

# Convolution, Deconvolution, and Mean Field Annealing Suitable for Analog VLSI

Griff L. Bilbro, *Senior Member, IEEE*, Lester C. Hall, Mark Clements, and Wentai Liu, *Senior Member, IEEE*

**Abstract**—We formulate several standard digital image processing operations as circuits suitable for implementation in real-time analog VLSI, including nonlinear piecewise-constant image restoration using mean field annealing as a global optimization technique. We report test results from an imaging chip that performs user-controlled convolution of the image. We use simulated results for mean field annealing (MFA) in SPICE to show that deconvolution could be implemented by rearranging the subcircuits that perform the corresponding convolution. We report the results of realizing deconvolution in this way on a printed circuit board.

**Index Terms**—Analog VLSI, CMOS image acquisition, deconvolution, image processing, MAP estimation, nonlinear Bayesian processing, piecewise smooth restoration, real-time, regularization.

## I. INTRODUCTION

REAL time image processing is limited to the simplest operations, such as sums of products, linear convolutions, and median filters. Many interesting problems in image processing and machine vision are numerically too intensive to compute algorithmically, either because they involve nonlinear image formation models such as in magnetic resonance imaging [6]–[8], or because they can be solved only by introducing nonlinear techniques such as simulated annealing [10], [17]. In spite of their computational complexity, such techniques are potentially more useful in images than in a time series, because images exhibit true discontinuities (occlusion edges) which cannot occur in band-limited time series data. Approximate solutions of the resulting global optimization problem are faster than simulated annealing and they also exhibit potentially useful parallelism [9], [13], [20], but they are still too slow for real-time applications, even when implemented on a 64-node hypercube parallel architecture [4]. Alternatively, for some special cases, it is possible to obtain results qualitatively similar to those from standard image processing techniques [5], [18]; however, those approaches do not exploit the mature and general technology that already exists in image processing and machine vision [11], [14], [16].

In this paper, we will consider implementing general image processing operations in analog VLSI. We are particularly

Manuscript received November 14, 1996; revised August 19, 1998. This work was supported in part by the Piedmont Triad Engineering Research Center at Winston-Salem State University, Winston-Salem, NC, and by the Center for Advanced Computation and Communication at North Carolina State University, Raleigh, NC. This paper was recommended by Associate Editor T. S. Fiez.

The authors are with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695-7911 USA.

Publisher Item Identifier S 1057-7130(99)01748-6.

interested in laying the foundation for incorporating nonlinear operations and global optimization into VLSI imaging.

Images degraded by imperfect sensors or transmission can be modeled usefully [12], [15], as the sum of random noise image  $n$  and a convolution of the true image  $f$  with a point spread function or blurring kernel  $h$ . We will write

$$g = h * f + n \quad (1)$$

for the degraded image  $g$  so that  $g_i$  for  $i = 1, 2, \dots, L$  is the  $i$ th pixel of the  $L$  pixels in image  $g$ . The additive noise  $n$  comprises  $n_i$  where  $i = 1, 2, \dots, L$ . The convolution of  $h$  with  $f$  is denoted by the  $*$  operator, defined by the relation

$$(h * f)_i = \sum_j h_{i-j} f_j \quad (2)$$

so that the  $i$ th pixel of the convolved image  $h * f$  is a weighted sum of pixels of  $f$ .

Circuit (A) in Fig. 1 shows one cell of a spatial convolution filter for one-dimensional (1-D) images. Two-dimensional (2-D) versions are obtained from these by adding connections as in Fig. 6. In either case, the complete filter comprises an array of identical cells. Input voltage  $V_{I,i}$  is applied at input node  $f(i)$  at the left-hand side of this cell and output voltage  $V_{O,i}$  appears at the right-hand side of the cell. The follower reproduces the input voltage at resistor  $R_{\text{local}}$  which connects the opamp buffer to the output node  $(h * f)(i)$  of cell  $i$ . The output nodes of cell  $i + 1$  and cell  $i$  are connected by resistor  $R_{\text{lateral}}$ . The output node of cell  $i$  is connected to cell  $i - 1$  below it by another  $R_{\text{lateral}}$  which is not shown in this cell. The sum of the currents entering output node  $(h * f)(i)$  is

$$0 = \frac{V_{I,i} - V_{O,i}}{R_{\text{local}}} + \frac{V_{O,i+1} - V_{O,i}}{R_{\text{lateral}}} + \frac{V_{O,i-1} - V_{O,i}}{R_{\text{lateral}}} \quad (3)$$

so that

$$0 = V_i \frac{1}{R_{\text{local}}} - V_{O,i} \left( \frac{1}{R_{\text{local}}} - \frac{2}{R_{\text{lateral}}} \right) + V_{O,i+1} \frac{1}{R_{\text{lateral}}} + V_{O,i-1} \frac{1}{R_{\text{lateral}}} \quad (4)$$

which can be written in matrix form as

$$-\lambda^2 \Delta^2 V_O + V_O = V_I \quad (5)$$

where the matrix  $\Delta_{i,i'} = \delta_{i,i'-1} - 2\delta_{i,i'} + \delta_{i,i'+1}$  is the second difference operator which is a discrete approximation to the second derivative operator, and  $\lambda = \sqrt{R_{\text{local}}/R_{\text{lateral}}}$  plays the role of a distance.

The output  $V_O$  is a smoothed version of the input  $V_I$ . This is easiest to see when there are many cells and the input array

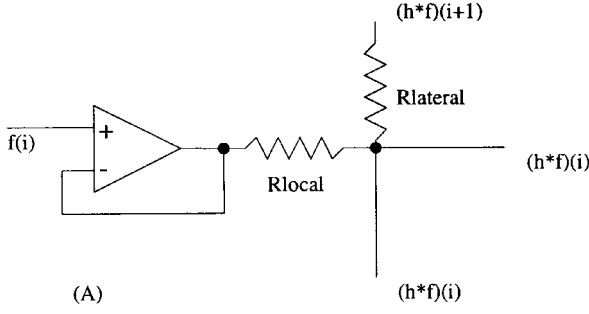
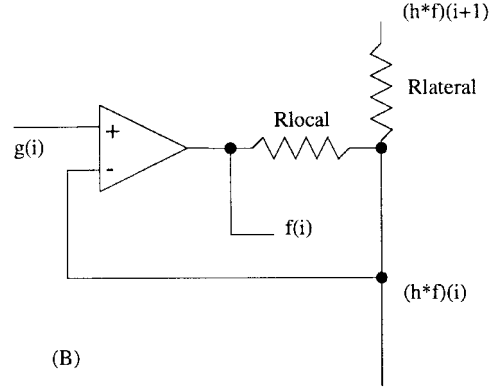


Fig. 1. Schematic for one cell of a convolution circuit.


 Fig. 2. Inverse filter for restricted  $h$  and negligible noise, as discussed in the text.

