
Computing Motion Using Analog and Binary Resistive Networks

James Hutchinson*
Jet Propulsion Laboratory

Christof Koch, Jin Luo, and Carver Mead
California Institute of Technology

To us, and to other biological organisms, vision seems effortless. We open our eyes and we “see” the world in all its color, brightness, and movement. Flies, frogs, cats, and humans can all equally well perceive a rapidly changing environment and act on it. Yet, we have great difficulties when trying to endow our machines with similar abilities. In this article, we describe recent developments in the theory of early vision that led from the formulation of the motion problem as an ill-posed one to its solution by minimizing certain “cost” functions. These cost or energy functions can be mapped onto simple analog and digital resistive networks. Thus, we can compute the optical flow by injecting currents into resistive networks and recording the resulting stationary voltage distribution at each node. These networks, which we implemented in complementary metal oxide semiconductor (CMOS) very large scale integrated (VLSI) circuits, represent plausible candidates for biological vision systems.

Motion

The movement of objects relative to eyes or cameras serves as an important

We can compute optical flow by injecting currents into resistive networks and recording the stationary voltage distribution at each node.

source of information for many tasks. We need motion to track objects and to determine whether an object is approaching or receding. Relative motion contains information regarding the three-dimensional structure of objects and allows biological organisms to navigate quickly and efficiently through the environment.

There exist two basic methods for computing motion. Intensity-based schemes

rely on spatial and temporal gradients of the image intensity to compute the speed and the direction in which each point in the image moves. The output is a velocity or motion vector field covering the entire image. The second method is based on the identification of special features in the image, called *tokens*, which are then matched from image to image. This method relies on the unambiguous identification of the tokens—for instance, corners—in each image frame before the matching occurs and only yields a velocity vector at the sparse token locations. Psychophysical evidence suggests that both systems coexist in humans.¹

The principal drawback of all intensity-based schemes lies in the data used—temporal variations in brightness patterns—which give rise to the perceived motion field, the *optical flow*. In general, the optical flow and the underlying velocity field, a purely geometrical concept, differ.² For example, a featureless rotating sphere will not give rise to any optical flow, because the brightness does not appear to change even though the velocity field is non-zero. Conversely, if a shadow moves across the same featureless but now stationary sphere, the optical flow is non-zero although the velocity field is zero. Apart from such situations, the estimated opti-

*Hutchinson is now with Thinking Machines Corp.

cal flow will be very nearly identical to the underlying velocity field, if strong enough gradients exist in the image. In this article, we assume that such strong gradients exist, as they do for most natural scenes, and consider how we can compute the velocity field using simple resistive networks.

Aperture problem. Let us derive an equation relating the change in image brightness to the motion of the image.² We denote the image at time t by $I(x, y, t)$. Let us assume that the brightness of the image is constant over time:

$$\frac{dI}{dt} = 0 \quad (1)$$

This will be true, for instance, if a rigid object translates in space (assuming orthographic projection), but not if it rotates. On the basis of the chain rule of differentiation, Equation 1 transforms into

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (2)$$

$$I_x u + I_y v + I_t = \nabla I \cdot \mathbf{v} + I_t = 0$$

where we define the velocity \mathbf{v} as $(u, v) = (dx/dt, dy/dt)$, and where I_x , I_y , and I_t are the partial derivatives of the brightness I with respect to x , y , and t . Because we assume that we can compute these spatial and temporal image gradients, we now have a single linear equation in two unknowns, u and v , the two components of the velocity vector.

In other words, this equation by itself is not sufficient to determine the velocity field. Figure 1 graphically illustrates this *aperture problem*. Any measuring system with a finite aperture, whether biological or artificial, can only sense the velocity component perpendicular to the edge or along the spatial gradient $(-I_t/|\nabla I|)$. The component of motion perpendicular to the gradient cannot, in principle, be registered. The problem remains unchanged even if we measure these velocity components at many points throughout the image. For each measurement, we recover one equation with two unknowns.

Smoothness assumption. Formally, this problem can be characterized as ill-posed.³ Hadamard introduced this concept to describe problems in mathematical physics that (1) have no solution at all, (2) have no unique solution, or (3) do not depend continuously on the initial data. Inverse problems, such as computer tomography, represent ill-posed prob-

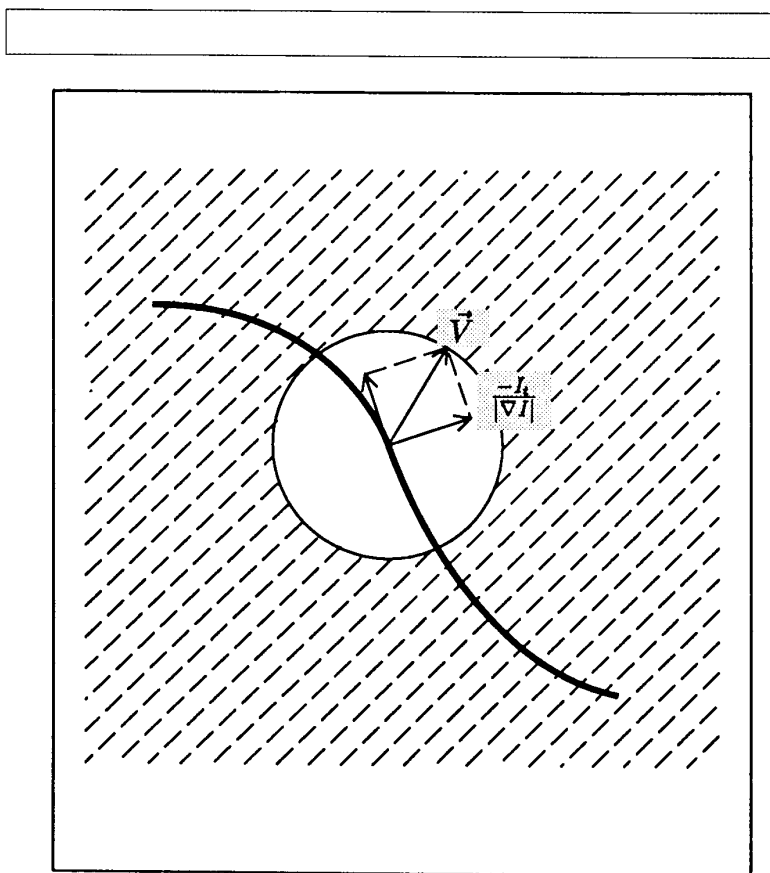


Figure 1. The aperture problem of motion. Any system with finite aperture, whether of biological or artificial origin, can only measure the velocity component $-I_t/|\nabla I|$ along the spatial gradient ∇I . Motion perpendicular to the gradient will not be visible, except by tracking salient features in the image.¹

lems. All problems in *early vision*, which we define as the set of processes that recover the properties of the visible three-dimensional surfaces from the two-dimensional intensity arrays on retinæ or cameras, are ill-posed. For example, binocular stereo and interpolating surfaces from sparse and noisy data are ill-posed, because in the former many and in the latter infinitely many solutions exist.

