COMPRESSIVE SENSING ON A CMOS SEPARABLE TRANSFORM IMAGE SENSOR

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

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250

ABSTRACT

This paper discusses the application of a computational image sensor, capable of performing separable 2-D transforms on images in the analog domain, to compressive sensing. Instead of sensing and transmitting raw pixel data, this image sensor first projects the image onto a separable 2-D basis set. The inner products computed in these projections are computed in the analog domain using a computational focal-plane and a computational analog vector-matrix multiplier. Since this operation is performed in the analog domain, components such as the analog-to-digital converters can be taxed less when a only subset of correlations are performed. Compressed sensing theory prescribes the use of a pseudo-random, incomplete basis set, allowing for sampling at less than the Nyquist rate. This can reduce power consumption or increase frame rate.

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

1. HARDWARE FOR COMPRESSIVE IMAGE SENSING

The standard model for sensing and sampling information includes the requirement of sampling at the Nyquist rate. This is necessary to uniquely convey all the information in the signal being sensed. Often, preexisting knowledge can reduce the amount of data required to uniquely capture the information in the signal. But, without a mechanism to capitalize on a priori knowledge in the sensing process, the sensor and communication hardware must exhaustively sense, process, and transmit information at the Nyquist rate. A compression stage can ease the throughput requirements of communication channels, which is especially critical for wireless sensors, but the advantages are only seen by the stages that follow the compression stage. These advantages translate to lower power consumption and smaller sizes.

More significant reductions in power and hardware complexity can be achieved if data reduction is performed earlier in the sensing chain. The reductions are a result of reducing the data throughput across more stages in the sensing system. In the extreme case, where data reduction is done at the front end of the system, all stages receive these benefits. This translates to less total system communication and possibly less computation required at the sensing device. Offloading computational complexity, like decoding, to the receiver is often more efficient since the receiving system often has relaxed power and area constraints, as is the case with distributed wireless sensor networks utilizing a central processing node.

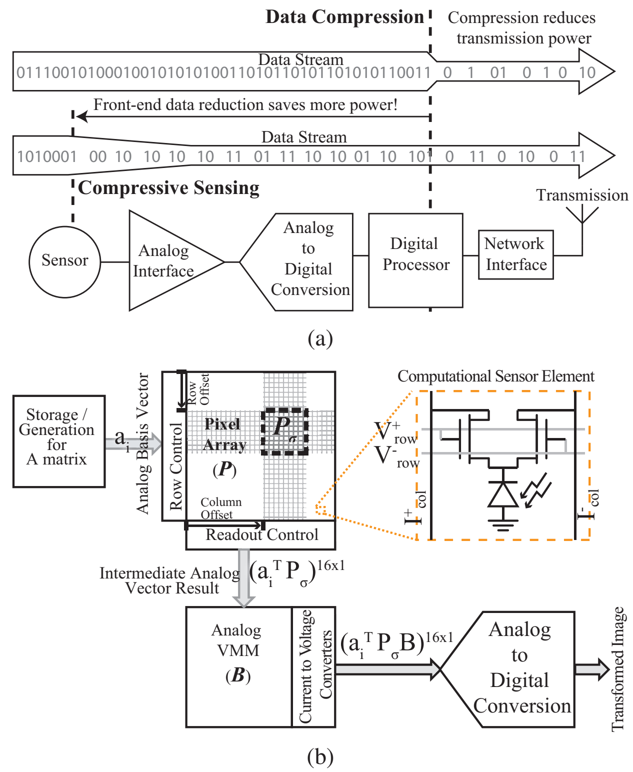

Fig. 1: Compressive Sensing system design (a) Total data manipulation and power is reduced in the chain from sensor to transmitter by sampling less often instead of just compressing data in the digital domain. (b) Separable transform image sensor hardware platform with the capability to capture reduced data sets through projections onto reconfigurable sets of basis functions.