$V_I$  is smoothly varying, so that a continuous approximation holds. In that limit, the Fourier transform of (5) is

$$\lambda^2 k^2 \tilde{V}_O(k) + \tilde{V}_O(k) = \tilde{V}_I(k) \quad (6)$$

or

$$\tilde{V}_O(k) = \frac{\tilde{V}_I(k)}{1 + \lambda^2 k^2} \quad (7)$$

which is a low-pass filter in the (spatial) Fourier domain. In direct space, this particular filter is the result of convolving the input with the blurring kernel (normalized to have unit integral)

$$h(x) = \frac{1}{2} \lambda e^{-|x|/\lambda} \quad (8)$$

as can be verified by computing its Fourier transform. This circuit therefore attenuates spatial frequencies higher than  $1/\lambda$  but passes lower spatial frequencies. Even though the preceding analysis quantitatively holds only in the continuous limit, this low-pass behavior is qualitatively correct for a discrete array.

The simplest algorithmic approach to estimating  $f$  given the blurred image  $g$  and the blurring kernel  $h$  is to attempt to ignore the noise  $n$  and solve

$$g = h * f \quad (9)$$

in the Fourier domain. The resulting inverse filter is unstable [15]. The Wiener filter for this problem is stable but cannot preserve sharp edges [15], [19]. We will investigate linear and nonlinear Bayesian techniques for removing noise without degrading edges. For independent, identically distributed Gaussian noise, the standard deviation  $\sigma_i = \sigma, \forall i$ . We restrict ourselves to the zero-mean case,  $\langle n_i \rangle = 0$  for  $i = 1, 2, \dots, L$ , since the case in which every pixel is shifted by a constant is not usually a problem in imagery.

Circuit (B) in Fig. 2 is one cell of a circuit that performs an inverse filter for spatial deconvolution. It contains the same resistive network as Circuit (A) but in the (vector) feedback path. Circuit (B) can be analyzed with Kirchoff's laws. The sum of the currents entering node  $(h * f)(i)$  is

$$0 = \frac{V_f(i) - V_{(h*f)(i)}}{R_{\text{local}}} + \frac{V_{(h*f)(i+1)} - V_{(h*f)(i)}}{R_{\text{lateral}}} + \frac{V_{(h*f)(i-1)} - V_{(h*f)(i)}}{R_{\text{lateral}}} \quad (10)$$

so that

$$-\lambda^2 \Delta^2 V_{(h*f)} + V_{(h*f)} = v_f \quad (11)$$

or

$$\tilde{V}_f(k) = (1 + \lambda^2 k^2) \tilde{V}_{(h*f)}(k) \quad (12)$$

for the array of internal voltages at nodes  $f(i)$ , which are adjusted by the array of opamps to make  $V_{(h*f)(i)}$  coincide with the input voltages at nodes  $g(i)$  in the direct domain, so that

$$\tilde{V}_f(k) = (1 + \lambda^2 k^2) \tilde{V}_g(k). \quad (13)$$

This result is satisfactory only when the power spectrum of the noise falls faster than  $1/k^2$ .

Another important operation in digital image processing is the least mean square (LMS) estimate. An LMS inverse filter minimizes the mean squared error  $H_1 = \sum_i [(g - h * f)_i]^2$  by solving the vector equation

$$\nabla_{\mathbf{f}}(H_1[f|g]) = h^\top * (g - h * f) = 0 \quad (14)$$

where the superscript  $\top$  indicates the matrix transpose. This is equivalent to solving

$$h^\top * h * f = h * g \quad (15)$$

for the pseudo-inverse

$$f = (h^\top * h)^{-1} * h * g. \quad (16)$$

Circuit (C) in Fig. 3 shows one cell of a circuit that implements an LMS filter. Circuit (C) can be analyzed formally as before, but it can also be understood directly in terms of the behavior of Circuits (A) and (B). In Circuit (C), the output of the opamp is the desired estimate. This signal is blurred by the same resistive network as before and this blurred estimate  $(h * f)(i)$  is fed back to a circuit function which outputs the difference between the input applied at node  $g(i)$  and  $(h * f)(i)$ . This difference is then convolved with  $h$  again before it is input to the opamp (with its negative input tied to 0), so that the opamp adjusts its output at  $f(i)$  to make  $f * (g - h * f) = 0$ , a condition equivalent to (15) for symmetric  $h$ .

The pseudo-inverse estimate of  $f$  produced by Circuit (C) is stable for a larger class of kernels, but is still ill-conditioned

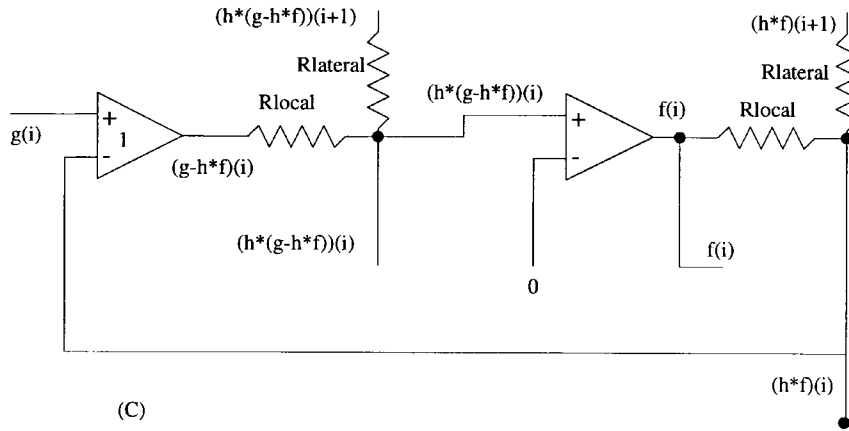


Fig. 3. LMS deconvolution filter for general  $h$  as discussed in the text.

because of the same sensitivity to small high spatial frequency variations in the input  $g$ .

Regularization techniques are typically used to condition such problems. These techniques exploit additional (prior) information to counteract high frequency noise. Bayesian techniques such as *maximum a priori* (MAP) belong to this class but enjoy the advantage of (at least starting from) rigorous results from statistics and probability. MAP estimates that preserve sharp edges while removing noise lead to multimodal (i.e., nonconvex) global optimization problems.

Circuit (D) implements a MAP estimation of  $f$  and can be operated in such a way to use mean field annealing (MFA) to select deeper minima in the resulting multimodal optimization problem. MFA is a kind of continuation method related to simulated annealing and is particularly effective for imaging problems. Circuit (D) can be understood just as Circuit (C), except that the vector of estimates  $f$  is adjusted to make the vector  $h * (g - h * f) + \nabla R = 0$ , where  $R = R[f]$  is the regularization functional.

In this approach to image processing, we define the optimal estimate as the image that minimizes

$$H[f] = H_1 + H_2 = \sum_i |g - h * f|_i^2 + \gamma \sum_i R[f]_i \quad (17)$$

where  $H_2$  contains a problem-dependent constant  $\gamma$  and the sums are over all pixels [1], [3], [10], [13]. A convenient  $H_2$  for piecewise-constant restoration in one dimension is

$$R[f]_i = -\exp\left(-\frac{(D * f)_i^2}{T}\right) \quad (18)$$

where  $T$  is an artificially introduced temperature-like variable that can be gradually reduced (i.e., annealed) and  $D$  is a difference kernel, so that  $(D * f)_i = f_{i+1} - f_i$ . (Since the sum in (17) is over all pixels, it is sufficient to include only terms involving right differences.) For 2-D images, a similar term involving vertical differences must be included in the argument of the exponential.

