Analog Models for Early Vision

Thesis by

John G. Harris

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1991

(Defended May 30, 1991)

Acknowledgments

- Thanks to Christof for many things, especially for his rapid-fire ideas and the constant enthusiasm he exudes. Thanks also to the rest of Koch Lab for help and comfort over the years. Research in the Koch lab is funded by a National Science Foundation grant IST-8700064, DDF-II funds from the Jet Propulsion Laboratory at the California Institute of Technology, Office of Naval Research Young Investigator and National Science Foundation Presidential Young Investigator Awards, the James S. McDonnell Foundation, Rockwell International Science Center and the Hughes Aircraft Artificial Intelligence Center. Thanks also to the Hughes AI Center for supporting me with a fellowship during my years at Caltech.

- Of course, none of this would have been possible without Carver for laying the framework upon which this research has been built. He serves as a model for us all to follow. Thanks also to everyone in Carver's lab for their help during my years at Caltech. Hewlett-Packard provided computing support. All chips were fabricated through MOSIS with DARPA's support.

- My thesis committee (Christof, Carver, Al Barr, David Van Essen, and Rod Goodman) provided valuable criticism. Candi Hochenedel and Andy Moore graciously read earlier drafts of this thesis. Jin Luo helped with the fuse chip and especially for built the hardware for scanning data on and off the chip. Thanks to Massimo Sivilotti for writing many of the Latex macros which were used in formatting this thesis. Mass

also wrote WoL, the VLSI layout editor that was used to design all the chips in this thesis. Thanks to Shih-Chii Liu, Bimal Mathur, and John Wyatt for many collaborations during this work. John Wyatt pointed out the theorems dealing with stability and uniqueness of nonlinear resistive networks.

- Thanks to my mother and father for their constant love and encouragement. They have taught me and my brothers the value of education.

- Finally, I thank Anita for bringing so much happiness to my life.

Abstract

Analog models provide a novel framework for understanding and developing algorithms for computer vision. This thesis introduces several extensions to well-known resistive network techniques for solving early vision problems. First, *constraint boxes* are developed as a general methodology for mapping regularization-based algorithms onto stable analog hardware. These multiterminal resistor systems solve low-level vision problems by minimizing a global Lyapunov energy. Second, a circuit element called the *resistive fuse* is introduced to extend these networks for discontinuity detection. This is the first hardware circuit that explicitly implements line-process discontinuities. Since resistive fuse networks must minimize a non-convex energy function that may contain local minima, complex annealing or continuation methods are necessary for adequate solutions of the problem. Third, the *tiny-tanh* network is proposed as a new mechanism for discontinuity detection that is not plagued by problems with local minima. A piece-wise constant segmentation is performed through minimization of a convex Lyapunov energy.

Contents

Chapter 1

Introduction

This thesis studies analog models for early vision. These analog models are an active research topic for two reasons. First, we want to build artificial vision systems for use in robotics, factory automation, parts inspection, and other industrial applications. These analog models are useful because they lead toward fast, low-power analog VLSI hardware. Second, we study these models in order to understand the information processing tasks necessary for artificial and biological vision systems.

Analog models provide a novel framework for understanding and developing algorithms for computer vision. This thesis describes examples of analog networks for several early vision problems such as image and depth segmentation. Many real-world applications become feasible because of the speed, size, cost, and power consumption of these networks. Besides these advantages, however, experimentation with continuous-time nonlinear circuits has led to some fundamental insights relevant to computer vision. Powerful analog algorithms thus developed may prove useful even if a computer vision researcher is limited to simulating the analog hardware on a digital computer.

The outline of this thesis is as follows:

- Chapter 1 introduces regularization theory and resistive networks.
- Chapter 2 develops *constraint boxes* as a general methodology for mapping regulari-

zation based algorithms to stable analog hardware. These systems find a solution by minimizing a global convex Lyapunov energy. Convexity implies that there are no other stable points in the energy surface other than a single global minimum.

- Chapter 3 describes an experimental analog VLSI chip for smooth surface interpolation from sparse depth data. Subtraction constraint boxes are used to enforce a second-order or "thin-plate" smoothness constraint on the data. This algorithm, as well as most regularization-based approaches, oversmoothes discontinuities.

- Chapter 4 describes the *resistive fuse*, the first hardware circuit that explicitly implements line process discontinuities. A two-dimensional fuse circuit successfully segments data which is scanned into the chip. However, resistive fuse networks must minimize a non-convex energy function that may contain local minima. As in computer vision, complex annealing or continuation methods are necessary for adequate solutions of the problem.

- Chapter 5 discusses an entirely new way of dealing with discontinuities that is not plagued by problems with local minima. The *tiny-tanh* network smoothes and segments data by minimizing a convex energy functional. The tiny-tanh approach is demonstrated in analog hardware.

- Chapter 6 speculates on some future directions of this work and summarizes the thesis.

Several subthreshold CMOS circuits are shown and described in this thesis. Rather than provide an exhaustive tutorial on this subject, the reader is strongly recommended to review Mead's excellent textbook on analog VLSI and neural systems [Mead, 1989]. (See also [Vittoz, 1985]). Furthermore, since the emphasis of this thesis has been the development of new algorithms, much can be learned by readers totally unfamiliar with analog VLSI. The ideas developed here have utility independent of any particular implementation technology.

1.1 Analog Models

Analog computation is not a new concept. In the 1950s, electronic analog computers were widely used to solve many types of mathematical problems, especially ordinary differential equations and boundary-value problems (for example, see [Karplus, 1958] or [Levine, 1964]). These analog systems provided a faster and cheaper solution to these problems than the crude digital systems of the day. Digital computers were required whenever greater accuracy was needed than the accuracy analog computers could provide.

The idea of using analog circuits for solving vision problems was explicitly stated by Horn, when he proposed the use of a grid of resistors to find the inverse of the discrete approximation to the thresholded Laplacian [Horn, 1974]. Later, Knight built an analog network for convolving an image with a Difference-of-Gaussian filter. Gaussian convolution was implemented using a resistor-capacitor network [Knight, 1983]. As charge diffused according to the diffusion equation, the diffusion time determined the width of the Gaussian.

Knight also proposed a MOS switching network that alternately opened switches to share charge in the horizontal and vertical directions [Knight, 1983]. This network implemented a binomial approximation to a Gaussian. A similar idea was implemented by Sage in a CCD process [Sage, 1984] [Sage and Lattes, 1987]. After an image was acquired by an on-chip imager, it was shifted to a convolver portion of the chip. Charge would then be alternately shared in the x and y directions until the correct amount of smoothing was performed. Yang implemented the same binomial approximation by efficiently performing these charge sharing operations as the image was being shifted out [Yang, 1990].

Poggio and Koch [1985] show how standard regularization algorithms map onto resistive networks. Exploiting Kirchhoff's and Ohm's law, they proved that the minimum of the regularized, quadratic cost functional is equivalent to the state of least power dissipation in an appropriate linear resistive network, where the data are given by injecting current into certain nodes and the solution by the stationary voltage distribution. For an overview of the use of analog circuits for early vision, see [Koch, 1989] and [Horn, 1988].

The work in this thesis relies upon the pioneering development of subthreshold, analog CMOS VLSI circuits for various sensory tasks by Carver Mead [Mead, 1989]. Mead is interested in understanding biological systems, and he and coworkers have developed numerous silicon models including the retina [Mead and Mahowald, 1988] [Mead, 1989], the cochlea [Lyon and Mead, 1988], and a circuit for auditory localization [Lazzaro and Mead, 1989]. Since analog VLSI circuits operate under many of the same power and communication restrictions imposed upon their biological counterparts, it is hoped that the silicon models will provide insights into these biological information processing systems.

1.2 Regularization Theory

Numerous problems in early vision are ill-posed [Poggio et al., 1985]. A problem is well-posed if its solution

1. exists,
2. is unique, and
3. depends continuously on the data.

A problem is ill-posed if it fails at least one of the above criteria. For example, many edge detection techniques amount to thresholding the magnitude of some sort of derivative of intensity data. Since derivatives amplify noise in the signal, edge detection is ill-posed because the result does not depend continuously on the data.

Consider

$$d = Au \tag{1.1}$$

where A is a known operator. We must solve the inverse problem of finding u from the measured data d which is typically ill-posed. Of course, the forward problem of finding d from u is straightforward and well-posed since A is known. The idea behind regularization is to use *a priori* information to constrain the problem in such a way that there is only one

unique solution that depends continuously on the data. Typically, an energy functional is created that consists of two terms. The data term forces the solution to be close to the data and the stabilizing term imposes constraints (such as smoothness) upon the solution. The solution u is the function that minimizes

$$||Au - d||^2 + \lambda||Pu||^2 \tag{1.2}$$

where the norms are usually quadratic and the stabilizing functional $||Pu||$ is typically a smoothness assumption. The regularization parameter λ controls the tradeoff between closeness of the solution to the data versus the degree of regularization. Many early vision algorithms can be described as examples of regularization even though most of these algorithms were developed before the unifying theory of regularization was applied to vision. Examples exist in stereo [Marr and Poggio, 1976], surface interpolation [Grimson, 1981] [Terzopoulos, 1983], optical flow [Horn and Schunck, 1981] [Hildreth, 1984], shape from shading [Ikeuchi and Horn, 1981], and edge detection [Poggio et al., 1986].

We will use surface interpolation as a problem to study regularization theory and analog networks. Reconstructing a surface from sparse sensory data is a well-known problem in computer vision. Early vision modules typically supply sparse depth, orientation, and discontinuity information. The surface reconstruction module incorporates these sparse and possibly conflicting measurements of a surface into a consistent, dense depth map. Suppose sparse and noisy data d_i are sampled randomly from a surface u. Since there are infinitely many surfaces u that could have led to these depth measurements, the problem is ill-posed. The usual regularizing method is to pick a stabilizing functional that enforces smoothness. If we use quadratic norms in Equation 1.2, we must minimize the following energy functional $E(u)$ with respect to all possible functions u:

$$E(u) = \int G(x)\,(d(x) - u(x))^2\,dx + \lambda \int (u')^2 dx \tag{1.3}$$

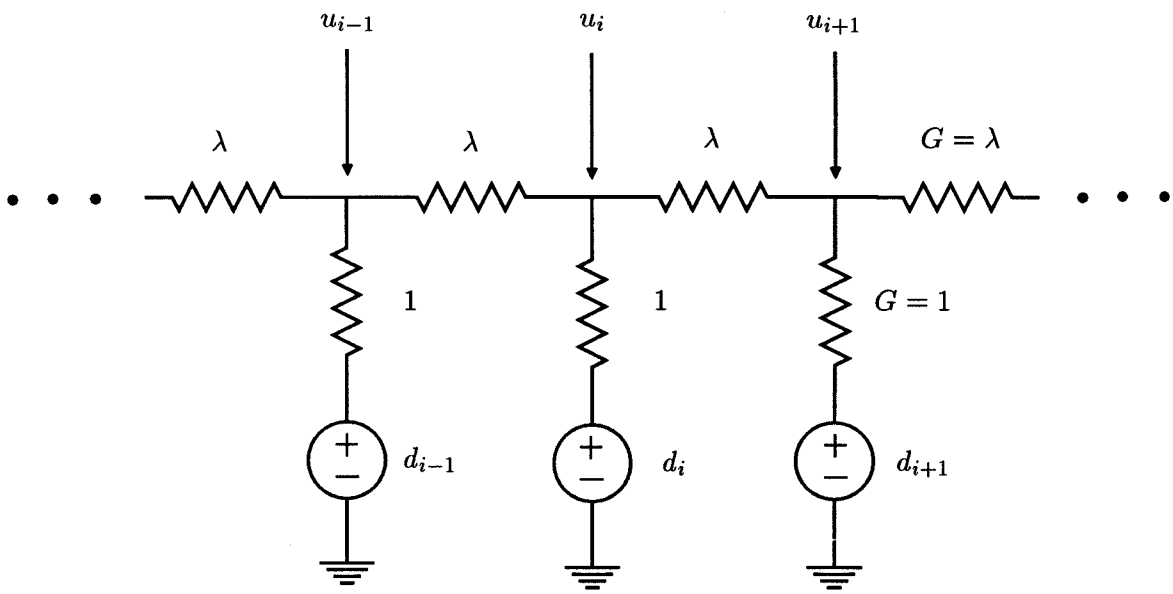


Figure 1.1: Resistive network for smoothing or interpolation. Data values d_i are supplied by voltage sources connected to the network with unity conductance resistors. Resistors with conductance λ implement the smoothness assumption. The stationary voltage values u_i are the smoothed outputs of the network.

It is assumed throughout this thesis that $G(x) = 1$. In discrete form, this equation turns into

$$E(u) = \sum_i (d_i - u_i)^2 + \lambda \sum_i (u_{i+1} - u_i)^2 \tag{1.4}$$

The first term ensures that the solution will be close to the data while the second term implements the smoothness constraint. Minimizing this term forces neighboring u_i to be similar. The typical approach in computer vision is to discretize time and repeatedly descend the gradient of the energy until a solution is found. This procedure is time-consuming even on the most powerful parallel computers. The next section describes an analog circuit that provides a continuous-time solution to this problem.

1.3 Resistive Networks

Consider the circuit shown in Figure 1.1. Noisy input data are provided as voltages d_i and the smoothed output values given by the voltages u_i. Resistors with conductance λ connect each u_i value with its neighbors. The total power dissipated by these resistors in the network

is equal to $\lambda \sum (u_{i+1} - u_i)^2$, which is identical to the second term of Equation 1.4. Unity conductance resistors couple u_i to d_i and the power dissipated by these resistors is identical to the first term in Equation 1.4. Therefore, the network dissipates exactly the power given by Equation 1.4. According to Maxwell's minimum heat theorem [Maxwell, 1891], the stationary voltage distribution is the solution to the minimization problem. Since all the resistors are linear, the node voltages are guaranteed to converge to a single unique set of values that minimize the total power dissipation of the circuit.

Resistor networks of this sort have a long history. Kirchoff [1845] describes what was probably the first conductive sheet system using a thin sheet of copper [Karplus, 1958]. Later, Kayan built an electrical model of thermal conducting systems using conducting paper [Kayan, 1945]. These metallic sheet simulations were hampered by the difficulty of obtaining uniform resistivity throughout the sheet. Evidently, it was difficult to reduce mismatches to below 10%. Karplus describes systems that use high-grade paper which reduced variations to below 2%.[1] Numerous systems have been constructed using various materials such as metallic screens, conductive fabrics, conductive paint on glass, graphite and wax models, and even conductive liquids. Later researchers used lumped circuit elements to approximate the continuous field properties of conductive sheets. Depack [1947] and Redshaw [1948] present some of the first published reports of successful solutions of Laplace's and Poisson's equation using this technique.

A hexagonal extension of the 1-D network shown in Figure 1.1 has been built in analog VLSI [Koch et al., 1990b] [Luo et al., 1989]. This 48x48 pixel network successfully performs interpolation and smoothing of noisy and possibly sparse data. This network used Mead's saturating resistor as its resistor element. This device is often called HRES (Horizontal RESistor) since it was originally designed to model the horizontal cells in the mammalian retina [Mead, 1989]. The I-V curve of this resistor is linear for small voltage drops and

[1] Interestingly, because of the high resistivity of paper compared to graphite, equipotential regions could be specified by drawing pencil lines on the paper.

saturates for large voltage drops. The slope of the linear region, i.e., the conductance for small voltage drops, can be varied over five orders of magnitude. The measured I-V curve has the form of a hyperbolic tangent as shown in Figure 1.2B. The next section discusses the implications of using nonlinear resistors, such as Mead's saturating resistor, on network stability and uniqueness.

1.4 Stability

Stability is an important issue to address in these analog models. Wyatt and Standley show how oscillations may occur when stable elements are coupled together with a resistive network [Wyatt et al., 1989]. They give criteria for guaranteeing the stable design of such systems. Since we are constructing devices out of highly nonlinear transistor building blocks, stability and uniqueness are pertinent issues.

As pointed out by Poggio and Koch [Poggio and Koch, 1984], the notion of minimizing power in linear networks implementing quadratic "regularized" algorithms must be replaced by the more general notion of minimizing the total resistor co-content [Millar, 1951] for nonlinear networks. For a voltage-controlled resistor characterized by $I = f(V)$, the co-content is defined as

$$J(V) = \int_0^V f(V')dV' \tag{1.5}$$

For a linear resistor, $I = GV$, the co-content is given by $\frac{1}{2}GV^2$, which is just half the dissipated power $P = GV^2$. The co-content for linear and saturating resistors is plotted in Figure 1.2.

If the *internal* dynamics of the incrementally active resistor circuit are neglected (that is, if we neglect the fact that our circuits consist of transistors with their own, relatively fast, temporal behavior), for any voltage input and any initial condition a purely passive resistive network will *not* oscillate indefinitely but must eventually settle to *some* stationary state (for more general stability results, see [Wyatt et al., 1989]). Purely passive means

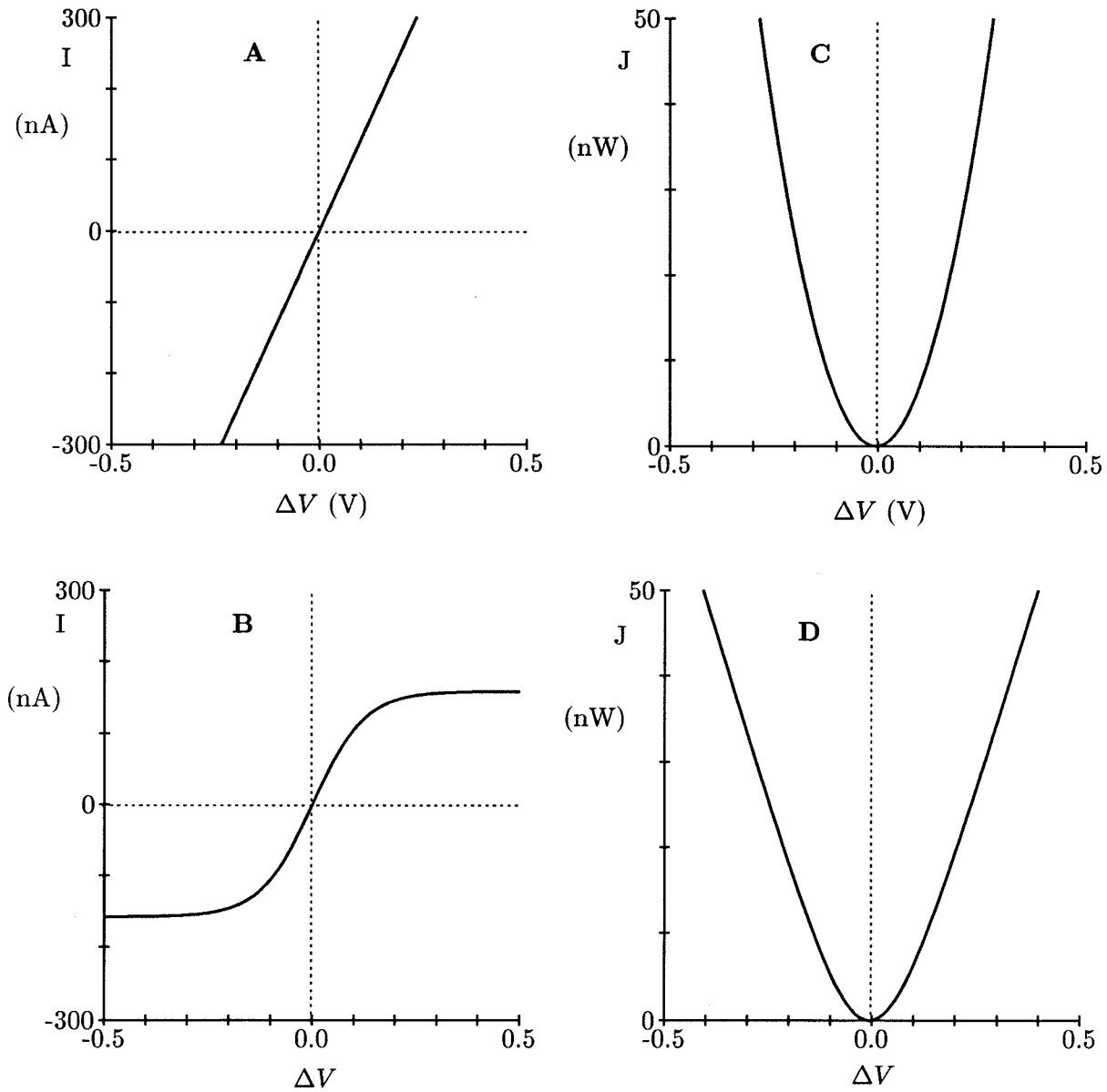


Figure 1.2: Theoretical I-V curves for a linear resistor (A) and a measured I-V curve for Mead's saturating resistor (B). Integrating numerically over these curves gives the co-content of the linear resistor (C) and the saturating resistor (D). Co-content is defined by Equation 1.5 and represents generalized power for nonlinear systems. The co-content for the linear resistor is equivalent to half the dissipated power, and thus a quadratic function in ΔV, while the co-content for the saturating resistor becomes a linear function of ΔV as $|\Delta V| \rightarrow \infty$.

