Proc. Natl. Acad. Sci. USA
Vol. 83, pp. 4263–4267, June 1986
Biophysics

Analog "neuronal" networks in early vision

(surface reconstruction/motion segmentation/analog hardware/parallel computing)

CHRISTOF KOCH*†, JOSE MARROQUIN†‡, AND ALAN YUILLE†

*Center for Biological Information Processing, E25-201, and †Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139

Communicated by John J. Hopfield, January 27, 1986

ABSTRACT **Many problems in early vision can be formulated in terms of minimizing a cost function. Examples are shape from shading, edge detection, motion analysis, structure from motion, and surface interpolation. As shown by Poggio and Koch [Poggio, T. & Koch, C. (1985) *Proc. R. Soc. London, Ser. B* 226, 303–323], quadratic variational problems, an important subset of early vision tasks, can be "solved" by linear, analog electrical, or chemical networks. However, in the presence of discontinuities, the cost function is nonquadratic, raising the question of designing efficient algorithms for computing the optimal solution. Recently, Hopfield and Tank [Hopfield, J. J. & Tank, D. W. (1985) *Biol. Cybern.* 52, 141–152] have shown that networks of nonlinear analog "neurons" can be effective in computing the solution of optimization problems. We show how these networks can be generalized to solve the nonconvex energy functionals of early vision. We illustrate this approach by implementing a specific analog network, solving the problem of reconstructing a smooth surface from sparse data while preserving its discontinuities. These results suggest a novel computational strategy for solving early vision problems in both biological and real-time artificial vision systems.**

This study addresses the use of simple analog networks to implement and solve problems in early vision, such as computing depth from two stereoscopic images, reconstructing and smoothing images from sparsely sampled data, and computing motion. Within the last years, computational studies have provided promising theories of the computations necessary for early vision (for partial reviews, see refs. 1–5). A number of early vision tasks can be described within the framework of standard regularization theory (5). Standard regularization analysis can be used to solve these problems in terms of quadratic energy functionals that must be minimized. Previous work by Poggio and Koch (6) showed how to design linear, analog networks for solving regularization problems with quadratic energy functions. The domain of applicability of standard regularization theory is limited, however, by the convexity of the energy functions, which makes it impossible to deal with problems involving true discontinuities. Such problems can be described by nonconvex energy functions involving binary line processes (7–10). More recently Marroquin (11) has proposed an approach to early vision based on the use of Markov random-field models and Bayes estimation theory (11, 33). We will show how these algorithms map naturally onto very simple resistive networks.

There has been considerable interest in the computational properties and capabilities of networks of simple, neuronal-like elements (12–15). Recently, Hopfield and Tank (16) have shown that analog neuronal networks can provide fast, next-to-optimal solutions to a well-characterized but difficult optimization problem, the "traveling salesman problem." In this paper we show that networks of simple, analog, or hybrid processing elements can be used to give fast solutions to a number of early vision problems.

SMOOTH SURFACE RECONSTRUCTION

Surface reconstruction is a typical problem of early vision that can be formulated in terms of minimizing a quadratic energy function (17, 18). It occurs in several situations. For example, if a stereo algorithm computes depth values only at specific locations in the image, for instance along edges, the surface must be interpolated between these points. Another instance occurs when the data is given everywhere but is noisy and needs to be smoothed. Grimson (17) studied surface interpolation in the context of stereo matching. He considered a stereo algorithm (19) in which isolated primitive features—zero crossings corresponding to significant events in the images—were matched, yielding a depth value at the feature points. He then proposed an interpolation scheme to obtain depth values thoughout the image that corresponds to fitting a thin flexible plate through the observed data points. Both Grimson's (17) and Terzopoulos's (18) interpolation schemes can be described in terms of standard regularization theory (5): the energy or cost function $E(f)$ to be minimized—derived from the inverse problem $Bf = d + n$, where the data d and the linear operator B are known, n represents some noise process, and f is to be computed—is given by

$$E(f) = \|Bf - d\|^2 + \alpha\|Sf\|^2, \qquad [1]$$

where the first term gives a measure of the distance of the solution to the data and the second term corresponds to the regularizer needed to make the problem well-posed (α is a "regularization" parameter and S a linear operator). For surface interpolation, B is a diagonal matrix with elements equal to 1 at those locations where the depth is known and 0 at all others. The stabilizer S corresponds to the operator associated with a membrane or thin plate, depending on the kind of smoothing desired. E can be reformulated as