The effect of  $H_2$  can be understood by considering  $R$  for a pair of adjacent pixels at extreme values of  $(D * f)_i^2$ : when  $(D * f)_i^2 \ll T$  the pixel values differ by much less than  $T$  and  $H_2$  increases almost quadratically with increasing argument. On the other hand, when  $(D * f)_i^2 \gg T$ ,  $H_2$  becomes

approximately independent of the difference in pixel values. The combined effect is to blur adjacent pixels with similar values, but to preserve significant differences at occlusion edges.

Circuit (D) in Fig. 4 regularizes the solution of Circuit (C) by minimizing the posterior error  $H[f|g] = H_1[f|g] + H_2[f]$  with respect to  $f$ , where the regularization term  $H_2[f]$  can be thought of in Bayesian terms as the negative of the logarithm of a prior probability distribution of  $f$  [1], [3], [10]. The gradient of  $H[f|g]$  appears in Circuit (D) as

$$(\nabla_f H)_i = -(h * (g - h * f))_i - (S(f_i - f_{i-1}) - S(f_{i+1} - f_i)). \quad (19)$$

The function  $S$  can be chosen to be linear or nonlinear; either can be used to suppress noise, but edges can be preserved only with a nonlinear  $S$ . For linear  $S(\Delta f) = \gamma \Delta f$ , where  $S$  is multiplication by some real number  $\gamma$ , the last two terms of (19) become the second difference in

$$(\nabla_f H)_i = -(h * (g - h * f))_i - \gamma(f_{i+1} - 2f_i + f_{i-1}) \quad (20)$$

which penalizes  $f$ 's that exhibit high frequency spatial variation.

For nonlinear  $S$ , such as for  $\gamma \tanh(\Delta f/T)$ , or for the derivative of  $R$  in (18), the behavior of the filter circuit is more interesting and can produce an estimate  $f$  that is *piecewise smooth* with sharp edges between smooth regions. Unfortunately in this case, Circuit (D) in Fig. 4 is multistable and can settle in any of several states, each corresponding to some (typically suboptimal) local minimum of  $H$  where  $\nabla_f H$  vanishes. MFA can be used to drive the circuit to the lowest accessible minimum of  $H$  and can be implemented in Circuit (D) by continuously decreasing the voltage at  $T$  from a large value down to zero in each image frame.

## II. SIMULATION OF MFA DECONVOLUTION IN 1-D

Fig. 5 shows the result of simulating an MFA continuation in one dimension. An image was constructed with an interior three-pixel foreground at one volt  $f_2 = f_3 = f_4 = 1$ , and a disjoint seven-pixel background at zero volt  $f_1 = f_5 = f_6 = f_7 = f_8 = f_9 = f_{10} = 0$ . The image was then blurred by convolving with a kernel of  $\lambda = \sqrt{2}$  using SPICE on a circuit containing ten copies of Circuit (A) with resistor values

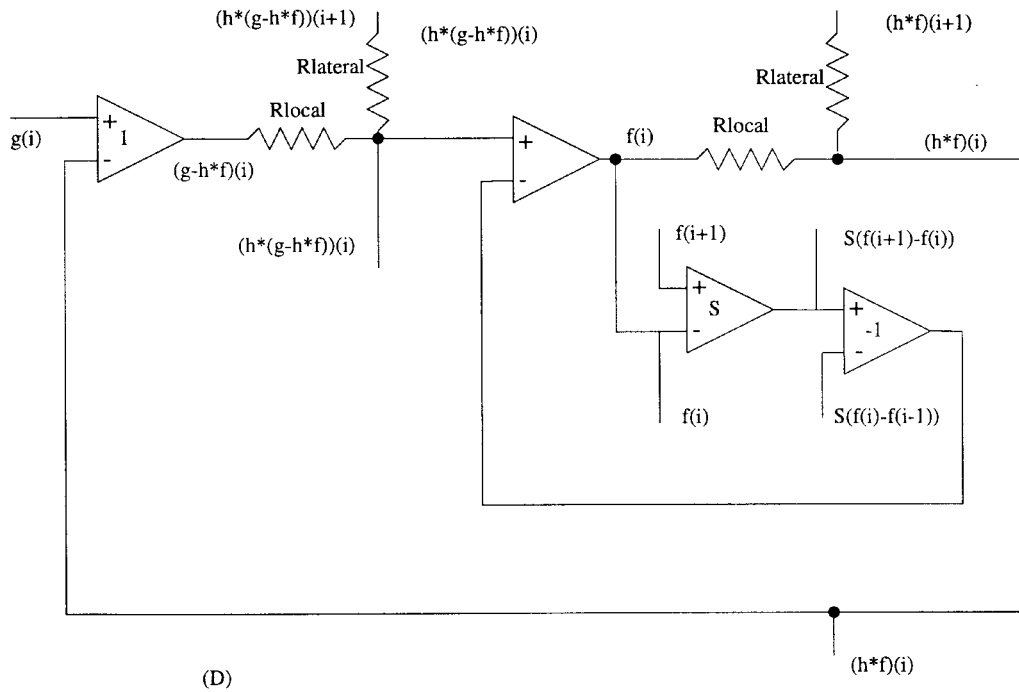


Fig. 4. Bayesian deconvolution filter that can be operated to perform MFA as discussed in the text.

