

# A Random Projection Imager for Visual Pattern Classification in Analog VLSI

Seth Bridges, Jeremy Holleman,  
Ania Mitros, and Chris Diorio  
University of Washington  
Dept. of Computer Science and Engineering  
Seattle, WA, USA  
Email: seth@cs.washington.edu

Miguel Figueroa  
Universidad de Concepción  
Department of Electrical Engineering  
Concepción, Chile  
Email: mfigueroa@die.udec.cl

**Abstract**—In this paper, we present a novel CMOS imager architecture that implements the random projection dimensionality reduction algorithm in the focal plane. We employ analog signal processing techniques to achieve low-power operation and our imager can readily integrate with known low-power VLSI classifiers. We fabricated a 20x20 pixel prototype of our 4.2mm<sup>2</sup> imager in 0.35μm CMOS that performs 1GOPS while consuming 1.25mW of power from a 5V supply.

## I. INTRODUCTION

A number of low-power applications such as wireless sensors, embedded biometrics, and micro-power robots could be enhanced through the addition of a visual pattern classification system. However, the detection and classification of complex visual patterns such as faces, characters, or scenes is a computationally-intensive task, commonly performed by transferring images from a digital camera to a microprocessor or digital signal-processor for processing. This requires a large amount of power, which is incompatible with the severe power restrictions that these applications impose.

To create a classification system that consumes orders of magnitude less power than the systems described above, we can directly integrate a hardware classifier, such as the reconfigurable learning array we presented in [1], with an imager on a single chip. Such an implementation reduces power consumption by, first, eliminating the transfer of large amounts of data from the camera to the processor and second, utilizing a low-power hardware classifier that uses custom analog circuits to compute. However, because the size of the classifier scales directly with the input dimensionality, it is essential to integrate a dimensionality-reduction scheme into the imager to remove redundancy in the data and improve the efficiency of the classification. Implementing this scheme in the focal-plane of an imager reduces power by efficiently computing the dimensionality-reduction operation using analog devices; additionally, allowing the classifier to operate on lower-dimensional data reduces the circuit die-area and power dissipation of the classifier.

Analog VLSI implementations of dimensionality reduction enable good classification performance at much lower power and die-area costs than their digital counterparts. Previous analog chips for image classification implement local feature-

extraction operations using small (e.g. 3x3) spatial filters [2] which are useful for computing simple features such as lines or contrast regions, but are inadequate for the direct classification of complex visual patterns required for face and character recognition.

In this paper, we present a CMOS imager that embeds dimensionality reduction functionality in the focal plane using the random projection algorithm [3]. We can combine the imager with our reconfigurable analog VLSI classifier [1] to perform low-power, real-time face recognition. Our imager uses analog circuits to project the incident images to a lower-dimensional space and includes additional features such as on-chip pixel-output normalization and mismatch compensation to enable accurate pattern classification. We fabricated a prototype of our 4.2mm<sup>2</sup> imager in 0.35μm CMOS, which performs up to 1GOPS while consuming 1.25mW on a 5V supply and providing an output dynamic range greater than 62dB.

## II. RANDOM PROJECTION

The computationally expensive nature of classification algorithms often requires that the dimensionality of the input data be reduced to make the learning operation more efficient or even feasible. Dimensionality reduction algorithms for classification applications must maintain mutual similarities between vectors in the reduced space [4]. In other words, vectors that are similar/dissimilar in the original space should remain similar/dissimilar in the reduced space.

A common linear projection technique for dimensionality reduction that meets this requirement is Principal Components Analysis (PCA), an algorithm that computes the orthogonal projection matrix that best preserves the directions of maximal variance in the data. Unfortunately, PCA requires learning continuously-valued weights from training data, which is difficult to implement in analog VLSI because of the need for persistent analog storage and global feedback signals.