$$L(V) = \frac{1}{2}\sum_{i,j} T_{i,j}V_iV_j + \sum_i V_iI_i, \qquad [2]$$

by identifying the matrix T with $2(B^TB + \alpha S^TS)$, V with f, I_i with $-2B^Td$ and dropping the constant term d^Td. In an electric circuit implementation, V_i corresponds to the voltage (relative to ground) at the processing element i, henceforth termed "neuron i," I_i to a current injected into neuron i and $T_{i,j}$ to the conductance of the connection between neuron i and j. If no connection exists between i and j, $T_{i,j}$ is set to 0. Finally, every neuron is "grounded" by a resistance R_i (with $T_{i,i} = \Sigma_j T_{i,j} + 1/R_i$). L now corresponds to the total power dissipated by the circuit as heat (6). We can also interpret this

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

‡Present address: Gutierrez Zamora 260, Las Aguilas, Mexico, D.F. Mexico.

expression as the Lyapunov function of the network. To introduce dynamics into this network, we associate with every neuron a capacity C_i in parallel with R_i. The equation describing the change in the potential is then given by:

$$C_i \frac{dV_i}{dt} = -\frac{\partial L}{\partial V_i}. \quad [3]$$

For the case of quadratic regularization principles, $L(V)$ is positive definite quadratic, and therefore the system will always converge to the unique energy minimum. In other words, every quadratic variational principle of the type shown in Eq. 1 can be solved with an appropriate electrical network, where the connections can be implemented by linear Ohmic resistances and the data is given by injecting currents.

LINE PROCESSES

However, quadratic variational principles have limitations. The main problem is the degree of smoothness required for the unknown function that is to be recovered. For instance, the surface interpolation scheme outlined above smoothes over edges and thus fails to detect discontinuities (Fig. 1).

Marroquin (9) has proposed a scheme to overcome this difficulty (see also refs. 7, 10, and 18). Following ref. 8, he used a probabilistic formulation of the surface reconstruction problem; the behavior of a piecewise smooth surface is modeled by using two coupled Markov random fields (20): a continuous-valued one that corresponds to the depth f_i at location i and a binary one, whose variables are located at sites between the depth lattice (see Fig. 2*A*). The function of this unobservable "line process" is to indicate the presence or absence of a discontinuity between two adjacent depth elements. By using Bayes theory, it is found that the maximum *a posteriori* estimate of the surface corresponds to the global minimization of an "energy" function. In one dimension this function is given by

$$E(f, h) = \sum_i (f_{i+1} - f_i)^2(1 - h_i) + c_D \sum_i (f_i - d_i)^2 + c_L \sum_i h_i. \quad [4]$$

Here the h_i corresponds to the line process. If the gradient of f becomes too large [i.e., $(f_{i+1} - f_i)^2 > c_L$, where c_L is some fixed parameter], it becomes cheaper to break the surface and put in a line—paying the "price" c_L—rather than to interpolate smoothly. The line process h_i introduces local minima into the energy function, making the problem nonquadratic. The term $(f_i - d_i)^2$, describing the difference between the measured data d_i and the approximated surface value f_i, is weighted by c_D, which depends on the signal-to-noise ratio. If d_i is very reliable, then $c_D >> 1$. For two-dimensional images more terms are required in Eq. 4. In ref. 9, the minimization of the optimization problem was carried out by using simulated annealing.

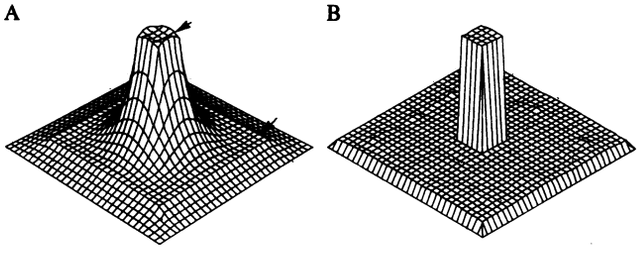

FIG. 1. Smooth-surface and piecewise-smooth-surface reconstruction from noisy and sparse data. (*A*) Three-dimensional representation of a reconstructed and smoothed surface from sparse observations using a quadratic energy expression corresponding to interpolating with a membrane. The depth is sampled randomly every second point along the ridge in the main plane of the figure and along the rim of the "tower" (see arrows). The observations are assumed to be corrupted by a Gaussian noise ($\sigma = 0.25$). The state of the system is shown after five time constants of the linear depth network. (*B*) Piecewise-smooth-surface reconstruction from the same data set after one time constant of the line process network τ. Both surfaces were computed by using an analog network with constant coupling.