Front end data reduction is exactly what compressive sensing enables [1–4]. Compressive sensing exploits the knowledge that the signal or image we are acquiring is *sparse* in a known transform domain (e.g. the wavelet domain). In other words, there are fewer degrees of freedom in the signal than the Nyquist rate requirement implies, so fewer samples are needed to capture the signal. Presently, in the majority of vision systems, the data throughput required through most of the system is much larger than entropy rate of the signals being processed. This suggests that fewer bits could be used to represent the signal in the system. As a result, compressive sensing is particularly well-suited for image sensing applications, and the development of hardware well-suited to compressive sensing is critical to realizing the anticipated power and size savings or increased per-

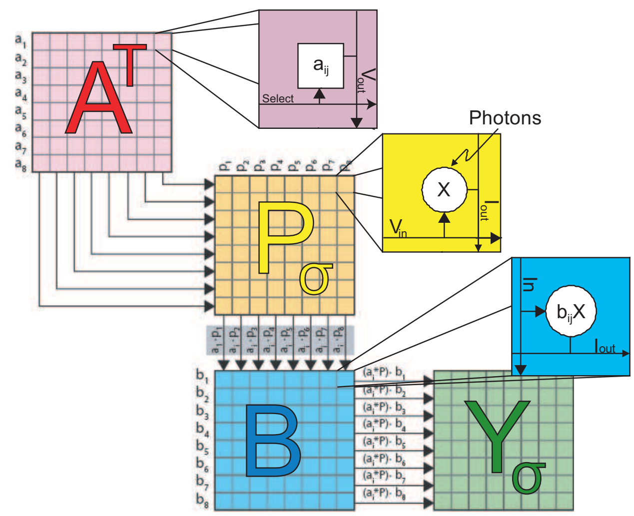

Fig. 2: Block matrix computation performed in the analog domain. Illustrated here as an 8×8 block transform, both a computational pixel array and an analog vector-matrix multiplier are used to perform signal projection before data is converted into the digital domain.

formance, such as the single-pixel camera discussed in [5].

While several technology options exist for image sensing applications, CMOS-based image sensors, also called imagers, share essentially the same manufacturing processes as those used for standard VLSI implementations. Complex computational circuitry can therefore be combined with the sensors and interface circuitry. This paper discusses the capability of a computational image sensor to implement compressive sensing operations. The structure implements a computational architecture similar to that in [6]. The current image sensor design was implemented on a 22.75 mm^2 die in a standard .35 μm CMOS process. The resolution is 256×256 with a pixel size of 8 $\mu\mathrm{m} \times$ 8 $\mu\mathrm{m}$.

The fundamental capability of this image sensor 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 16×16 pixel sub-region of the image under transform. This separable transform operation is demonstrated in hardware to be sufficient to perform compressive sensing.

2. TRANSFORM IMAGE SENSOR

The separable transform image sensor uses a combination of focal-plane processing performed directly in the pixel, and an on-die, analog, computational block to perform computation before the analog-to-digital conversion occurs.

The first computation is performed at the focal plane, in the pixels, using a computational sensor element shown in Fig. 1(b). 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. This operation is represented in Fig. 2 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 computational elements.

The second computation is performed in an analog vector-matrix multiplier (VMM) [7]. This VMM may be designed so that it accepts input form all of the columns of the pixel array, or it can be designed with multiplexing circuity 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 row output. Using the same automatic current summation as the P matrix, a parallel set of weighted summations occur, resulting in the second matrix operation.

3. SENSING WITH A DECORRELATED BASIS SET

The transform image sensor gives us a large degree of flexibility in the choice of basis used in the acquisition. Since our goal is to acquire the image using as few basis functions as possible, one possible choice would be the discrete cosine transform (DCT) basis set, Fig. 3 . The DCT is the basis used by the popular JPEG compression standard. Its effectiveness stems from it tendency to *compact* the energy in the image to the low-frequency basis coefficients. The idea is that since the high-frequency coefficients are small, they can be ignored (not sampled) without too much loss.

