other minor interfacing components. Physically, the imager board is a daughter card of the FPGA motherboard. The FPGA is configured to implement a soft-core processor and collection of custom hardware interface components described in VHDL. The supporting components handle timing-critical signal flow, allowing for more manageable but less timing-sensitive C code to be used on the soft-core processor. A supporting VHDL module also handles interfacing to the external USB IC for communicating with a computer. The user interface to the system is a set of MATLAB functions which encapsulate the IC control. The user has the ability to program an arbitrary set of coefficients into the A and B matrices. A DCT is an example of such a transformation matrix. The user can then capture the image, which undergoes the on-chip transformation.

CMOS imaging technology can be implemented on standard, relatively low-cost CMOS processes. CMOS implementation enables integration of large amounts of computational circuitry with sensor and interface circuitry. By integrating circuit components into the image sensor, such as in-pixel ADCs [25], CMOS technology has become competitive in the high-end camera market. Other aggressive circuit integrations in CMOS imaging technology provide additional computationally significant gains, such as random image access [26], dynamic field of view capabilities [27], multi-resolution image sensing [28], and biologically inspired abilities such as edge enhancement [29].

The system IC, as shown in Fig. 4(a), is composed of the following: a random-access analog memory, row and column selection control, a computational pixel array, logarithmic current-to-voltage (I-V) converter, an analog vector-matrix multiplier, and a bidirectional I-V converter. Fig. 4(c) shows the die photo of the transform imager IC. We will describe these blocks in the following subsections.

A. Light Sensor and Interfacing

Fig. 4(b) shows a schematic of an 8×1 pixel tile. Each pixel is a photosensor and a differential transistor pair, providing both a sensing capability and a multiply-accumulate operation. The output of each pixel is a differential current¹ and it represents a multiplication of the light intensity falling on the photosensor by a weighting value represented by a voltage input. Pixels along a given row of the image plane share a single, differential, voltage input, which sets the multiplication factor for the row. Pixels along a column share a differential-current output line. Since the outputs of the pixels are currents, and currents onto a node sum together, an addition operation is performed. More specifically, this is a weighted summation, also known as a dot product. Within each tile is a switch which selectively allows the pixels in the tile to output to the column. When deselected, the pixels' currents are switched off of the column's output line and onto a separate fixed-potential line. Since only a sub-portion of the rows of the imager are read at a time, these switches result in a $1/8^{\text{th}}$ reduction of parasitic capacitance introduced by the deactivated

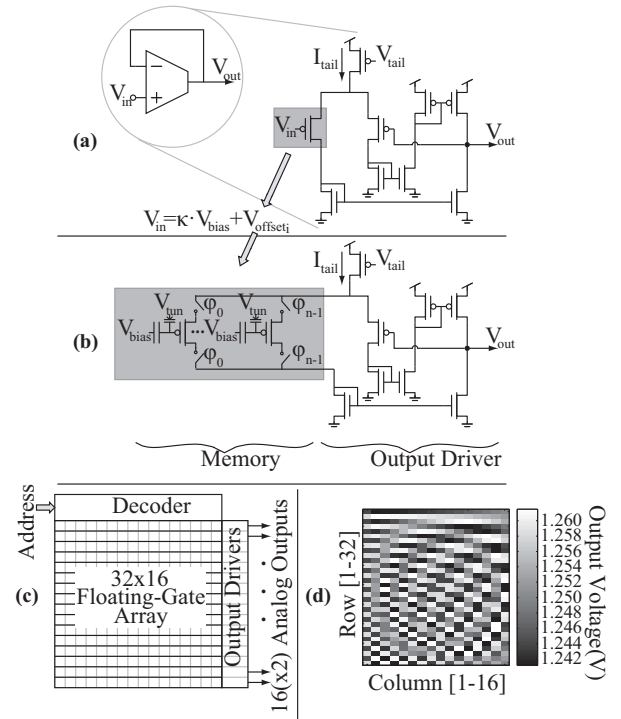


Fig. 5: Circuitry and Results from our Programmable Analog Waveform Generation. (a) Circuit Diagram for a Basic voltage buffer. (b) By replacing the input transistor with a selectable analog float-gate transistors, we transform the buffer circuit into an analog waveform generator. (c) Full Analog Memory Bank. (d) Measured Results of DCT Programmed as Differential Pairs. The differential errors were within $400 \mu\text{V}$, approximately our measurement precision.

pixels' drain junctions. Furthermore, these parasitic junctions introduce unwanted currents to the output line, since they themselves are photo diodes. Therefore, these switches reduce parasitic capacitances and currents to improve SNR.