Here we sketch another method based on refs. 12 and 16. Hopfield's idea was to solve combinatorial optimization problems by allowing the binary variable to vary continuously between 0 and 1 and to introduce terms in the energy function that forced the final solution to one of the corners of the hypercube $[0, 1]^N$. Briefly, let the output variable V_i for neuron i have the range $0 < V_i < 1$ and be a continuous and monotonic increasing function of the internal state variable U_i of the neuron i: $V_i = g_i(U_i)$. A typical choice is $g(U_i) = (1 + e^{-2\lambda U_i})^{-1}$. Like before, the "strength" of the connection between i and j is given by the matrix element $T_{i,j}$, and each neuron has an associated capacitance C_i and resistance R_i (with $\tau = R_iC_i$). The resulting charging equation that determines the rate of change of U_i is

$$C_i \frac{dU_i}{dt} = \sum_j T_{i,j}V_j - \frac{U_i}{R_i} + I_i, \quad [5]$$

where I_i can be considered as fixed input to neuron i. Hopfield introduces the quantity

$$E = -\frac{1}{2}\sum_{i,j} T_{i,j}V_iV_j + \sum_i \frac{1}{R_i}\int_0^{V_i} g_i^{-1}(V)dV - \sum_i I_iV_i \quad [6]$$

and shows that E is a Lyapunov function of the system, as long as $T_{i,j}$ is symmetric (12). In other words, by using the update of Eq. 5 $[C_i(dU_i/dt) = -(\partial E/\partial V_i)]$, the time evolution of the system is a motion in state space that seeks out minima in E and comes to a stop at such points. The relation between the stable states of the continuous model and those of the binary ones, in which the output of every neuron can be either 0 or 1, is governed by λ. For $\lambda \to \infty$, g_i tends to either 0 or 1.

Following ref. 16, we shall map the binary line processes $h_{i,j}$ and $v_{i,j}$ into continuous variables bounded by 0 and 1. One possibility for choosing an associated cost function is outlined below. This function has four contributors: the term implementing a membrane type of surface interpolation E_I together with the data term E_D, the line potential term E_L, and the gain term E_G:

$$E_I + E_D = \sum_{i,j}(f_{i,j+1} - f_{i,j})^2(1 - h_{i,j}) + c_D\sum_{i,j}(f_{i,j} - d_{i,j})^2. \quad [7a]$$

$$E_L = c_V\sum_{i,j} h_{i,j}(1 - h_{i,j}) + c_P\sum_{i,j} h_{i,j}h_{i,j+1} + c_C\sum_{i,j} h_{i,j} + c_L\sum_{i,j} h_{i,j}\cdot[(1 - h_{i+1,j} - v_{i,j} - v_{i,j+1})^2 + (1 - h_{i-1,j} - v_{i-1,j} - v_{i-1,j+1})^2] \quad [7b]$$

$$E_G = c_G\sum_{i,j}\int_0^{h_{i,j}} g_{i,j}^{-1}(h_{i,j})dh_{i,j}. \quad [7c]$$

Here $f_{i,j}$, $v_{i,j}$, and $h_{i,j}$ correspond to the depth, the vertical line process, and the horizontal line process, respectively.¶ Note that the summation in E_D only includes nodes where measurements are available. The first term in E_L forces the line process to the corners of the hypercube—i.e., to either 0 or 1. The second term penalizes the formation of adjacent

¶We have only derived the energy expressions for $h_{i,j}$. The expression for $v_{i,j}$ can be obtained by replacing $h_{i,j}$ with $v_{j,i}$ and i with j and vice versa.