The low-frequency DCT coefficients do capture the smooth regions of the image using very few terms. However, the edges in the image are diffused across all frequencies, making them harder to represent using the DCT. If we look at a DCT approximation of an image as we increase the number of terms, we get a "smooth" approximation relatively quickly, but then it takes many terms to capture the edge details, and often the approximations suffer from "ringing" (Gibbs phenomena).

We can build up better approximations to the image using the wavelet transform. While the smooth parts of the image are approximated just as well (if not better) as when using the DCT, the locality of wavelets results in much milder edge effects. An extremely accurate approximation of a medium-size image (one megapixel, say) can be constructed from something like 3% of the wavelet coefficients. As such, the wavelet transform lies at the heart of nearly every competitive image compression algorithm.

Translating the success of the wavelet transform in image compression to sensing is not straightforward. The reason is that there is a subtle difference in the nature of the DCT and wavelet models: the same DCT coefficients (roughly) will be important for every image. The DCT approximation is linear, consisting of a projection onto a *fixed* subspace. Wavelet approximations, however, are nonlinear; significant wavelet coefficients tend to cluster around edges in the image, and hence their locations can change drastically from image to image. While we would like to use our image sensor to sense a small number of wavelet coefficients, we have no idea beforehand *which* coefficients will be the ones to measure.

Recent work in the field of Compressive Sensing [1–4] suggests a non-adaptive sensing strategy that fully exploits the approximation power of the wavelet transform. Instead of trying to match the basis set to the structure we are expecting in the image, we do the exact opposite: we use a basis consisting of (seemingly) random waveforms.

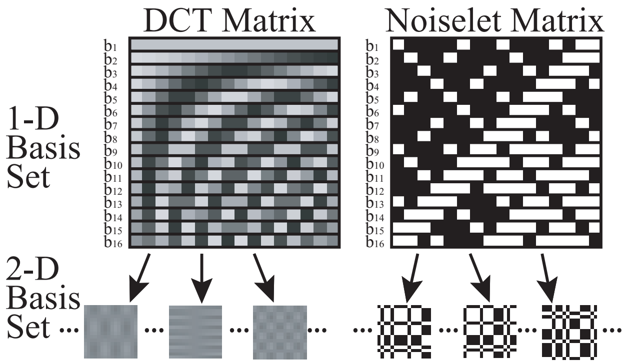

Fig. 3: DCT and noiselet basis sets. The DCT 2D basis functions are structured to correlate with different spatial frequencies in images. The inner products with the different DCT basis functions are generally non-uniform, since most of the energy in images lies in the low frequency components. The noiselets basis are decorrelated with most image features and with reconstruction basis functions, making each noiselet basis function statistically as significant as any other.

Each basis coefficient that we measure is thus a random combination of all the pixels in the image (or of pixels in the sub-block the image sensor is concentrating on). From this series of *random measurements*, we can untangle the important features in the image using convex optimization.

In this paper, the imager samples the image in the *noiselet* domain [8]. Noiselets are an orthogonal basis of waveforms which for our intents behave like random waveforms (see [3] for a more detailed discussion). The noiselet transform is also fast, it requires $O(n \log n)$ operations, which makes relatively large problems computationally feasible.

4. COMPRESSIVE SENSING RECONSTRUCTION PROCESS

To recover the image from the (relatively small) number of random measurements, we need to do more than simply invert a transform. Since the data is undersampled, there are many configurations of pixels that could explain what we have measured. Very few of these, however, have the structure (smooth regions separated by edges) we expect from a real-world image.