B. Random Access Analog Memory

A compact analog memory structure was used to implement a storage for the A matrix, as shown in Fig. 5. It uses analog floating gates to store the coefficients of the transform matrix, which means that no digital memory or DACs are required to feed the analog weighting coefficients to the computational pixel array. The use of several DACs along with digital memory would be costly in size and power. Building the memory storage element into the voltage generation structure avoids unnecessary signal handling and conversion, saving size and power.

The basic structure of the analog memory is an amplifier connected as a follower, Fig. 5(a). However, one of the differential pair transistors has been replaced with a reprogrammable bank of selectable analog floating-gate PFETs (FGPFET), Fig. 5(b). Each FGPFET shares the same input V_{bias} , but is programmed to a particular voltage offset that sets the desired output voltage. The programming procedure inherently avoids issues of voltage offsets due to mismatches in the transistors and in the op-amp itself by directly monitoring the output in the programming cycle. The use of floating-gate transistors, which act much like standard transistors that have a programmable threshold voltage instead of a fixed one, is

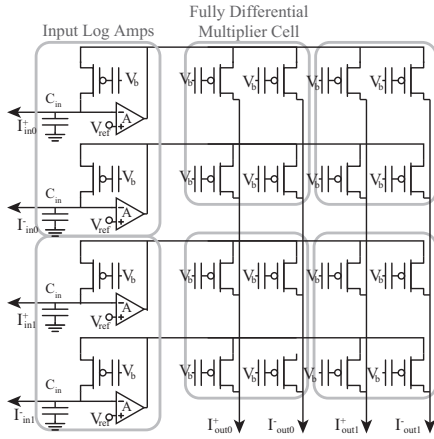


Fig. 6: Circuit Diagram of our Floating-Gate Vector-Matrix Multiplier (VMM) Components. We use this circuit to perform the second matrix multiplication on the image.

discussed in [30]. Generating 16 differential outputs, where the signal is encoded in the difference of two voltages, requires 32 amplifier structures. The storage of a 16×16 differential values requires a total of 32 rows \times 16 columns of floating gates. Stacking the amplifiers atop each other creates a 2-D array of floating-gates in a convenient structure for parallel addressing and fits well into floating-gate array programming schemes.

C. Current-Based Vector-Matrix Multiplication Design

The vector-matrix multiplier performs multiplication using addition on a logarithmic scale. First, the so-called logarithmic current-to-voltage converters generate voltages that are logarithmically proportional to the currents received from the pixel plane. These voltages are passed to an array of elements that create output currents that are exponentially related to the the voltages received from the current-to-voltage converters. These elements have individually programmed references that effectively shift the scales by which the input voltages are interpreted. This scale shift in the voltage domain corresponds to multiplication in the current domain. Current can vary over several orders of magnitude without saturating or distorting the behavior of the system while voltage range and precision is limited. Therefore, it is advantageous that the inputs and outputs of this hardware scheme are currents and that the intermediate signal is a voltage on a logarithmic—therefore compressed—scale.

With these concepts in mind, the back end circuitry of the imager was designed to handle the large line capacitances and high dynamic range of the pixel array’s output. Fig. 6 shows logarithmic transimpedance amplifiers on the left; the amplifiers sense and logarithmically convert the pixel current to a voltage. The logarithmic conversion is made possible by the subthreshold exponential voltage-to-current relationship of the feedback MOSFET, much like previous BJT or diode implementations [31]. The internal amplifiers, with labeled gain A_v , serve a dual purpose: they buffer the outputs of the converter, providing the current for the load transistors, and they create a large loop gain, fixing the input voltage. In addition, they lower the effective input impedance seen at

the drain of the feedback transistor by the gain, A_v . This low impedance generation is critical to sensing low currents in the presence of large capacitance. The amplifiers can even be matched by programming.

Unfortunately, the power consumption of this topology is roughly proportional to the dynamic range the circuit is designed to support. This stems from the need to maintain stability in the feedback loop [32]. Since the dynamic range is several orders of magnitude, significant costs are incurred in order to support the full range. To alleviate this, an automatic gain control (AGC) amplifier was integrated into the feedback loop, reducing power consumption dependence on dynamic range support. This is also discussed in [32]. Since subthreshold source conductance scales with input current, the gain A can be allowed to drop with higher input currents while still maintaining the low input impedance and stability. The AGC amplifier lowers its gain at higher output voltages, which correspond to larger input currents.