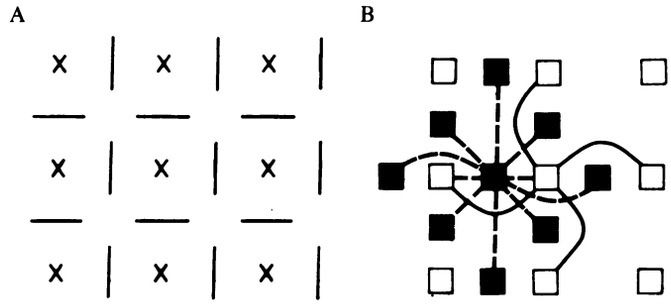

FIG. 2. (*A*) The two-dimensional lattice of line processes (lines) and depth points (crosses). Each depth value is enclosed on all four sides by a line process. (*B*) The local connections between neighboring line processes (filled squares) and the depth lattice (open squares).

parallel lines, while the third term represents the cost that need be paid for the introduction of every single line. The fourth term is an interaction term that favors continuous lines and penalizes both multiple-line intersections and discontinuous-line segments. The gain term forces the line process inside the hypercube $[0,1]^N$. Fig. 2*B* illustrates the connections required to implement this energy function within the line and the depth lattices. Following ref. 12, we choose the following update rule:

$$\frac{df_{i,j}}{dt} = -\frac{\partial E}{\partial f_{i,j}} \quad \text{and} \quad \frac{dm_{i,j}}{dt} = -\frac{\partial E}{\partial h_{i,j}}, \qquad [8]$$

where $m_{i,j}$ is the internal state variable for the processing elements corresponding to the horizontal line process—that is, $h_{i,j} = g(m_{i,j})$. It is easy to see that for this update, the total energy will always decrease. The system will evolve in such a manner as to find a minimum of E. Note that, different from refs. 12 and 16, our energy function contains cubic terms (e.g., $f_{i,j+1} \cdot f_{i,j} \cdot h_{i,j}$).

SIMULATION RESULTS AND HEURISTICS

An analog network for a 32 by 32 image was simulated on a digital computer. For the simulation, two sets of parameters are important. First is the weight of the term describing the interaction among the line processes, E_L, versus the weight of the smoothing term (equal to 1). This ratio determines the limiting depth gradient beyond which no more interpolation takes place. Decreasing the importance of the line interaction term E_L versus the smoothing term encourages the formation of lines at smaller and smaller depth gradients. This number requires some rough estimate of the limiting depth gradient for which no smooth interpolation should occur. The second parameter is the relative weight of the individual components of E_L and E_G. Fortunately, the choice of these parameters does not seem to vary from image to image. Results in this paper refer to $c_V = 0.5$, $c_L = 4.0$, $c_P = 5.0$, $c_C = 1.0$, and $c_G = 0.5$. As boundary conditions we set $h_{i,j}$ and $v_{i,j}$ to 1 along the boundaries of the square. As initial conditions we set the internal state variable of the line process neurons, $m_{i,j}$, to 0 and all horizontal and vertical lines in the image to 0.5.

The final state of the network should approximate as closely as possible the state of lowest energy. Since f, h, and v are independent variables, the solution can be found by minimizing $E(f, h, v)$ for a given arrangement of the line processes by varying f [since $E(f)$ is a quadratic function for fixed h and v]. These considerations lead us to adopt the following strategy. After the depth lattice is "initialized" with the sparsely sampled depth data, the network computes the smoothest surface assuming all line processes set to 0. Subsequently, the depth network is updated 10 times for every single update of the line process network. Functionally, this is equivalent to assuming that the line process network is stationary or substantially slower than the depth network. However, independent of the relative speed of these two processes, $E(f, h, v)$ is always a Lyapunov function. Fig. 1 illustrates the difference between smooth-surface and piecewise-smooth-surface interpolation (for equivalent results, see also ref. 11).

How does the choice of λ affect our results? For large values of λ—that is, in the high-gain limit—the line processes will almost always be either 0 or 1, while in the low-gain limit, values of $h_{i,j}$ and $v_{i,j}$ will be distributed around 0.5, and the energy function of Eq. **7** will be almost quadratic. Experimentally, we found satisfactory convergence for $\lambda > 5$ (in Figs. 1, 3, and 4, $\lambda = 16$). Choosing different values of λ clustered around this value affected mainly the convergence time—since logical "decisions" (i.e., 0 and 1) are computed faster for higher λs—but not the final solution to any significant extent. Lowering λ below 0.1 prevented the formation of lines altogether—at least within reasonable times.