How can we make these problems well-posed, with unique solutions depending continuously on the data? One method of "regularizing" ill-posed problems involves restricting the class of admissible solutions by imposing appropriate constraints.³ Applying this method to

motion, we argue that, in general, objects are smooth—except at isolated discontinuities—undergoing smooth movements. Thus, in general, neighboring points in the world will have similar velocities. The projected velocity field should reflect this fact. We therefore impose on the velocity field the constraint that it should be the smoothest (in a given sense) while satisfying the data. As the measure of smoothness we choose the square of the velocity field gradient:

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \quad (3)$$

A variational functional provides the most

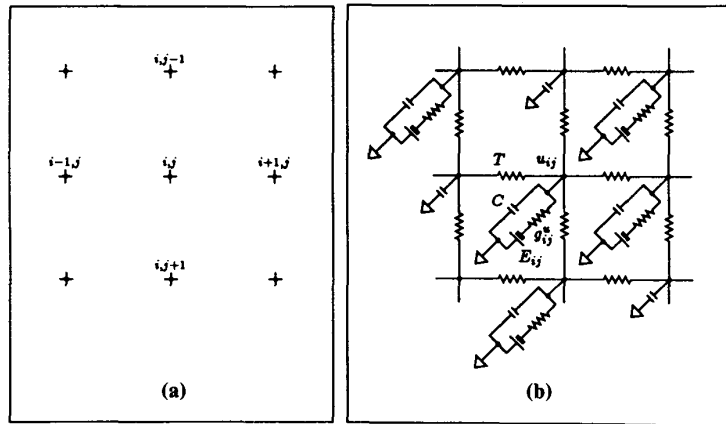


Figure 2. Rectangular grid and resistive network. (a) Rectangular grid for solving the discrete version of Equation 4. (b) Part of the resistive network minimizing the discrete approximation of the energy function E in Equation 4. We assume the conductance T connecting neighboring nodes to be constant. Each node connects to a variable battery E_{ij} via a conductance g_{ij}^u . Parasitic capacities (on the order of 0.1 picofarad) give the circuit its dynamic behavior. The final network consists of two such resistive networks superimposed, where corresponding nodes are connected via a variable conductance T_{c-ij} , as in Figure 3b. Once the batteries E_{ij} and conductances g_{ij}^u and g_{ij}^v have been set, the network will converge—following Kirchhoff's laws—to the state of least power dissipation that corresponds to the solution of the variational Equation 4.

general way to formulate the problem.² The final velocity field (u, v) minimizes

$$E(u, v) = \iint (I_x u + I_y v + I_t)^2 + \lambda \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] dx dy \quad (4)$$

where the regularization parameter λ is inversely dependent on the signal-to-noise ratio. The first term describes the fact that the final solution should follow as closely as possible the measured data, whereas the second term imposes the smoothness constraint on the solution. The degree of minimization of one or the other term is governed by λ . With accurate data, violating the first term should be "expensive" and λ will be small. Conversely, with unreliable data (low signal-to-noise ratio), much more emphasis will be placed on the smoothness term. Horn and Schunck² first formulated this variational approach to the motion problem.

The energy $E(u, v)$ is quadratic in the

unknown u and v . It then follows from standard calculus of variation that the associated Euler-Lagrange equations will be linear in u and v :

$$\begin{aligned} I_x^2 u + I_x I_y v - \lambda \nabla^2 u + I_x I_t &= 0 \\ I_x I_y u + I_y^2 v - \lambda \nabla^2 v + I_y I_t &= 0 \end{aligned} \quad (5)$$

We now have two linear equations at every point. Our problem is therefore completely determined. We could now use a number of iterative techniques, such as steepest descent, to solve these equations. Instead, we pursue a different path.

Analog resistive networks. Let us assume that we are formulating Equations 4 and 5 on a discrete two-dimensional grid, such as the one shown in Figure 2a. Equation 5 then transforms into

$$\begin{aligned} I_{xij}^2 u_{ij} + I_{xij} I_{yij} v_{ij} - \lambda (u_{i+1j} + u_{ij+1} \\ - 4u_{ij} + u_{i-1j} + u_{ij-1}) \\ + I_{xij} I_{tij} &= 0 \end{aligned}$$

$$\begin{aligned} I_{xij} I_{yij} u_{ij} + I_{yij}^2 v_{ij} - \lambda (v_{i+1j} + \\ v_{ij+1} - 4v_{ij} + v_{i-1j} + v_{ij-1}) \\ + I_{yij} I_{tij} &= 0 \end{aligned} \quad (6)$$

where we replaced the Laplacian with its five-point approximation on a rectangular grid. We now show that this set of linear equations can be solved naturally using a simple resistive network. Let us apply Kirchhoff's current law to the center node of the resistive network shown in Figure 2b. We then have the following update equation:

$$\begin{aligned} C \frac{du_{ij}}{dt} &= T(u_{i+1j} + u_{ij+1} - 4u_{ij} + \\ u_{i-1j} + u_{ij-1}) &+ g_{ij}^u (E_{ij} - u_{ij}) \end{aligned} \quad (7)$$

Let us now assume that we have two such resistive networks superimposed, with the node ij in the upmost network connected—via a conductance T_{c-ij} —to the appropriate node ij in the bottom network (see Figure 3b). We then have two equations similar to Equation 7 with a coupling term $T_{c-ij}(v_{ij} - u_{ij})$, where v_{ij} is the voltage at node ij in the bottom network. If we assume that the resistive network has converged to its final state, $du_{ij}/dt = 0$ and $dv_{ij}/dt = 0$, we see that both equations are identical with Equation 6, if we identify

$$\begin{aligned} T &\rightarrow \lambda \\ T_{c-ij} &\rightarrow -I_{xij} I_{yij} \\ g_{ij}^u &\rightarrow I_{xij} (I_{xij} + I_{yij}) \\ g_{ij}^v &\rightarrow I_{yij} (I_{xij} + I_{yij}) \\ E_{ij} &\rightarrow \frac{-I_t}{I_{xij} + I_{yij}} \end{aligned} \quad (8)$$

Once we set the batteries and the conductances to the values indicated in Equation 8, the network will settle—following Kirchhoff's laws—into the state of least power dissipation. The associated stationary voltages correspond to the solution sought: u_{ij} is equivalent to the x component and v_{ij} to the y component of the optical flow field. A unique and stable solution always exists, even if some of the conductances have negative values. In fact, many of the conductances connecting the lower and the upper networks (T_{c-ij}) and the conductances associated with the batteries (g_{ij}^u and g_{ij}^v) will be negative, because the sign of I_x and I_y can be either negative or positive. As we will see, this poses no serious problems, given the technology we have chosen to build resistances. (See the sidebar "Parallel computer implementation.")