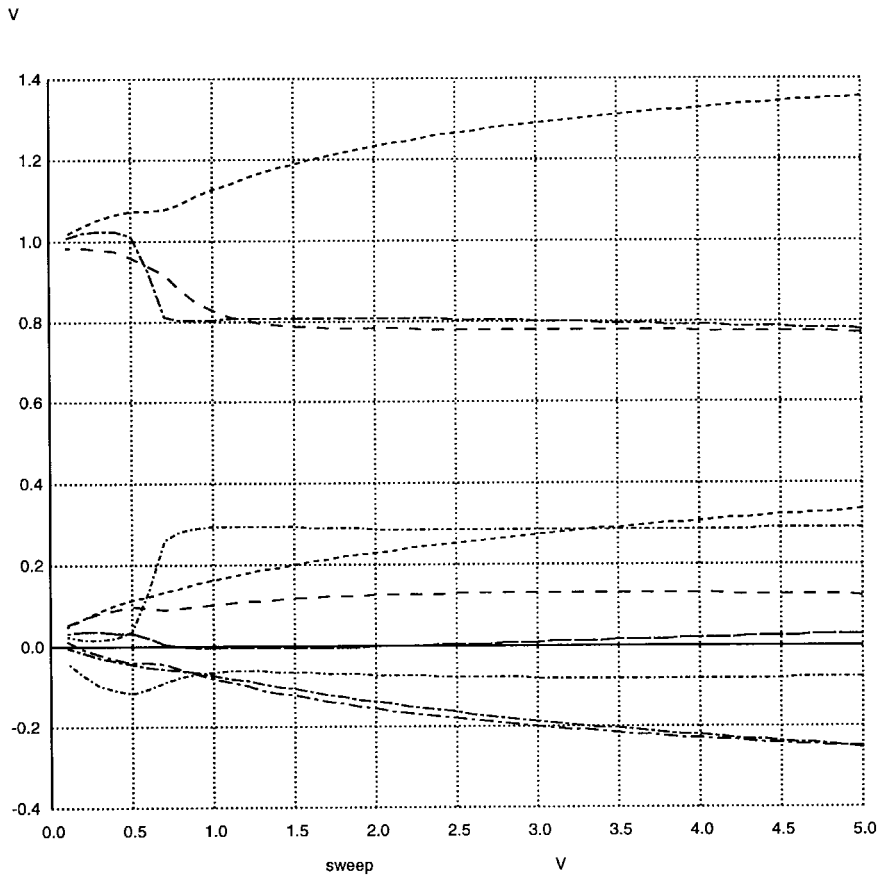


Fig. 5. Simulation results for MFA deconvolution on a one dimensional image of ten pixels as discussed in the text. The ordinate is the voltage representing the annealing temperature and is read from high initial value at right to a low final value at the left. The top three and low seven voltages exhibit the added noise which gradually vanishes as each signal converges to its true value.

$R_{local} = 50 \text{ k}\Omega$  and  $R_{lateral} = 25 \text{ k}\Omega$ . The resulting blurred signal was further corrupted by adding high-frequency noise sampled from a Gaussian distribution with zero mean voltage and standard deviation of 0.1 V to get  $g$ .

Deconvolution was performed by an array of ten copies of Circuit (D) in Fig. 4 with a nonlinear circuit element corresponding to (18). The temperature-like control voltage  $T$  was reduced from  $v = 5 \text{ V}$  down to  $v = 0.1 \text{ V}$  to obtain Fig. 5,

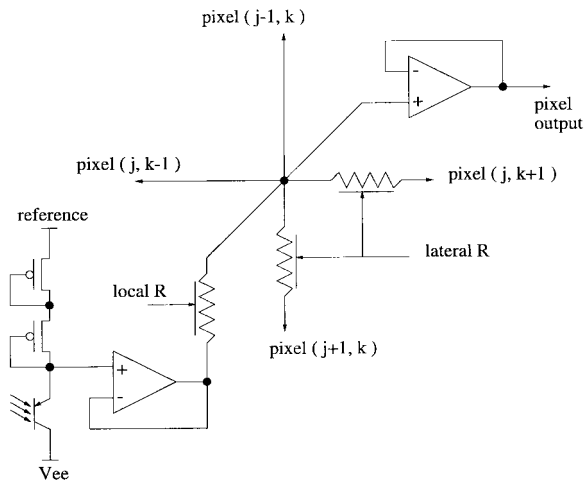


Fig. 6. Schematic for the  $ij$  pixel and connections to neighboring pixels.

which is best read from right to left (from high  $T$  to low  $T$ ) as in algorithmic annealing. At voltages corresponding to high  $T$ , Circuit (D) behaves like the (linear) LMS deconvolution filter because a small signal approximation holds for the exponentials of (18) for the  $R$  in Circuit (C). The right-hand side of Fig. 5 is therefore the LMS deconvolution. This LMS estimate varies 20% from the correct values of +1 and 0, because LMS deconvolution amplifies the 10% added error. As the temperature-like voltage is reduced, the estimates for the three pixels of the foreground converge to 1.0 V and the values for the remaining background pixels converge to 0.0 V, to within a few percent. The largest error is 0.07 V and the average error is 3% of the range of 0 to 1 V; this is more than a factor of five better than a LMS filter. Other simulations show that errors are even smaller at lower values of the temperature-like voltage, but these are difficult to simulate in SPICE.

### III. SIMULATION OF CONVOLUTION IN ANALOG VLSI

We simulated a chip that acquires a 2-D image and performs a 2-D convolution on it using analog processing elements in each pixel shown schematically as in Fig. 6. The calculations for all the pixels are performed in parallel. Voltage-controlled local and lateral resistors are used to determine the convolution kernel.

Each pixel comprises a buffered photodetector circuit, a local voltage-controlled resistor, two lateral voltage-controlled resistors, and a pixel output buffer amplifier. The photodetector circuit [18] consists of a vertical p-n-p transistor which is loaded by two series diode-connected pMOS devices, as shown in Fig. 6. Incident photons create electron-hole pairs in the bipolar device, resulting in a photocurrent proportional to the light intensity. The photocurrent drives the two load devices, and the voltage across them is buffered to the local resistor which is connected to the output node. The MOS devices are biased in the subthreshold region, where the relationship between voltage and current is approximately logarithmic over a range of approximately four orders of magnitude. The output node is connected to neighboring output nodes by two lateral resistors (down and right) and by two wires (left and up)

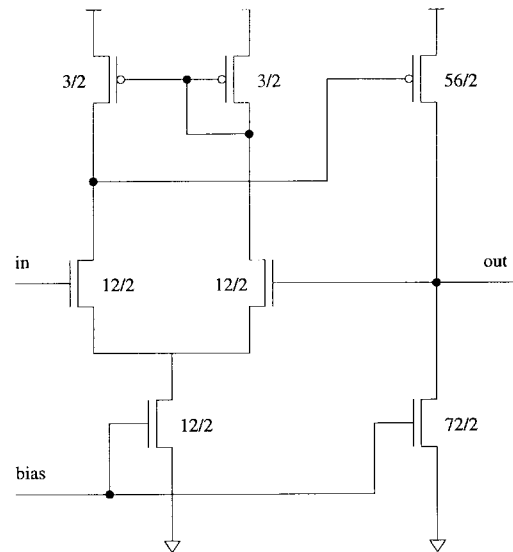


Fig. 7. Schematic for opamps used in each pixel.

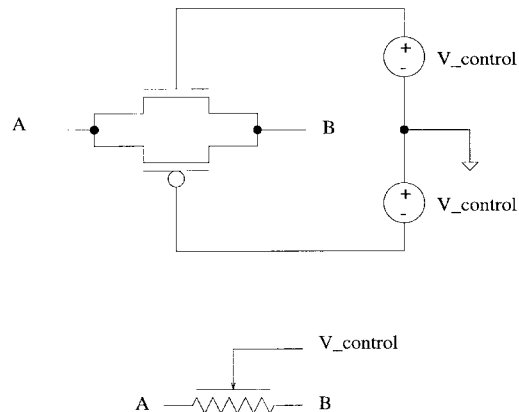


Fig. 8. Schematic and symbol for the voltage-controlled resistors which provide user control over the convolution kernel.

and is buffered to external connections. All local resistors are controlled by a single control voltage  $V_{\text{local}}$ , and all lateral resistors are controlled by a second control voltage  $V_{\text{lateral}}$ . The supply voltages are +9 and -9 V.

The two-stage CMOS operational amplifier shown in Fig. 7 in a unity-gain feedback configuration is used to buffer the photodetector and the output. The opamp transistors are sized to provide sufficient low frequency current over a wide input voltage range.