Figs. 3 and 4 show more complicated synthetic images where the depth is sampled randomly at, on the average, every third point and the measurement process is corrupted by Gaussian noise. This condition represents the most stringent performance test of our algorithm. Contrariwise, when the depth is assumed to be known at edges, as for instance in the Marr and Poggio stereo algorithm (19), the network converges rapidly to a good solution. However,

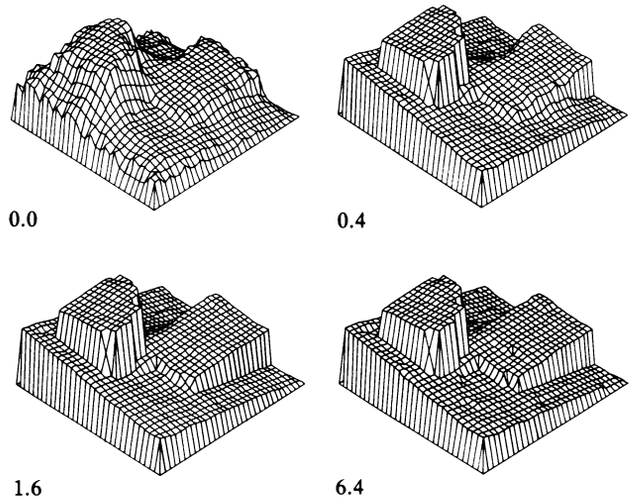

FIG. 3. Temporal evolution of the states of the network for a sparsely sampled synthetic scene containing three slanting rectangles. On the average, every third point is sampled. These measurements are corrupted by Gaussian noise ($\sigma = 0.25$). (*Upper Left*) Initial state of the network after smoothing. Subsequent illustrations show the changing states of the network, with time specified in terms of τ. Note, that in order to reconstruct such objects, there is a critical number of sampling points *per* object, below which the reconstruction yields ambiguous results. A variable coupling was assumed.

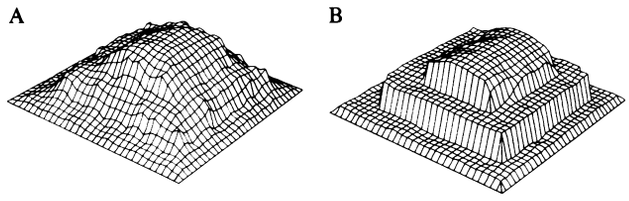

FIG. 4. Smooth-surface (*A*) and piecewise-smooth-surface (*B*) reconstruction of a sparsely sampled synthetic scene containing both flat and curved surfaces. Smoother surface interpolation can be obtained by the use of a higher-order stabilizer, such as the thin plate stabilizer (10). The network is shown after 1.6τ (for more details, see Fig. 3).

when the depth measurements were assumed to occur randomly throughout the image, we introduced the following *ad hoc* procedure to reconstruct the original, fully sampled, image as closely as possible. Most images contain, unlike Fig. 1, more than a single depth scale. One way to scan for different depth values is to change the weight of E_L during a simulation run. We multiplied E_L by a factor $1/K(t)$, where $K(t)$ starts out small—typically at 0.1—and increases linearly until a given saturation threshold. In other words, initially the formation of lines is strongly penalized, encouraging a smooth interpolation everywhere except at very steep disparity gradients. Subsequently, by paying a smaller price for the formation of lines, the surface will break at smaller depth gradients (see also refs. 10 and 18). The final state of the network is independent of the speed at which $K(t)$ changes, as long as it increases slowly enough (adiabatic approximation). A second method to achieve next-to-optimal solutions involves varying λ during a run (16). Both methods yield roughly similar results.

ANALOG NETWORKS FOR EARLY VISION

The results we have presented indicate the plausibility of using graded networks of simple neuron-like processing elements to "solve" constraint satisfaction problems in early vision that can be formulated as minimization of an energy function or as finding optimal Bayesian estimates. They include surface interpolation (9, 10, 17), edge detection (21), shape from shading (22), velocity field estimation (23, 24), color (34), and structure from motion (25, 26). The solutions we obtain using analog networks are similar to the solutions obtained using simulated annealing or other algorithms derived from estimation theory (9, 11).