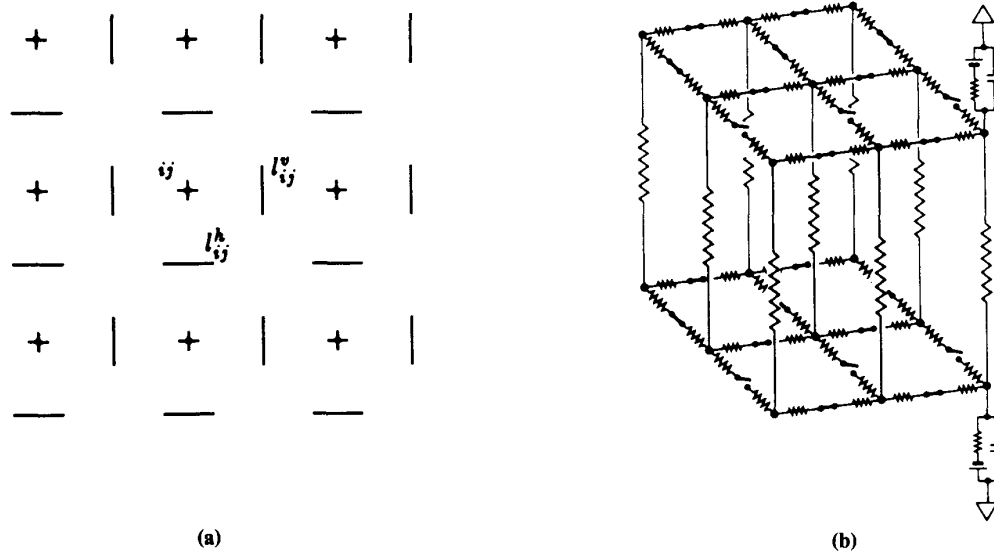


Figure 3. Rectangular grid with line processes and hybrid network. (a) The location of the horizontal (l_{ij}^h) and vertical (l_{ij}^v) line processes relative to the rectangular motion-field grid. (b) The hybrid resistive network, computing the optical flow in the presence of discontinuities. The conductances T_{c-ij} , which connect both grids, depend on the brightness gradient, as do the conductances g_{ij}^u and g_{ij}^v , which connect each node with the battery. For clarity, only two such elements are shown. The battery E_{ij} depends on both the temporal and the spatial gradient and is zero if no brightness change occurs. The x component of the velocity, u , is given by the voltage in the top network, while the y component of the velocity, v , is given by the voltage in the bottom network. A high voltage value at location i, j will spread to its four neighboring nodes. The degree to which voltage spreads depends on the value of the fixed conductance, T , given by the inverse of the signal-to-noise ratio. Binary switches, which make or break the resistive connections between nodes, implement motion discontinuities, because an arbitrary high voltage (velocity) will not affect the neighboring site across the discontinuity. An extended horizontal motion discontinuity is indicated. These switches could be under the control of distributed digital processors. Analog CMOS implementations of the line processes also are feasible.⁴

The sequences in Figures 4, 5, and 6 illustrate the resulting optical flow for synthetic and natural images. Figure 4c illustrates the initial velocity data and the velocity component perpendicular to the image gradient. Figure 4d shows the resulting smooth optical flow. As discussed by Horn and Schunck,² the smoothness constraint leads to a qualitatively correct estimate of the velocity field. Thus, one undifferentiated blob appears to move to the lower right and one blob to the upper left. However, at the occluding edge where both squares overlap, the smoothness assumption results in a spatial average of the two opposing velocities, and the estimated velocity is very small or zero.

Parallel computer implementation

We simulated the behavior of these networks for both synthetic and natural images by solving the previous circuit equations at each node. As boundary conditions, we copied the initial velocity data at the edge of the image into the nodes lying directly adjacent to but outside the image (zero normal derivative). We estimated the spatial and temporal derivatives I_x , I_y , and I_t using a discrete eight-point approximation.

Given the high computational cost associated with solving these elliptical equations, we used parallel com-

puters of the Hypercube family: the 32-node Mark III Hypercube at the Jet Propulsion Laboratory and a 4- and a 16-node Ncube in the laboratory at Caltech. Even though we used a variable time-step algorithm, convergence times were slow (10 minutes for a 128×128 image). Solving Equation 4 is similar to solving Poisson's equation. Thus, the number of iterations required to converge is proportional to n^2 (on an $n \times n$ -pixel image). A multigrid approach will greatly speed up the performance.

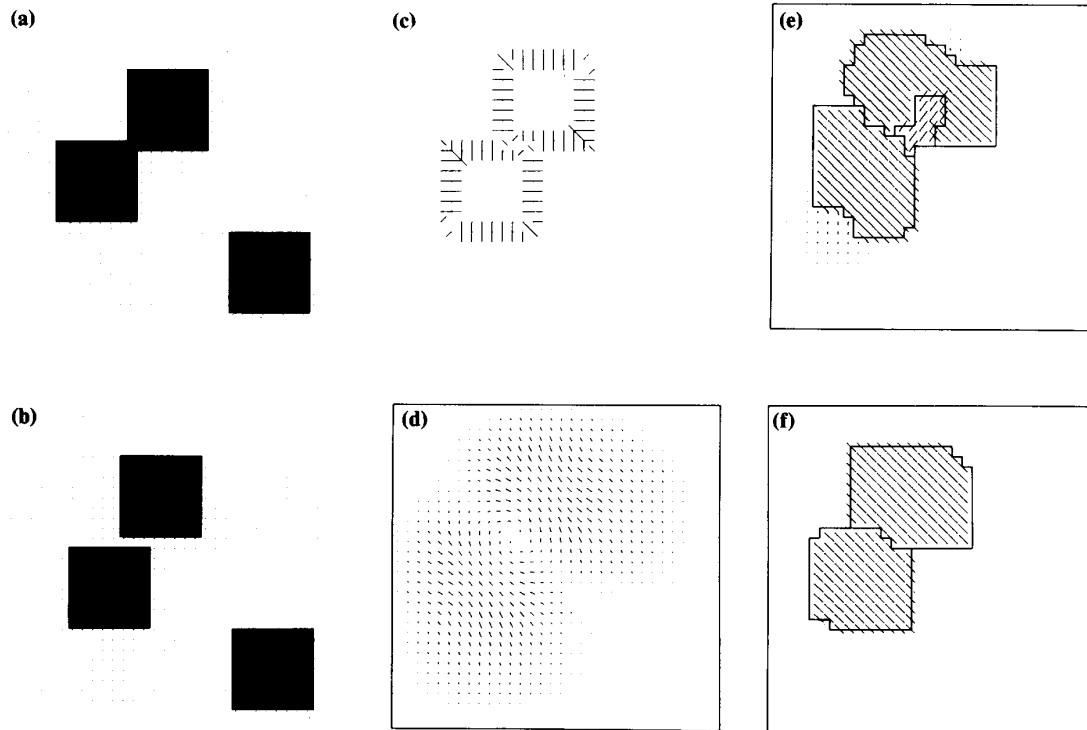


Figure 4. Motion sequence using synthetic data. (a) and (b) Two 32×32 -pixel images of three high-contrast squares on a homogeneous white background. Only the two squares on the upper left are displaced. (c) The initial velocity data. The insides of both squares contain no data. (d) The final state of the network after 240 iterations, corresponding to the smooth optical flow field. The algorithm gives a qualitatively correct estimate of the velocity field. Note, however, the vanishing velocity estimate at the occluding edges where the two moving squares overlap, caused by the averaging property of the smoothness constraint. Moreover, the moving objects are not delineated in the flow field, because the algorithm smooths over the figure-ground motion discontinuity. (e) Optical flow in the presence of motion discontinuities (indicated by solid lines). Numerous line processes are turned on in the area where the moving objects overlap. The formation of discontinuities along continuous contours is explicitly encouraged. (f) Discontinuities are strongly encouraged to form at the location of intensity edges.⁵ This additional constraint leads to the correct velocity field. The location of these discontinuities facilitates object segmentation at a later stage of visual analysis. Both (e) and (f) show the state of the hybrid network after six analog-digital cycles.

In parts of the image where the brightness gradient is zero and thus no initial velocity data exist (for instance, in the interiors of the two squares), the velocity estimate is simply the spatial average of the neighboring velocity estimates. These empty areas eventually will fill in from the boundary, similar to the flow of heat for a uniform flat plate with "hot" boundaries.