The voltage-controlled resistors are implemented by parallel nMOS and pMOS transistors whose gates are driven by control voltages with symmetrical values above and below ground, as shown in Fig. 8. By using both device types with symmetrical control voltages, the MOS body effect can be largely cancelled, and so a true floating resistor can be created. These resistors have a fairly linear control range that spans about a factor of four in resistance values. Therefore, we have a 16:1 range of resistance ratios resulting in a 4:1 range in  $\lambda$  values for the convolution kernel.

We laid out the chip using MAGIC. The physical design of each pixel cell is illustrated in Fig. 9. The square area

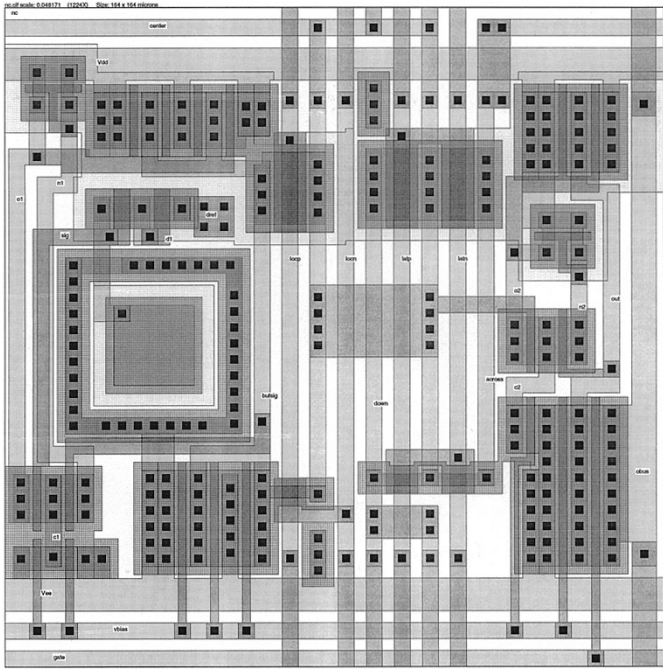


Fig. 9. MAGIC layout of one pixel of the image sensor with user controllable low-pass filtering.

near the left center of the cell is the phototransistor, and the two diode-connected load transistors are located immediately above it. The pMOS and nMOS transistors of the photodetector buffer amplifier are placed in the upper left and lower left areas, respectively. Just to the right of the buffer are the two transistors of the local resistor, and further to the right are the two lateral resistors. Running vertically through the middle of the cell are the four resistor control voltages. The connection to the adjacent pixel below is routed between the control voltages at the lower center of the cell and the connection to the pixel to the right is seen in the upper right corner. The wire in the upper left portion of the cell provides the connection from the pixels to the left and above. Finally, the output buffer, multiplexer gate transistor, and output bus compose the rightmost quarter of the cell layout.

The unblurred input and the blurred output of an array of  $16 \times 48$  pixels was simulated. For the unblurred input of Fig. 10 the simulated output from the convolution chip is shown in Fig. 11 for  $V_{\text{local}} = \pm 9$  V and  $V_{\text{lateral}} = \pm 3$  V.

We simulated the transient behavior of a group of pixels, including realistic timing and control logic (but with a simplified model of the phototransistor), for many values of the voltages  $V_{\text{local}}$  and  $V_{\text{lateral}}$ .  $RC$  oscillations could be observed at the output when the inputs representing photocurrents were abruptly changed, but we could produce no oscillation with any characteristic time that exceeded  $1 \mu\text{s}$ . We conclude that the proposed blurring circuit is faster than is necessary for real time operation, which requires only that each cell provide stable outputs appropriate to changes in photocurrent inputs within a few tens of milliseconds. For convolution, the settling time of any group of pixels is  $O(\lambda^2)$  and is independent of the size of the array for more than  $O(\lambda^2)$  pixels, so the system speed will always be limited by I/O rather than by

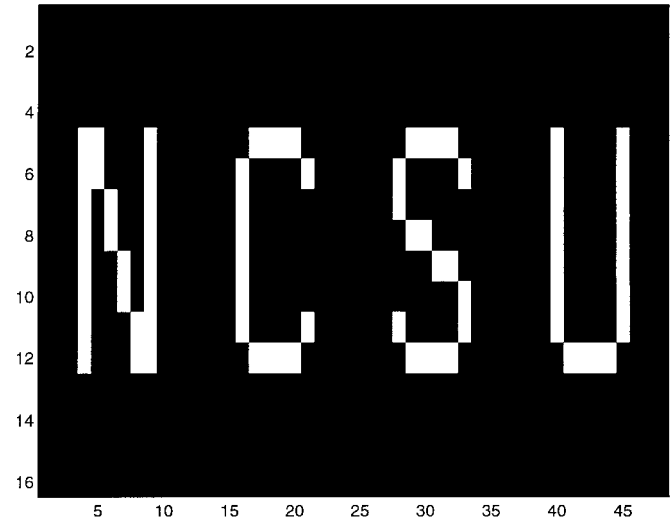


Fig. 10. Input to the simulation of a  $16 \times 48$  cell convolution chip. The numbers along the side and bottom are the cell indices.

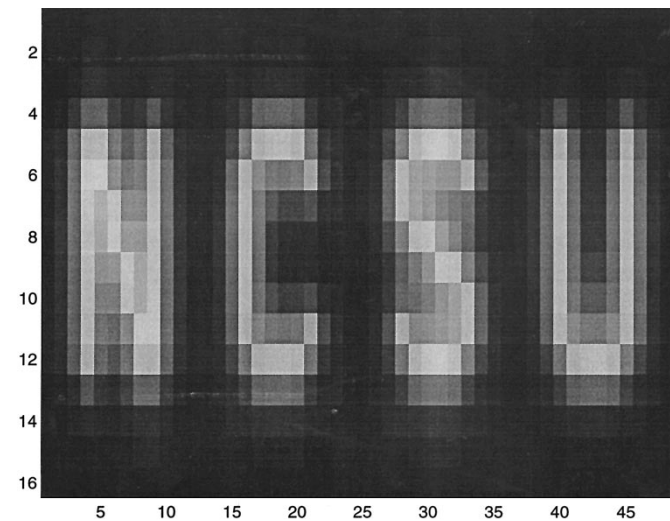


Fig. 11. Simulated output from the convolution chip for  $V_{\text{local}} = \pm 9$  V and  $V_{\text{lateral}} = \pm 3$  V. The numbers along the side and bottom are the cell indices as in Fig. 10.

the settling time for cells. Our  $16 \times 48$  cell circuit will operate at video frame rate (30 frames/s) with our current I/O design. Much larger arrays would require interleaved I/O, but the image processing operations are relatively local (i.e., they occur over a distance of  $O(\lambda)$  pixels) and therefore do not incur any additional communication delay. Our experience with algorithmic convolution suggests that for  $\lambda \geq 2$  and for more than  $256 \times 256$  pixels, convolution will be faster in analog VLSI than on a general purpose serial computer such as a Sun Ultra 1. The complexity of algorithmic convolution increases as  $\lambda^2$  up to large enough  $\lambda$  to prefer digital FFT. Image processing programs we have written in C and run on a 200-MHz Pentium cannot do *any* arithmetic processing at 30 frames/s for 8-bit images as small as  $512 \times 512$  pixels.