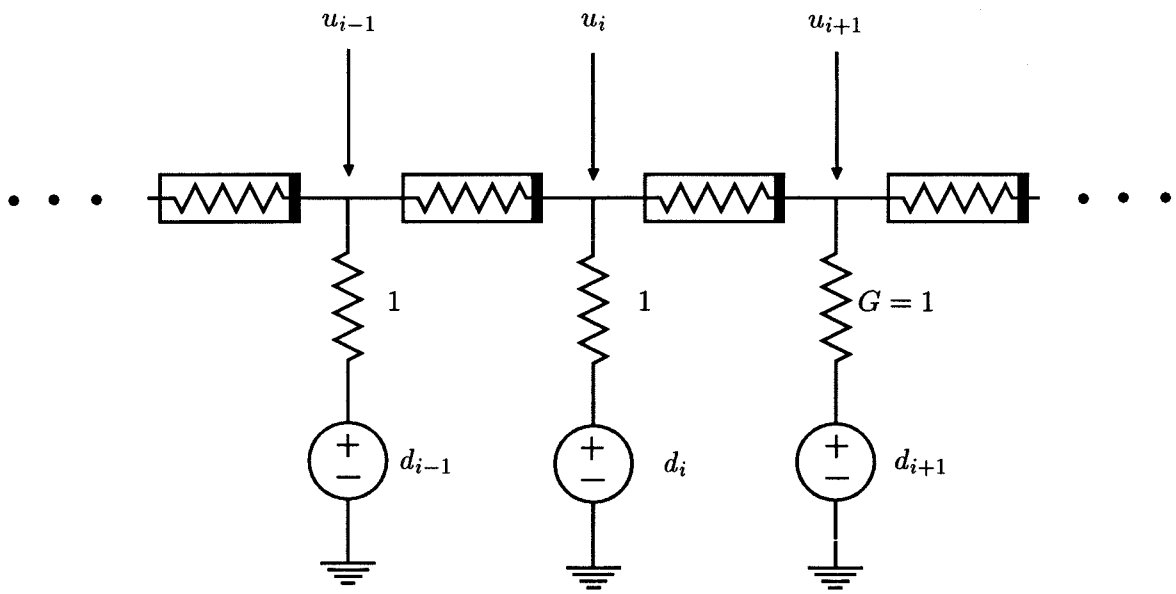


Figure 1.3: Nonlinear resistive network for smoothing or interpolation. The smoothing resistors in Figure 1.1 have been replaced by nonlinear resistors.

that each resistor's I-V curve only passes through the first and third quadrants as well as the origin. In other words, all resistive elements must dissipate power and no energy can be created. This conclusion holds even if parasitic (nonlinear but positive) capacitors are distributed arbitrarily throughout the network, provided there are no inductors. The total co-content of all resistors will act as a Lyapunov function [Harris et al., 1989].

1.5 Uniqueness

Given that nonlinear resistive networks can be shown to converge to a stable state under certain conditions, what can be said about the uniqueness of this stable state for arbitrary nonlinear resistive networks? It can be shown that a network of arbitrary topology consisting of strictly incrementally passive resistors (i.e., where $dI/dV > 0$ for all V) and ideal voltage and current sources has at most one solution, given by the unique minimum of the co-content [Harris et al., 1989]. This theorem first appeared in [Duffin, 1947]; see also [Birkhoff and Diaz, 1956]. A more recent treatment can be found in [Hasler and Neirynck, 1986] or [Chua et al., 1987].

In the network shown in Figure 1.3, each node is connected to exactly two nonlinear resistors and one linear resistor. Suppose the nonlinear resistors are described by an arbitrary nonlinear function $I = f(V)$. The co-content of the system is given by

$$J(u) = \frac{1}{2}\sum_i (u_i - d_i)^2 + \sum_i \int_0^{u_{i+1}-u_i} f(V)dV \tag{1.6}$$

For a network consisting of passive nonlinear resistive elements, a sufficient condition to guarantee a unique stationary network solution is

$$\frac{\partial^2 J}{\partial u_i^2} > 0 \tag{1.7}$$

for all u_i. This implies:

$$1 + f'(u_i - u_{i+1}) + f'(u_i - u_{i-1}) > 0 \tag{1.8}$$

In the worst case, this condition is satisfied if $f'(V) > -1/2$ for all values of V. For the 2D Cartesian case $f'(V) > -1/4$ for all values of Δv. For sparse data, or an arbitrarily connected nonlinear resistive network, this condition reduces to requiring the sign of the derivative of the I-V characteristic to be strictly positive. Because the I-V curve of Mead's saturating resistor has the form $I = I_0 \tanh(V/V_0)$ [Mead, 1989], its derivative is always positive. Thus, replacing ideal linear resistors with Mead's saturating resistors will not cause additional solutions to appear. This treatment again neglects the possible effect of the internal dynamics of the saturating resistor on the stability of the network.

We have shown how resistive elements implement the smoothness constraint required for regularizing ill-posed problems in early vision. The next chapter discusses constraint boxes that are analog models for implementing more powerful constraints.

Chapter 2

Constraint Boxes

This chapter introduces *constraint boxes*—a useful mechanism for solving computational problems that also provides a direct mapping to stable analog VLSI circuits.[1] This chapter introduces a theory of constraint boxes that can be used to implement powerful constraints that are useful in early vision.

Analog networks that solve most regularizable early vision problems can be designed with networks consisting solely of linear resistances and batteries [Poggio and Koch, 1985]. Unfortunately, many times these networks contain negative resistances that are troublesome to implement in analog hardware. Also, battery values may be complex functions (i.e., products or quotients) of measurable image quantities. This chapter proposes a new way of implementing the resulting analog networks using a theory of least-squares constraint boxes. Using constraint boxes, all nominal resistance values are data-independent and input voltages are relevant image quantities.

These least-square constraint boxes are contrasted with the "exact" constraint methods used in constrained optimization problems [Platt, 1990] [Arrow et al., 1958].

[1] Portions of this chapter have already been published [Harris, 1988] [Harris, 1989].

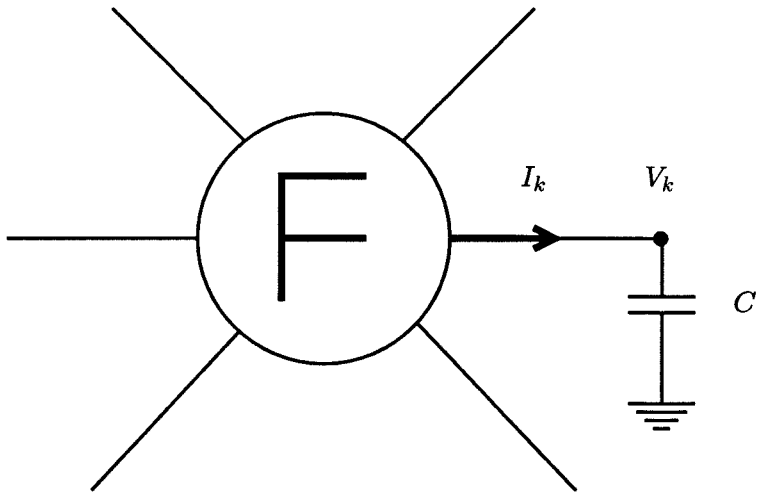


Figure 2.1: A generic constraint box enforces a constraint F on its voltage terminals. For the kth voltage terminal, a current I_k is generated that tends to push the voltage V_k in the direction which best satisfies the constraint F. The capacitor is added to provide dynamics to the system.

2.1 Single Constraint Boxes

A constraint box such as the one shown in Figure 2.1 imposes a constraint on the voltages of its terminals. A general constraint is given by

$$F(V_1, V_2, ...V_i....V_n) = 0 \tag{2.1}$$

V_i represents the voltage at the ith terminal. Intuitively, the constraint box senses the voltages at its terminals and feeds back currents that drive the voltages in the right directions to satisfy the constraint. The current supplied by the constraint box to the ith voltage terminal is

$$I_i = C\frac{dV_i}{dt} = -\frac{\partial F}{\partial V_i}FG \tag{2.2}$$

where G is an arbitrary constant that reflects the "conductance" of the constraint box. C is the capacitance at each node. We will assume $C = 1$ without loss of generality. It will be shown that for this choice of currents given by Equation 2.2 the constraint box minimizes an energy of

$$E = \frac{1}{2}GF^2 \tag{2.3}$$

which is the square error in the constraint equation scaled by the conductance G.

To prove this, take the time derivative of the energy and show that it is always decreasing:

$$\frac{dE}{dt} = \frac{1}{2}G\frac{d}{dt}F^2 \tag{2.4}$$

$$= GF\frac{dF}{dt} \tag{2.5}$$

By using the chain rule,

$$\frac{dE}{dt} = GF\sum_i \frac{\partial F}{\partial V_i}\frac{dV_i}{dt} \tag{2.6}$$

$$= -\sum_i (\frac{dV_i}{dt})^2 \tag{2.7}$$

since $dV_i/dt = -(\partial F/\partial V_i)FG$ from Equation 2.2. Thus dE/dt is always negative (or zero) and therefore E is always decreasing (or stationary). If G is always positive, the energy $E \geq 0$ and therefore is bounded. G and the capacitance at each node determine the speed of convergence of a single constraint box. We can force a constraint box to obey Kirchhoff's current laws by constraining all of the currents into the constraint box to sum to zero, i.e., $\sum I_i = 0$. This may require adding an extra terminal to the constraint box.[2]

2.2 Networks of Constraint Boxes

Imagine the arbitrary network of constraint boxes shown in Figure 2.2. The constraint enforced by the jth constraint can be written as

$$F_j(V_1, V_2, ...V_i....V_n) = 0 \tag{2.8}$$

V_i represents the voltage at the ith terminal of the jth constraint box. The constraint box current equation gives the current to the ith voltage terminal from the jth constraint box

[2] Actual constraint box implementations in CMOS will generally require additional connections to power and ground.

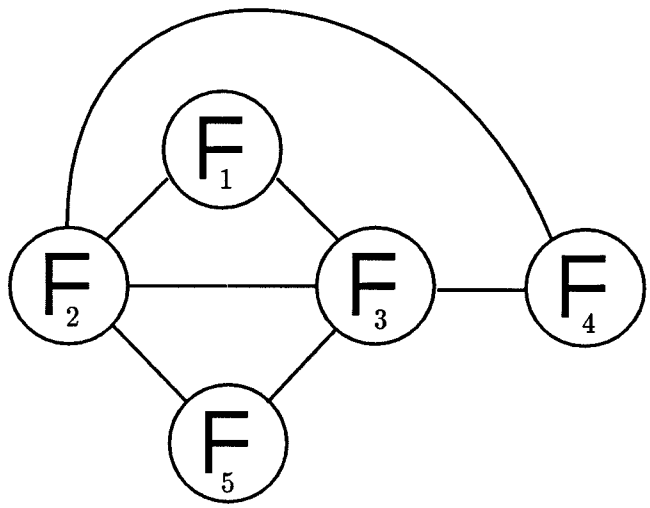


Figure 2.2: Arbitrary network of constraint boxes. It is shown in the text that any arbitrary collection of constraint boxes minimizes the sum of the squares of the errors in all of the constraints.

as

$$I_{ij} = -\frac{\partial F_j}{\partial V_i} F_j G_j \tag{2.9}$$

The constant G_j is the conductance of the jth constraint box. By Kirchhoff's current law, the total current supplied to node j equals the sum of all the currents supplied by the constraint boxes, so

$$\begin{aligned} C\frac{dV_i}{dt} = I_i &= \sum_j I_{ij} \\ &= -\sum_j \frac{\partial F_j}{\partial V_i} F_j G_j \end{aligned} \tag{2.10}$$

The total energy minimized will be the weighted sum of the squares of all of the constraint equations. Therefore,

$$E = \frac{1}{2}\sum_j G_j F_j^2 \tag{2.11}$$

Taking the time derivative of this energy yields

$$\frac{dE}{dt} = \frac{1}{2}\sum_j G_j \frac{d}{dt}(F_j)^2 \tag{2.12}$$

$$= \sum_j G_j F_j \frac{dF_j}{dt} \tag{2.13}$$

$$= \sum_j G_j F_j \sum_i \frac{\partial F_j}{\partial V_i}\frac{dV_i}{dt} \tag{2.14}$$

because of the chain rule. Substituting in Equation 2.10 (assuming C=1) and rearranging gives

$$\frac{dE}{dt} = \sum_i \frac{dV_i}{dt}\sum_j G_j F_j \frac{\partial F_j}{\partial V_i}$$

$$= -\sum_i (\frac{dV_i}{dt})^2 \tag{2.15}$$

which is always negative or zero. Therefore, a network of constraint boxes minimizes the weighted sum of the squares of all of the constraint equations utilizing simple gradient descent.

2.3 Examples of Constraint boxes

2.3.1 The Resistor

One of the simplest constraints to enforce is $A = B$, where A and B are the voltages of two nodes in a network. Of course, a simple method for enforcing this constraint exactly is to hook a wire from A to B, but this method does not generalize for more complicated constraints and does not degrade gracefully with conflicting constraints. The constraint we wish to enforce can be rewritten as

$$F(A, B) = A - B = 0 \tag{2.16}$$

Applying the constraint box current rule gives

$$\begin{aligned} I_A &= -GF\frac{\partial F}{\partial A} = -G(A-B) \\ I_B &= -GF\frac{\partial F}{\partial B} = G(A-B) \end{aligned} \tag{2.17}$$

Notice that these currents are exactly those that would be supplied by a resistor with conductance G connecting A to B. Therefore, the resistive networks discussed in Chapter 1 are a special case of these constraint box networks.

2.3.2 The Subtraction Constraint Box

The next chapter discusses a surface interpolation scheme that requires an element that enforces the constraint that $A - B = C - D$, which leads to the following error equation:

$$F(A,B,C,D) = (A-B) - (C-D) = 0 \tag{2.18}$$

Assuming that $G = 1$ and applying constraint box theory yields

$$\begin{aligned} I_A &= -F\frac{\partial F}{\partial A} = -(A-B) + (C-D) \\ I_B &= -F\frac{\partial F}{\partial B} = (A-B) - (C-D) \\ I_C &= -F\frac{\partial F}{\partial C} = -(A-B) + (C-D) \\ I_D &= -F\frac{\partial F}{\partial D} = (A-B) - (C-D) \end{aligned}$$

where I_A, I_B, I_C and I_D represent feedback currents that must be generated by the device.

Figure 2.3 shows a transistor realization of the above equations. As long as the differences $|A-B|$ and $|C-D|$ are small, we can assume that the differential pairs are working in their linear region. In subthreshold operation, the following constraint is implemented:

$$e^{V_1/(kT/q)}(A-B) = e^{V_2/(kT/q)}(C-D) \tag{2.19}$$

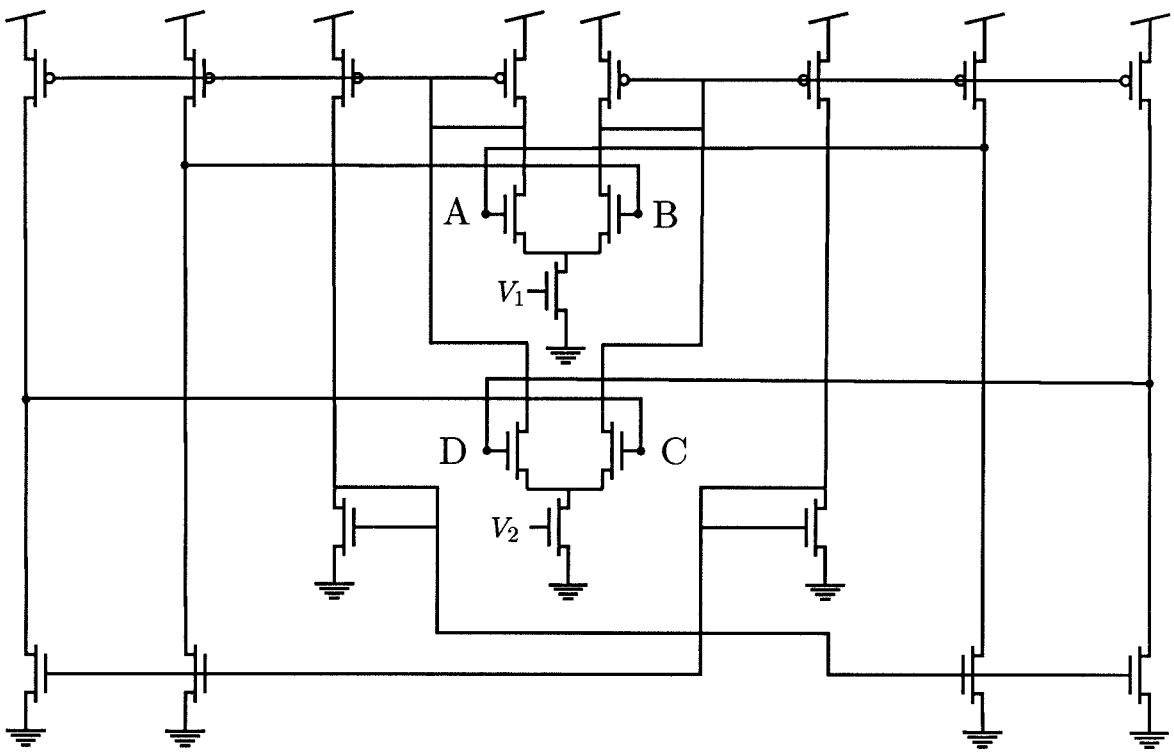


Figure 2.3: CMOS implementation of the subtraction constraint circuit. This device implements the constraint that $A - B = C - D$. V_1 and V_2 are bias voltages which control the relative weighting of $A - B$ vs $C - D$.