The log amp plays an integral role in the analog vector-matrix multiplier (VMM), which performs the B matrix multiplication. As shown in Fig. 6, every FGPFET in the array, coupled with the respective row’s log amp, forms a wide range, programmable gain current mirror. Rather than utilize voltage movement on the gates of transistors, this current mirror uses changes on of the source voltages of the transistors to perform signal transfer, as in [24], minimizing power law errors caused by mismatches in gate-to-surface coupling. Each quadruplet of VMM FGPFETs corresponds to one coefficient in B . For a fully differential multiplication, w , the programmed gains for a quadruplet are set to

$$\begin{bmatrix} 1 + w/2 & 1 - w/2 \\ 1 - w/2 & 1 + w/2 \end{bmatrix}.$$

All VMM transistors along a row share the same input signal and perform their respective multiplications. The output currents are summed along the columns. The resulting differential current output vector is a vector-matrix multiplication, vB . The work in [33] implements a similar VMM structure used in a classifier circuit.

D. Logarithmic, Bidirectional, Current-to-Voltage Conversion

Since the output of the VMM is a differential current, and a single value is required for the final output, a differential to single-ended conversion was required. With the desire to maintain the ability to process wide dynamic range signals, a logarithmic conversion was sought. Because the resolution of a logarithmic signal is proportional to the signal, it is desirable to remove the common-mode component of the signal before the conversion.¹ To perform the precursory current subtraction that converts the differential signal to a single-ended signal, a current mirror, utilizing the source node for signal propagation as in the VMM, is used to negate one of the currents. Though a gain error may occur due to threshold voltage mismatch in the current mirrors, this is accounted for when programming the corresponding column of the VMM. The resulting current signal can be positive or negative in direction. Hence, a special converter was designed to perform logarithmic conversion on

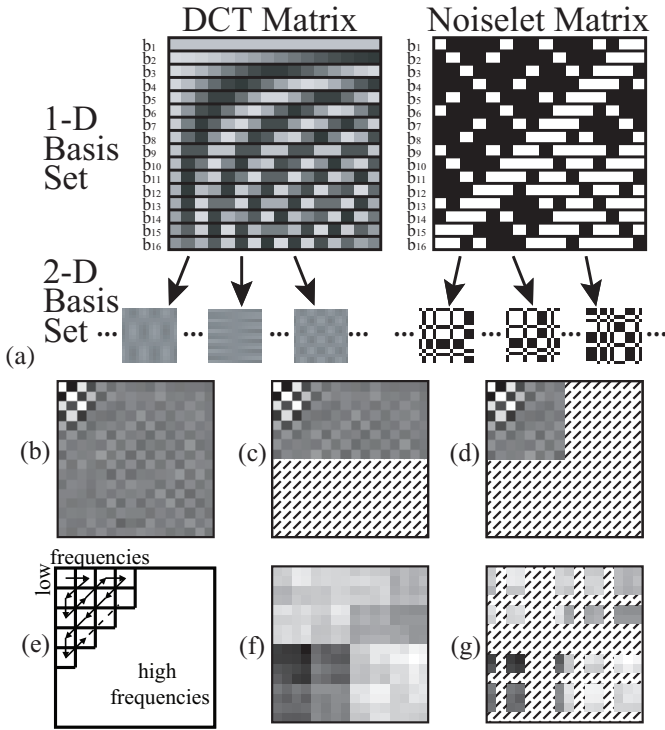


Fig. 7: Basis Functions and Selection of Transform Coefficients. (a) The 2D basis functions are outer products of pairs of 1-D basis vectors. The DCT basis functions are structured to correlate with different spatial frequencies in images. (b) A 2D matrix of output coefficients representing the inner products with the different 2D DCT basis functions is generally non-uniform, since most of the energy in images lies in the low frequency components. (c,d) Skipping 1-D basis vectors eliminates columns and rows of the output matrix. (e) More typical Zig-zag selection of DCT coefficients. (f) The noiselets basis are decorrelated with most image features and with reconstruction basis functions, making each independent noiselet basis function statistically significant as any other and giving an even spread of energy. (g) Random skipping of 1-D basis vectors.

a negative or positive signal [34]. As the input current deviates from zero, the converter approximates a logarithmic compression. This bi-directional converter is very useful in applications where support for large dynamic range is essential and small currents must be sensed at moderate to high bandwidths.

V. COMPRESSIVE IMAGING AND RECONSTRUCTION

In this section, we compare two different image acquisition strategies using the transform imager. The imager acquires 256×256 pixel images; the hardware implementation we discuss here breaks the image into 256×16 pixel blocks, and measures a certain number of transform coefficients in each of these blocks. In principle, the number of coefficients can vary from block to block — a useful feature if there are certain regions of the image we are more interested in than others — but the experiments in this section keep this number fixed.