#### IV. A CONVOLUTION CHIP

We fabricated an  $8 \times 8$  version of this convolution chip, shown in Fig. 12. On the actual chip, the interior pixels are

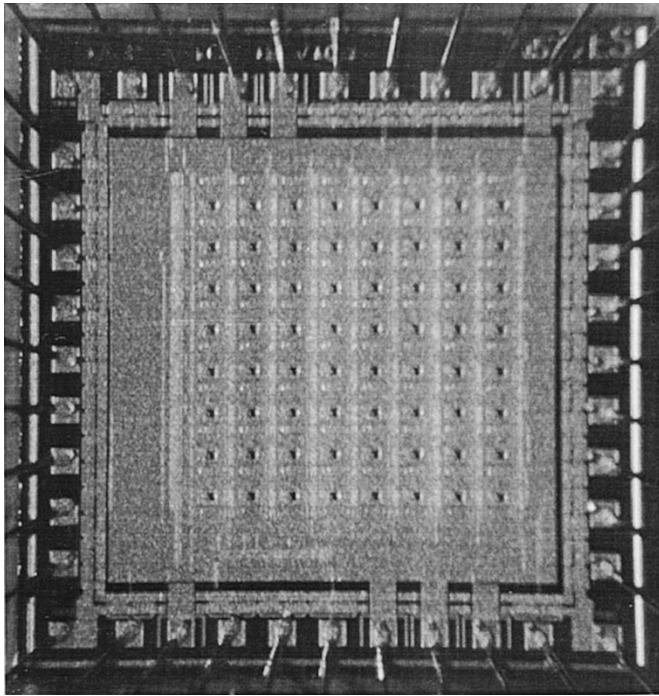


Fig. 12. Photograph of  $8 \times 8$  convolution chip.

connected as described above. At each edge of the array, all eight nodes are connected, through lateral resistors, to an externally accessible bus. Sixteen extra lateral resistors were added for the top and left edges. This arrangement allows flexibility in exploring the effects of alternative boundary conditions for the convolution.

Each pixel has an area of  $160 \mu\text{m} \times 160 \mu\text{m}$ . The entire circuit covers about  $1.8 \text{ mm} \times 1.8 \text{ mm}$  and was implemented in a  $2\text{-}\mu\text{m}$  CMOS n-well two-metal process. The second metal layer is used as a light shield over all circuitry except the vertical p-n-p. This layer also serves as the conductor for the photodetector reference voltage, so that a third metal layer would simplify the wiring. Except for the photodetector, the circuit scales with feature size. We believe that the photodetector can also be scaled directly, but we have not yet shown this. Therefore, a conservative estimate of the size of each pixel in a  $0.5\text{-}\mu\text{m}$  three-metal process is  $80 \mu\text{m} \times 80 \mu\text{m}$ .

Since it is not possible to give all 64 pixels a dedicated output pin, the outputs are multiplexed off the chip one row at a time. A 3-to-8 decoder is used to select a single pixel row to be gated onto the eight output lines. The decoder is implemented in static CMOS logic as is shown for one corner of the chip in Fig. 13. This same scheme would suffice up to  $64 \times 64$  pixels, but larger arrays would require interleaved output.

Using a microscope, we illuminated the convolution chip with the cross-shaped pattern shown in Fig. 14, for which the convolution effect ( $\lambda$ ) was minimized by setting  $V_{\text{localN}} = 5 \text{ V}$ ,  $V_{\text{localP}} = 0 \text{ V}$ ,  $V_{\text{lateralN}} = 0 \text{ V}$ , and  $V_{\text{lateralP}} = 5 \text{ V}$ . Using the MOSIS characterization for this run and for these voltages,  $R_{\text{local}} = 8.5 \text{ k}\Omega$  and  $R_{\text{lateral}}$  is effectively infinite, so that  $\lambda = 0$ . The output of the chip is inverted from the incident light signal and for this small  $\lambda$  the chip produces an image with no filtering (i.e., smoothing) at all. Fig. 15 was obtained

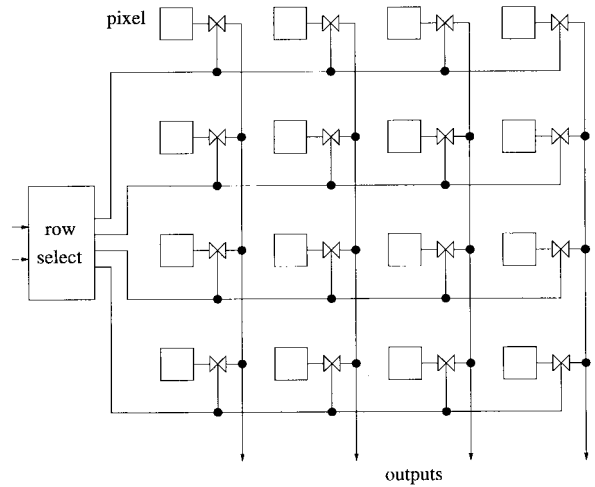


Fig. 13. Row select logic for output multiplexing (only lower left  $4 \times 4$  corner is shown here).

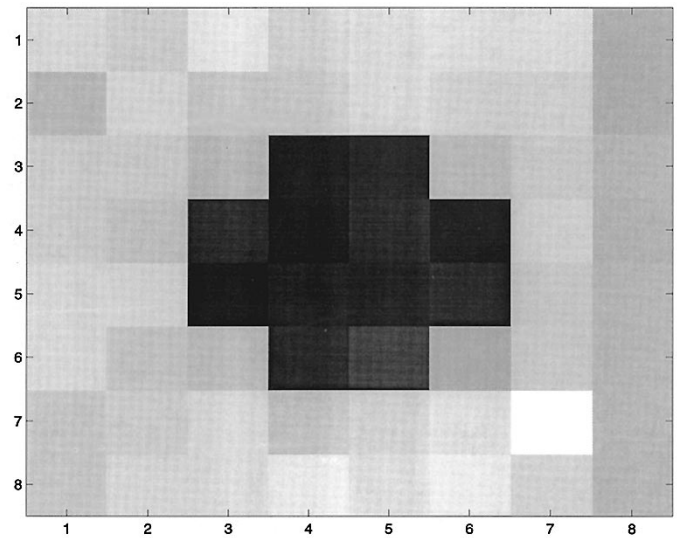


Fig. 14. Output from the convolution chip with minimal convolution.

with the control voltages set to  $V_{\text{localN}} = 3 \text{ V}$ ,  $V_{\text{localP}} = 2 \text{ V}$ ,  $V_{\text{lateralN}} = 2.3 \text{ V}$ , and  $V_{\text{lateralP}} = 2.7 \text{ V}$ , which corresponds to a predicted  $R_{\text{local}} = 120 \text{ k}\Omega$  and  $R_{\text{lateral}} = 55 \text{ k}\Omega$  to obtain a predicted  $\lambda^2 = R_{\text{local}}/R_{\text{lateral}} = 2.18$  and  $\lambda = 1.48$ , but both the resistors are somewhat nonlinear for these voltages. Examination of Fig. 15 indicates an actual value of  $\lambda \approx 1.7$  or  $R_{\text{local}}/R_{\text{lateral}} \approx 3$ .