Random projection [3], another linear technique that can maintain mutual similarities between vectors, has been useful in applications such as face recognition [5], motif finding [6], and document categorization [4]. Random projection is more amenable than PCA to implementation in VLSI because it does

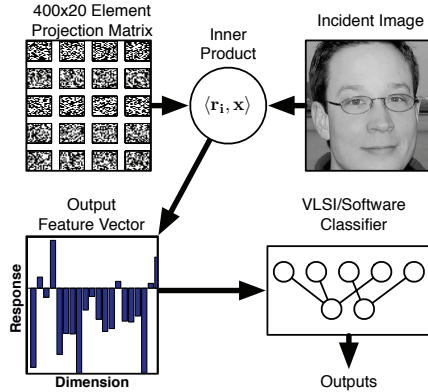


Fig. 1. Random projection for image classification. The initialization of the chip involves sampling/programming a set of binary-valued projection coefficients. For this paper, coefficients are always chosen with equal probability from  $\{-1,1\}$ . Then, in parallel, the chip computes the inner product of the incident image with each of the random kernels to produce a feature vector that can train or be classified by a VLSI or software classifier.

not require learning, instead using data-independent binary-valued weights and strictly local computation. The algorithm linearly maps each input data vector  $\mathbf{x}$  from a  $d$ -dimensional space onto a smaller  $k$ -dimensional space ( $d = 400$  and  $k = 20$  in our prototype) by multiplying it with a projection matrix  $\mathbf{R}$  of dimensions  $d \times k$ , whose elements are binary and randomly chosen.

Fig. 1 illustrates how our chip implements random projection. Each pixel of the incident image is one dimension of a 400-dimensional vector and each row of the projection matrix  $\mathbf{R}$  defines a single projection direction. Continuously and in parallel, the chip computes the inner product of the image with each of the twenty rows of the projection matrix. The 20-d output feature vector comprises the scalar outputs of the inner product operations.

Fig. 2 illustrates how we use the feature vectors that the chip produces to classify images. The center column of the figure is set of template vectors (shown with the corresponding image) of four different individuals taken with our imager. The test feature vector is compared with each feature vector in the training set and the distance between the vectors defines a match likelihood, shown in the right of the figure.

### III. IMAGER ARCHITECTURE AND IMPLEMENTATION

Our imager implements the random projection algorithm by integrating a photodiode, the associated projection matrix coefficients, and the projection multipliers into each pixel. This architecture allows the matrix multiplication to be distributed over the entire imager with the required additions performed by summing currents on global wires. The binary coefficients are digitally programmable and, for additional configuration options, each pixel can be individually disabled. Our imager provides differential voltage/current outputs, the ability to remove the effects of varying global lighting conditions, and increased output dynamic range through the use of electrically-calibrated pixels.

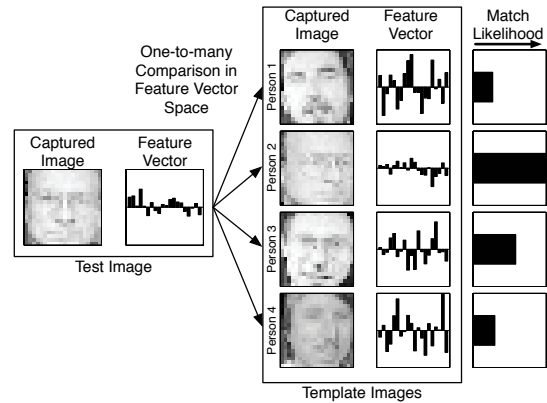


Fig. 2. Classification in the feature vector space. A test image is compared to a set of template images by measuring the distance between the samples in the feature vector space. The likelihood of a match, shown at right, is a function of the distance between the vectors. In this figure, the test image is a photo of “Person 2” and the large match likelihood of this template image indicates a correct match.

Fig. 3 shows a diagram of our system, including the pixel block diagram. To adapt the imager response to varying global illumination conditions, a normalization circuit sums the pixel output currents using the mirror formed by M1-M5. An amplifier compares the total imager current  $I_{\text{sum}}$  to a reference  $I_{\text{ref}}$  and generates an error signal  $V_{\text{mean}}$  that is fed back to the pixels and ensures  $I_{\text{sum}}$  is kept equal to  $I_{\text{ref}}$ .

Each pixel uses the error signal  $V_{\text{mean}}$  to compute its output current which is multiplied by each of the 20 coefficients of the random projection matrix stored locally in SRAM. A local, non-volatile calibration memory compensates for pixel-output offsets introduced by device mismatch. The chip can either output the feature vector currents directly or convert them to differential voltage outputs buffered by on-chip track/hold circuits.

#### A. Pixel:

Fig. 4 shows a schematic of the pixel circuit. A normalization circuit subtracts the output voltage of a continuous-time logarithmic photosensor from the global output mean of the imager. The resulting current is mirrored to an array of differential pairs M9-M10 (M11-M12) that multiply each copy of the current by the local projection matrix coefficients. The output currents from each multiplier are summed across the imager on common wires. Each coefficient is stored locally in SRAM, which gives us the flexibility to configure different projections.