The first strategy is based on the discrete cosine transform. In each block, we measure M 2D DCT coefficients in the manner illustrated in Fig. 7(d). The observation indexes are in a block with the DC coefficient in the upper left-hand

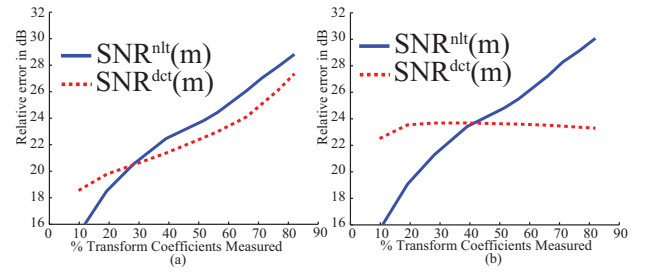


Fig. 8: PSNR of reconstruction vs. percentage of used transform coefficients. Reconstruction from using fewer coefficients compared to (a) a reconstruction using all coefficients and (b) an idealized, denoised image constructed using all coefficients. As expected, retaining a small number of DCT coefficients gives better performance than using a similar number of noiselet transform coefficients since the signal is concentrated in the low frequencies. However, as more DCT coefficients are used, the SNR drops in (b) because the analog system contributes an equal noise with each additional coefficient but less and less additional signal. When more coefficients are used, the noiselet-based reconstruction performed better. It should be noted that we optimized the representation of the noiselets in our analog system utilizing the fact that they consist of only -1 's and 1 's. The Noiselet-based reconstruction also benefits from a reconstruction algorithm that optimizes over the entire image.

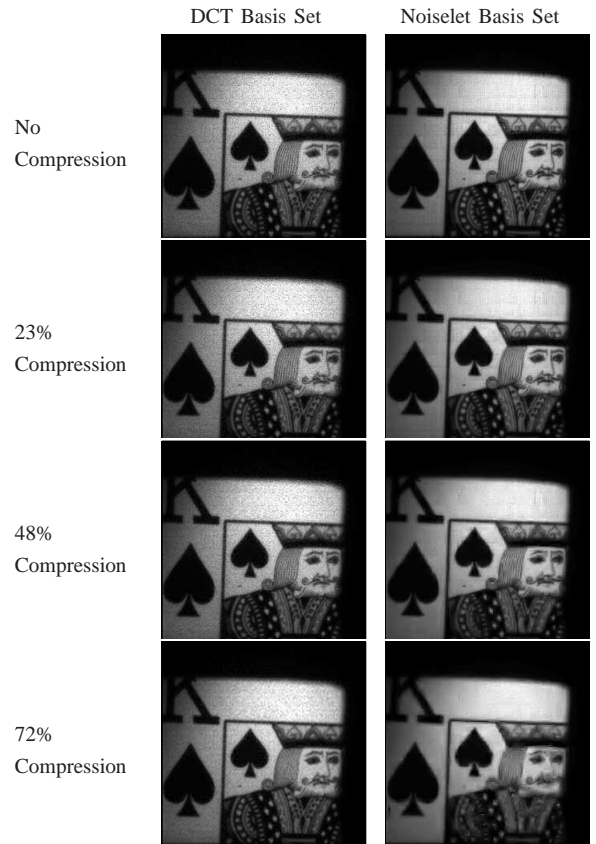


Fig. 9: Reconstruction results using DCT and noiselet basis sets with various compression levels. The image sensor measured 16×16 blocks of the image projected onto DCT and noiselet basis functions. Subsets of the data were taken and used to reconstruct the shown images using a pseudo-inverse for incomplete DCT measurements and a nonlinear-total-variance-minimization algorithm for the noiselets.

corner; we move toward high frequencies by iteratively adding rows and columns to this block. The corresponding ϕ_k in (1) are 2D sinusoids, as shown in Fig. 7(a). After these

measurements are off-loaded from the sensor, the image is reconstructed by taking an inverse DCT transform with the unobserved coefficients set to zero. The reconstruction is the image with *minimum energy* whose DCT coefficients match what we have measured. Sample reconstructions are shown in Fig. 9.