Figs. 14 and 15 have been scaled independently to maximize contrast in each figure. The actual voltages output in the first case were from 0.65 to 2.92 V with a range of 2.27 V. In the second case, measured output voltages were from 1.43 to 2.01 V, giving a much smaller range of 0.58 V due to blurring.

Also evident in Figs. 14 and 15 is the fixed pattern noise due to pixel variations that is typical in CMOS imagers. As a side effect, the convolution also smoothes out this noise for large enough  $\lambda$  although this effect is not significant in Fig. 15. Nevertheless, even for imagers that filter their output, correlated double sampling would provide superior results for

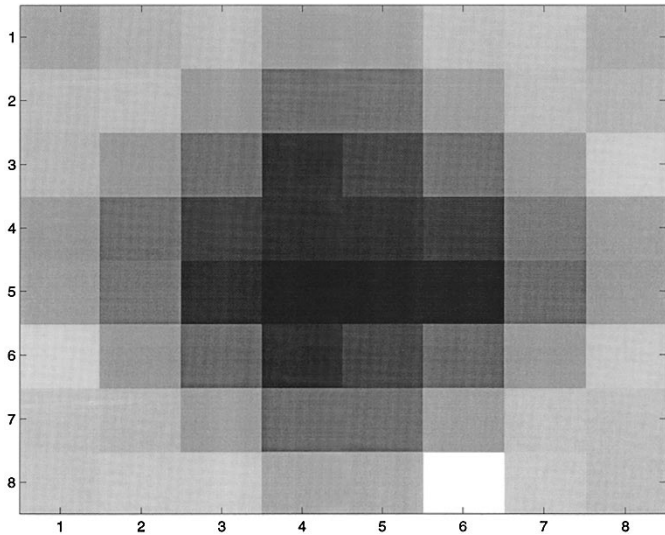


Fig. 15. Output from the convolution chip with visible convolution.

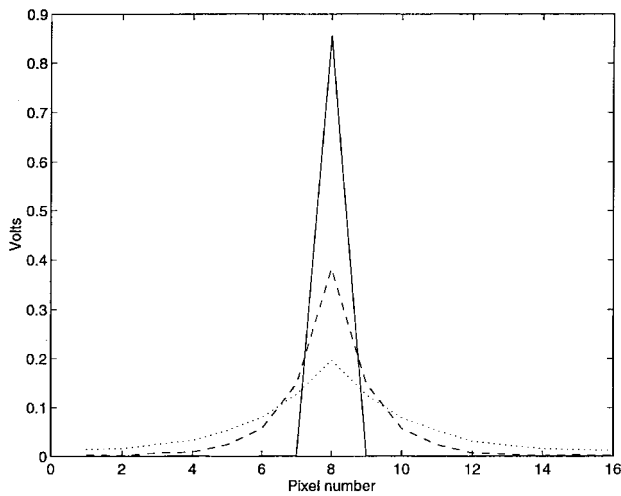


Fig. 16. Measured input signal to PCB, *Dashed line*: convolution  $\lambda \approx 1$  pixels,  $10\text{-k}\Omega$  local resistors, and  $10\text{-k}\Omega$  lateral resistors. *Dotted line*: convolution  $\lambda \approx 4.5$  pixels,  $10\text{-k}\Omega$  local resistors, and  $2.2\text{ k}\Omega$  lateral resistors.

fixed pattern noise, since it is more specific and does not reduce spatial resolution. Correlated double sampling is completely compatible with the filtering imagers we are proposing and would be interesting to include in subsequent chips.

### V. A DECONVOLUTION PCB

We implemented three 1-D 16-pixel image processing modules on PCB using commercial IC's: a 16 pixel 1-D deconvolution analog circuit with user-adjustable inputs containing 16 copies of Circuit (D) in Fig. 4, a convolution PCB that contains 16 copies of Circuit (B) in Fig. 1 with replaceable resistor arrays to provide user-controllable  $\lambda$  for blurring, and a 16-pixel dc vector signal generator that provides 16 potentiometer-controlled voltages. Input power conditioning is provided by an LM317T linear voltage regulator. All other active circuit functions are performed by LM324 quad single supply low-power opamps. A 9-V battery provides power.

Fig. 16 shows an (spatial) impulse input (solid line) and convolved outputs for two different blurring kernels, obtained

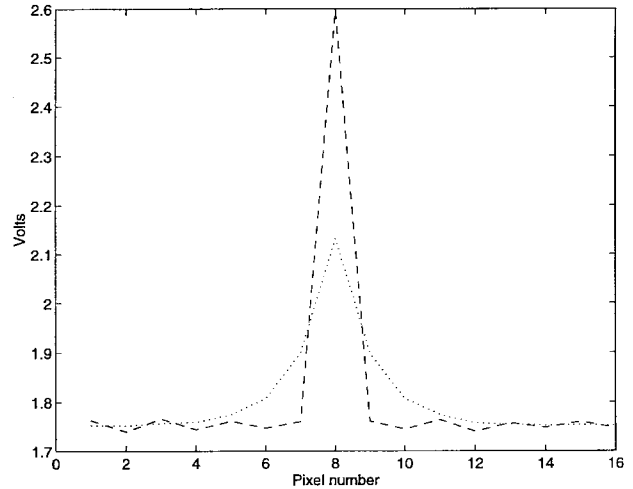


Fig. 17. *Dashed line*: measured input to deconvolution array. *Dotted line*: the corresponding deconvolution as measured from the PCB for the smaller kernel ( $10\text{-k}\Omega$  source resistors and  $10\text{-k}\Omega$  transverse resistors).

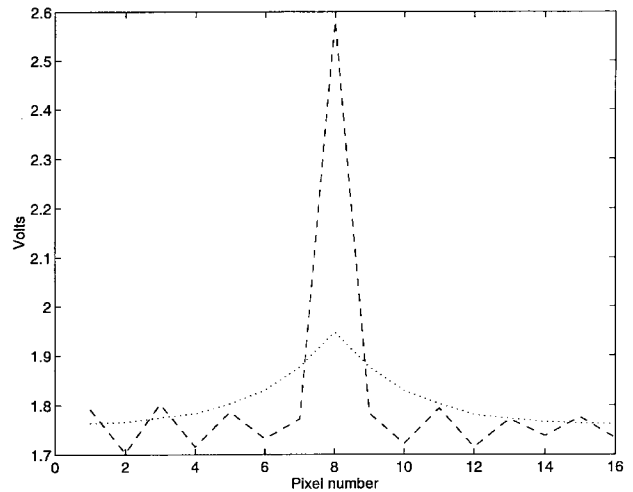


Fig. 18. *Dashed line*: convolved input to deconvolution array. *Dotted line*: the corresponding deconvolution as measured from the PCB for the larger kernel ( $10\text{-k}\Omega$  source resistors and  $2.2\text{-k}\Omega$  transverse resistors). Note the high spatial frequency noise with amplitude of about  $0.05\text{ V}$  in the deblurred output.

by manually swapping the lateral resistors between measurements.