To compensate for the fixed-pattern noise introduced by device mismatch, each pixel features a local nonvolatile calibration circuit based on the floating-gate pFET [7] M1-M2. To calibrate each pixel, we first remove the charge stored on the floating gate using global Fowler-Nordheim tunneling. We then uniformly illuminate the visual field of the imager and set the global input  $V_{\text{mean}}$  to a fixed voltage. Finally, we apply  $10\mu\text{s}$  digital pulses on  $V_{\text{inj}}$  to selectively add electrons to the floating gate using hot-electron injection until the output

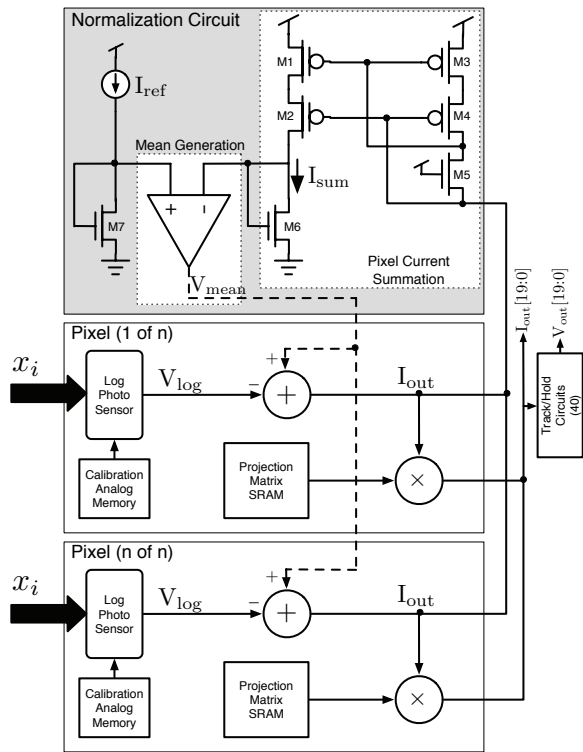


Fig. 3. System Diagram. Our imager comprises an array of pixels and a normalization system that generates a feedback signal to fix the total imager output current at  $I_{\text{ref}}$ . The feature vector output can be read directly as currents or as voltages from a bank of track/hold circuits.

current reaches a predetermined level.

Fig. 5 shows chip data of the transfer functions ( $I_{\text{out}}$  in Fig. 4) of 8 pixels pre- and post-calibration. Our calibration successfully removes DC offsets in the pixel output for the calibration point ( $I_{\text{out}} = 50\text{nA}$ ), but gain mismatch still accounts for smaller variations over the entire range of the output. Our single floating-gate device is unable to fully compensate for this effect.

### B. Normalization:

We normalize each pixel output to gain insensitivity to global illumination change and to achieve robust classification performance. To achieve this normalization operation, we generate a global mean signal  $V_{\text{mean}}$  using a feedback circuit that compares the total imager output current to an external reference  $I_{\text{ref}}$ . We choose  $I_{\text{ref}} = I_b \times \text{numPixels}/2$  to maximize the differential pair's linear range.

Fig. 6 illustrates the twenty-dimensional feature vector (taken directly from our chip) for a single image over varying global illumination conditions. In Fig. 6(a), the normalization operation is disabled and each dimension of the feature vector output varies as a function of the total imager output current. Fig. 6(b) shows that enabling normalization keeps each dimension of the output constant with a resolution greater than 6 bits.

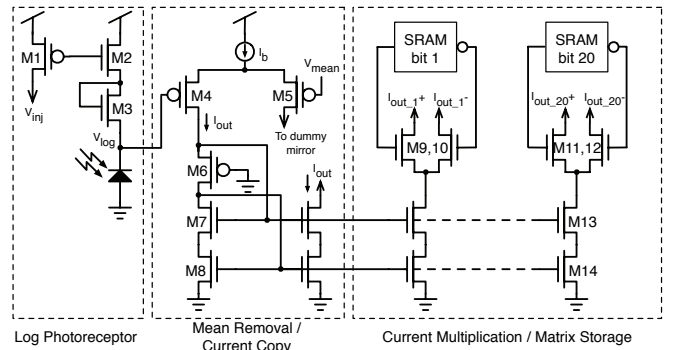


Fig. 4. Pixel schematic. Our pixel generates an output current proportional to the log light intensity with the imager mean removed. The pixel replicates that current and multiplies it by locally stored coefficients using switches.