Another problem that maps very naturally onto our networks is the recovery of the two-dimensional velocity field. In general, only one independent measurement of the change in image brightness is available from an image sequence, while the velocity field has two components (*aperture problem*). In order to recover the global velocity field, various constraints have

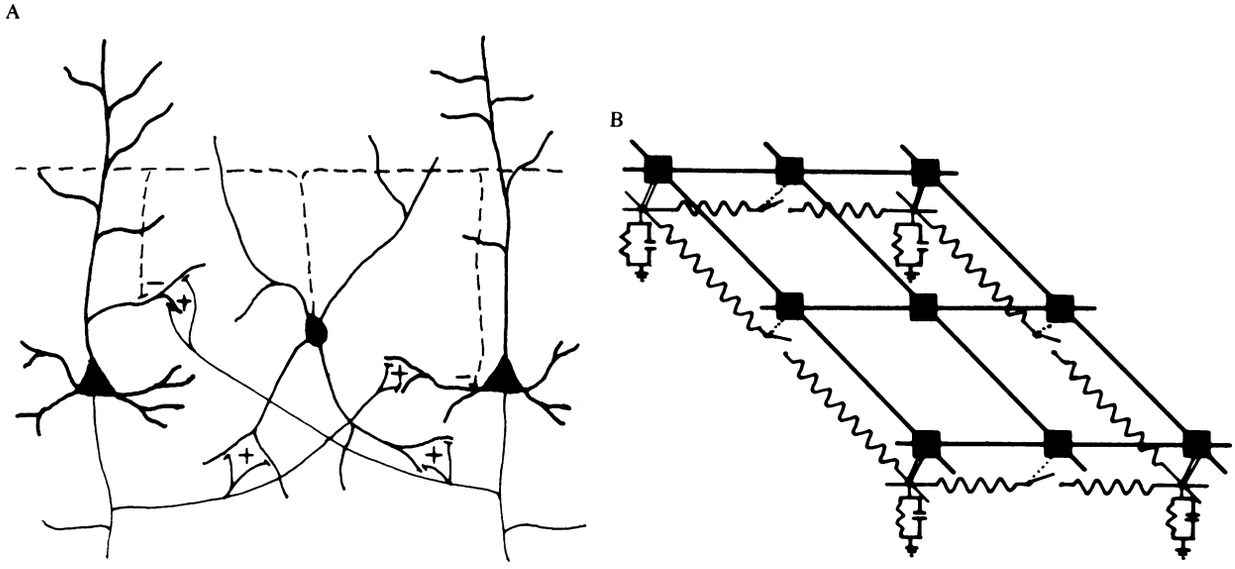

FIG. 5. (*A*) A neuronal implementation of the analog network for the piecewise-smooth-surface interpolation problem. The depth values are assumed to be represented in a two-dimensional network of pyramidal cells within the visual cortex. These pyramidal cells excite, via intracortical arborization that can extend over many hundreds of microns, neighboring pyramidal cells in a roughly symmetrical way. The measured depth at isolated points or edges is computed at an earlier stage and relayed to the pyramidal cells via excitatory synapses. The line processes are implemented via inhibitory GABAergic (GABA = γ-aminobutyric acid) interneurons, in this case basket cells. Their axon ascends from the cell body, usually giving off extensive horizontal collaterals at several levels. We assume that the synapse of the interneuron onto the dendrites of the pyramidal cells vetos locally the excitatory signal from a neighboring pyramidal cell, on the basis of the nonlinear and local interaction between excitatory and silent or shunting inhibitory synapses (see ref. 28). GABAergic interneurons also contact other GABAergic interneurons, thus providing the substratum for the interaction between line processes (see Fig. 2*B*). Basket cells receive input from neighboring pyramidal cells, computing locally the difference between the electrical activity in adjacent pyramidal cells. Activation (i.e., firing) of the basket cells signals the presence of a discontinuity in the visual scene. Information regarding the location of these edges obtained from different sources (for instance, by using the flow field) can be incorporated in a natural way into this network by exciting the appropriate interneuron. Since the pyramidal cells code for depth, their dynamic firing range should be large compared with the dynamic range of the interneurons. (For the anatomy of cortical cells, see ref. 29.) (*B*) A circuit implementation of the hybrid network for the piecewise-smooth-surface interpolation problem. The linear resistive network minimizes the quadratic energy function for a specific setting of resistances (i.e., line processes). The digital processors of a very simple "single instruction multiple data" array processor—shown here in a "connection machine"-like architecture (ref. 30; indicated as solid squares)—set or break the resistive connections in the analog network (with the help of some transistors) as a function of the current state of the analog network and of the previous state of the digital processors. Note that sampling the current depth value requires a analog-to-digital conversion and thus increases the convergence time. When an observation is present at site i,j a positive current of value $c_D d_{i,j}$, corresponding to the measured depth, is injected at this node, which must also be connected to ground via a conductance of value c_D. This scheme combines the speed of a simple analog network with the versatility of a digital processor and would permit real-time execution of vision algorithms.