Measured results from the deconvolution module are shown in Figs. 17 and 18 for the blurred signals shown in Fig. 16. Fig. 17 shows that the output signal is numerically almost identical to the original unblurred input for the smaller kernel. Fig. 18 shows that for the larger kernel the output signal begins to exhibit the high spatial frequency noise expected from a simple inverse filter.

### VI. CONCLUSION

We formulated several standard image processing operations in terms of circuits suitable for implementation in analog VLSI.

We simulated a  $16 \times 48$  pixel chip for user-controlled convolution, and argued that larger versions of such a chip could be expected to run at video frame rate and that chips larger than  $256 \times 256$  pixels are expected to exceed algorithmic processing speeds.



We fabricated a  $8 \times 8$  convolution chip and reported favorable results for user-controlled convolution in analog VLSI.

We presented SPICE simulations of piecewise-constant MFA image restoration for a 10-pixel 1-D image; this is a more complex image operation that cannot be performed algorithmically in real time even for  $32 \times 32$ -pixel images.

We fabricated a 16-pixel one-dimensional PCB arrays that perform deconvolution in real time.

By jointly considering theory, simulations, test results, and experience with algorithmic image processing, we argued that convolution, deconvolution, and even MFA piecewise-constant image restoration can be realized similarly in analog electronics, and that moderately sized analog VLSI implementations would provide higher performance than algorithmic approaches, especially for more complex image processing operations.

#### REFERENCES

- [1] J. E. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. R. Statis. Soc. B*, pp. 192–236, 1974.
- [2] G. L. Bilbro, T. K. Miller, W. E. Snyder, D. E. Van den Bout, M. W. White, and R. C. Mann, "Optimization by the mean field approximation," in *Advances in Neural Network Information Processing Systems 1*. San Mateo, CA: Morgan-Kaufman, 1989, reprinted from 1988 NIPS, Denver, CO.
- [3] G. L. Bilbro and W. E. Snyder, "Applying mean field annealing to image noise removal," *J. Neural Network Comput.*, pp. 5–17, Fall 1990.
- [4] G. L. Bilbro, W. E. Snyder, and R. C. Mann, "The mean field minimizes relative entropy," *J. Opt. Soc. Amer. A*, vol. 8, no. 2, pp. 290–294, Feb. 1991.
- [5] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.
- [6] A. D. Elster, *Magnetic Resonance Imaging: A Reference Guide and Atlas*. New York: Lippincott, 1986.
- [7] S. J. Garnier, G. L. Bilbro, J. W. Gault, and W. E. Snyder, "The effects of various basis image priors on MR image map restoration," *J. Math. Imaging Vision*, vol. 5, no. 1, pp. 21–41, 1995.
- [8] ———, "Magnetic resonance image restoration," *J. Math. Imaging Vision*, vol. 5, no. 1, pp. 7–19, 1995.
- [9] D. Geiger and F. Girosi, "Coupled Markov random fields and mean field theory," in *Advances in Neural Network Information Processing Systems 2*, D. S. Touretzky, Ed. San Mateo, CA: Morgan-Kaufman, 1990.
- [10] D. Geman and S. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, pp. 721–741, Nov. 1984.
- [11] R. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1987.
- [12] H. P. Hiriyanaiiah, "Signal reconstruction using mean field annealing," Ph.D. dissertation, North Carolina State Univ., Raleigh, 1990.
- [13] H. P. Hiriyanaiiah, G. L. Bilbro, W. E. Snyder, and R. C. Mann, "Restoration of piecewise constant images via mean field annealing," *J. Opt. Soc. Amer. A*, vol. 6, no. 12, pp. 1901–1912, Dec. 1989.
- [14] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.
- [15] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [16] M. D. Levine, *Vision in Man and Machine*. New York: McGraw-Hill, 1985.
- [17] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Amer. Stat. Assoc.*, vol. 82, no. 397, pp. 76–89, Mar. 1987.
- [18] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.
- [19] N. Wiener and P. Masani, Eds., *Norbert Wiener: Collected Works—Vol. 1*. Cambridge, MA: MIT Press, 1976.
- [20] Y. T. Zhou, R. Chellappa, and A. Vaid, "Image restoration using a neural network," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1141–1151, July 1987.



**Griff L. Bilbro** (M'85–SM'94) received the B.S. degree in physics from Case Western Reserve University, Cleveland, OH, and the Ph.D. degree in 1977 from the University of Illinois at Urbana-Champaign, IL, where he was a National Science Foundation Graduate Fellow in physics.

He designed computer models of complex systems in industry until 1984, then accepted a research position in the Department of Electrical and Computer Engineering, North Carolina State University (NCSU), Raleigh. He joined the faculty of NCSU in 1992 as an Associate Professor. He has published works on image analysis, global optimization, neural networks, and device physics. He is a member of the Electronics Research Laboratory and the Center for Advanced Computation and Communication at NCSU. His current interests include RF devices and analysis circuits.

**Lester C. Hall**, photograph and biography not available at the time of publication.



**Mark Clements** received the B.S. and M.S. degrees in electrical engineering in 1988 and 1994, respectively, from North Carolina State University (NCSU). He is currently working toward the Ph.D. degree in electrical engineering at NCSU.

From 1988 to 1992, he worked as an Applications Engineer in the Logic Analyzer Products Group of Tektronix, Inc. Since 1993, he has served as a Research Assistant in the Department of Electrical and Computer Engineering at NCSU. His research interests include new architectures for sampling and generating very high speed data, circuit techniques for high performance VLSI, and circuits for retinal prostheses and image processing.



**Wentai Liu** (S'78–M'81–SM'93) received the B.S.E.E. degree from National Chiao Tung University, the M.S.E.E. degree from National Taiwan University, Taiwan, and the Ph.D. degree in computer engineering from the University of Michigan at Ann Arbor in 1983.

Since 1983, he has been on the faculty of North Carolina State University, where he is currently an Associate Professor of Electrical and Computer Engineering. As a consultant he has developed several VLSI CAD tools for microelectronic companies. He holds three U.S. patents and has led a research group on wave pipelining and high-speed digital circuit design at North Carolina State University. As a pioneer in the research area of CMOS wave pipelining and timing optimization, he has been invited to present research results in Germany, Brazil, and Taiwan. He has coauthored a book entitled *Wave Pipelining: Theory and CMOS Implementation*, (Norwell, MA: Kluwer Academic, 1994). His research interests include high speed VLSI design/CAD, microelectronic sensor design, high speed communication networks, parallel processing, and computer vision/image processing.

Dr. Liu received an IEEE Outstanding Paper Award in 1986. He is a council member of the *IEEE Solid-State Circuits Society*.