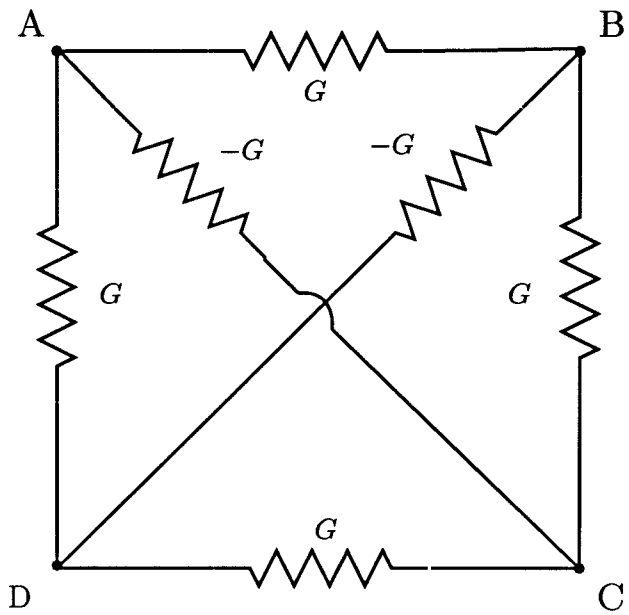


Figure 2.4: This resistor network also implements the constraint $A-B = C-D$ at the expense of constructing negative resistors.

The bias voltages V_1 and V_2 allow either side of the equation to be scaled arbitrarily. This circuit has been implemented in a standard CMOS process and successfully tested. It is used as a building block for the spline interpolation circuit discussed in Chapter 3.

An alternative implementation using resistors is shown in Figure 2.4. This circuit also minimizes the same constraint F^2 given above. Note, however, that negative resistors are required. Negative resistors are notoriously difficult to build and control stably.

2.3.3 The Multiplication Constraint Box

Many times a multiplication constraint is necessary. For example, the Tanner motion chip enforces a multiplication constraint to compute the uniform velocity of an image [Tanner, 1986] [Tanner and Mead, 1986]. In 1D, the basic constraint necessary is $k_1 A = k_2$, where k_1 and k_2 are arbitrary fixed constants. The constraint equation is

$$F(A) = k_1 A - k_2 \tag{2.20}$$

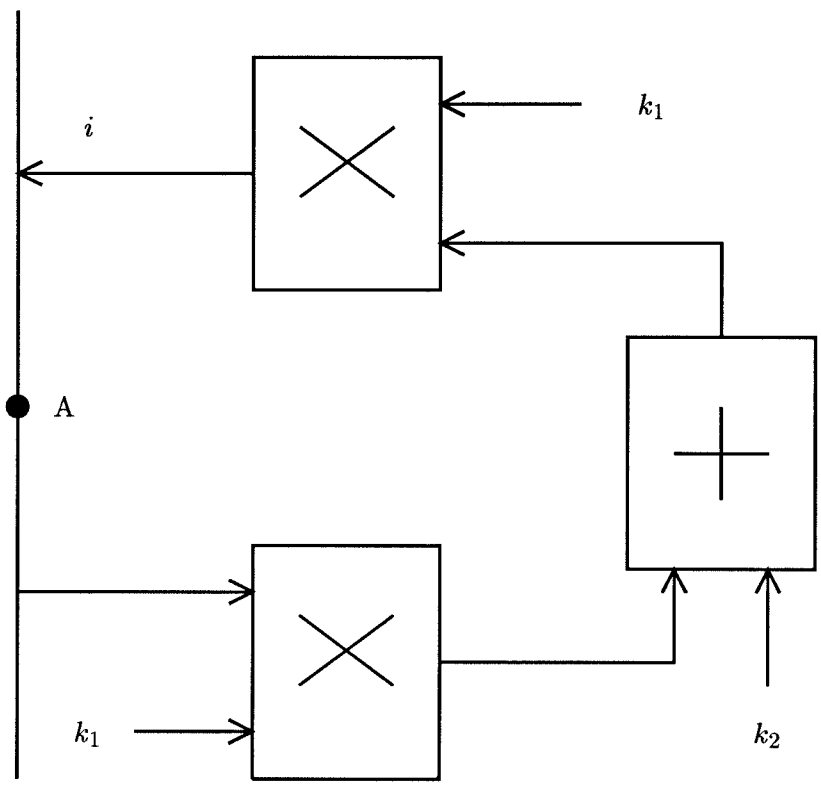


Figure 2.5: The Tanner-like multiplication constraint box implements the constraint that $k_1 A = k_2$ where k_1 and k_2 are input voltages and A is the output voltage.

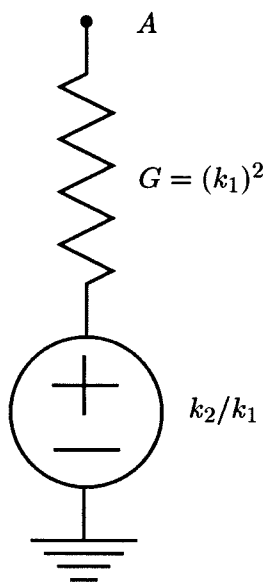


Figure 2.6: This resistor network also implements the constraint $k_1 A = k_2$ where k_1 and k_2 are input voltages and A is the output voltage. Although the resistor value is non-negative, the resistor and the voltage values are nontrivial functions of k_1 and k_2.

Now the current equation is

$$I_A = -F\frac{\partial F}{\partial A} = -k_1(k_1 A - k_2) \tag{2.21}$$

Figure 2.5 depicts a multiplication constraint box very similar to that used by Tanner.

Various strategies for implementing Figure 2.5 have been investigated. Building a compact four-quadrant multiplier in analog VLSI that is quantitatively accurate is a difficult task because of transistor mismatches. An alternative circuit using resistors is depicted in Figure 2.6. The resistor required is data-dependent but always positive. However, a voltage source is needed whose value is the four-quadrant quotient of two voltages. This design would also have to contend transistor mismatches.

Hutchinson et al. have proposed a analog network for computing optical flow from two images [Hutchinson et al., 1988]. The constraint-box methodology has been used to simplify this proposal to one in which all resistors are positive and data-independent [Harris et al., 1990b]. Furthermore, all voltage source values are relevant image quantities.

Dennis studied the use of analog networks to solve optimization problems. He used diodes to enforce inequality constraints [Dennis, 1959]. More recently, Chua has extended the approach of Dennis and actually built optimization circuits using discrete op amps [Chua and Lin, 1984]. In both of these approaches, negative resistors are still necessary.

2.4 Exact Constraint Boxes

Least squares constraint boxes are related to ideas from constrained optimization. It is rare in early vision processing to require constraints to be satisfied exactly. Omnipresent noise and inaccuracies present in the inputs and processing circuitry make it unlikely that constraints should be enforced exactly. Suppose, however, that there are hard and soft constraints in a particular problem. The hard constraints ($\hat{F}_i$) must be satisfied exactly while soft constraints (F_j) can be approximately fulfilled. In other words we must, minimize the following energy functional:

$$E = \frac{1}{2}\sum_j G_j F_j^2 \tag{2.22}$$

while satisfying $\hat{F}_i = 0$ for all values of i. One solution would be to implement $\hat{F}_i$ as a set of constraint boxes, each with conductance $\hat{G}$. This would lead to the following augmented energy function:

$$E = \frac{1}{2}\sum_i \hat{G}\hat{F}_i^2 + \frac{1}{2}\sum_j G_j F_j^2 \tag{2.23}$$

The conductance factor $\hat{G}$ can be increased to obtain arbitrary accuracy in the constraints $\hat{F}_i$, provided that the constraints do not conflict. $\hat{F}_i$ will be identically equal to zero in the limit as $\hat{G} \rightarrow \infty$. This technique is called the penalty method in the constrained optimization literature [Platt, 1990] [Gill et al., 1981]. The penalty method requires solving a very stiff set of differential equations causing instability problems for both numerical differential equation solvers and analog hardware [Platt, 1990].

Another method for satisfying constraints exactly is to minimize an absolute-value con-

straint energy instead of the more familiar quadratic. This would result in

$$E = \frac{1}{2}\sum_i \hat{G}|\hat{F}_i| + \frac{1}{2}\sum_j G_j F_j^2 \tag{2.24}$$

For values of $\hat{G}$ above a certain finite value, $\hat{F}_i$ will be satisfied exactly [Platt, 1990] [Gill et al., 1981]. This method will be discussed in more detail in Chapter 3 and actually used in working hardware in Chapter 5.

Another way to enforce the constraints $\hat{F}_i$ exactly is to use Lagrange multipliers. Under this formulation, the following energy would be minimized:

$$E = \sum_i \lambda_i \hat{F}_i + \frac{1}{2}\sum_i \hat{G}\hat{F}_i^2 + \frac{1}{2}\sum_j G_j F_j^2 \tag{2.25}$$

If gradient ascent is performed on λ, a small value of $\hat{G}$ solves the problem [Platt, 1990] [Arrow et al., 1958] [Bertsekas, 1976]. The $\frac{1}{2}\sum_i \hat{G}\hat{F}_i^2$ term is needed to guarantee stability.

Chapter 3

Coupled Depth/Slope Network

This chapter discusses a biharmonic surface interpolation chip using constraint boxes. The coupled/depth slope network minimizes a second-order or "thin-plate" energy of the surface.[1]

3.1 Coupled Depth/Slope

Reconstructing a surface from sparse sensory data is a well-known problem in computer vision. Early vision modules typically supply sparse depth, orientation, and discontinuity information. The surface reconstruction module incorporates these sparse and possibly conflicting measurements of a surface into a consistent, dense depth map.

The coupled depth/slope model provides a novel solution to the surface reconstruction problem [Harris, 1987]. Figure 3.1 depicts a high-level schematic of the circuit. The d_i voltages represent noisy and possibly sparse input data, the u_is are the smooth output values, and the p_is are the explicitly computed slopes. The vertical data resistors (with unity conductance) control the confidence in the input data. In the absence of data these resistors are open circuits. The horizontal chain of smoothness resistors (each with conductance λ) forces the derivative of the data to be smooth. This model is called the coupled depth/slope

[1] Portions of this chapter have already been published [Harris, 1986] [Harris, 1987] [Harris, 1989].

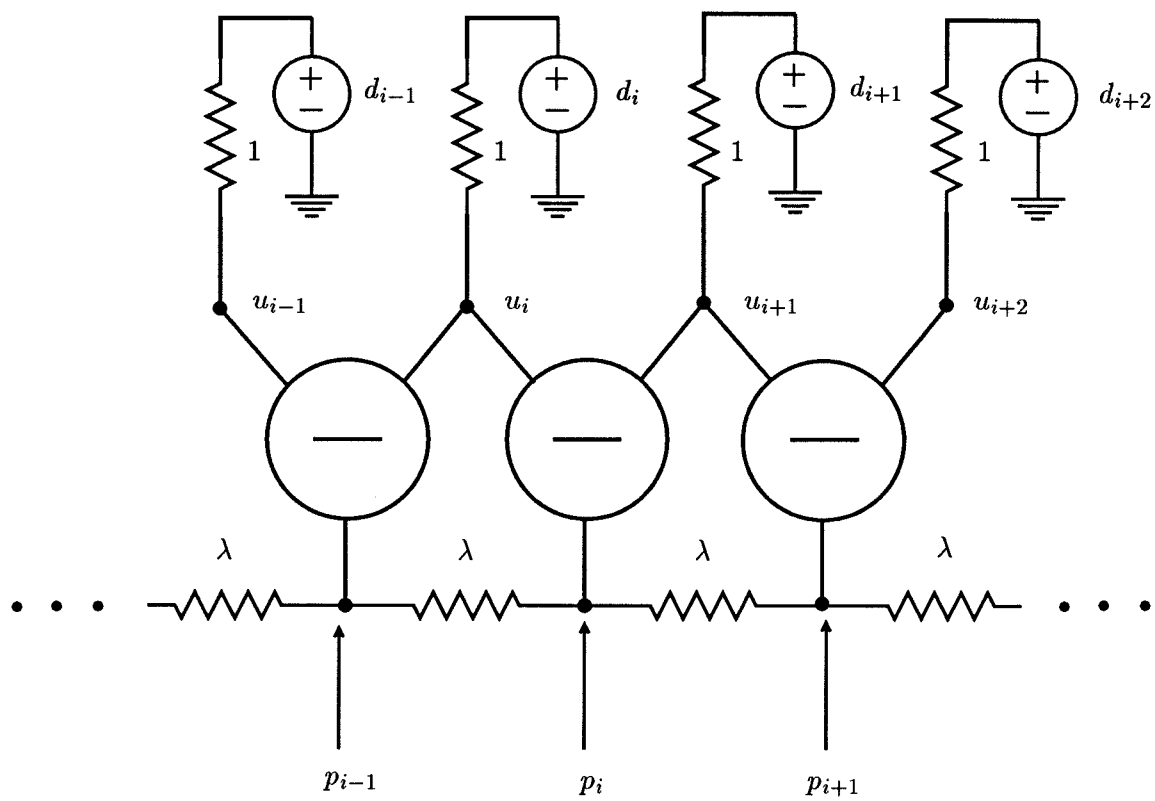


Figure 3.1: The coupled depth/slope model. The data d_i are supplied to the network through unity conductance resistors. The subtract constraint boxes compute an explicit representation of the derivatives p_i which are smoothed by resistors with conductance λ. The stationary voltage values u_i are the final smooth outputs.

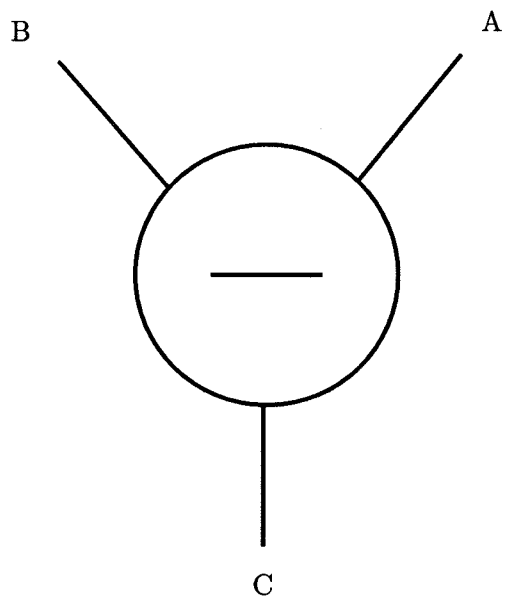


Figure 3.2: Single subtract constraint box. This device enforces the constraint that $A - B = C$.

model because of the coupling between the depth and slope representations provided by the subtraction elements. The subtractors explicitly calculate a slope representation of the surface. Any depth or slope node can be made into a constraint by fixing a voltage source to the proper location in the network. Intuitively, any sudden change in slope is smoothed out with the resistor mesh.

The tri-directional subtractor device (developed in the last chapter) is responsible for the coupling between the depth and slope representations. The subtract constraint box is shown schematically in Figure 3.2. If nodes A and B are set with ideal voltage sources, then node C will be forced to $A - B$ by the device. This circuit element is unusual in that all of its terminals can act as inputs or outputs. If nodes B and C are held constant with voltage sources, then the A terminal is fixed to $B + C$. If A and C are input, then B becomes $A - C$. This element can be constructed with the $A - B = C - D$ constraint box (shown in Figure 2.3) by setting the D terminal to a constant reference voltage that defines "zero" slope. C can now be positive or negative with respect to the reference voltage. When further constraints are added, this device minimizes an energy proportional to $(A-B-C)^2$. The energy minimized by the constraint box is different, in general, from the actual power

dissipated by its transistor implementation. In the limiting case of a continuous network, the total energy is

$$E = \int (u-d)^2 + G\left(\frac{du}{dx} - p\right)^2 + \lambda\left(\frac{dp}{dx}\right)^2 dx \tag{3.1}$$

The three terms arise from the power dissipated in the subtractors and in the two different types of resistors. Energy minimization techniques and standard calculus of variations have been used to formally show that the reconstructed surfaces, u, satisfy the 1D biharmonic equation between input data points [Harris, 1987]. In the two-dimensional formulation, u is a solution of

$$\nabla^2\nabla^2 u = 0 \tag{3.2}$$

This interpolant, therefore, provides the same results as minimizing the energy of a thin plate, which has been commonly used in surface reconstruction algorithms on digital computers [Grimson, 1981] [Terzopoulos, 1983].

In discrete terms, this energy becomes

$$E(u) = \sum_i (u_i - d_i)^2 + G(u_{i+1} - u_i - p_i)^2 + \lambda(p_i - p_{i+1})^2 \tag{3.3}$$

The variable G is a measure of the conductance of the constraint box. As the value of G increases, the constraint box accuracy becomes more exact. This is an implementation of the penalty method in constrained optimization [Gill et al., 1981].

3.2 Hardware Implementation

An eight-node 1D network of the form shown in Figure 3.1 was designed and fabricated in 3μm CMOS. Three important components of the model must be mapped to analog VLSI: the two different types of resistors and the subtractors. The vertical confidence resistors are built with simple transconductance amplifiers (transamps) connected as followers. The

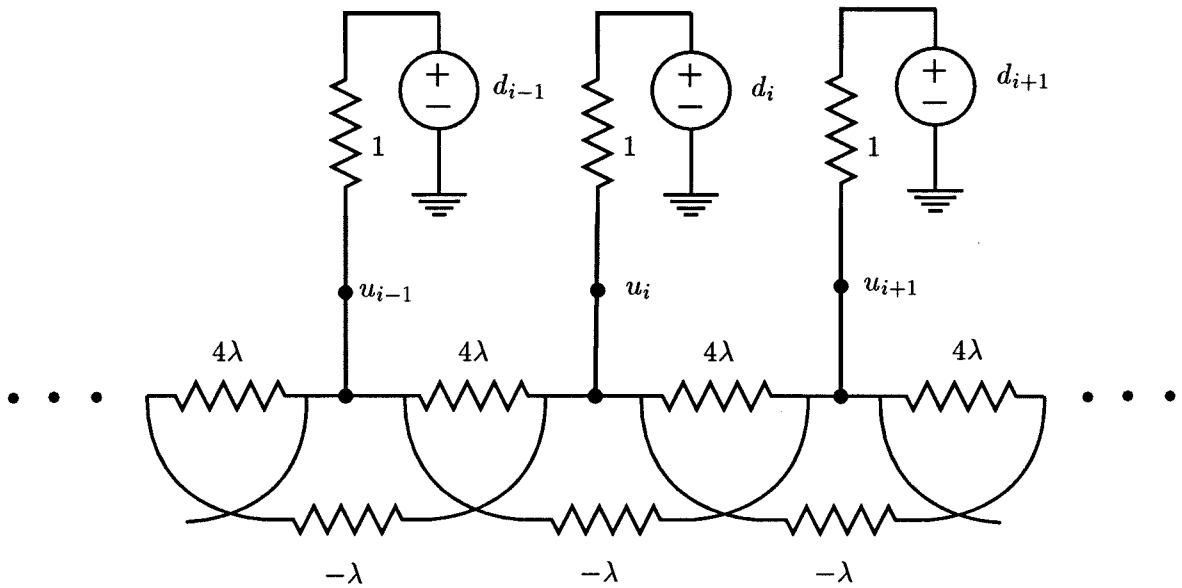


Figure 3.3: A negative-resistor solution to the 1D biharmonic equation. This circuit is equivalent to the network in Figure 3.1 in the limit as the conductance of the constraint boxes (G) goes to infinity.

bias voltage of the transamp follower determines its conductance and therefore signifies the certainty of the data. If there are no data for a given location, the corresponding transamp follower is turned off. The horizontal smoothness resistors are implemented with Mead's saturating resistor [Mead, 1989]. Since conventional CMOS processes lack adequate resistive elements, we are forced to build resistors out of transistor elements. The bias voltage for Mead's resistor allows the effective conductance of these circuit elements to vary over many orders of magnitude.