been utilized, such as assuming that the reconstructed motion field is the smoothest field compatible with the known velocity components (23, 24). We propose that the use of line processes marking the boundary between moving objects will greatly ameliorate this problem. Thus, an image consisting of several overlapping figures moving within the plane superimposed on a stationary background will have a smooth velocity field everywhere but at the boundaries of the moving figures where the line processes will be "turned on."

The single most important advantage of analog networks is their speed. Typical convergence times are on the order of several time constants. The convergence time does not depend *per se* on the size of the image array but rather on the size of the largest patch of smooth surface without associated depth measurements in the image: it takes on the order of n^2 time constants for information to propagate across such a n-pixel-wide smooth patch. This behavior contrasts favorably with simulated annealing (27). The principal drawback of our method is that there is no guaranty that the network will converge to the state of lowest energy. We can only show experimentally that the computed solutions seem reasonable compared with solutions obtained with other algorithms. As has been pointed out (16), the main reason for this good behavior is the smoothing of the solution space upon transforming the problem from a discrete, binary space into a continuous one.

Fig. 5*A* shows a hypothetical implementation of our analog network within the visual cortex that is in accordance with known cortical anatomy and physiology. However, what are the implications of our results for the architecture of artificial vision systems? The feasibility of building analog very-large-scale integrated (VLSI) systems has been demonstrated by Tanner and Mead (see ref. 31). They have built a single cMOS (complementary metal-oxide-semiconductor)-motion detection chip that combines a grid of high-performance bipolar photosensors with purely analog components. On the basis of graded light variations, their chip extracts the velocity vector of a spatially uniform flow field.

Purely analog networks do have one major drawback with regard to conventional programmable processors. Once built, it is difficult to change their parameters—such as the specific form of the penalizing terms or the stabilizer. Thus, every task requires a dedicated analog network. A possible solution to this dilemma is a mixed *hybrid* architecture. The basic idea is to combine a regular grid of simple, serial programmable processors communicating locally with a linear resistive network. The digital processors, corresponding to the line elements, can break the resistive connection between two neighboring analog processors with the aid of a simple switch (Fig. 5*B*). Coupled binary and smoothly varying Markov random fields map naturally onto this type of architecture (8, 11, §). The hybrid machine has two basic cycles. In the analog cycle, the processors inject current—corresponding to the depth measurements—into the analog network. Subsequently, the resistive network finds the unique (6) smoothest surface, *given a certain distribution of lines*—mimicked by the breaking of resistances between nodes. In the digital cycle, the digital processor network will read out the current voltage at every node in the resistive network and compute a new estimate for the binary line process using conventionally programmed digital software. Stochastic optimization techniques (11, 27, 32, §) could easily be implemented here. Subsequently, the processors will set or break the appropriate connections in the analog network and the machine will switch back into the analog cycle.

We thank John Hopfield, Francis Crick, Tom Knight, Carver Mead, Tomaso Poggio, Eric Saund, and Demitri Terzopoulos for many useful discussions and suggestions. The Center's support is provided in part by the Sloan Foundation and in part by Whitaker College. Support for the Artificial Intelligence Laboratory's research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research Contract N00014-80-C-0505. J.M. is supported by the Army Research Office under contract ARO-DAAG29-84-K-0005. C.K. is supported by a grant from the Office of Naval Research, Engineering Psychology Division to Tomaso Poggio.