The second strategy is based on the theory of compressive sensing discussed in Section II. Inside of each block, we measure M 2D noiselet coefficients. The basis functions are formed by taking pairwise outer products of the 1D noiselet basis for \mathbb{R}^{16} ; some of these are shown in Fig. 7(a). The noiselet coefficients are chosen, as shown in Fig. 7(g) by selecting M_r rows and M_c columns at random (with $M = M_r M_c$), and observing the 2D noiselet coefficients at the intersection of these rows and columns. After these measurements are off-loaded from the sensor, the image is reconstructed by solving (3) with Ψ as an undecimated Daubechies-4 three-scale wavelet transform¹ and $\epsilon = 0.01 \cdot \|Y\|_2$. Sample reconstructions are shown in Fig. 9.

Care must be taken in judging the results; as we are working from real data, there is no “true” underlying image with which to compare. Moreover, the experimental setup suffers slight physical changes between the DCT and noiselet acquisitions, and as a result the underlying images are slightly different from one another. With this in mind, we measure the accuracy of a reconstruction from partial measurements with respect to the reconstruction from a full set of 65,536 measurements in Fig. 8(a). Using $P^{\text{nl}t}$ to denote the image reconstructed by collecting a full set of noiselet coefficients and applying the inverse noiselet transform, and $\hat{P}_M^{\text{nl}t}$ to denote the ℓ_1 reconstruction from M coefficients per block, we define the signal-to-noise ratio as

$$\text{SNR}^{\text{nl}t}(M) = -20 \log_{10} \left(\frac{\|\hat{P}_M^{\text{nl}t} - P^{\text{nl}t}\|_2}{\|P^{\text{nl}t}\|_2} \right).$$

We define the SNR^{dct} , the signal to noise ratio for the DCT reconstructions, in the same manner. Fig. 8 plots SNR^{dct} and $\text{SNR}^{\text{nl}t}$ for values of M ranging from $M = 25$ ($\approx 10\%$ of the measurements, or a $\approx 90\%$ compression rate) to $M = 210$ ($\approx 82\%$ of the measurements, or a $\approx 18\%$ compression rate).

From these plots we can see that in terms of relative SNR the CS acquisition strategy starts to outperform the classical DCT coded acquisition at a compression rate of about 70%. Visually, the CS reconstructed images are much cleaner starting at compression rates of about 75%.

The CS reconstructions also benefit from the fact that the basis functions are binary valued (± 1) — this created a larger operating signal range in the analog circuitry. This is evidenced by the fact that the full noiselet reconstruction in Fig. 9 is noticeably cleaner than the full DCT reconstruction. As a result, the CS reconstructions compare even more favorable than the SNR plot, Fig. 8(a), would suggest, as they are being judged against a better image. Fig. 8(b) shows what the SNR

¹While the measurement operator Φ acts on each 16×16 block independently, the transformation Ψ acts on the 256×256 image as a whole. This stitches the blocks together very naturally, avoiding undue artifacts in the reconstruction.

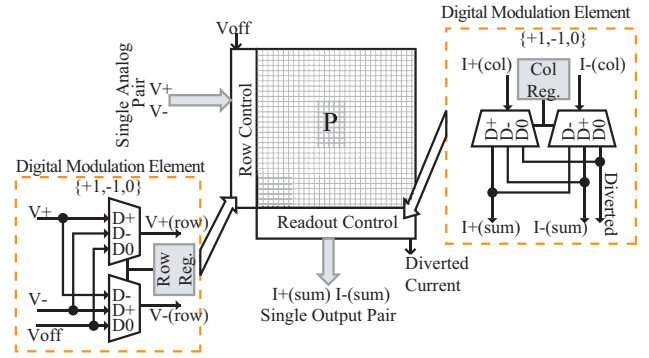


Fig. 10: Small additions to the pixel plane periphery circuitry allowed full 256×256 transforms using coefficients of 1, -1 , and 0. Negations are achieved with differential signals by swapping signals in the pair. Differential input voltages are nulled by setting the pair of signals to the same value while currents are best diverted from the summed outputs. The output pairs signals could have been swapped after an I-V conversion and converted back to currents.

curves look like if compared against a full acquisition that has been denoised by some mild thresholding in undecimated Daubechies-4 wavelet domain (the same transform used for Ψ in the CS reconstructions). We see that the performance of the DCT acquisition scheme essentially saturates after observing about 30% of the coefficients, while the noiselet CS reconstructions steadily improve.

We close this section by mentioning some ways the architecture could be modified to realize higher compression ratios. The first would be to implement a hybrid acquisition scheme which forms a coarse approximation to the image by directly measuring a small number of DCT coefficients, and then fills in the details by measuring noiselet coefficients and reconstructing using ℓ_1 minimization. Although not implemented for this paper, this type of hybrid acquisition has outperformed noiselet-only acquisitions in numerical experiments (see [11]). Another way to achieve higher compression ratios is to increase the block size of the imager, which would allow us to take advantage of the structure of the image at a larger scale. As discussed in Section VI, this architectural modification is being implemented at the current time.