The circuit shown in Figure 3.3 computes the same solutions as the coupled depth/slope network in the limit as $G \rightarrow \infty$. Interestingly, a 2-D implementation of this idea was implemented in the 1960s using inductors and capacitors. Proper choice of the frequency of alternating current allowed the circuit elements to act as pure positive and negative impedances. Unfortunately, negative resistances are troublesome to implement, especially

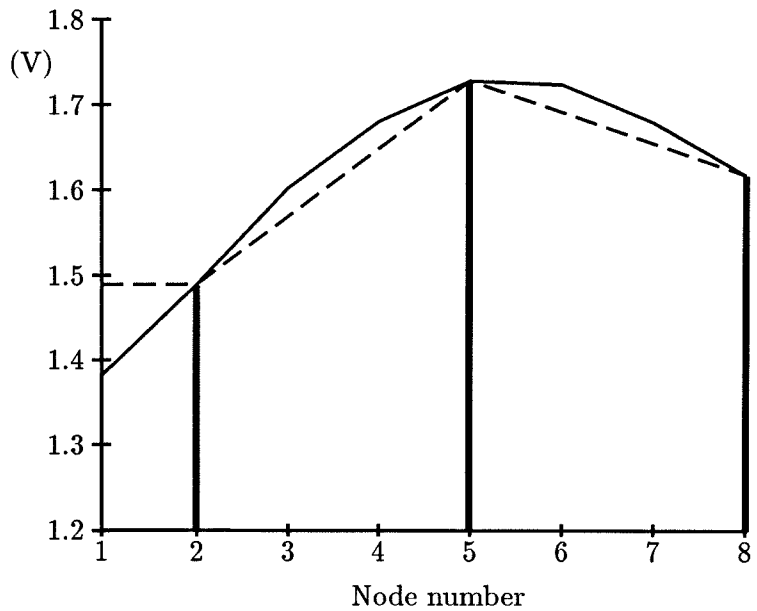


Figure 3.4: Measured data from the second-order chip (solid line) and simulated first-order result (dashed line). The second-order interpolant is smoother and also extrapolates beyond the data points.

in analog VLSI. Recently, a 2D approximation of this network was constructed in analog VLSI [White and Abidi, 1989] [Kobayashi et al., 1991]. Their chip dissipates over 2 Watts of power; presumably much of this is necessary for the negative resistor elements. One of the big advantages of using constraint boxes to implement early vision algorithms is that the resulting networks do not require negative resistances.

Figure 3.4 shows a sample output of the circuit. Data (indicated by vertical dashed lines) were supplied at nodes 2, 5, and 8. As expected, the chip finds a smooth solution (solid line) that extrapolates beyond the known data points. As discussed in Chapter 1, a single resistive grid minimizes the first-order or membrane energy of a surface. A disadvantage of the coupled depth/slope implementation is its sensitivity to transistor offsets due to the nature of the derivative operation. The first-order resistive network implementation using Mead's saturating resistor is guaranteed to have zero voltage offset.

Figure 3.4 also shows the simulated performance of a first-order energy or membrane energy minimization. Data points are again supplied at nodes 2, 5, and 8. In contrast to the second-order chip results, the solution (dashed line) is much more jagged and does not

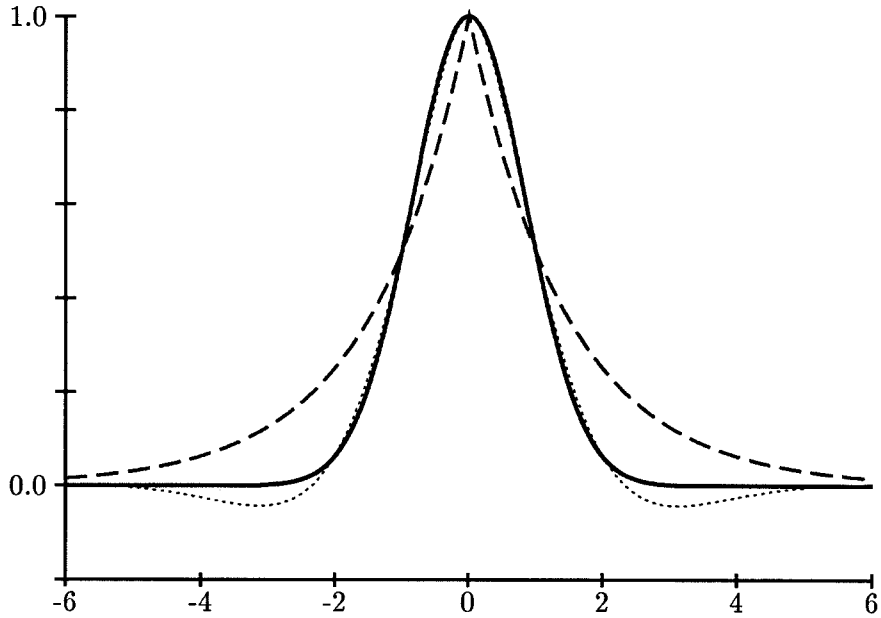


Figure 3.5: Graphical comparison of 1D analytic Green's functions for first-order (dashed line), second-order (dotted line) and Gaussian (solid line).

extrapolate outside of the known data points (for example, see node 1). Interestingly, psychophysics experiments support the smoother interpolant used by the second-order coupled depth/slope chip [Grimson, 1981]. Unlike the second-order network, the first-order network is not rigid enough to incorporate either orientation constraints or orientation discontinuities [Terzopoulos, 1983].

Image smoothing is a special case of surface interpolation where the data are given on a dense grid. The first-order network is a poor smoothing operator. A comparison of the analytic Green's functions of the first- and second-order networks is shown in Figure 3.5 (the first-order shown with a dashed line and the second-order with a solid line). Note that the analytic Green's function of the second-order network (solid line) and that of standard Gaussian convolution (dotted line) are nearly identical. This fact was pointed out by Poggio, Voorhees, and Yuille [1986], when they suggested the use of the second-order energy to regularize the edge detection problem. Gaussian convolution has been claimed by many authors to be the "optimal" smoothing operator and is commonly used as the first stage of edge detection [Marr and Hildreth, 1980] [Poggio et al., 1986]. Though the

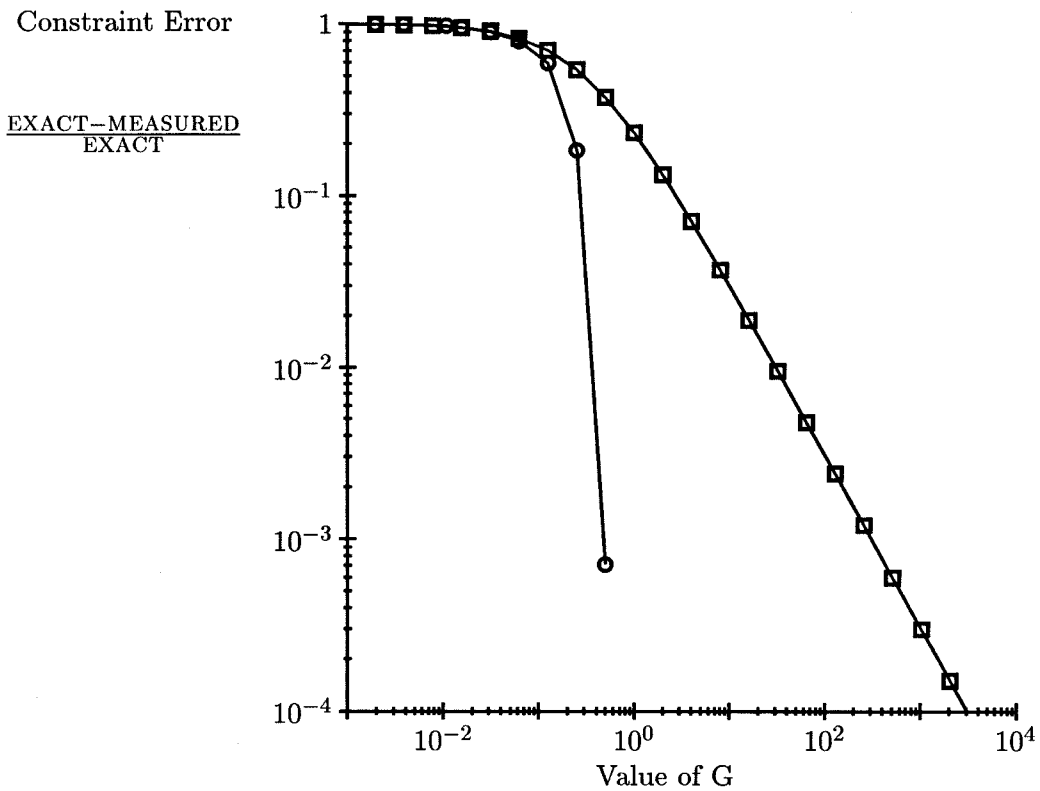


Figure 3.6: Inaccuracy of the constraint box versus G for simulation of the coupled depth/slope network. The penalty method (Equation 3.4) is shown with squares and the absolute value penalty method (Equation 3.5) shown with circles.

second-order network can be used to smooth images, Gaussian convolution cannot be used to solve the more difficult problem of interpolation between sparse data points.

3.3 Constrained Optimization Approach

The hardware described in this chapter minimizes the following energy functional when all the devices are operating in the linear range:

$$E(u) = \sum_i (u_i - d_i)^2 + G(u_{i+1} - u_i - p_i)^2 + \lambda(p_i - p_{i+1})^2 \tag{3.4}$$

The variable G is the measure of the conductance of the constraint box. As the value of G increases, the constraint box computation becomes more exact. An infinite value of G is necessary for the constraint to be satisfied exactly. An alternative is to use the absolute

value constraint. This energy becomes

$$E(u) = \sum_i (u_i - d_i)^2 + G|u_{i+1} - u_i - p_i| + \lambda(p_i - p_{i+1})^2 \tag{3.5}$$

As discussed in Chapter 1, a small finite value of G will suffice to ensure that the constraint is fulfilled exactly. Both the quadratic and the absolute value methods were simulated. Figure 3.6 compares the accuracy of the two methods for various values of G. As expected, the absolute value method reduces the error in the constraint equation to within machine precision even for small values of G.

The coupled depth/slope network has a natural generalization that enforces smoothness constraints to any arbitrary level of derivative of the data. Furthermore, discontinuities of any order can be detected with resistive fuses on each level of derivative. For example, discontinuities in the first-derivative, i.e., creases, can be detected. Higher-order smoothing and discontinuity detection has been demonstrated for edge detection, image segmentation, and surface interpolation [Liu and Harris, 1989]. Recently, Suter has studied the higher-order smoothness problem [Suter, 1990a]. His approach uses the exact constraint boxes proposed by Platt instead of the penalty method [Suter, 1990b].

Chapter 4

Resistive Fuse

This chapter describes the *resistive fuse* circuit—the first hardware circuit that explicitly implements either analog or binary line processes in a controlled fashion. We have successfully designed and tested an analog CMOS VLSI circuit that contains a 20x20 network of resistive fuses implementing piece-wise smooth surface interpolation. The segmentation ability of this network is demonstrated for various 2-D inputs. Finally extensions of these ideas are discussed for the detection of outliers and for segmentation of temporal signals.[1]

4.1 Line Processes

Standard regularization theory provides a basis for solving most early vision problems but algorithms that depend solely on a smoothness assumption perform badly at discontinuities. In surface reconstruction, for example, an inflexible surface smoothness constraint either blurs discontinuities or else causes unrealistic overshoot effects wherever there are breaks in the surface. Resistors and constraint boxes can map most regularization theory algorithms onto stable analog hardware, but they must be extended to deal with discontinuities.

Many recent approaches combine discontinuity detection with surface reconstruction by

[1] Portions of this chapter have already been published [Harris et al., 1989] [Harris and Koch, 1989] [Harris et al., 1990a].

defining an energy function of the surface as a nonconvex energy function that includes binary line processes. Line processes provide a systematic method of selectively "breaking" the smoothness constraints imposed by standard regularization theory. Geman and Geman worked with binary line processes on a Markov field lattice in reconstructing surfaces from noisy images [Geman and Geman, 1984]. They also used simulated annealing to minimize the resulting nonconvex energy. Marroquin used two coupled Markov random fields to solve the surface reconstruction problem—one for analog depth values and the other for binary line processes [Marroquin, 1987].

Rather than use time-consuming stochastic annealing methods, Koch, Marroquin and Yuille later showed how the reconstruction problem can be solved using analog networks where the discontinuities are allowed to vary continuously between 0 and 1 [Koch et al., 1987]. The energy function of this network is minimized using similar updating equations from Hopfield's work [Hopfield, 1984]. Koch et al. used a first-order energy model (membrane) for reconstructing the surface between sparse depth points of simple synthetic images. Blake and Zisserman used similar energy functions to locate discontinuities in natural images using the membrane model [Blake and Zisserman, 1987]. They also extended their work using a plate model to locate discontinuities in synthetic images. They obtain the minimum of the energy function by an approach they called the GNC (gradient non-convexity) method. A convex approximation to the problem is first constructed and then the energy landscape is continuously transformed into its true form.

A number of authors have used deterministic methods to find the (local) minima of non-convex variational functionals, with next-to-optimal results (e.g., [Koch et al., 1987] [Terzopoulos, 1986] [Chou and Brown, 1986]). Line processes were extended and modified to account for discontinuities in depth, texture, optical flow and color [Marroquin, 1987] [Marroquin et al., 1987] [Hutchinson et al., 1988] [Poggio et al., 1988] [Zhou and Chellappa, 1988]. The principal drawback of all of these methods is the computational expense involved in minimizing the associated non-quadratic cost functionals, in

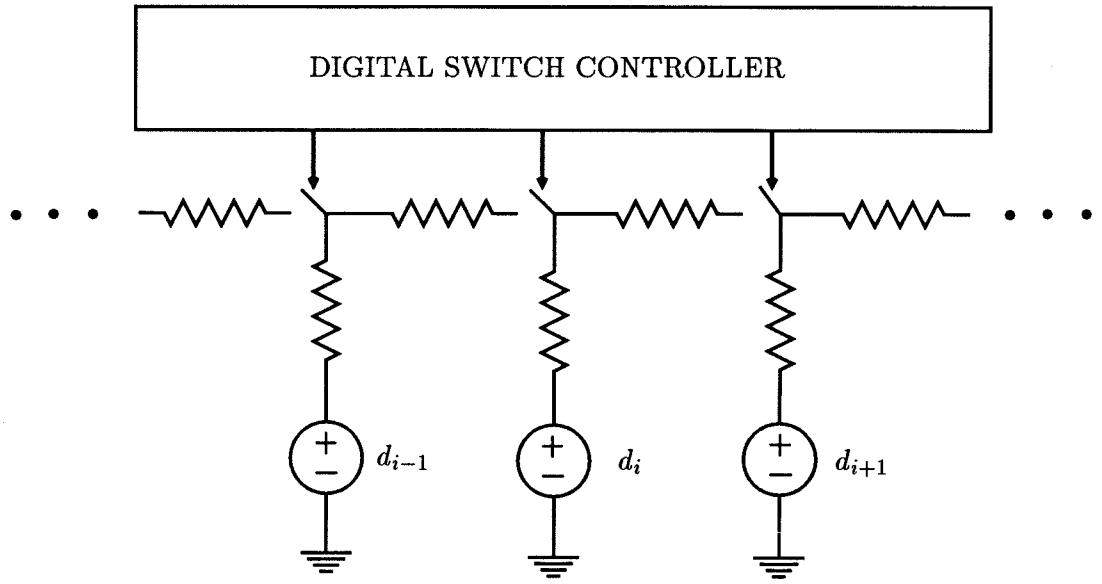


Figure 4.1: Resistor network with digital switching network, similar to proposal in [Koch et al., 1987].

particular when numerous constraints are incorporated.

Including binary line processes ℓ into the functional of Equation 1.4 leads to

$$E(u) = \sum_i (d_i - u_i)^2 + \lambda \sum_i (u_{i+1} - u_i)^2 (1 - \ell_i) + \sum_i \alpha \ell_i \tag{4.1}$$

where α is an additional free parameter. The second term in this functional has been modified to implement the constraint that surfaces should vary smoothly for small values of the surface gradient. If all variables, with the exception of u_i, u_{i+1}, and ℓ_i, in Equation 4.1 were held fixed and $\lambda(u_{i+1} - u_i)^2 < \alpha$, it would be "cheaper" to pay the price $\lambda(u_{i+1} - u_i)^2$ and set $\ell_i = 0$ than to pay the larger price α. However, if the difference becomes too steep, the line process is switched on, i.e. $\ell_i = 1$, and the "price" α is paid. The functional of Equation 4.1 is non-convex and a large number of both stochastic and deterministic methods have been designed to find optimal or nearly optimal solutions for this and similar functionals [Geman and Geman, 1984] [Marroquin et al., 1987] [Koch et al., 1987] [Blake and Zisserman, 1987] [Terzopoulos, 1983] [Terzopoulos, 1986].

One straightforward manner to implement line discontinuities is via binary switches, breaking the resistive connections among neighboring nodes. Figure 4.1 depicts a pro-

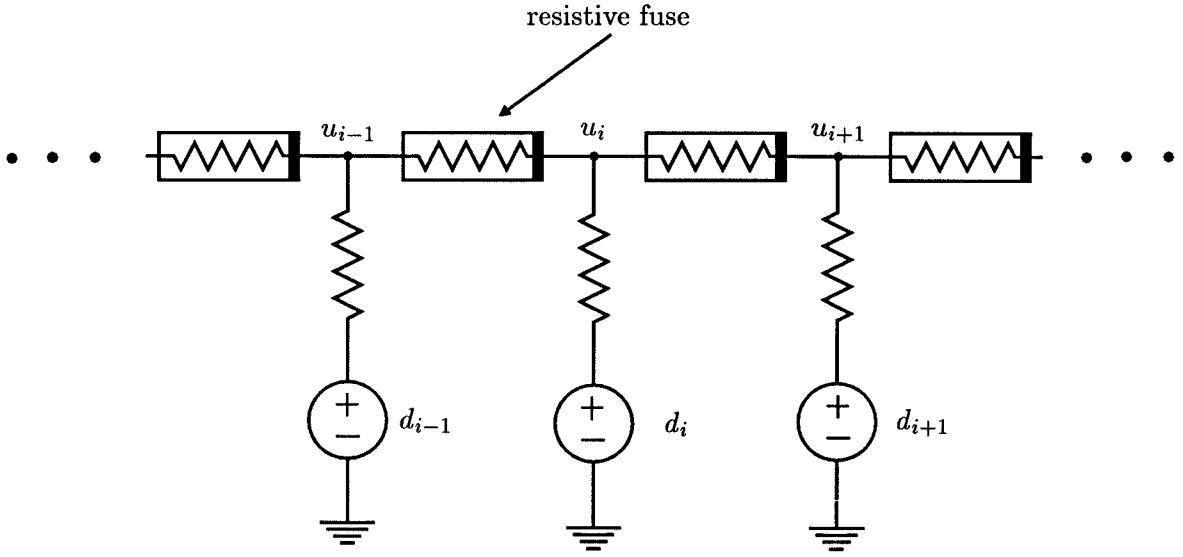


Figure 4.2: 1D fuse network.

posal for implementing these line processes in hardware, i.e., see [Koch et al., 1987] and [Hutchinson et al., 1988]. Switches are added to the resistive network of Figure 1.1. Each switch has an associated digital processor that controls the opening and closing of the switch in either a deterministic or in a stochastic fashion, depending on the value of the voltage across the switch as well as on the states of neighboring switches. Such an analog-digital implementation is quite difficult to implement within conventional two-dimensional silicon circuits, due to the high amount of connectivity among nodes and the complexity of integrating the analog and digital processes. This mixed analog-digital circuit can be replaced by a single analog non-linear resistor, the "resistive fuse" as shown in Figure 4.2.