There are several popular models to quantify this structure. One model, motivated by the successes in image compression, is sparsity in the wavelet domain. Another model, and one which tends to produce slightly better results in practice, is that typical images tend to have small *total variation* compared to their energy. The total variation of an $n \times n$ pixel image x is given by

$$\mathrm{TV}(x) = \sum_{i,j=1}^{n} \left((x_{i+1,j} - x_{i,j})^2 - (x_{i,j+1} - x_{i,j})^2 \right)^{1/2}, \quad (1)$$

where $x_{i,j}$ is the pixel in the ith row and jth column.

The recovery procedure searches for the image with smallest total variation which explains the measured basis coefficients we have observed. Viewing the image to be recovered as a vector in $\mathbb{R}^{n^2}$, the measurements process can be written compactly in matrix notation as

$$y = \Phi x_0, \quad (2)$$

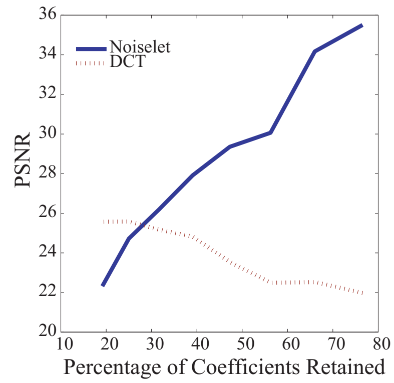

Fig. 4: PSNR of reconstruction vs. percentage of used transform coefficients. As expected, retaining a small number of DCT coefficients gives better performance than using a similar number of noiselet transform coefficients since the signal is concentrated in the low frequencies. However, as more DCT coefficients are used, the SNR drops 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 performs better. This is likely because the noiselets consist of only -1 and 1, and thus can be scaled to maximally use the full analog range. The Noiselet-based reconstruction also benefits from a reconstruction algorithm that optimizes over the entire image.

where $x_0 \in \mathbb{R}^{n^2}$ is the "true" image, the $m \times n^2$ matrix Φ is constructed by stacking the m measurement basis functions — each of which is also a vector in $\mathbb{R}^{n^2}$ — on top of one another, and y is the m vector containing the observations. Given y, we reconstruct the image by solving the following optimization program:

$$\min_x \mathrm{TV}(x) \quad \text{subject to} \quad \|\Phi x - y\|_2 \le \epsilon. \quad (3)$$

The program (3) balances two criteria: we want the recovered image to have small total variation, but we also want it to explain what we have observed (we should only consider x such that $\Phi x \approx y$). The parameter ϵ can be adjusted by the user to properly weight each of these criteria.

The program (3) is convex, and is what is known as a second-order cone program. The solution can be computed using standard *interior-point methods* [9–11]. State-of-the-art solvers can recover images with millions of pixels on a standard desktop computer.

5. RESULTS

The analog computational system described was used to sense images as projected onto programmed basis sets. The raw pixel-by-pixel data is never transferred through the system. Instead, the two-step computational process at the front end of the system projects the image onto selected basis and outputs the inner products from this process, which will be refereed to as the transform coefficients hereafter. The output of the image sensor IC is therefore the representation of the image in the selected vector space. Performing a subset of the complete projections can either reduce power consumption or increase frame-rate.

In the experiments, a complete set of transform coefficients were collected, and the reduced collection was simulated by discarding measured values. The nonlinear recovery algorithm discussed was used to reconstruct the images captured with Noiselet measurement functions. A pseudo-inverse was used to reconstruct images from