The sequence in Figure 5 also illustrates

the effect of varying the conductance T between neighboring points. As we place more confidence in the measured data (small λ), the coupling between neighboring nodes decreases because T decreases, and the optical flow becomes more inhomogeneous, better reflecting the correct velocity field. As the data becomes less reliable (large λ), more smoothing occurs until little spatial variation exists (see Figure 3).

Motion discontinuities. The smoothness assumption of Horn and Schunck² regularizes the aperture problem and leads to the qualitatively correct velocity field inside moving objects. However, this approach fails to detect the locations at which the velocity changes abruptly or discontinuously. Thus, this strategy smooths over the figure-ground discontinuity or completely fails to detect the boundary between two objects with differing veloc-

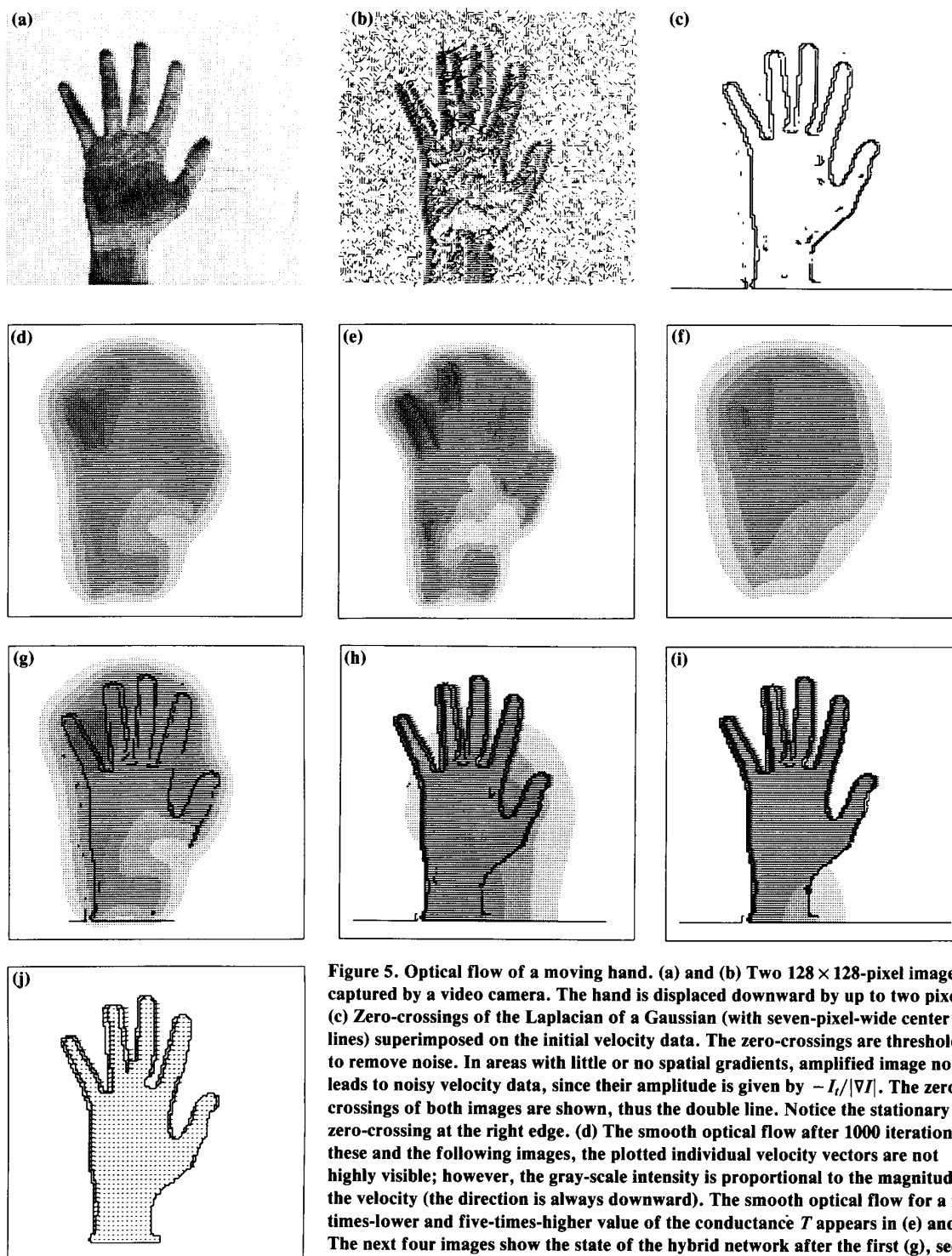


Figure 5. Optical flow of a moving hand. (a) and (b) Two 128×128 -pixel images captured by a video camera. The hand is displaced downward by up to two pixels. (c) Zero-crossings of the Laplacian of a Gaussian (with seven-pixel-wide center lines) superimposed on the initial velocity data. The zero-crossings are thresholded to remove noise. In areas with little or no spatial gradients, amplified image noise leads to noisy velocity data, since their amplitude is given by $-I_c/|\nabla I|$. The zero-crossings of both images are shown, thus the double line. Notice the stationary zero-crossing at the right edge. (d) The smooth optical flow after 1000 iterations. In these and the following images, the plotted individual velocity vectors are not highly visible; however, the gray-scale intensity is proportional to the magnitude of the velocity (the direction is always downward). The smooth optical flow for a five-times-lower and five-times-higher value of the conductance T appears in (e) and (f). The next four images show the state of the hybrid network after the first (g), second (h), fifth (i), and ninth (j) analog-digital cycles. In the final image, the fingers have a higher velocity than does the hand itself. It takes several cycles for the motion discontinuities to "creep" around the outline of the hand.