4.2 Resistive Fuse Theory

The appropriate current-voltage relationship of a *binary resistive fuse* is illustrated in Figure 4.3a. As long as the voltage drop across this device is below a threshold, the current through the nonlinear resistor is linearly related to the voltage across it. Once past the voltage threshold, the fuse open-circuits (hence the name "fuse"), and the current is zero. Unlike conventional electrical fuses, however, the resistive fuse can "reconnect" after open-

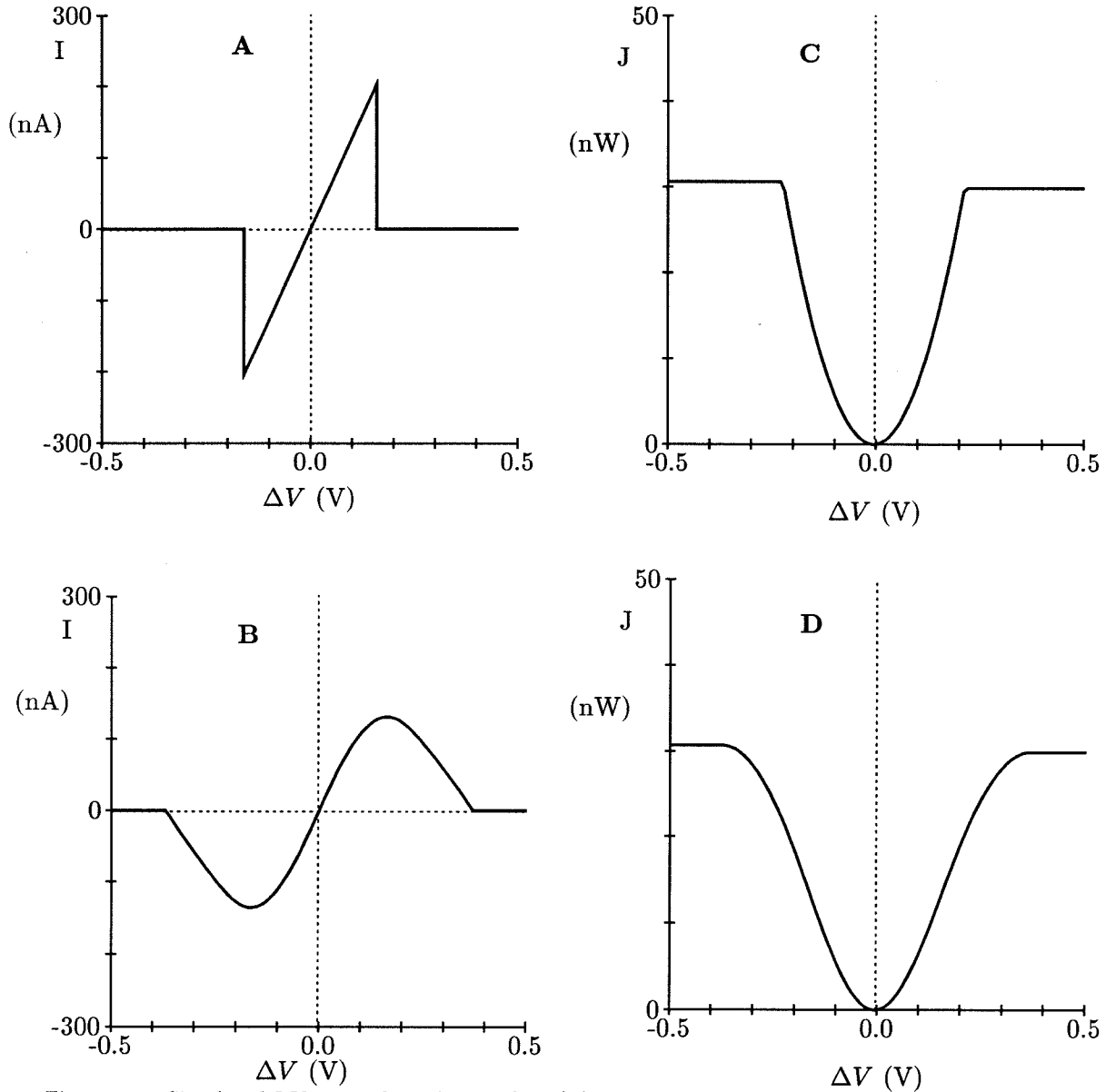


Figure 4.3: Simulated I-V curve for a binary fuse (A) and an experimentally measured I-V curve for an analog resistive fuse (B). Integrating numerically over these curves gives the co-content J for the binary (C) and the analog fuse (D).

circuiting as long as the voltage across the fuse drops again below the threshold. This two-terminal device therefore implements the high-level constraint that surfaces should be smooth unless their neighboring values differ by more than $\sqrt{\alpha/\lambda}$, at which point the surface will break. Figure 4.3c can be interpreted as a plot of the co-content ($J(u,\ell)$) as a function of the depth at locations u_i and u_{i+1} and as a function of the discontinuity ℓ_i. The values of the surface and of the line discontinuities are assumed to be fixed at all other locations. As long as $\lambda(u_{i+1}-u_i)^2 \leq \alpha$, the function J is quadratic in the gradient. However, once $|u_i - u_{i+1}|$ exceeds the gradient limit $\sqrt{\alpha/\lambda}$, J remains flat at $J=\alpha$, independent of the magnitude of $u_i - u_{i+1}$ [Blake and Zisserman, 1987].

The measured I-V relationship of the analog fuse is shown in Figure 4.3b. The most salient difference from the binary fuse is the smooth flanks, where the current decreases smoothly to zero for increasing values of the voltage gradient (in contrast with the discontinuity in the I-V relationship for the binary fuse). In this region, the slope conductance dI/dV will be negative (Figure 4.15). The measured I-V curve can be related directly to the concept of analog line discontinuities of Koch et al.[1987]. The key idea is that, following Hopfield and Tank [1985] in their neural network implementation of the Traveling Salesman Problem , binary discontinuities are mapped onto continuous "neurons," whose output is constrained to lie between 0 and 1. The input-output relationship of these "discontinuity neurons" is governed by the sigmoidal function $V = \mathrm{g}(U)$, where $\mathrm{g}(U)$ is a strictly monotonic function, usually taken to be

$$\mathrm{g}(U) = \frac{1}{1+e^{-2\eta U}} \tag{4.2}$$

with the "gain" $\eta > 0$. The network converges to a stationary solution using a steepest-descent rule. It is straightforward to derive an analog version of resistive fuses with the following I-V relationship

$$I = f(V) = \left[1 - g(V^2 - \alpha)\right] V \tag{4.3}$$

Our measured I-V curve for the fuse (Figure 4.3b) implements a similar function. For

$\eta \rightarrow \infty$, the function g becomes binary and $f(V)$ of Equation 4.3 approaches the form of the binary fuse (Figure 4.3a).

The analytical form of the I-V relationship of the resistive fuse element we used in our simulations (see below) can be rigorously derived using a deterministic Mean Field Approximation to the underlying stochastic Markov Random Field model of piecewise smooth surface interpolation [Geiger and Girosi, 1989]. In this approximation, common to statistical mechanics [Huang, 1963], the interaction among neighboring values of u and ℓ is replaced by the interaction among neighboring mean values of u and ℓ. If we re-interpret the results of Geiger and Girosi within our electrical circuit framework, we arrive at the constitutive relationship of the analog resistive fuse given in Equation 4.3.

4.3 Hardware Implementation

The circuit schematic for the original analog resistive fuse is shown in Figure 4.4. The circuitry above the dotted line in the figure is Mead's saturating resistor with a p-type pullup transistor that sets the nominal resistance of the fuse. In subthreshold operation, the current through a transistor varies exponentially with the gate-to-source voltage. Thus, the voltage V_B produces a current I_B equal to:

$$I_B = I_0 e^{\kappa(V_{DD} - V_B)} \tag{4.4}$$

All voltages are assumed to be normalized by kT/q. The variable κ is a process-dependent parameter that reflects the inability of the gate to be 100% effective in reducing the barrier potential [Mead, 1989]. I_0 is a constant that includes the width and length of the transistor as well as process-dependent fabrication parameters. Letting $I_F = I_B$, the I-V relation of the resistor can be derived as:

$$I_{FUSE} = \frac{I_F}{2} \tanh\left(\frac{\Delta V}{2}\right) \tag{4.5}$$

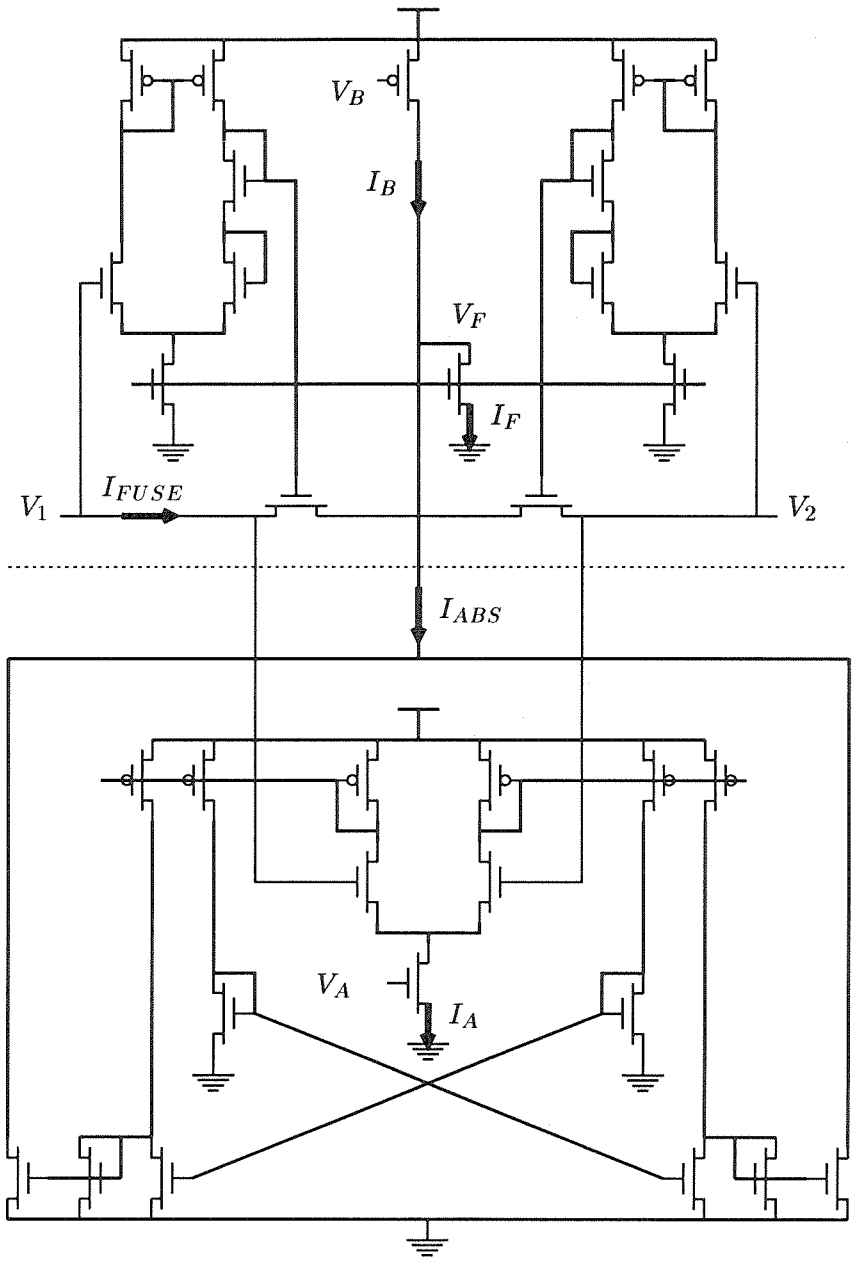


Figure 4.4: Schematic of the analog fuse circuit. The nonlinear, voltage-controlled resistance is seen across the V_1 and V_2 terminals. The circuitry above the dotted line is a saturating resistor [Mead, 1989] with V_B controlling the nominal amount of resistance. The circuit below the dotted line is a saturating absolute-value circuit that turns off the resistor for large $|V_1 - V_2|$. V_A determines the magnitude of the current drawn by the absolute-value circuit.

where $\Delta V = V_1 - V_2$. For small ΔV this portion of the circuit operates as a linear resistor with a resistance of

$$R = \frac{4kT/q}{I_F} \tag{4.6}$$

Because we are working in the subthreshold region, I_F, and thus the resistance, can be varied over five orders of magnitude. For large ΔV the resistor saturates and provides a constant current of $I_F/2$. A measured I-V curve for this circuit was shown in Figure 1.2B.

The circuit below the dotted line in the figure performs a saturating absolute-value operation. This portion of the circuit is enabled by the voltage V_A, which creates a current I_A equal to:

$$I_A = I_0 e^{\kappa V_A} \tag{4.7}$$

The positive parts of the outputs of a dual-output wide-range transconductance amplifier are combined to create a current of:

$$I_{ABS} = I_A \tanh\left(\frac{\kappa|\Delta V|}{2}\right) \tag{4.8}$$

By Kirchhoff's current law, the current I_F is:

$$I_F = \lfloor I_B - I_{ABS} \rfloor \tag{4.9}$$

where the symbols $\lfloor\ \rfloor$ are defined as

$$\begin{aligned} \lfloor x \rfloor &= x \quad \text{if} \quad x \geq 0 \\ &= 0 \quad \text{if} \quad x < 0 \end{aligned}$$

Substituting Equation 4.8 and Equation 4.9 into Equation 4.5, gives

$$I_{FUSE} = \frac{1}{2}\left\lfloor I_B - I_A \tanh\left(\frac{\kappa|\Delta V|}{2}\right)\right\rfloor \tanh\left(\frac{\Delta V}{2}\right) \tag{4.10}$$

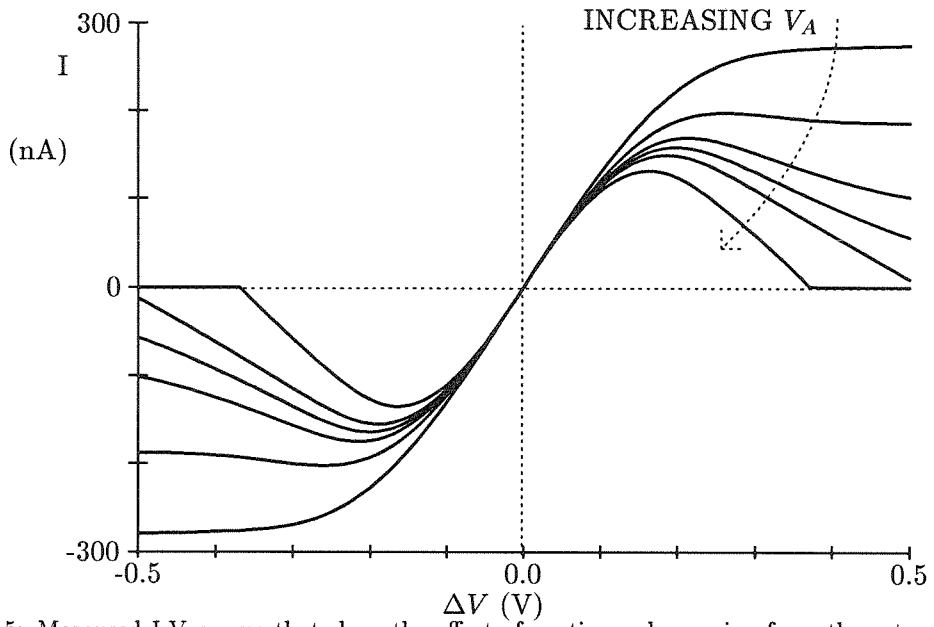


Figure 4.5: Measured I-V curves that show the effect of continuously varying from the saturating characteristic to that of the fuse curve. V_B was set to 4V and V_A was varied from 0V to 2V. When $V_A = 0$, the resulting I-V curve is identical to that of Mead's saturating resistor.

When $|\Delta V|$ is small, the fuse acts as a linear resistor whose nominal resistance is set by I_B. When $|\Delta V|$ is large, I_A increases above the current supplied by the p-type pull-up, and V_F is pulled to ground, shutting off the resistor. In between these extremes, the fuse exhibits a gradual transition.

Figure 4.5 shows a family of curves measured by varying V_A while keeping V_B constant. By varying V_A in this way, the circuit's I-V characteristic can be continuously and smoothly changed from that of a saturating resistor to the fuse I-V curve. Setting $V_A = 0$ gives $I_A = 0$, disabling the absolute-value circuit, and giving the fuse a saturating I-V relationship (Figure 1.2B).

Integration of the I-V curves in Figure 4.5 gives the family of co-content curves shown in Figure 4.6. For small ΔV the co-content is quadratic and for large ΔV the co-content saturates at a constant value. Instead of saturating for large voltage differences, the co-content of the saturating resistor increases linearly with voltage. From Chapter 1 it is known that networks of resistors with positively sloped I-V curves are guaranteed to converge to a unique minimum value of the co-content. By changing the control voltage, we are warping

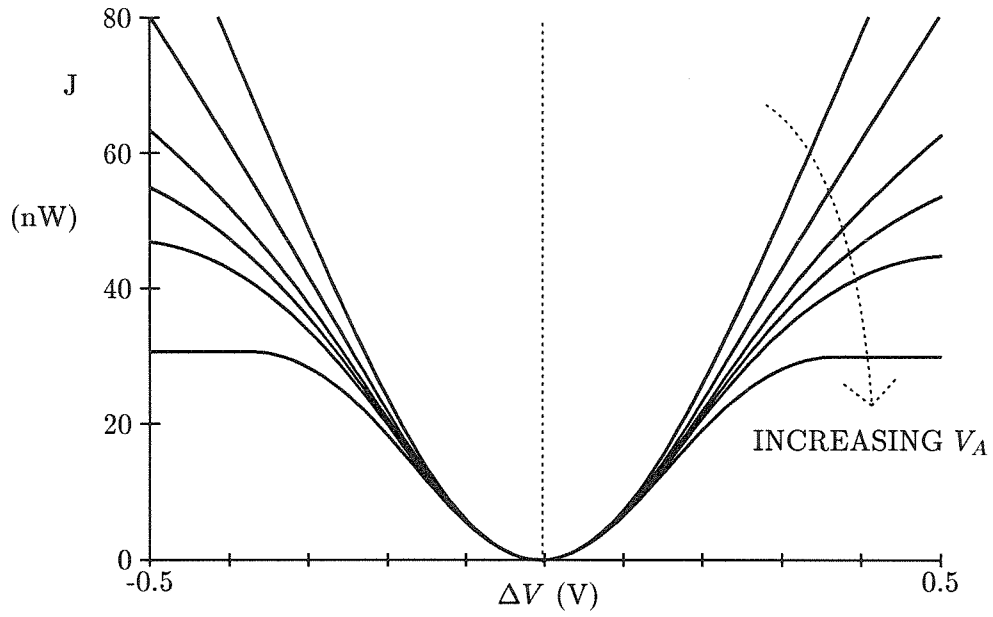


Figure 4.6: Co-content functions: each curve was numerically integrated from the family of curves in Figure 4.5. Continuously varying the co-content curves in this way performs a useful computation that is explored further in Figure 4.16 and Figure 4.17.

the energy landscape in a continuous fashion ("continuation method") from one containing a unique global minimum to one containing many local minima.