VI. DISCUSSION AND EXTENSIONS OF THE IMAGER ARCHITECTURE

The imaging architecture discussed was born from a general idea to perform image computations assisted by focal-plane processing without too much hardware at the pixel-level that would sacrifice area and sensitivity. The hardware implementation and compressive sensing application presented thus far represent one realization and application of a general, extensible architecture.

Mathematically, the architecture computes the Frobenius inner product (matrix dot product) of the incoming 2-D image with any matrix, F , that can be formed as the outer product of two vectors. This is a vector-matrix-vector multiplication, which is the basis for matrix operations stemming block-wise matrix transforms implemented in [30], [35] as well as operations like convolutions and image-wide projections onto subspaces.

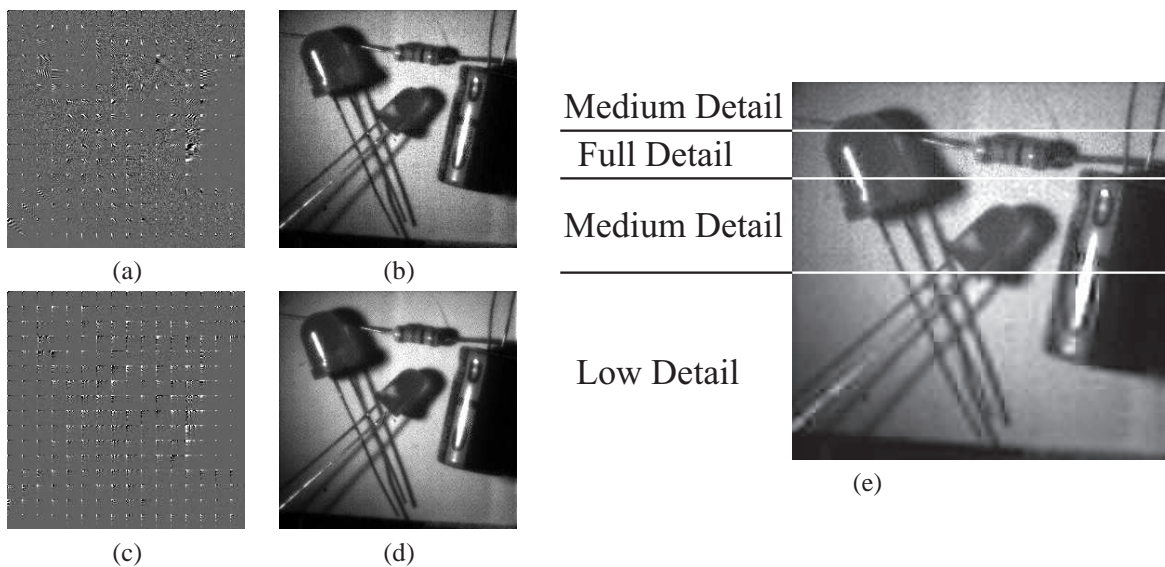


Fig. 11: Capture of transform coefficients. (a) Complete measured DCT Transform Data. (b) Ideal Reconstruction from the complete DCT Transform Data (c) Complete measured Haar Transform Data. (d) Ideal Reconstruction from the complete Haar Transform Data (d) Multiple resolutions of a scene can be captured in the same frame, here using a DCT capture mode, by capturing fewer coefficients in parts of the image. This can be adapted temporally, per-frame. This allows capturing of parts of the visual field with different spatial and temporal resolutions.

Showing that this architectural approach to imaging reaches beyond traditional block-based image processing, Fig. 2 shows data from an operation performed on this transform imager—a convolution implementing an edge-detection filter. Omitting hardware details, it is conceptually enough to understand that the previous block-based architecture had to be extended to support operating on overlapping blocks of the image, rather than strictly adjacent blocks. Eliminating the fixed block boundaries avoids, among other things, the associated edge artifacts.

While the traditional block-based processing genre (the uniform, repeated application of an operation to blocks of the image) is pursued to reduce hardware area and development time, it is not fundamental. In fact one alternative image-wide mode of operation was included on this imager for further study. While in the first mode, the kernel F is restricted to be a size of 16×16 , with elements of analog values in the range $[-1, 1]$, and it operates on a selectable 16×16 block of the image—this is the mode of operation described thus far. In another mode, F is restricted to the values $-1, 0$, and 1 . In this mode, the inner product is computed on all 256×256 elements; so the imager is capable of performing full-image transformations. Fig. 10 illustrates the second mode of operation.