1. Barrow, H. G. & Tenenbaum, J. M. (1978) in *Computer Vision Systems*, eds. Hanson, A. & Riseman, E. (Academic, New York), pp. 3–26.
2. Brady, J. M. (1982) *Comput. Surv.* **14,** 3–71.
3. Ballard, D. H., Hinton, G. E. & Sejnowski, T. J. (1983) *Nature (London)* **306,** 21–26.
4. Marr, D. (1982) *Vision* (Freeman, San Francisco).
5. Poggio, T., Torre, V. & Koch, C. (1985) *Nature (London)* **317,** 314–319.
6. Poggio, T. & Koch, C. (1985) *Proc. R. Soc. London, Ser. B* **226,** 303–323.
7. Blake, A. (1983) *Pattern Recognition Lett.* **1,** 393–399.
8. Geman, S. & Geman, D. (1984) *IEEE Trans. Pattern Analysis & Machine Intelligence* **6,** 721–741.
9. Marroquin, J. (1984) *Artificial Intelligence Laboratory Memo*, (Massachusetts Institute of Technology, Cambridge, MA), No. 792.
10. Terzopoulos, D. (1986) *IEEE Trans. Pattern Analysis & Machine Intelligence* **8,** 129–139.
11. Marroquin, J. (1985) *Artificial Intelligence Laboratory Memo* (Massachusetts Institute of Technology, Cambridge, MA), No. 839.
12. Hopfield, J. J. (1984) *Proc. Natl. Acad. Sci. USA* **81,** 3088–3092.
13. Kohonen, T. (1977) *Associative Memory* (Springer, Heidelberg).
14. Palm, G. (1982) *Neural Assemblies* (Springer, Heidelberg).
15. Ullman, S. (1979) *Comput. Graph. Image Proc.* **10,** 95–105.
16. Hopfield, J. J. & Tank, D. W. (1985) *Biol. Cybern.* **52,** 141–152.
17. Grimson, W. E. L. (1981) *From Images to Surfaces: A Computational Study of the Human Early Visual System* (MIT Press, Cambridge, MA).
18. Terzopoulos, D. (1983) *Comp. Graph. Image Proc.* **24,** 52–96.
19. Marr, D. & Poggio, T. (1979) *Proc. R. Soc. London, Ser. B* **204,** 301–328.
20. Wong, E. (1968) *SIAM J. Appl. Math.* **16,** 4–14.
21. Poggio, T., Voorhees, H. & Yuille, A. (1985) *Artificial Intelligence Laboratory Memo* (Massachusetts Institute of Technology, Cambridge, MA), No. 776.
22. Horn, B. K. P. & Brooks, M. (1985) *Artificial Intelligence Laboratory Memo* (Massachusetts Institute of Technology, Cambridge, MA), No. 815.
23. Hildreth, E. C. (1984) *The Measurement of Visual Motion* (MIT Press, Cambridge, MA).
24. Horn, B. K. P. & Schunck, B. G. (1981) *Artif. Intell.* **17,** 185–203.
25. Grzywacs, N. & Yuille, A. (1986) *Artificial Intelligence Laboratory Memo* (Massachusetts Institute of Technology, Cambridge, MA), No. 888.
26. Ullman, S. (1979) *The Interpretation of Structure from Motion* (MIT Press, Cambridge, MA).
27. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983) *Science* **220,** 671–680.
28. Koch, C., Poggio, T. & Torre, V. (1982) *Philos. Trans. R. Soc. London, Ser. B* **298,** 227–264.
29. Martin, K. A. C. (1985) in *Cerebral Cortex*, eds. Jones, E. G. & Peters, A. (Plenum, New York), Vol. 2, pp. 241–284.
30. Hillis, W. D. (1985) *The Connection Machine* (MIT Press, Cambridge, MA).
31. Tanner, J. (1986) Dissertation (California Institute of Technology, Pasadena, CA).
32. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. (1953) *J. Chem. Phys.* **21,** 1087–1092.
33. Marroquin, J., Mitter, S. & Poggio, T. (1985) in *Proceedings of the Image Understanding Workshop*, ed. Baumann, L. S. (Science Applications International Corp., Washington, DC), pp. 293–309.
34. Poggio, T. (1985) in *Proceedings of the Image Understanding Workshop*, ed. Baumann, L. S. (Science Applications International Corp., Washington, DC), pp. 25–39.