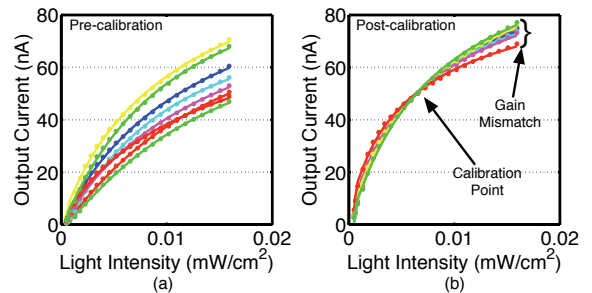


Fig. 5. Pixel calibration. Each pixel has a floating-gate transistor (M1 in Fig. 4) that we program to remove DC offsets in the pixel transfer function. These chip data show the transfer functions for 8 individual pixels both (a) pre-calibration and (b) post-calibration.

## IV. EXPERIMENTAL RESULTS

Fig. 7(a) shows a micrograph of the  $20 \times 20$  pixel prototype of the architecture we fabricated in a  $0.35\mu\text{m}$  double-poly four-metal CMOS process. Our prototype is capable of reducing its 400 pixel input to a twenty-dimensional feature vector. The imager core occupies  $3.31\text{mm}^2$  and the entire design (including the forty sample/hold circuits) occupies  $4.2\text{mm}^2$ . The power consumption for the chip configuration used in these experiments is  $1.25\text{mW}$  from a  $5\text{V}$  supply. Each pixel uses an  $I_b = 10\text{nA}$  and creates 21 copies of approximately half this current for a total of  $115\text{nA}$  ( $575\text{nW}$ ) per pixel ( $230\mu\text{W}$  for the entire imager). The remainder of our power dissipation, about  $1\text{mW}$ , is from our output track/hold circuits.

We tested the performance of our imager in a face-recognition task using the ORL face database [8] containing 10 frontal face images for each of 40 individuals. We used a Dell 1703FP LCD screen to provide our imager stimulus. To classify the feature vectors, we implemented a one-nearest-neighbor classifier [9] in software and trained it on the chip output, randomly selecting half of the data for training and testing on the remaining half. In addition to capturing the feature vectors from our chip, we used a software implementation of random projection for a baseline comparison. Fig. 8(a) shows the classification accuracy as a function of

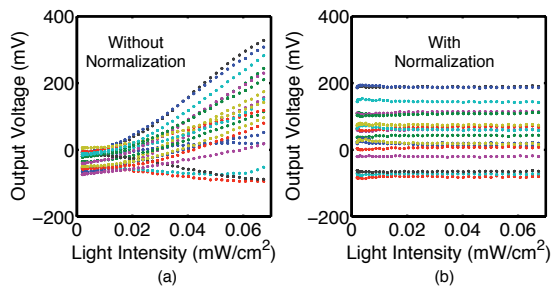


Fig. 6. On-chip normalization. This data shows the chip output for a single image whose global illumination was scaled to simulate varying global illumination both (a) without normalization and (b) with normalization. The on-chip normalization feedback loop keeps the outputs constant within more than 6 bits of resolution.

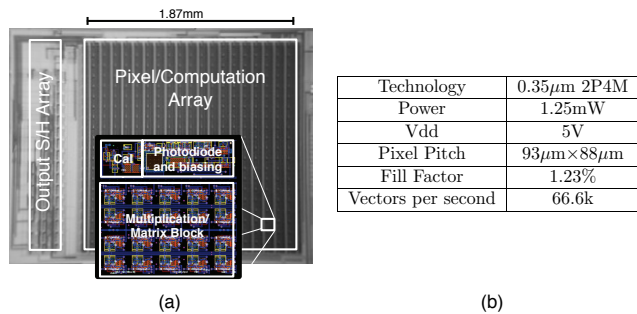


Fig. 7. Imager architecture. (a) Micrograph of our 2-D array of pixels that computes the convolution of stored binary-weighted kernels. (b) Key statistics of our fabricated prototype.

the output dimensions. On average, our chip performs within 3% of the classification accuracy of a software implementation of the algorithm over the entire range, suggesting that the non-idealities of our analog implementation did not introduce significant errors.