Even further flexibility in the capture process is available given our architecture. For one, subregions of the image can be captured instead of the entire image on a per-frame basis. This would be motivated by previous information about where regions of interest are in the image and the desire for lower power or higher frame rates. Coupling predictive algorithms into the determination of regions of interest provides significant data reduction opportunities.

Even more interesting is a multiresolution approach where sampling functions are constructed to capture the most detail

in areas of high interest and less detail in other areas. This non-uniform sampling approach achieves foviating imaging and maintains peripheral awareness. Such a result would look like that in Fig. 11. Looking at the operation on each block, a typical matrix transform is one where the transformation matrix is constructed from an orthogonal basis set. This is used, for instance, to remap a spatial representation to a frequency representation. Again, a DCT is one such transformation, which does a good job of creating an image representation which segments high and low frequency components in the image. Both a block-based DCT and block-based Haar capture are shown in Fig. 11. Since most image energy is in the lower spatial frequency domain, a DCT does what is called energy compaction. This means for an $N \times N$ block, a majority of the image's energy is represented by less than N^2 DCT coefficients. This can be seen by the sparsity of the DCT result in Fig. 2. The user can choose to only calculate and capture the coefficients representing the majority of the image energy. Applying a subset of the rows of A can be considered mathematically equivalent to zeroing rows of the output, Fig. 7(c), but saves time and power by avoiding unnecessary capture, conversion, and processing. By deactivating columns of the B^T matrix and turning off corresponding ADC's, Fig. 7(d), further savings are achieved.

VII. CONCLUSION

We demonstrated a computational sensor IC capable of a unique and flexible set of sampling modes applicable to compressive imaging. The capabilities of the IC to reconfigurably sense and processes data in the analog domain provides a versatile platform for compressive sensing operations. To demonstrate the platform, images were sensed through projections onto noiselet basis functions that utilize a binary coefficient set, $\{1, -1\}$, and DCT basis functions that

use a range of coefficients. The recent work in the field of compressive sensing enabled effective image reconstruction from a subset of the measurements taken. The fundamental architecture is flexible and extensible to adaptive, foveal imaging and adaptive processing in combination with non-adaptive compressive sensing.

We illustrated the critical circuit components necessary to implement a separable transform image sensor. The final speed of the entire system, including image acquisition and programming, are in testing, but the newly-designed components provide the proper foundation for implementation of a large separable transform imager. The IC results thus far have shown successful system-level current-based sensing and computation with wide dynamic range.

We described a flexible hardware platform which we are developing for dynamically capturing and filtering images. Combining reprogrammable analog processing at the sensor level with reconfigurable digital control allows system resources to be maximally utilized. A flexible MATLAB interface allows users to couple image processing with creative and dynamic algorithms for capturing visual information.