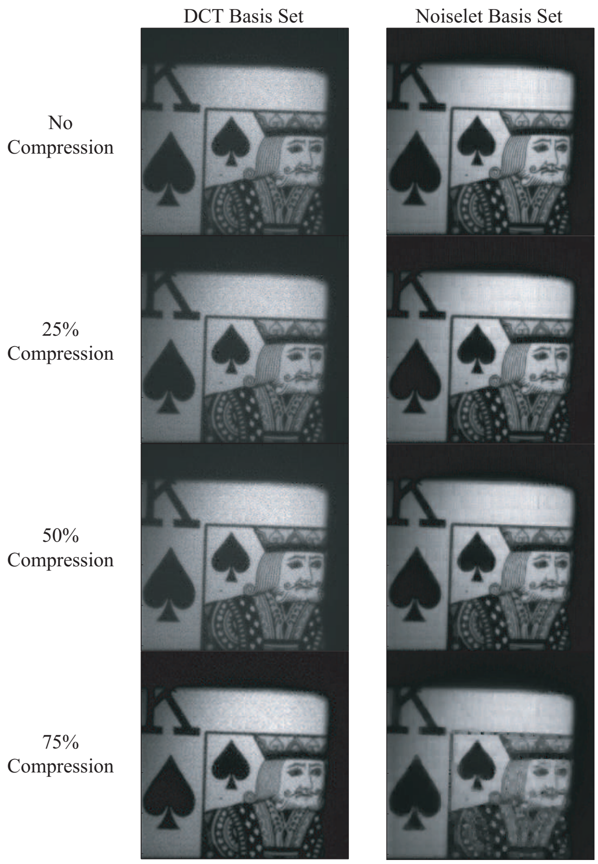

Fig. 5: Reconstruction results using DCT and noiselet basis sets with various compression levels. The image sensor measured 16×16 blocks of the image projected onto DCT and noiselet basis functions. Subsets of the data were taken and used to reconstruct the shown images using a pseudo-inverse for incomplete DCT measurements and a nonlinear-total-variance-minimization algorithm for the noiselets.

incomplete DCT measurements. Since the exact original image is not available, reconstructed images corresponding to incomplete collection were compared against denoised versions of images created from complete coefficient collection.

At high levels of compression, retaining few transform coefficients, the DCT representation lead to better peak signal-to-noise ratio (PSNR), Fig.4 and Fig.5. This is possible because the predefined DCT coefficient removal process exploits the knowledge of where energy compaction occurs in the DCT domain. In the case of the noiselets, higher transform coefficient retention lead to better performance, surpassing the DCT results in quality. It is expected that every transform coefficient in the noiselet domain statistically contributes the same signal and noise power to the resulting image as any other coefficient. In the case of DCT transform coefficients, the coefficients representing high spatial frequencies contribute the same noise as the coefficients representing low frequencies, but they contribute little signal power. In this case, where the reference images were denoised and have little high frequency information overall, the high frequency components contributed negatively to the SNR. Additionally, the noise in the DCT images is higher than the noiselets because the DCT basis functions are smaller in magnitude than those of the noiselets when implemented in this analog system. The basis functions are constrained to a linear input range of the analog computational elements. Since the noiselet functions consist of only 1's and -1's, they use the fullest signal range of the system, resulting in better signal to noise ratio. Moreover, the noiselet-based reconstruction benefits from a reconstruction algorithm that optimizes even across block boundaries. The analysis of the system behavior is ongoing.

6. CONCLUSION

In this paper, 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.

7. 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] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "An architecture for compressive imaging," in *Image Processing, 2006 IEEE International Conference on*, 8-11 Oct. 2006, pp. 1273–1276.

[6] A. Bandyopadhyay, J. Lee, R. Robucci, and P. Hasler, "Matia: a programmable 80 /spl mu/w/frame cmos block matrix transform imager architecture," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 3, pp. 663–672, March 2006.

[7] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nw/mhz, 128/spl times/32 current-mode programmable analog vector-matrix multiplier with over two decades of linearity," in *Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004*, 3-6 Oct. 2004, pp. 651–654.

[8] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," *Applied and Computational Harmonic Analysis*, vol. 10, no. 1, pp. 27 – 44, 2001/01/. [Online]. Available: http://dx.doi.org/10.1006/acha.2000.0313

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[10] D. Goldfarb and W. Yin, "Second-order cone programming methods for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 27, no. 2, pp. 622–645, 2005.

[11] E. Candès and J. Romberg, "ℓ_1-magic," Software available at www.l1-magic.org, 2006.