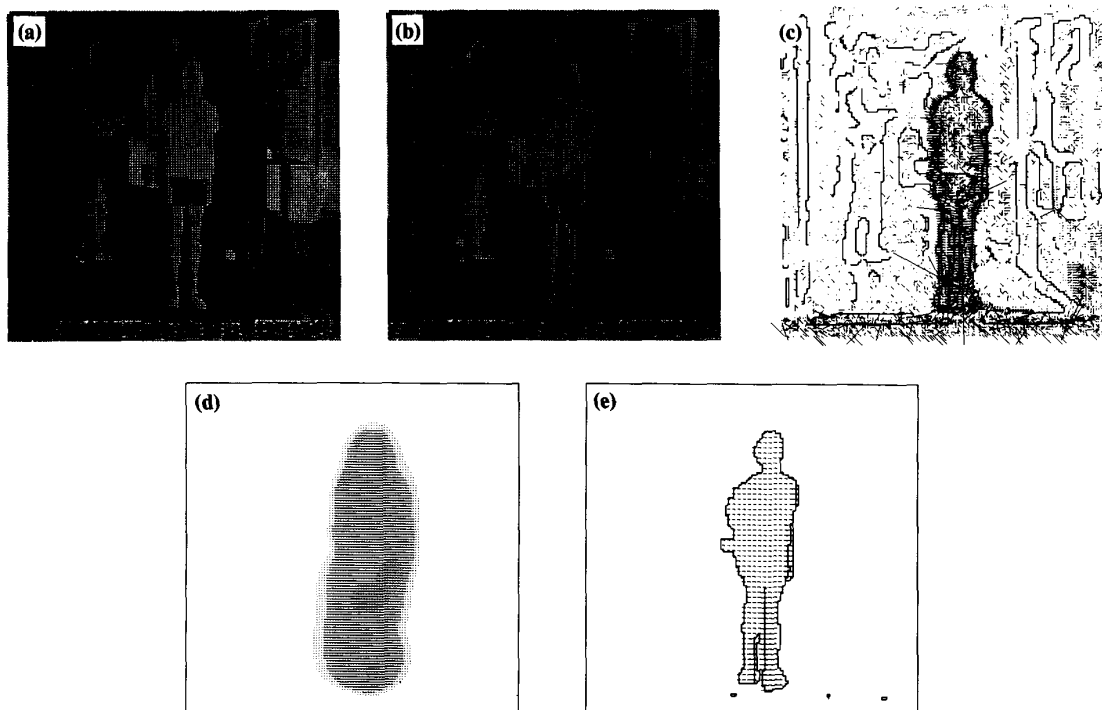


Figure 6. Optical flow of a moving person. (a) and (b) Two 128×128 pixel images captured by a video camera. The person in the foreground is moving toward the right while the person in the background is stationary. The noise in the lower part of the image is a camera artifact. (c) Thresholded zero-crossings superimposed on the initial velocity data. (d) The smooth optical flow after 1000 iterations. Note that the noise in the lower part of both video images is completely smoothed away. (e) The final piecewise smooth optical flow after 13 analog-digital cycles. The velocity field is subsampled to improve visibility. With the exception of a square appendage at the right hip, the optical flow field shown corresponds to the correct velocity field. The appendage, caused by the edge-detection scheme lumping part of the garbage can in the background with the contour of the person, represents an instance of what psychophysicists term motion capture. More recently, we have successfully computed the optical flow field for images with many, partially occluding, moving people.

ities, because the algorithm combines velocity information across motion boundaries. We argue that motion discontinuities are the most interesting locations in any image, because they indicate where one object ends and another one begins. Motion as well as intensity discontinuities are vital for solving the critical object segmentation problem that occurs at a subsequent stage of the image understanding process.

Various researchers have attempted to prevent the smoothing constraint from taking effect across strong velocity gradients.⁶ Geman and Geman⁷ proposed a successful strategy for dealing with discon-

tinuities. They exploited an analogy between statistical mechanics and images, whereby the intensity values at each picture element and the presence of discontinuities are viewed as states of particles on a lattice. We can assign an "energy" function to this system and compute its most likely state.

In this article, we do not rigorously develop this approach, based on Bayesian estimation theory.^{7,8} Suffice it to say that a priori knowledge (for instance, that the velocity field in general should be smooth) can be formulated in terms of a Markov random field model of the image. (In a Markov random field, the conditional

probability that a given variable at location i, j has a particular value f_{ij} depends only on the values of f in a neighborhood of i, j .) Given such an image model, and given noisy data, we then estimate the "best" flow field by some likelihood criterion. The one we use here is the maximum a posteriori estimate, although other possible criteria have certain advantages.⁸ Maximizing the a posteriori probability yields the best solution. We can show this to be fully equivalent to minimizing an expression such as Equation 4.

To reconstruct images consisting of piecewise constant segments, Geman and Geman⁷ further introduced the powerful

idea of a line process l (see also Blake and Zisserman⁹). For our purposes, we assume that a line process can occupy one of two states: "on" ($l = 1$) or "off" ($l = 0$). Line discontinuities are located on a regular lattice set between the original pixel lattice (see Figure 3a), such that each pixel ij has one horizontal l_{ij}^h and one vertical l_{ij}^v line process associated with it. If the appropriate line process is turned on, the smoothness term between the two adjacent pixels will be set to zero.

To prevent line processes from forming everywhere and to incorporate additional knowledge regarding discontinuities into the line processes, we must include an additional term $V_c(l)$ in the new energy function:

$$E(u, v, l^h, l^v) = \sum_{ij} (I_x u_{ij} + I_y v_{ij} + I_l)^2 + \lambda \sum_{ij} (1 - l_{ij}^h) [(u_{i+1j} - u_{ij})^2 + (v_{i+1j} - v_{ij})^2] + \lambda \sum_{ij} (1 - l_{ij}^v) [(u_{ij+1} - u_{ij})^2 + (v_{ij+1} - v_{ij})^2] + V_c(l) \quad (9)$$

V_c contains a number of terms penalizing or encouraging specific configurations of line processes:

$$V_c(l) = C_c \sum_{ij} l_{ij}^h + C_p \sum_{ij} l_{ij}^h [l_{ij+1}^h + l_{ij+2}^h] + C_l V_l(l) \quad (10)$$

plus the corresponding expression for the vertical line process l_{ij}^v (obtained by interchanging i with j and l_{ij}^h with l_{ij}^v). The first term (C_c) penalizes each introduction of a line process, because the cost C_c has to be "paid" every time a line process is turned on. The second term prevents the formation of parallel lines. If either l_{ij+1}^h or l_{ij+2}^h is turned on, this term will tend to prevent l_{ij}^h from turning on. The third term (C_l) embodies the fact that, in general, motion discontinuities occur along extended contours and rarely intersect. We adopt the function given by Koch et al.¹⁰ favoring the formation of motion discontinuities along extended contours and penalizing both multiple line intersections and isolated discontinuities.

We obtain the optical flow by minimizing the cost function in Equation 9 with respect to both the velocity field $v = (u, v)$ and the line processes l^h and l^v . However, unlike before, this cost or energy function is nonconvex, since it contains cubic and possibly higher terms (in V_c). Geman and Geman resorted to annealing, a statistical optimization technique, to find the ground state of their system. If annealing is applied appropriately, the system con-

verges with probability converging to one to the global maximum.⁷ However, the length of the required convergence times makes any practical application expensive.

To find an optimal solution to this non-quadratic minimization problem, we follow the approach used by Koch et al.¹⁰ and use a purely deterministic algorithm, based on solving Kirchhoff's equations for a mixed analog and digital network.^{8,11} Our algorithm exploits the fact that for a fixed distribution of line processes, the energy function of Equation 9 is quadratic. Thus, we first initialize the analog resistive network (see Figure 3b) according to Equation 8 and with no line processes on. The network then converges to the smoothest solution. Subsequently, we update the line processes by deciding at

each site of the line process lattice whether the overall energy can be lowered by setting or breaking the line process. We always accept the state of the line process corresponding to the lower energy configuration: l_{ij}^h will be turned on if $E(u, v, l_{ij}^h = 1, l^v) < E(u, v, l_{ij}^h = 0, l^v)$; otherwise, $l_{ij}^h = 0$. This computation requires only local information. Breaking the appropriate resistive connection between the two neighboring nodes switches on the line processes. After the completion of one such analog-digital cycle, we reiterate and compute the smoothest state of the analog network for the newly updated distribution of line processes.

Although we have no guarantee that the system will converge to the global minimum, given our use of a gradient descent

Restricting motion discontinuities to edges

As edges we use the zero-crossings of a Laplacian of a Gaussian convolved with the original image. Marr and Hildreth¹² have shown that these locations usually correspond to physical edges. We threshold these zero-crossings (based on the square of the gradient) in order to remove spurious zero-crossings caused by noise and "weak" edges. Other edge detection algorithms should work equally well.

We now add a new term $V_{z-c_{ij}}$ to our energy function E , such that $V_{z-c_{ij}}$ is zero if l_{ij} is off or if l_{ij} is on

and a zero-crossing exists between locations i and j . If $l_{ij} = 1$ in the absence of a zero-crossing, $V_{z-c_{ij}}$ is set to a large positive number (in our case, 1000).