REFERENCES

- [1] E. Candès and T. Tao, "Near-optimal signal recovery from random projections and universal encoding strategies?" *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5245, December 2006.
- [2] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. on Pure and Applied Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [3] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–986, June 2007.
- [4] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [5] W. B. Pennbaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. Springer, 1993.
- [6] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed. Academic Press, 2008.
- [7] I. Daubechies, *Ten Lectures on Wavelets*. New York: SIAM, 1992.
- [8] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Proc. Mag.*, vol. 18, pp. 36–58, Sept 2001.
- [9] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," *Appl. Comp. Harmonic Analysis*, vol. 10, no. 1, pp. 27–44, 2001.
- [10] E. Candès and J. Romberg, "Signal recovery from random projections," in *Proc. SPIE Conference on Computational Imaging III*, C. A. Bouman and E. Miller, Eds., vol. 5674. San Jose, CA: SPIE, January 2005, pp. 76–86.
- [11] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Proc. Mag.*, pp. 14–20, March 2008.
- [12] R. Coifman and D. Donoho, "Translation-invariant de-noising," in *Wavelets and Statistics*, ser. Lecture Notes in Statistics, Antoniadis/Oppenheim, Ed. Springer-Verlag, 1995, vol. 103.
- [13] E. Candès, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Analysis and Appl.*, vol. 14, no. 5, pp. 877–905, 2008.
- [14] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Proc. Mag.*, vol. 25, no. 2, pp. 83–91, 2008.
- [15] D. Graham-Rowe, "Pixel power," *Nature Photonics*, vol. 1, pp. 211–212, 2007.
- [16] W. L. Chan, K. Charan, D. Takhar, K. F. Kelly, R. G. Baraniuk, and D. M. Mittleman, "A single-pixel terahertz imaging system based on compressed sensing," *Appl. Phys. Lett.*, vol. 93, p. 112105, 2008.
- [17] N. P. Pitsianis, D. J. Brady, A. Portnoy, X. Sun, T. Suleski, M. A. Fiddy, M. R. Feldman, and R. D. TeKolste, "Compressive imaging sensors," in *Proc. SPIE Intelligent Integrated Microsystems*, vol. 6232, Orlanda, FL, 2006.
- [18] D. J. Brady, M. Feldman, N. P. Pitsianis, J. P. Guo, A. Portnoy, and M. Fiddy, "Compressive optical MONTAGE photography," in *Proc. SPIE Photonic Devices and Algorithms for Computing VII*, vol. 5907, 2005.
- [19] R. F. Marcia and R. M. Willet, "Compressive coded aperture superresolution image reconstruction," in *Proc. IEEE Int. Conf. Acoust. Speech Sig. Proc.*, April 2008, pp. 833–836.
- [20] R. Fergus, A. Torralba, and W. T. Freeman, "Random lens imaging," MIT CSAIL Technical Report, September 2006.
- [21] L. Jacques, P. Vanderghyest, A. Bibet, V. Majidzadeh, A. Schmid, and Y. Leblebici, "CMOS compressed imaging by random convolution," in *Proc. IEEE Int. Conf. Acoust. Speech Sig. Proc.*, Taipei, Taiwan, 2009, pp. 2877–2880.
- [22] J. Romberg, "Compressive sensing by random convolution," *SIAM J. Imaging Sci.*, vol. 2, no. 4, pp. 1098–1128, 2009.
- [23] J. Haupt, W. Bajwa, G. Raz, and R. Nowak, "Toeplitz compressed sensing matrices with applications to sparse channel estimation," *Submitted to IEEE Trans. Inform. Theory*, August 2008.
- [24] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nm/mhz, 128x/32 current-mode programmable analog vector-matrix multiplier with over two decades of linearity," in *Proc. IEEE Custom Integr. Circuits Conf.*, 3-6 Oct. 2004, pp. 651–654.
- [25] D. X. D. Yang, A. E. Gamal, B. Fowler, and H. Tian, "A 640x512 cmos image sensor with ultrawide dynamic range floating-point pixel-level adc," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 339–342, Dec. 1999.
- [26] O. Yadid-Pecht, R. Ginosar, and Y. Shacham-Dimand, "A random access photodiode array for intelligent image capture," in *Proc. Conv. Elect. Electron. Eng. Israel*, 5-7 March 1991, pp. 301–304.
- [27] B. Pain, C. Sun, C. Wrigley, and G. Yang, "Dynamically reconfigurable vision with high performance cmos active pixel sensors (APS)," in *Proc. IEEE Sensors*, vol. 1, June 2002, pp. 21–26.
- [28] S. Kemeny, R. Panicacci, B. Pain, L. Matthies, and E. Fossum, "Multiresolution image sensor," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 4, pp. 575–583, 1997.
- [29] A. Andreou and K. Boahen, "A 590,000 transistor 48,000 pixel, contrast sensitive, edge enhancing, cmos imager-silicon retina," in *Proc. Conf. Adv. Research VLSI*, 27-29 March 1995, pp. 225–240.
- [30] A. Bandyopadhyay, J. Lee, R. Robucci, and P. Hasler, "Matia: a programmable 80 μ w/frame cmos block matrix transform imager architecture," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 663–672, March 2006.
- [31] R. McFadyen and F. Schlereth, "Gain-compensated logarithmic amplifier," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, vol. VIII, Feb 1965, pp. 110–111.
- [32] A. Basu, R. W. Robucci, and P. E. Hasler, "A low-power, compact, adaptive logarithmic transimpedance amplifier operating over seven decades of current," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 10, pp. 2167–2177, 2007.
- [33] S. Chakrabarty and G. Cauwenberghs, "Sub-microwatt analog vlsi trainable pattern classifier," *IEEE J. of Solid-State Circuits*, vol. 42, no. 5, pp. 1169–79, 2007.
- [34] R. Robucci, J. Gray, D. Abramson, and P. E. Hasler, "A 256x256 separable transform cmos imager," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2008, pp. 1420–1423.
- [35] T. Lee, L. K. Chiu, D. Anderson, R. Robucci, and P. Hasler, "Rapid algorithm verification for cooperative analog-digital imaging systems," in *Proc. Midwest Symp. Circuits Syst.*, 2007, pp. 1305–1308.