In another experiment, we evaluated the effectiveness of our normalization system under varying global lighting conditions. We repeated our previous experiment, but randomly scaled the brightness of the face images by a factor between 0.5 and 1.0. Fig. 8(b) shows the results of normalization on the classification performance for this experiment. With the normalization enabled, the chip performance is within 2% of the performance of the previous experiment (labeled “With norm, static” in Figure 8(b)). Without the normalization enabled, the classification performance drops by up to 30%.

To obtain an estimate of the maximum speed of our imager, we measured the output settling time in response to a step input applied to the global mean signal. The outputs settled to 90% of their final value within 15 $\mu$ s, corresponding to a frequency of 66.6kHz. For each output sample, the imager computes 16,000 multiply/add operations for a total of about 1GOPS.

To measure the output resolution of our imager, we first measured the output range and then stimulated the imager with a constant (static) image and resampled the output sample/hold amplifiers fifty times, sampling the differential output voltages

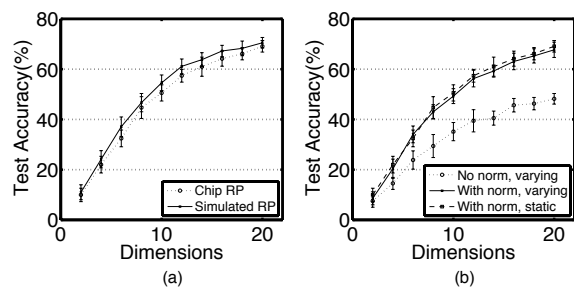


Fig. 8. Face classification performance. (a) This experiment compares the performances of our chip and a software simulation of the same algorithm. The data show our chip’s classification performance is within 3% of the software simulation of random projection. (b) Using the same dataset as in (a), but with the global illumination of the images modified randomly, the on-chip normalization system compensates for these illumination differences and improves classification performance by up to 42%.

from each amplifier for each of the fifty samples. The voltage range at this power level is 1.8V and the noise level is 1.34mV (RMS) corresponding to a dynamic range of more than 62dB per channel.

## V. CONCLUSIONS

We presented an architecture for embedding the random projection dimensionality reduction in the focal plane of a CMOS imager. Our low-power operation and accuracy when compared with software simulations of the same algorithm make this architecture attractive for wireless sensors and power constrained robotics applications. Future work will focus on scaling this architecture to accommodate larger imager resolutions and reducing the power dissipated by the output track/hold circuits.

## ACKNOWLEDGMENTS

Seth Bridges was partially funded by an Intel Ph.D. Fellowship. Miguel Figueroa was funded by FONDECYT grant No. 1040617. Additional funding was provided by Packard Foundation grant 1998-4401 and NSF ITR CCR-0086032.

## REFERENCES

- [1] S. Bridges, M. Figueroa, D. Hsu, and C. Diorio, “Reconfigurable Learning in Silicon,” in *Proc. of ESSCIRC05*, Grenoble, France, 2005.
- [2] V. Gruen and R. Etienne-Cummings, “Implementation of Steerable Spatiotemporal Image Filters on the Focal Plane,” *Circuits and Systems II: Analog and Digital Signal Processing*, *IEEE Trans. on*, vol. 49, no. 4, pp. 233–244, 2002.
- [3] D. Achlioptas, “Database-friendly Random Projections,” in *Proc. of the 20th Symp. on Principles of Database Systems*, 2001.
- [4] S. Kaski, “Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering,” in *IEEE Intl. Joint Conference on Neural Networks*, vol. 1, 1998, pp. 413–418.
- [5] N. Goel, G. Bebis, and A. Nefian, “Face Recognition Experiments with Random Projection,” in *SPIE Defense and Security Symposium*, 2005.
- [6] J. Buhler and M. Tompa, “Finding Motifs Using Random Projections,” in *Proc. of RECOMB’01*, 2001, pp. 69–76.
- [7] C. Diorio, D. Hsu, and M. Figueroa, “Adaptive CMOS: From Biological Inspiration to Systems-on-a-chip,” *Proc. of the IEEE*, 2002.
- [8] F. Samaria and A. Harter, “Parameterisation of a Stochastic Model for Human Face Identification,” in *Proc. of 2nd IEEE Workshop on Applications of Computer Vision*, 1994.
- [9] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997.