The fuse provides a mechanism for changing the threshold value. If we assume that the circuit is operating in the linear region of the two hyperbolic tangents, I_{FUSE} becomes twin parabolas of the form:

$$I_{FUSE} = \left\lfloor \frac{I_B}{4} - \kappa \frac{I_A}{8} |\Delta V| \right\rfloor \Delta V \tag{4.11}$$

This linear analysis indicates that the measured curve in Figure 4.3b consists of a parabola in each of the first and third quadrants. This current in Equation 4.11 is cut to zero for:

$$|\Delta V| \geq 2 \frac{I_B}{I_A} \frac{kT}{q\kappa} \tag{4.12}$$

I_{FUSE} reaches its extrema points at:

$$|\Delta V| = \frac{I_B}{I_A} \frac{kT}{q\kappa} \tag{4.13}$$

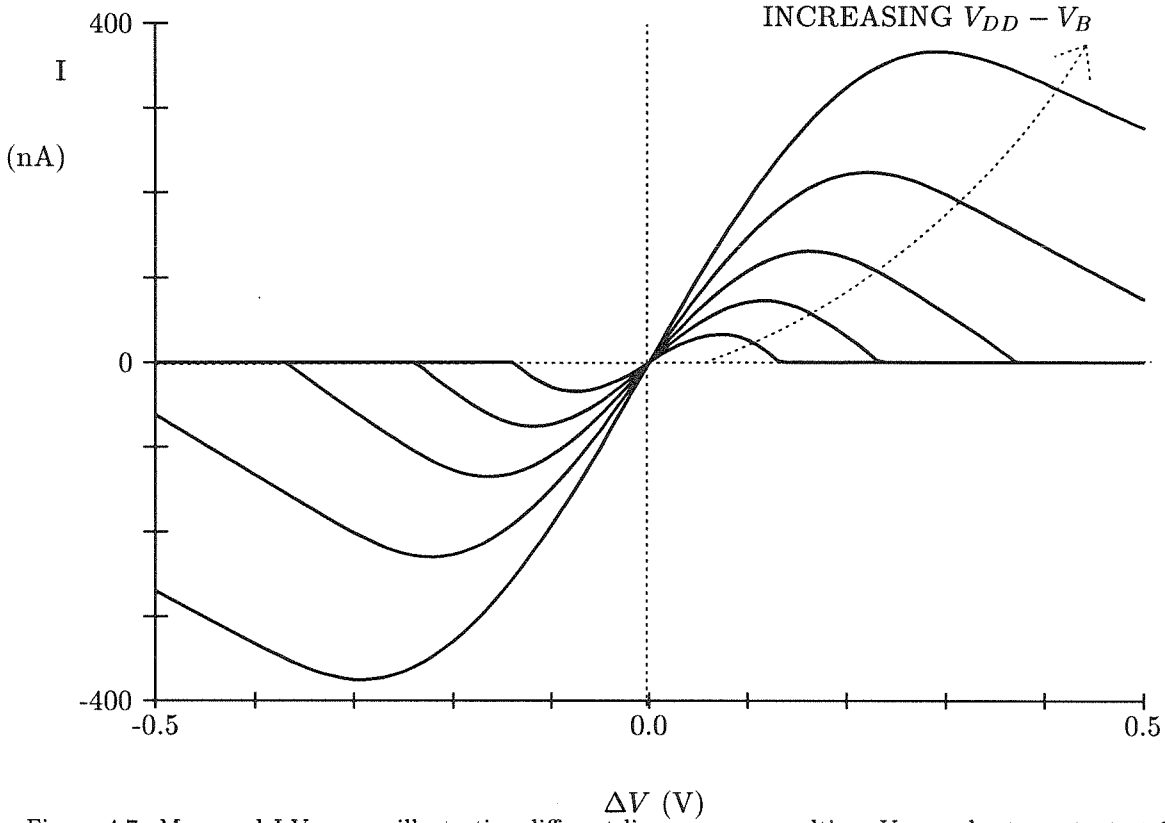


Figure 4.7: Measured I-V curves illustrating different line process penalties. V_A was kept constant at 2V and V_B was varied from 3.9V to 4.1V.

The extrema points can be set by the ratio of I_B to I_A. In subthreshold operation, the width of the saturating tanh curves is about 100mV. The extrema points can then only be varied from 0 to about ±100mV. For gate voltages above the threshold of the bias transistors, the width of the linear region of the hyperbolic tangent function increases by $V_{GS} - V_T$, where V_{GS} is the gate-to-source voltage and V_T is the threshold voltage of the bias transistors. Thus, by going slightly above threshold, the extremum point can be varied from 0 to about ±500mV. Figure 4.7 shows a family of I-V curves measured by varying V_B and holding V_A constant.

A binary fuse has been built and tested. The measured I-V curve of the binary fuse (shown in Figure 4.9) has a small incrementally active region. The schematic (shown in (Figure 4.8) is a minor modification of the analog fuse circuit. Instead of feeding the absolute-value current back to the resistor bias circuits, current is fed back to a pass gate

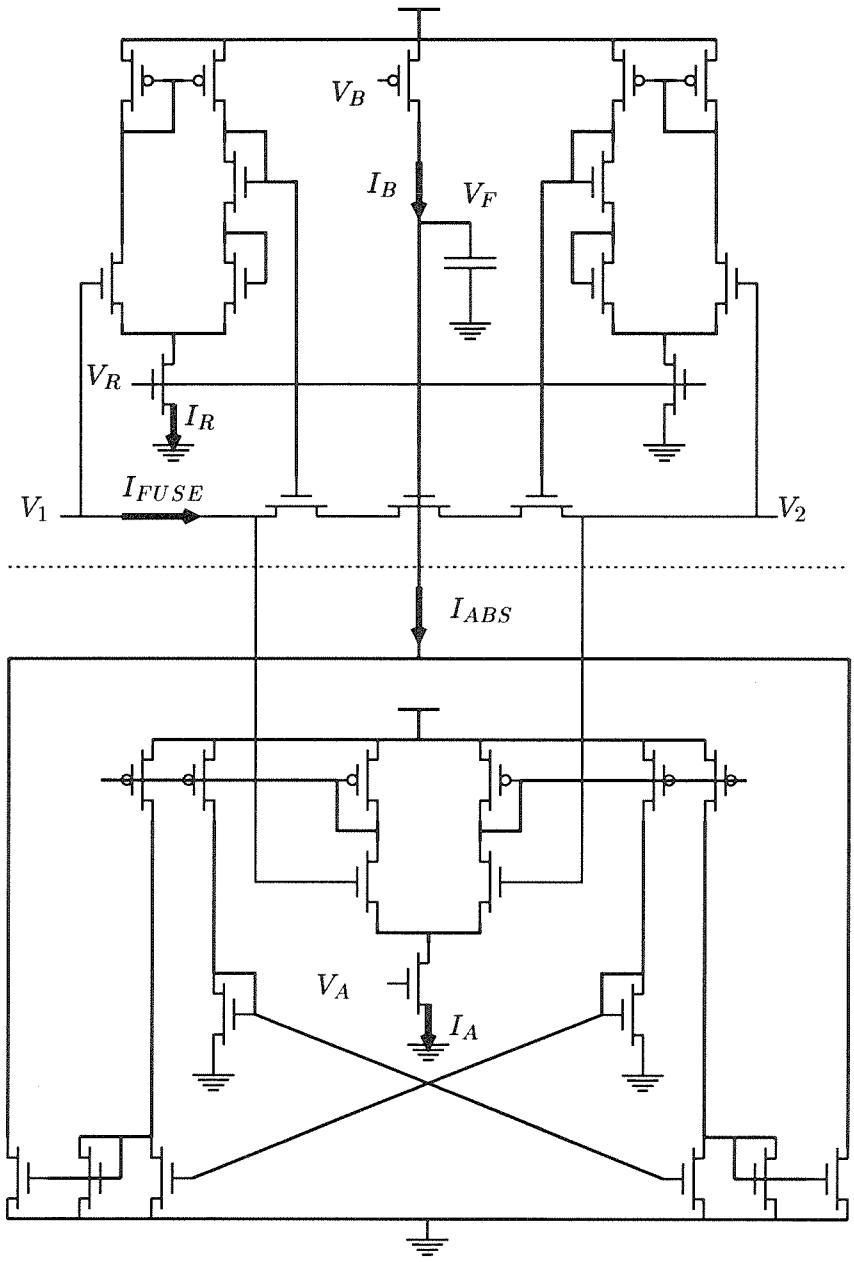


Figure 4.8: Modification of the fuse to obtain a binary characteristic. As before, a saturating resistor and an absolute-value circuit are combined to create a fuse. However, different from the circuit of Figure 4.4, the absolute-value circuit discharges the gate of a pass transistor that has been added in the resistance path. This pass gate acts as a binary switch that is opened or closed depending on whether or not the absolute-value current is greater than the threshold current provided by V_B. V_R provides independent control of the resistance of the fuse when the binary switch is closed. The parasitic capacitor at the high-gain node has been added to illustrate that the fuse has non-zero switching times.

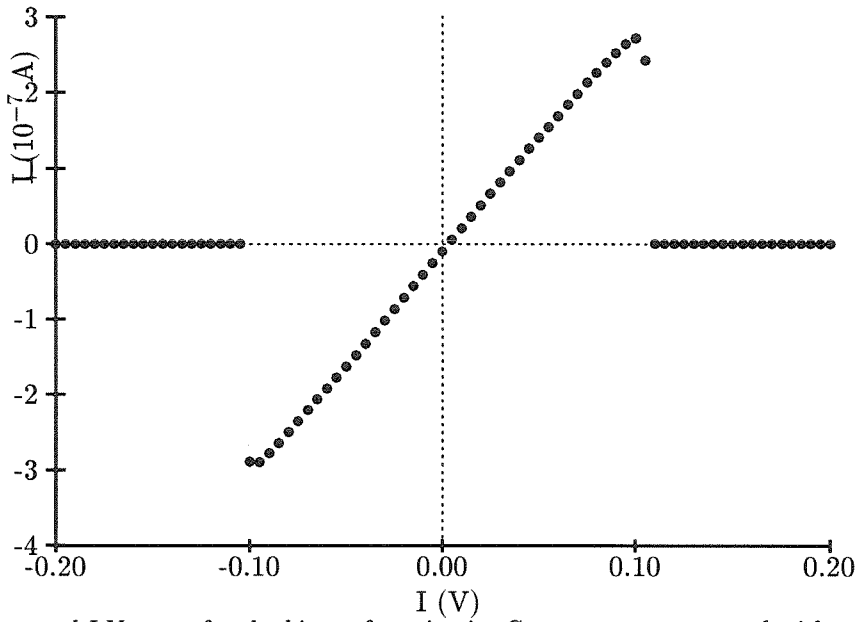


Figure 4.9: Measured I-V curve for the binary fuse circuit. Currents were measured with voltage steps of 5mV.

that acts as a binary switch in the current path. When $I_B > I_{ABS}$ the voltage (V_F) on the gate of the binary switch is charged to V_{DD}. On the other hand, when $I_B < I_{ABS}$, V_F is pulled to ground, effectively open-circuiting the resistor. The resistance of the resistor is controlled by V_R, which sets the bias current I_R. Notice that the current that controls the line process penalty is decoupled from the current that sets the resistance of the fuse. Assuming high-gain elements, the I-V equation for the high-gain fuse is given by:

$$I_{FUSE} = \frac{I_R}{2} \tanh\left(\frac{\Delta V}{2}\right) \quad \text{if} \quad I_A \tanh\left(\frac{\kappa|\Delta V|}{2}\right) < I_B \tag{4.14}$$

$$I_{FUSE} = 0 \quad \text{if} \quad I_A \tanh\left(\frac{\kappa|\Delta V|}{2}\right) \geq I_B \tag{4.15}$$

This implementation of the binary fuse shares an advantage with Mead's saturating resistor layout, because only one biasing circuit is needed for each node. This saves many transistors, especially in 2-D layouts. The low-gain fuse requires 33 transistors per connection, while the high-gain fuse requires only 21 transistors per connection plus 6 transistors per node. For a hexagonal mesh, each basic cell needs to contain one node plus half of the six

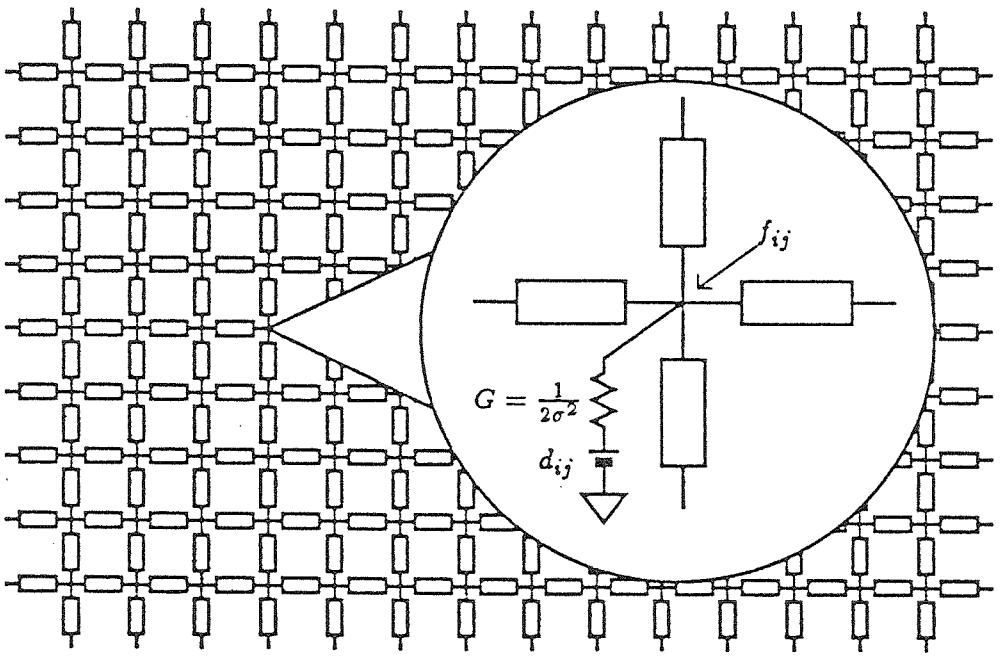


Figure 4.10: Schematic diagram of the 20 by 20 resistive fuse chip. The chip consists of a rectangular mesh of analog resistive fuse elements (shown in Figure 4.4). The data are given as battery values d_{ij} with the conductance G connecting the battery to the grid. If no data are available, $G = 0$. The output is the voltage u_{ij} at each node. Parasitic capacitances (not shown) provide the dynamics. A zero-slope boundary condition is assumed along the boundary. Data are read in/out via additional scanning circuitry (not shown).

neighboring connections, requiring a total of 69 transistors per cell for the high-gain fuse and 99 transistors per cell for the low-gain version.

4.4 Segmentation Performance

A 20 by 20 resistive fuse chip has been fabricated and tested. The analog fuse (shown in Figure 4.4) was used. Figure 4.11 shows experimental data from this chip. The input array which was scanned into the chip corresponds to a central tower on a flat plane corrupted by Gaussian noise (Figure 4.11a,d). The middle plots illustrate the resultant voltage distribution if the "fuses" are set to act as saturating resistors. The tower merges into the

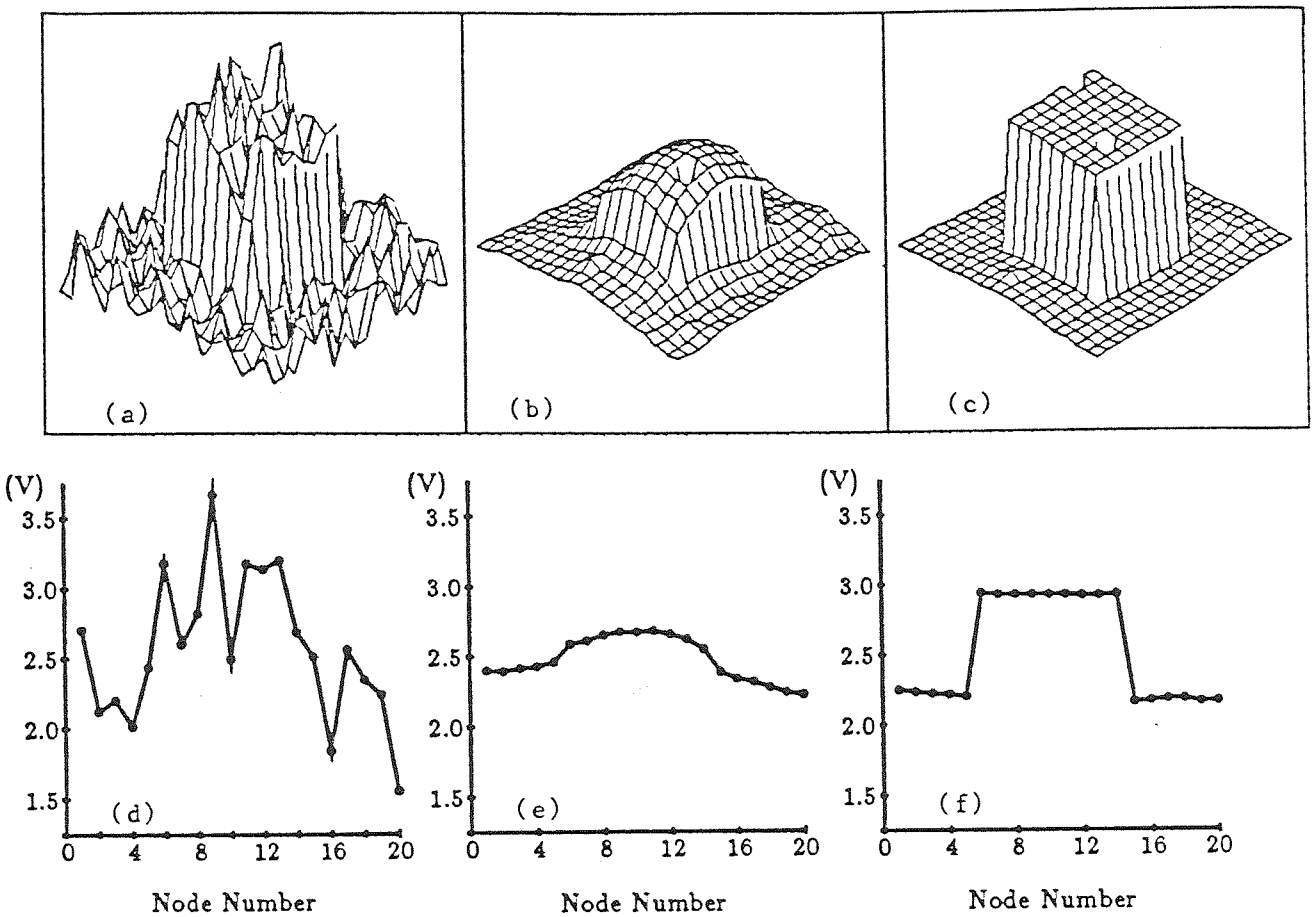


Figure 4.11: Experimental data from the chip. The scanned input data was a tower (corresponding to $d_{ij} = 3.0\ V$) rising from a plane (corresponding to 2.0 V) with superimposed Gaussian noise. **(a)** shows the input with the variance of the noise set to 0.2 V, **(b)** the voltage output using the fuse configured as a saturating resistance, and **(c)** the output when the I-V curve of the fuse has been varied from the saturating resistance to that of the analog fuse (following the arrow in Figure 4.5) as well as increasing the conductance λ. **(d)**, **(e)** and **(f)** illustrate the same behavior along a horizontal slice across the chip for $\sigma^2 = 0.4\ V$. The smoothing and segmentation abilities of the fuses are obvious. Notice that the amplitude of the noise in the last case (40% of the amplitude of the voltage step) is so large that a single filtering step on the input **(d)** will fail to detect the tower. Cooperativity and hysteresis are required for optimal performance. Notice the "bad" pixel in the middle of the tower (in **c**) which was due to a chip fabrication error. Its effect is localized to a single element.

plane, since no discontinuities prevent smoothing from occurring. The resistors around the perimeter of the tower are saturating but do not provide enough of a segmentation effect to filter the noise and preserve the step.[2] Figure 4.11c,f show that changing the I-V curve from that of a saturating resistor to that of an analog fuse enables the network to clearly segment the tower from the background. Numerical analysis as well as our empirical studies have shown that the smoothing abilities of resistive networks are robust to variations (caused by process variations) in the value of the resistances across the chip. Furthermore, point defects, such as the one shown in Figure 4.11c, induce line processes to break, thereby preventing the error from propagating.