This strategy effectively prevents motion discontinuities from forming at locations where no zero-crossings exist, unless strongly suggested by the data. Conversely, however, zero-crossings by themselves will not induce the formation of discontinuities in the absence of motion gradients.

Varying the 'amplitude' of motion discontinuities

When dealing with real data, the amplitude of velocity and, consequently, the amplitude of any motion discontinuity vary over a considerable range (as compared to the artificial situation in Figure 4). Our strategy in dealing with this problem involves varying the magnitude of the V_c term in Equation 9 by multiplying V_c with $1/K(t)$.¹⁰ Initially, $K(t)$ is small, but it then increases linearly until a given upper bound.

In other words, the formation of discontinuities is penalized initially,

encouraging a smooth interpolation everywhere except at very steep velocity gradients. Subsequently, by paying a smaller price for the formation of line processes, the optical flow will break at smaller velocity gradients. The final state of the network is independent of the speed at which $K(t)$ changes (adiabatic convergence). All other parameters remain constant and are identical for all simulations reported in this article.

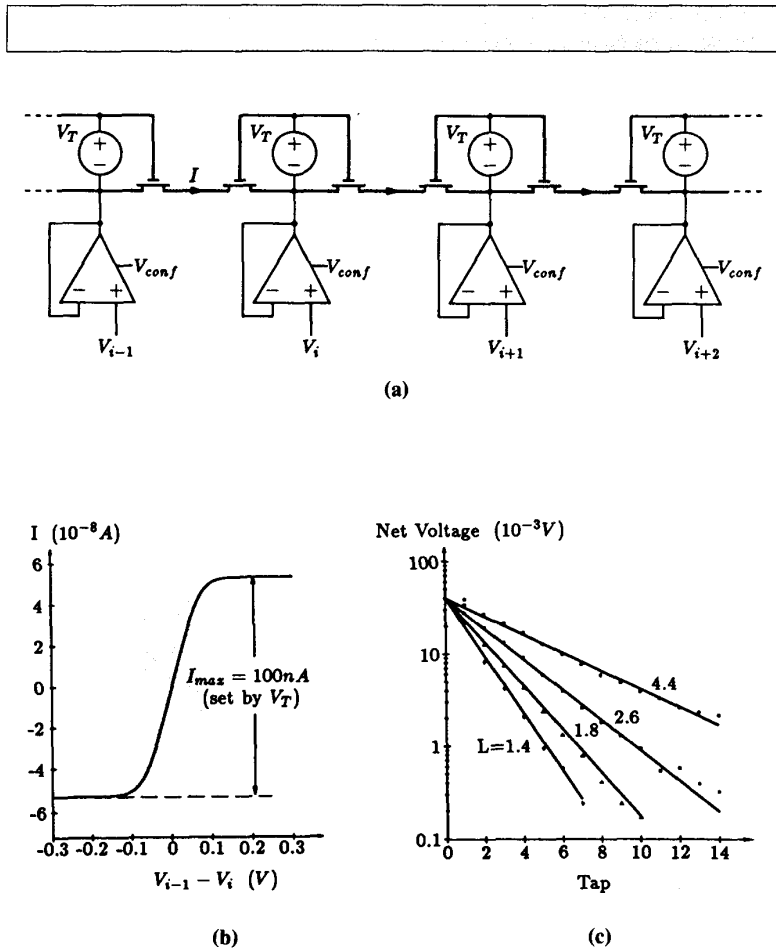


Figure 7. Basic resistance element in analog subthreshold CMOS technology. (a) Shows a one-dimensional cut through the resistive network of Figures 2 and 8. The active circuits—built out of nine transistors—within the shaded areas implement a variable nonlinear resistance. Each transconductance amplifier implements the conductance G (see Figure 8), whose value can be set by V_{conf} . The spatial response of such a network to a point voltage stimulus applied to the left-hand side is shown in (c). In an ideal tap line, the measured voltage values (points) should follow an exponential decay (lines). $V_T - V_{conf}$ sets the decay length L . The current-voltage characteristic of one such resistive element is illustrated in (b). The voltage V_T controls the maximum current and thus the slope of the resistance, which can vary between 100 k Ω and 10 G Ω . Many variations are possible.⁴

rule, the system seems to find next-to-optimal solutions (see Figures 4, 5, and 6) in about 10 to 15 analog-digital cycles. Furthermore, the algorithm must converge, because at each step the energy E is always reduced and E is bound from below. We compared statistical annealing with our deterministic method in the case of interpolating and smoothing sparsely

sampled data in the presence of discontinuity, where the underlying energy function is similar to E in Equation 9. Both methods converged to qualitatively similar solutions.¹⁰

The synthetic motion sequence in Figure 4 demonstrates the dramatic effect of the line processes. The optical flow outside the discontinuities approximately delineating

the boundaries of the moving squares is zero, as it should be (see Figure 4e). Where the two squares overlap, however, the velocity gradient is high and multiple intersecting discontinuities exist.

To restrict further the location of discontinuities, we adopt a technique used by Gamble and Poggio⁵ to locate depth discontinuities by requiring that depth discontinuities coincide with the location of intensity edges. In general, the physical processes and the geometry of the three-dimensional scene giving rise to the motion discontinuity will also give rise to an intensity edge. For example, moving physical objects occluding other objects will give rise to an image with edges at the occluding boundaries. In fact, only under laboratory conditions—for instance, using random dot patterns—does a motion discontinuity not coincide with intensity edges.

Figure 4f demonstrates that this strategy leads to the correct velocity field—with the exception of the corners—in addition to labeling all motion discontinuities. Figures 5 and 6 demonstrate our method on image pairs obtained with a video camera. See also the sidebars “Restricting motion discontinuities to edges” and “Varying the ‘amplitude’ of motion discontinuities.”

Analog VLSI networks

Even with the approximations and optimizations we previously described, the computations involved in this and similar early vision tasks require tens of minutes to hours on a large-scale parallel computer (see, however, Gamble and Poggio⁵). For the computations to be truly useful, we should be able to carry them out on a whole image in real time. Fortunately, modern integrated circuit technology gives us a medium in which we can realize extremely complex, analog real-time implementations of these computational metaphors.⁴

We can achieve a compact implementation of a resistive network using an ordinary CMOS process, provided the transistors run in the subthreshold range, where their characteristics are ideal for implementing low-current analog functions. We achieve the effect of a resistor by choosing the circuit configuration shown in Figure 7 rather than by using the resistance of a special layer in the process. We can control the value of the resulting resistance over five orders of magnitude by setting the bias voltages on the upper and

lower current source transistors. The current-voltage curve saturates above approximately 100 millivolts, a feature that we can use to advantage in many applications.

With small voltage gradients, we can treat the circuit as if it were a linear resistor, as shown by the shaded areas on the curves (Figure 7b). Conductances to signal input sources are implemented with transconductance amplifier followers, as shown in Figure 7a. Each amplifier injects a current into the network proportional to the difference between the local signal potential and the potential of the network. The effect of a conductance is thus achieved without drawing any current from the signal source. The value of the conductance is set by the transconductance control on the amplifier, which we can use to reflect the confidence assigned to the particular input. High conductance values give the network a short spatial-averaging scale, low values give a long averaging scale.