The I-V curves of the fuses in this example have been set to the form shown in Figure 4.3b. In this configuration, the network exhibits a hysteresis property in which two stable final states are possible. The two stable states correspond to segmenting or smoothing the step edge. The segmented stable state is shown as the solid line in Figure 4.16b. The smoothed stable state becomes essentially a flat horizontal line. The final state depends on the temporal history of the network. To ensure that the proper stable state is reached in a deterministic fashion, V_A is initially set to 0 V and then gradually moved to its final value.

Figure 4.12a shows a figure-eight pattern that was scanned into the chip. The height of the signal was 0.5V with evenly distributed additive noise of $\pm$0.25V. Figure 4.12b shows the measured voltage values that were scanned off the chip. Finally, a 20x20 piece of the Lena image (shown in Figure 4.13) was scanned into the fuse chip and processed.[3] These results are shown in Figure 4.14. Various levels of segmentation behavior were obtained from many edges (top left) to no edges (bottom right).

4.5 Annealing Schedules and Continuation Methods

Though the chord resistance (i.e., V/I) of the fuse circuit is always positive, its incre-

[2] Chapter 5 describes a modification of the saturating resistor that enhances its segmentation ability.

[3] As this image has historically been used in vision research, we include it here for comparison.

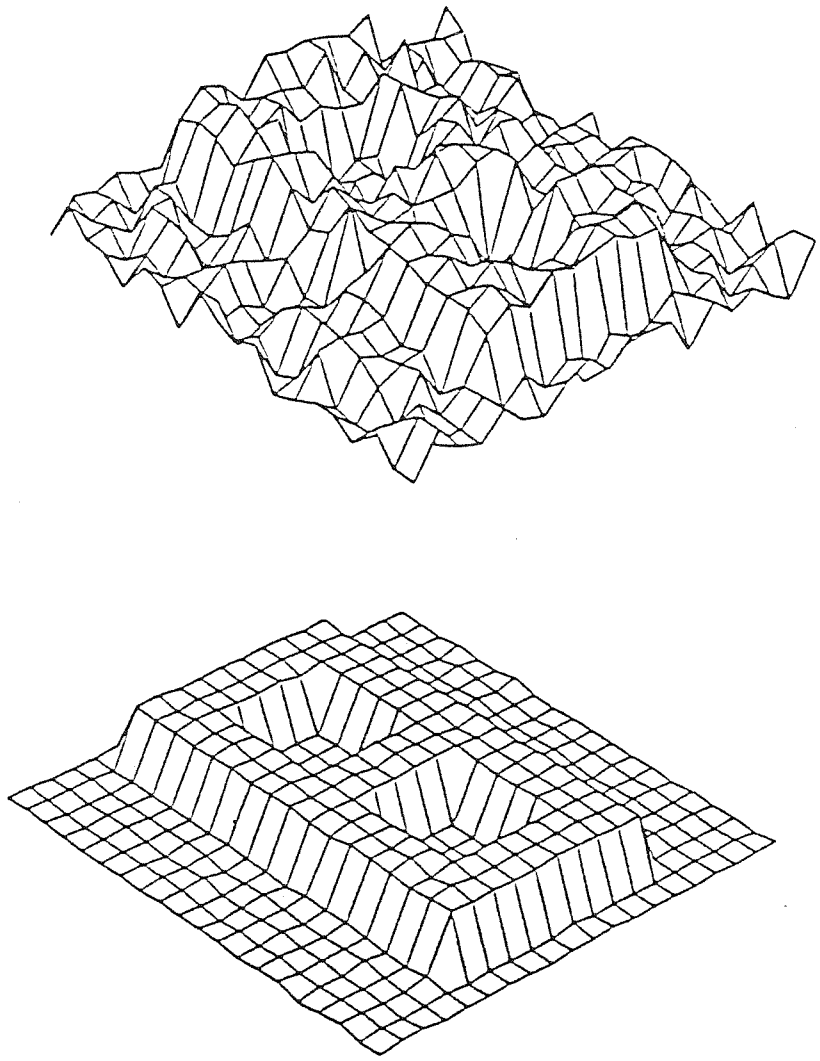


Figure 4.12: a) shows a figure-eight pattern that was scanned into the 20x20 resistive fuse chip. The height of the signal was 0.5V with evenly-distributed additive noise of ± 0.25V. b) shows the measured voltage values that were scanned off the chip.

Figure 4.13: 128x128 pixel Lena image.

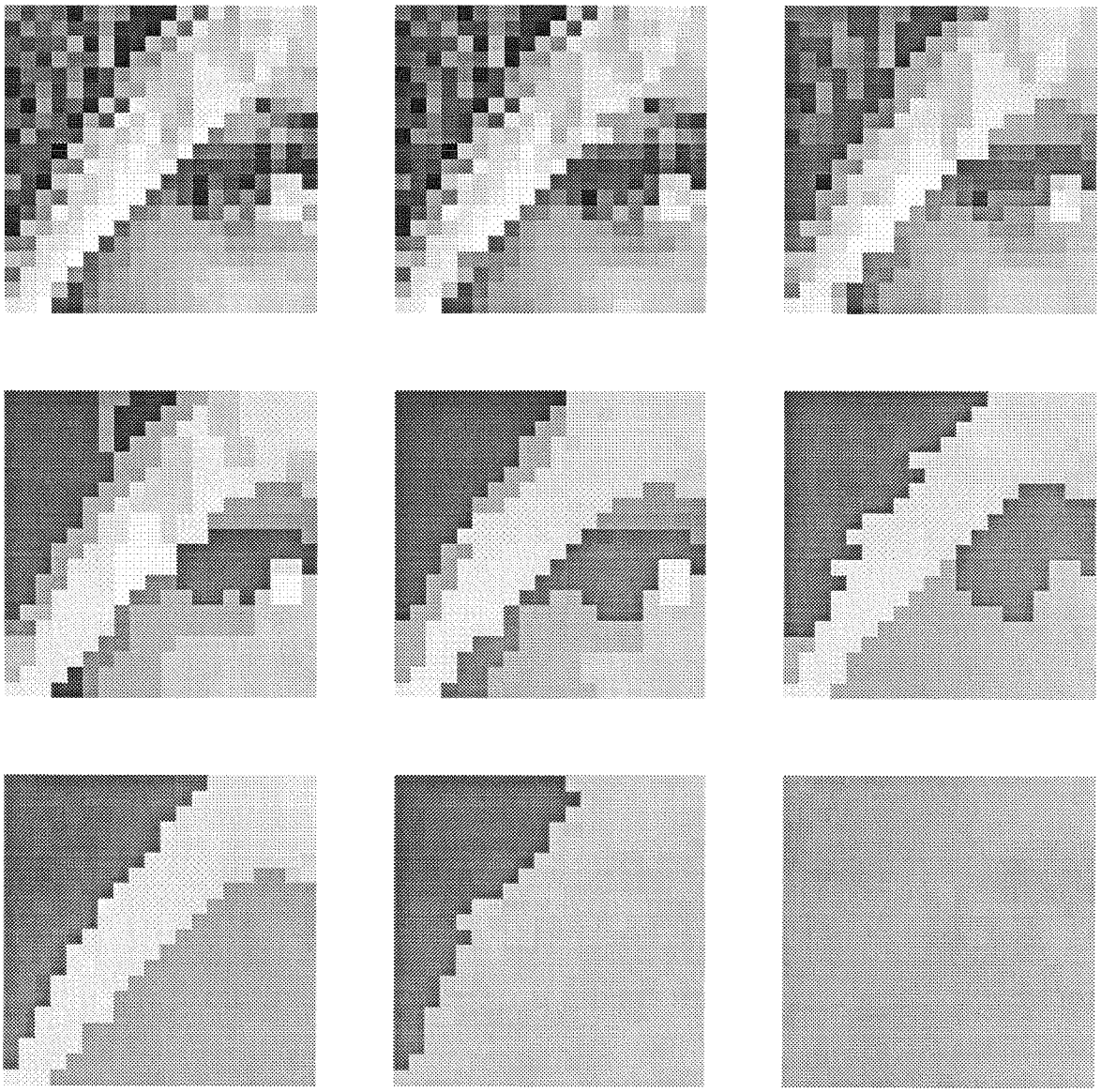


Figure 4.14: Segmented pieces of Lena image. A 20x20 patch from the image in Figure 4.13 was scanned into the resistive fuse chip. Various levels of segmentation behavior were obtained from many edges (top left) to no edges (bottom right).

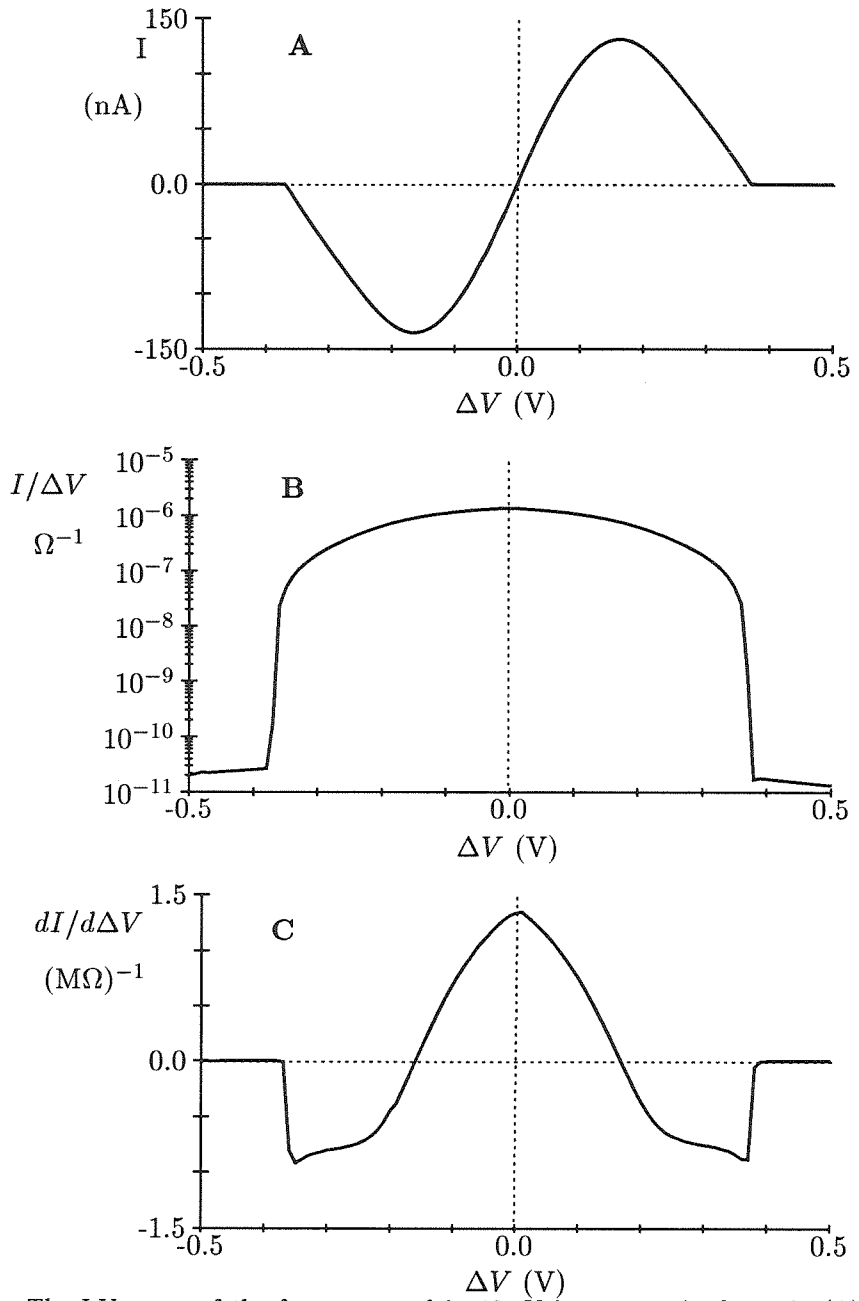


Figure 4.15: The I-V curve of the fuse measured in 10mV increments is shown in (A). (B) shows the numerically computed chord conductance, which is defined as $I/\Delta V$. Incremental conductance is defined to be $dI/d\Delta V$, which is the derivative of the I-V curve. (C) shows the incremental conductance computed using a two-point derivative approximation. Note the two regions of negative incremental conductance in (C).

mentally negative resistance regions (see Figure 4.15) raise doubts about the stability of networks of resistive fuse elements. However, as was discussed in Chapter 1, as long as the nonlinear resistors are externally *passive* (i.e., their I-V curves lie in the 1st and 3rd quadrants of the I-V plane) and if we can neglect the internal dynamics of the incrementally active resistor circuit, then for any voltage input and any initial condition, the network will *not* oscillate indefinitely but must eventually settle to *some* stationary state. This conclusion holds even if parasitic (positive) capacitances are distributed arbitrarily throughout the network, provided there are no inductors. This is a rather surprising result in view of the well-known instability problems with negative incremental resistance circuits.

As was noted in Chapter 1, the notion of minimizing the dissipated power in linear resistive networks must be replaced by the notion of minimizing the co-content (Equation 1.5) in nonlinear resistive networks. With incrementally active resistors, i.e., with regions of negative slope such as in the fuse, there will, in general, exist a number of stationary network solutions for a given input image, and uniqueness is no longer guaranteed.

The hysteresis properties of the network can be better understood through a load-line analysis of a much simplified circuit (Figure 4.16). The current through the fuse is plotted as a function of the voltage across the fuse. The simulated voltage source/resistor is also illustrated as a solid line, with the negative slope of this line given by the conductance G and the x-intercept given by the value of the voltage source E. A stability analysis reveals that the system possesses up to three equilibria. In the case illustrated in Figure 4.16A, the middle equilibrium is unstable and the voltage will tend toward the two stable solutions P1 and P2. Point P1 corresponds to segmentation, and P3 corresponds to smoothing. By increasing the value of the voltage source E (Figure 4.16B), only a single stable equilibrium point remains, corresponding to segmentation. Of course, stability cannot be guaranteed for negative values of G. The dotted-line curves show the effect of changing V_A.

Figure 4.17 shows the computed total co-content from the I-V curves shown in Figure 4.16. For Figure 4.17A, P1 is the global and P3 is only a local minimum, while P2

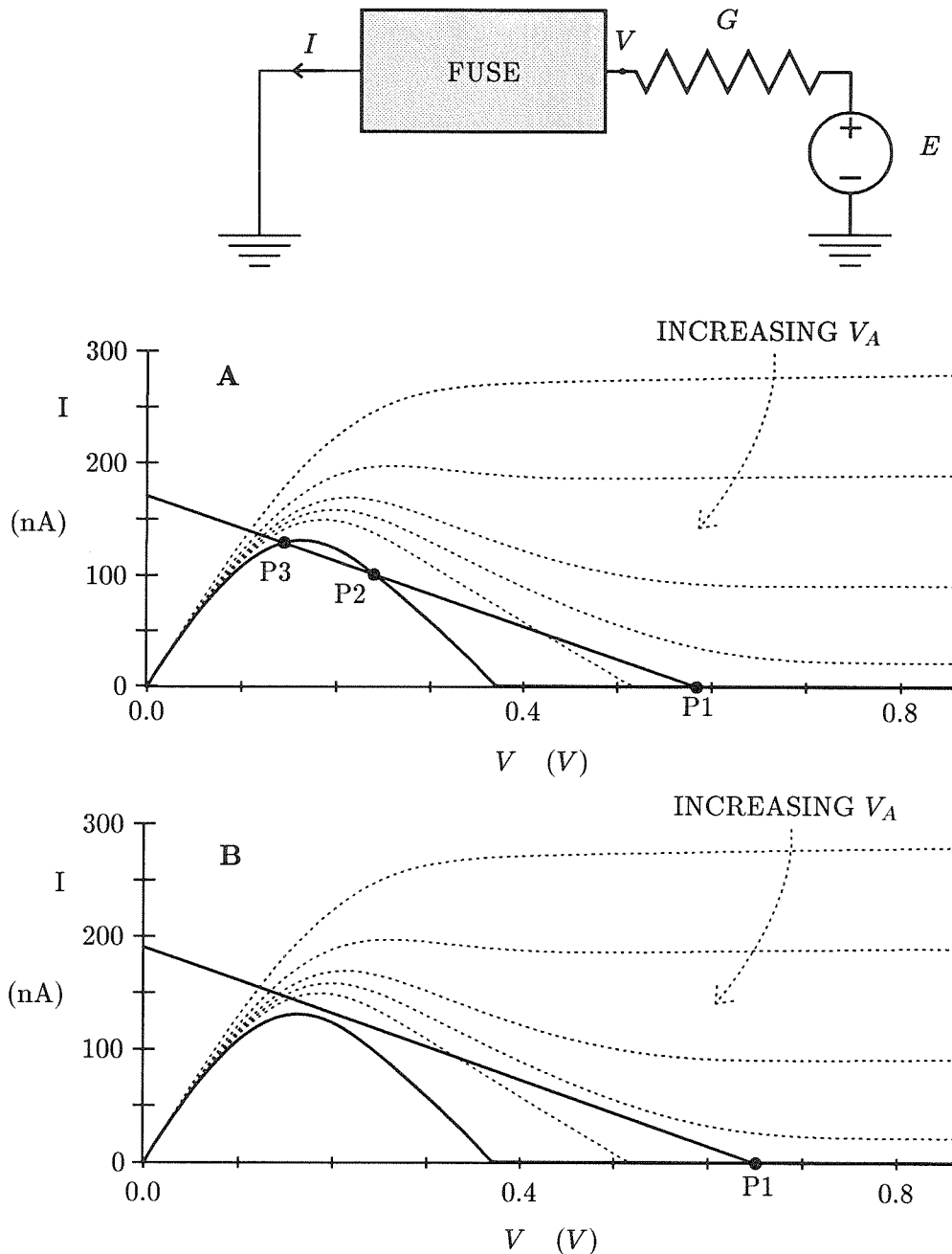
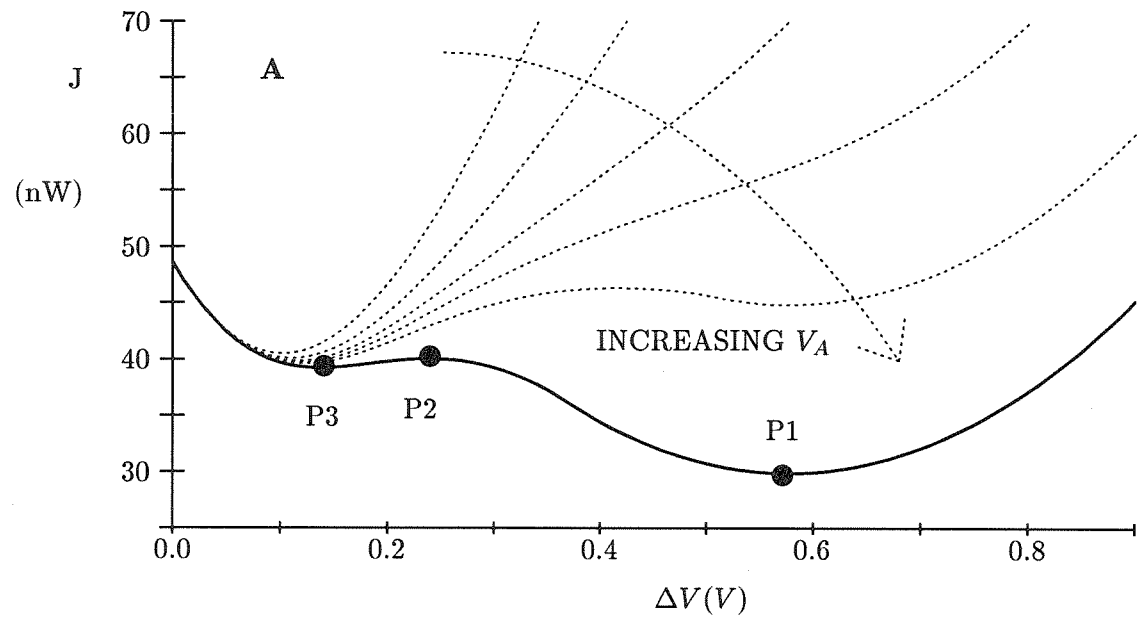


Figure 4.16: Simple load-line analysis shows that there can be up to three equilibrium points for the fuse/resistor circuit given above. The I-V curves for the measured fuse and the simulated voltage source/resistor are shown as solid lines. For plot A, points P1 and P3 are stable, and P2 is unstable. Voltages in the neighborhood of P2 will be driven to either P1 or P3. By increasing the value of the voltage source E, a single stable equilibrium point P1 remains (plot B). The dotted-line curves show the effect of changing V_A.