Figure 7c shows the spatial response of an experimental one-dimensional network to a point stimulus. We obtained the different values of averaging length L by appropriate settings of the amplifier transconductances. We can easily realize resistances with an effective negative resistance value.

Figure 8 shows the ideal configuration for a network implementation in two dimensions. Each point on the hexagonal grid is coupled to six equivalent neighbors. The high degree of symmetry of such an arrangement creates a nearly isotropic environment, free of many of the "preferred axis artifacts" introduced by an orthogonal grid. In addition, the larger connectivity allows a greater variation in effective resistor value caused by variations in transistor parameters.

Figure 9 shows a test chip implementing this network. Each node includes the resistor apparatus and a set of sample-and-hold circuits for setting the confidence and signal input voltages. In addition, an output amplifier enables measurement of the node voltage without disturbing the node itself. A scanning mechanism addresses both the sample-and-hold circuits and the output buffer, so the stored variables can be refreshed or updated, and the map of node voltages can be read out in real time.

A 48×48 silicon retina has been constructed that uses the hexagonal network of Figure 8 as a model for the horizontal cell layer in the vertebrate retina.¹³ In this application, the input potentials were the

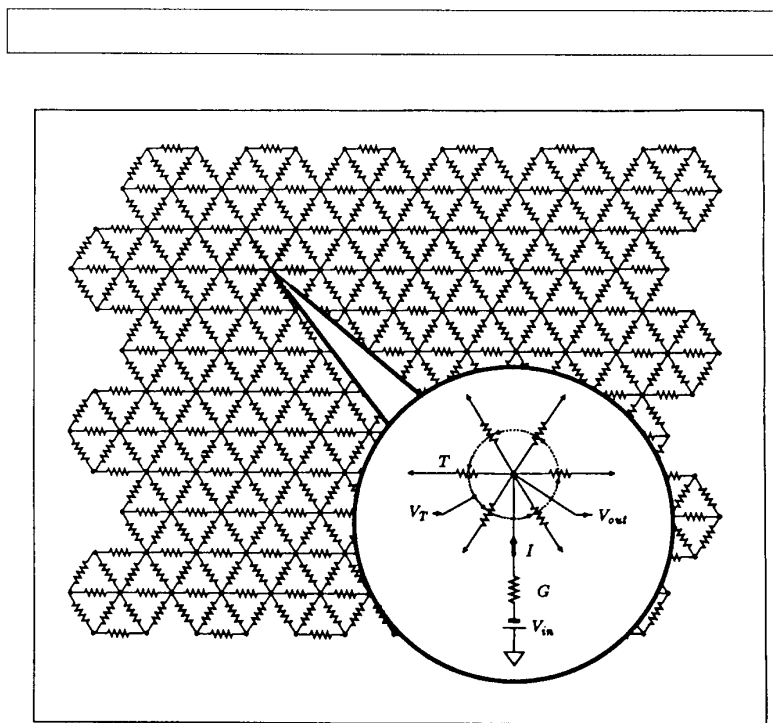


Figure 8. Circuit design for a resistive network for interpolating and smoothing noisy and sparsely sampled depth measurements.¹⁰ The basic version of this CMOS circuit contains 20×20 grid points on a hexagonal lattice. The individual resistive elements with a variable slope controlled by V_T , shown in Figure 7, correspond to the term governing the smoothness, λ . At those locations where a depth measurement d_{ij} is present, the battery is set to this value ($V_{in} = d_{ij}$) and the value of the conductance G is set to $1/(2\sigma^2)$, where σ^2 is the variance of the Gaussian noise process associated with the depth measurements. If no depth data are present at that node, G is set to zero. The voltage at each node corresponds to the discrete values of the smoothed surface fitted through the noisy and sparse measurements.^{8,10} The network for computing smooth optical flow minimizing E in Equation 4 via the network shown in Figure 3b is similar to this circuit.

outputs of logarithmic photoreceptors—implemented via phototransistors—and the potential difference across the conductance T formed an excellent approximation to the Laplacian operator.¹² This model results in the classical center-surround receptive field properties observed in the response of retinal ganglion cells. The circuit performs in real time.

We have demonstrated that the introduction of binary motion discontinuities into Horn and Schunck's² algorithm leads to a much

improved performance of their method, particularly for the optical flow in the presence of a number of moving objects. Moreover, we have shown that the appropriate computations map onto simple resistive networks.

We are now implementing these resistive networks in VLSI circuits, using sub-threshold CMOS technology. Many problems in early vision can be formulated in terms of similar nonconvex energy functions that need to be minimized, such as binocular stereo, edge detection, surface interpolation, and structure from motion.^{3,8,11} A similar approach to early

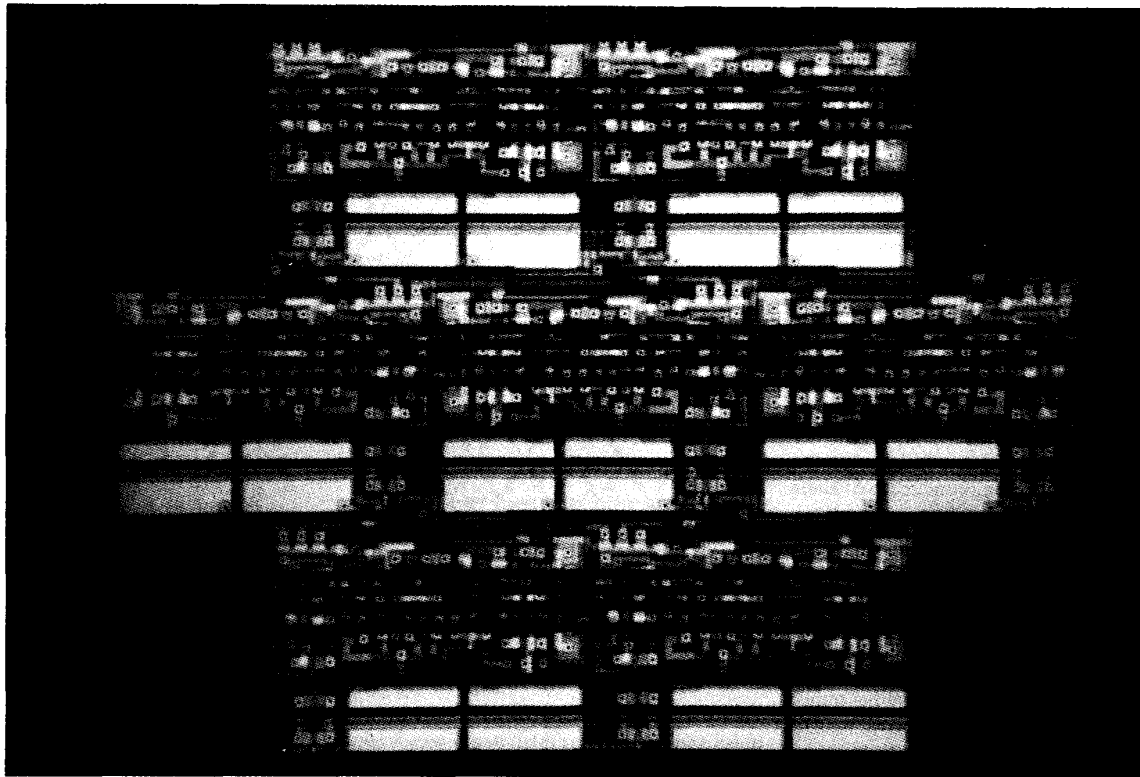


Figure 9. Partial layout of the CMOS chip implementing the resistive network shown in Figure 8, which interpolates sparsely sampled noisy data. Only seven cells (out of the 20×20 cell array) are shown in order to demonstrate the hexagonal grid. Each cell is dominated by the two capacities (approximately two picofarads each) for holding the depth data and its associated confidence value, consists of 46 transistors, and measures $180 \times 132 \lambda^2$. For a $\lambda = 1.5 \mu\text{m}$ production run, the total chip measures $5.4 \times 2.6 \text{ mm}$. If image acquisition-devices (phototransistors) are placed on the chip, the sample-and-hold circuitry can be eliminated, substantially reducing the area of the elementary cell.¹³

Fault tolerance

Hutchinson and Koch¹⁵ demonstrated the robustness of these resistive networks to component errors. In their circuit simulations of the resistive network for interpolating surfaces from noisy and sparsely sampled data (shown in Figures 2b and 8), they replaced each transversal conductance T by $T(1 + N(\sigma^2))$, where N is a zero-mean Gaussian probability distribution with variance σ^2 .

Due to the linearity of the network connections, errors "average out" and performance is only marginally

impaired even for widely varying conductances ($\sigma^2 = 0.5$). If nodes are pulled accidentally to ground, the line processes in their immediate neighborhood turn on because of the high voltage gradient, isolating these nodes and preventing error propagation. The saturation characteristic of our resistive elements outside their linear range (see Figure 7b) serves to prevent high current flows, because high voltage gradients between neighboring nodes induce only a constant maximal current.⁴

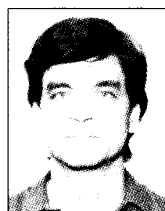
vision—using the fine-grained, mesh-type, single-instruction, multiple-data parallel Connection Machine instead of resistive networks—is being pioneered at MIT's Artificial Intelligence Laboratory in the Vision Machine project.^{5,14}

These networks share several features with biological neuronal networks. Specifically, they do not require a system-wide clock, they rely on many connections between simple computational nodes, they converge rapidly (within several time constants), and they are quite robust to hardware errors. (See the sidebar "Fault tolerance.")

Our networks consume moderate amounts of power, because each resistive

element operates in the millivolt and 10 nanoampere range. The entire retina chip¹³ requires about 100 μ W (the dominant power consumption lies in the photoconversion stage).

These features—real-time performance, low power consumption, robustness, and small spatial dimensions—make these circuits attractive for a variety of deep space missions. In collaboration with the Jet Propulsion Laboratory, we are currently evaluating the feasibility of such resistive network-based vision systems for autonomous vehicles to be used in the exploration of planetary surfaces, such as that of Mars. □



James Hutchinson has been a member of the technical staff at Thinking Machines Corp. since December 1987. He is currently implementing a document retrieval system for the Connection Machine. Previously, while at the Jet Propulsion Laboratory in Pasadena, he investigated artificial neural network applications on the Caltech/JPL Hypercube concurrent computer. His current interests include parallel computation, intelligent systems, and machine vision.

Hutchinson received the MS and BS in computer science from MIT in 1986.



Jin Luo is a staff research assistant in both computer science and biology at the California Institute of Technology. He studied analog VLSI under Carver Mead and received an MS in computer science from Caltech in 1987.

A citizen of the People's Republic of China, Luo received his BS in computer engineering from the Beijing Institute of Technology in 1983.

Acknowledgments

An early version of this model was developed and implemented in collaboration with A.L. Yuille (1987). Mathew Avalos and Andrew Hsu wrote the code for the Imaging Technology system, and Erik Staats for the Ncube. Koch is supported by an Office of Naval Research Young Investigator Award, grants from the NSF Advanced Engineering Program (EET-8714710 and IST-8700064), a grant from the Dept. of Energy (DE-FG03-85 ER25009), and the James Lee Powell Foundation. Mead is also supported by the Office of Naval Research and the System Development Foundation. A portion of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by NASA.



Christof Koch is an assistant professor of computation and neural systems at the California Institute of Technology. He is studying early vision algorithms, such as stereo, motion, and color, in both the mammalian visual system and in simple resistive networks for machine vision. His other major interest is the biophysics of computation—understanding the biophysical and biochemical mechanisms underlying information processing in nerve cells.

Koch studied physics and philosophy at the University of Tübingen in West Germany, receiving a PhD in 1982 at the Max Planck Institute for Biological Cybernetics.



Carver Mead is the Gordon and Betty Moore Professor of Computer Science at the California Institute of Technology, where he has taught for more than 20 years. A pioneer in the fields of solid-state electronics and the management of complexity in the design of VLSI circuits, he has been heavily involved in the development of a design methodology for VLSI. His recent work concerns modeling neuronal structures, such as the retina or cochlea, using analog VLSI systems.

Mead is a fellow of the American Physical Society, a member of the National Academy of Engineering, and a foreign member of the Royal Swedish Academy of Engineering Science, and has been elected to life membership in the Franklin Institute. He has also received a number of awards.

Readers may write to Koch at Caltech, Division of Biology 216-76, Pasadena, CA 91125.

References

1. E.C. Hildreth, *The Measurement of Visual Motion*, MIT Press, Cambridge, Mass., 1984.
2. B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, 1981, pp. 185-203.
3. T. Poggio, V. Torre, and C. Koch, "Computational Vision and Regularization Theory," *Nature*, Vol. 317, 1985, pp. 314-319.
4. C. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley Publishing Co., Reading, Mass., to appear in the summer of 1988.
5. E. Gamble and T. Poggio, "Integration of Intensity Edges with Stereo and Motion," *Artificial Intelligence Lab. Memo No. 970*, MIT, Cambridge, Mass., 1987.
6. H.H. Nagel, "On the Estimation of Optical Flow: Relations Between Different Approaches and Some New Results," *Artificial Intelligence*, Vol. 33, 1987, pp. 299-324.
7. S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 6, 1984, pp. 721-741.
8. J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic Solution of Ill-Posed Problems in Computational Vision," *J. Am. Statistics Assoc.*, Vol. 82, 1987, pp. 76-89.
9. A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, Mass., 1987.
10. C. Koch, J. Marroquin, and A. Yuille, "Analog 'Neural' Networks in Early Vision," *Proc. Natl. Academy of Sciences USA* 83, 1986, pp. 4263-4267.
11. A.L. Yuille, "Energy Functions for Early Vision and Analog Networks," *Artificial Intelligence Lab. Memo No. 987*, MIT, Cambridge, Mass., 1987.
12. D. Marr and E.C. Hildreth, "Theory of Edge Detection," *Proc. Royal Society of London*, Vol. B 207, 1980, pp. 187-217.
13. M.A. Sivilotti, M.A. Mahowald, and C.A. Mead, "Real-Time Visual Computations Using Analog CMOS Processing Arrays," in *1987 Stanford VLSI Conference*, P. Losleben, ed., 1987, pp. 295-312.
14. T. Poggio and the staff, "MIT Progress in Understanding Images," in *Image Understanding Workshop*, Vol. 1, DARPA, 1987, pp. 1-14.
15. J.M. Hutchinson and C. Koch, "Simple Analog and Hybrid Networks for Surface Interpolation," in *Neural Networks for Computing*, J.S. Denker, ed., American Institute of Physics, New York, 1986, pp. 235-239.