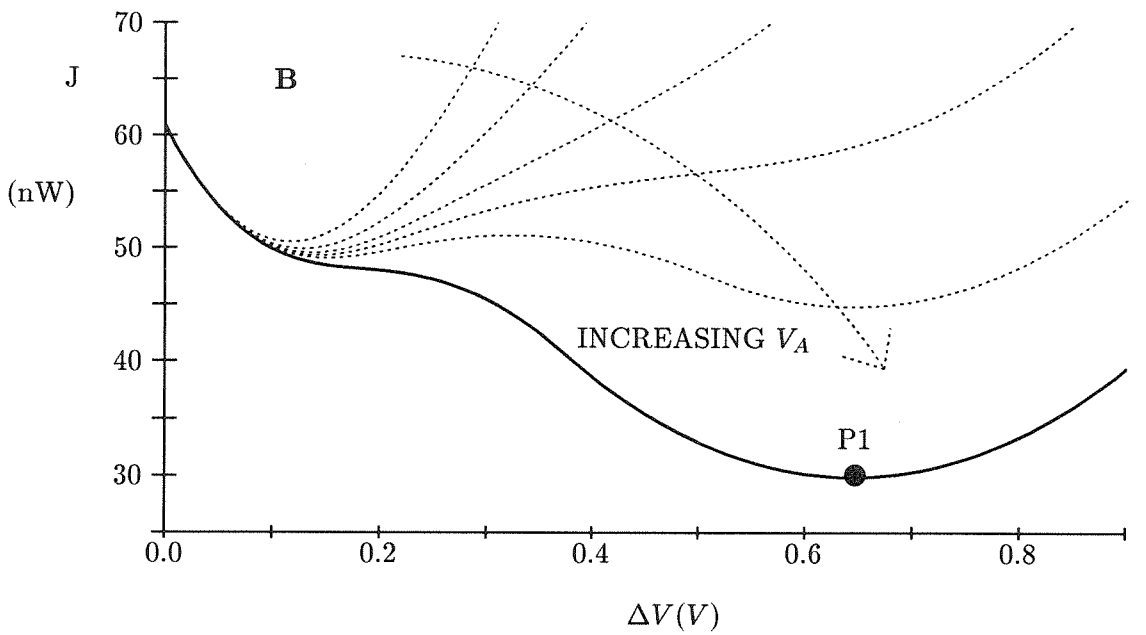


Figure 4.17: Computed total co-content J from the I-V curves shown in Figure 4.16. In plot A, P1 and P3 correspond to stable minima while P2 is an unstable maximum. In contrast, Plot B contains a single equilibrium point P1 that corresponds to a discontinuity. The dotted lines show the effect of increasing V_A.

corresponds to an unstable local maximum. In contrast, Figure 4.17B contains a single equilibrium point, P1, which corresponds to a discontinuity. The dotted lines show the effect of increasing V_A, deforming the energy surface from one with a single equilibrium point to one with two local minima. By using a continuation method in this fashion, discontinuities are deterministically located. Reasonable performance may be obtained by using a single setting of the fuse control voltages and keeping the voltages constant over time. This static approximation of the continuation method will still smooth small step edges while preserving large steps. However, medium steps, such as those simulated in Figure 4.16, can be either smoothed or segmented depending upon the temporal history of the network. This load-line analysis is a simplified version of the true dynamics of networks of fuse elements, but serves to illustrate the complexity of even a single fuse element circuit.

As seen above, mapping the necessary nonconvex energy minimization to analog VLSI does not avoid the problem of local minima. Typically, continuation methods are used that gradually warp the Lyapunov energy of the system from a convex one to the final desired energy, for example see [Ortega and Rheinboldt, 1970]. Continuation methods rely on the fact that the unique minima of the convex energy are close to the global minima of the desired energy.

Blake and Zisserman first explicitly proposed the use of an energy functional for the problem of surface interpolation and segmentation from which the line process ℓ is eliminated [Blake and Zisserman, 1987]. Its form is identical to that illustrated in Figure 4.3c. In order to find the minimum of this non-quadratic functional, Blake and Zisserman use their GNC algorithm as a continuation method. The idea behind this iterative method is to map the variational functional $J(u, \ell)$ to be minimized onto a family of functionals $J^*(u, \ell, t)$ with $t \in [0, 1]$, such that $J^*(u, \ell, t = 0)$ is given by some convex functional and $J^*(u, \ell, t = 1) = J(u, \ell)$. Instead of directly attempting to minimize the non-convex functional $J^*(u, \ell, 1)$, continuation methods involve first finding the unique solution to the convex functional $J^*(u, \ell, 0)$ and then applying some smooth transform (parameterized by

t) to continuously deform J^*. The minimum of the functional $J^*(u, \ell, t)$ is then used as a starting approximation when attempting to minimize $J^*(u, \ell, t + \Delta t)$ until the minimum to the desired functional $J^*(f, \ell, 1) = J(u, \ell)$ is reached. For the construction of J^*, Blake and Zisserman used a piece-wise polynomial of order two, whose functional dependency (for a fixed value of t) is similar to our measured co-content in Figure 4.3d.

There are three classes of continuation methods available for the hardware network we have constructed. The first is to continuously vary the I-V curve of our nonlinear element from that of a saturating resistor to that of the fuse (as shown in Figure 4.5). Another class of methods consists of decreasing the breakpoint threshold V_T of the fuse. Initially V_T is set at a high-value where no discontinuities are detected and then V_T is slowly lowered to its desired value. This method was used to generate the results shown in Figure 4.14. The final class of methods involves decreasing the conductance of the vertical confidence resistors. This method (termed λ_f annealing by Lumsdaine et al. [Lumsdaine and Wyatt, 1991]) is only applicable when there is dense and low-noise input data. Each of these methods is deterministic and guarantees that there will be one stable state at the start of the procedure. All three continuation strategies have been explored with the analog hardware and despite small differences, it does not seem to matter much which annealing strategy is used. The methods are very dependent on the nature of the problem and the type and amount of noise that is present. The chip prefers to smooth more than break, that is, when there are multiple stable states in a given configuration the network tends to jump to the smoother states but never jumps to a state with more breaks. Intuitively, it seems analog fuses should be preferred over binary fuses. Our intuition is that the smoother I-V curve of the analog fuse will provide a smoother energy surface and, therefore, the gradient descent will not tend to get stuck in bad local minima as easily. This conjecture has yet to be verified, either analytically or empirically.

In order for a fully automated system to implement any of these annealing strategies, elaborate clocking circuitry would be necessary to vary an on-chip voltage through a range

of values. Furthermore, since our analog technology naturally implements continuous-time systems with on-chip photoreceptors, it would be awkward to resort to a sampled-time system. This problem is especially evident when we try to recover the optical flow in the presence of motion discontinuities. Motion already contains a natural time scale that we would have to ensure is much longer than the time scale of our continuation methods. One alternative is to maintain a constant setting of the fuse circuit. The resulting network would still have local minima problems but would segment for large edges. Small moving edges in the input would tend to get stuck in one place in the output. Alternatively, the gradient of the original data could be used to set the resistance of the fuses. This network would perform well if one could guarantee that no noise point was farther away from its neighboring values than the minimum detectable edge strength. This continuous time network would have limited usefulness and would not extend to sparse depth and motion segmentation problems. Chapter 5 discusses a different segmentation network that operates in continuous time without requiring annealing methods of any kind.

4.6 Alternate Hardware Implementations

Since the fuse circuits described above were developed, there have been a number of different approaches and technologies used to build a better hardware implementation. Barman has designed a novel nonlinear externally-controlled switched-capacitor element [Barman, 1990]. Using this architecture, a network of resistors with any arbitrary I-V characteristic could be simulated. Workers at MIT have successfully fabricated a four-transistor resistive fuse using zero-threshold n-channel and zero-threshold p-channel CMOS transistors [Decker et al., 1991]. A resistive fuse that has regions in which it becomes a true negative resistor is useful for edge enhancement. An analog CMOS version of this *negative fuse* has been demonstrated [Liu and Harris, 1991].

Independently, Perona and Malik simulated diffusion networks of fuse-like resistors for segmentation of images [Perona and Malik, 1990]. The effective conductance of the resistors

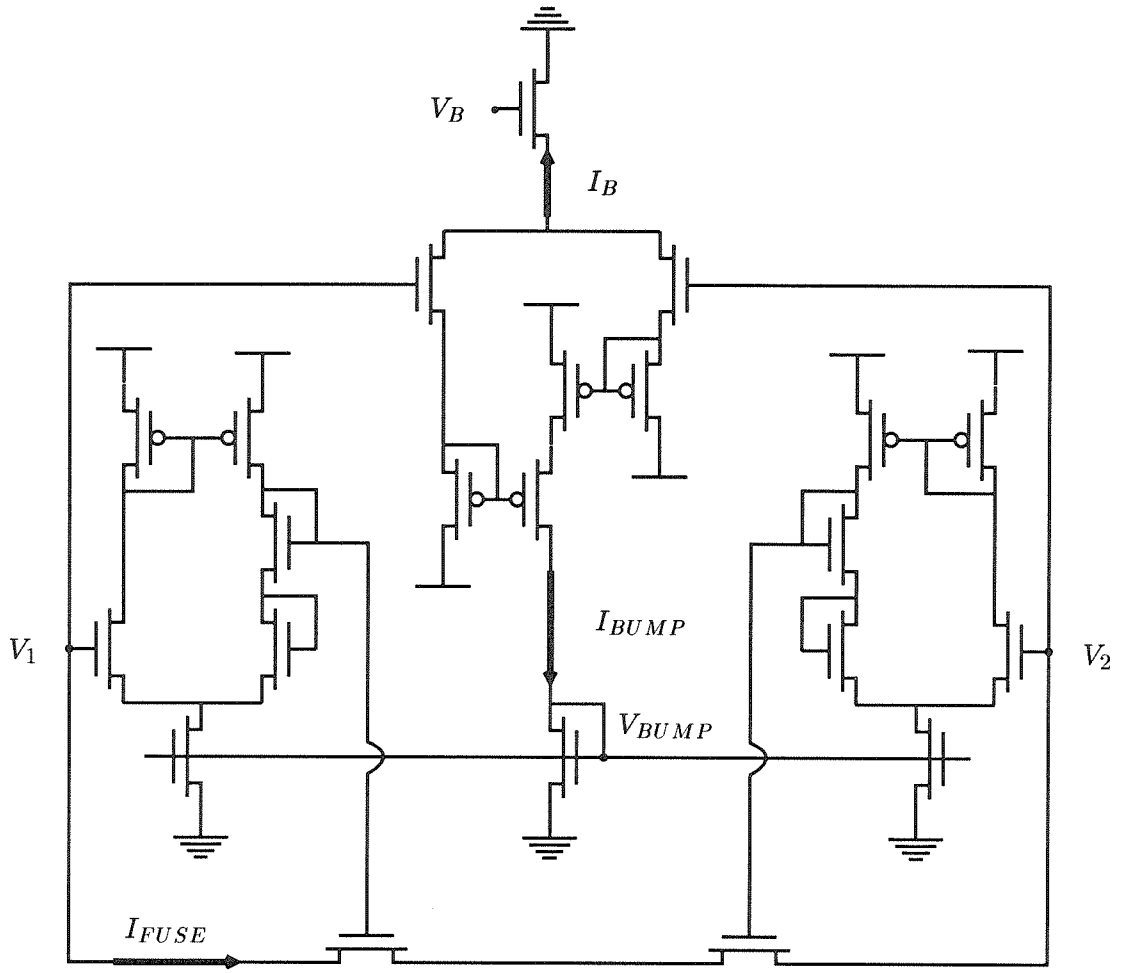


Figure 4.18: Bump fuse circuit.

is made to vary inversely with the square of the voltage drop. Each node voltage is initialized to the value of the noisy input data corresponding to that pixel. Ultimately, the network voltages converge to a single constant voltage. However, a human operator can stop the network at a certain point and, by running a simple postprocessing step, create a very impressive edge map. A hardware implementation of this method would be challenging because of the timing mechanisms and human control that would be required. Nordström has further refined these anisotropic diffusion methods by simulating diffusion networks with resistive constraint terms [Nordström, 1990]. His method, called biased anisotropic diffusion, is essentially a resistive fuse network.

Probably the most compact adjustable fuse design implemented so far is based on

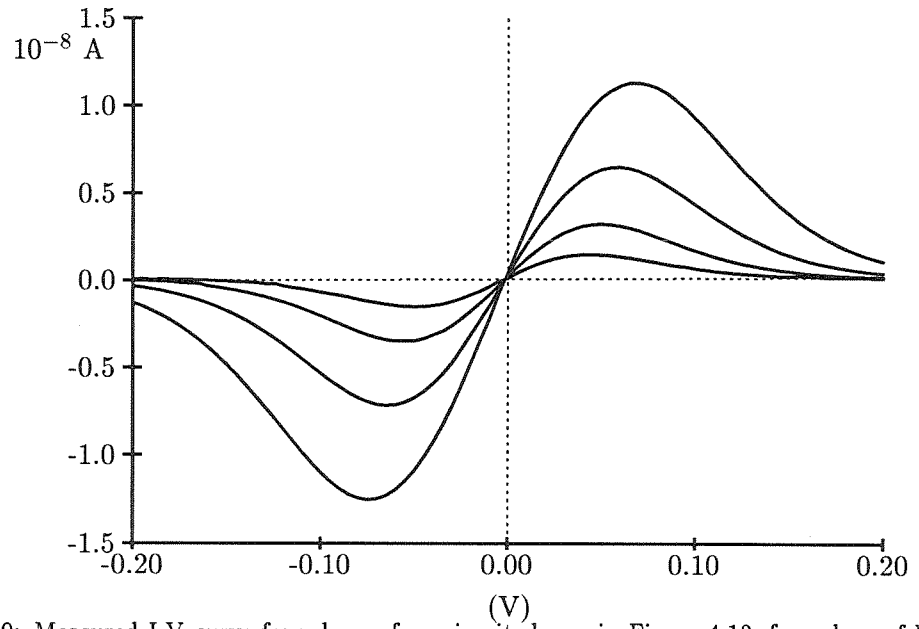


Figure 4.19: Measured I-V curve from bump-fuse circuit shown in Figure 4.18, for values of V_B = 0.80V, 0.85V, 0.90V, and 0.95V.

the "bump" circuit. The bump circuit, developed by Tobi Delbrück at Caltech, outputs a current that is inversely dependent on the magnitude of the input voltage difference. [Delbrück, 1991a] [Delbrück, 1991b]. Figure 4.18 shows an analog fuse schematic which uses the bump circuit to generate a bias current for Mead's saturating resistor. A fuse-like I-V characteristic is measured between the V_1 and V_2 terminals. In subthreshold operation I_{BUMP} is described by:

$$I_{BUMP} = \frac{I_B/4}{\cosh^2(\frac{\kappa(V_1-V_2)}{2})} \tag{4.16}$$

which reaches a maximum of $I_B/4$ when $V_1 = V_2$ and decays to zero as $|V_1 - V_2|$ increases. It is remarkable that a symmetric function can be created with such an asymmetric circuit. The effective width of the bump curve can be increased by running the circuit above threshold.[4] This current is mirrored and used as the bias current for Mead's saturating

[4]The symmetric above-threshold version of this circuit requires two additional transistors.

resistor. The bump-fuse I-V characteristic is therefore:

$$I_{FUSE} = \frac{I_B/8}{\cosh^2(\frac{\kappa(V_1-V_2)}{2})} \tanh(\frac{V_1 - V_2}{2}) \tag{4.17}$$

The resulting I-V curves are shown in Figure 4.19. The width of the linear region of the bump fuse can be increased by driving the circuit above threshold. A binary fuse can be constructed by using a thresholded bump circuit to control a pass gate in a similar manner to the binary fuse shown in Figure 4.8.

4.7 Constraint Fuses

A natural generalization of this thesis arises when we consider that constraint boxes are generalized resistors and that resistive fuses are resistors that break. Why not allow constraint boxes to break? This section discusses this idea of "constraint fuses". A special case of the constraint fuse has resulted in a network which discards outliers in noisy and possibly sparse data [Harris et al., 1991]. A resistive fuse is used as the data constraint term in a resistive grid network. This effectively isolates an input point from the rest of the network when the input point differs significantly from the neighborhood average. This idea has many similarities to the field of robust statistics [Huber, 1981] except that the algorithm presented here has a simple elegant embodiment in analog real-time VLSI hardware.

The need for robust estimation techniques in early vision processing has been clearly stated [Schunck, 1989]. The assumption of an additive Gaussian noise model leads to least squares methods which are frequently employed in regularization type networks. Outliers occur more often in real-world situations than the pure Gaussian noise model allows. For example, false stereo correspondence matches produce spurious depth estimates which can be eliminated through the use of robust statistical methods. Without outlier elimination, least squares surface reconstruction algorithms could potentially produce wildly inaccurate solutions. The recent workshop on robust methods in computer vision included papers on

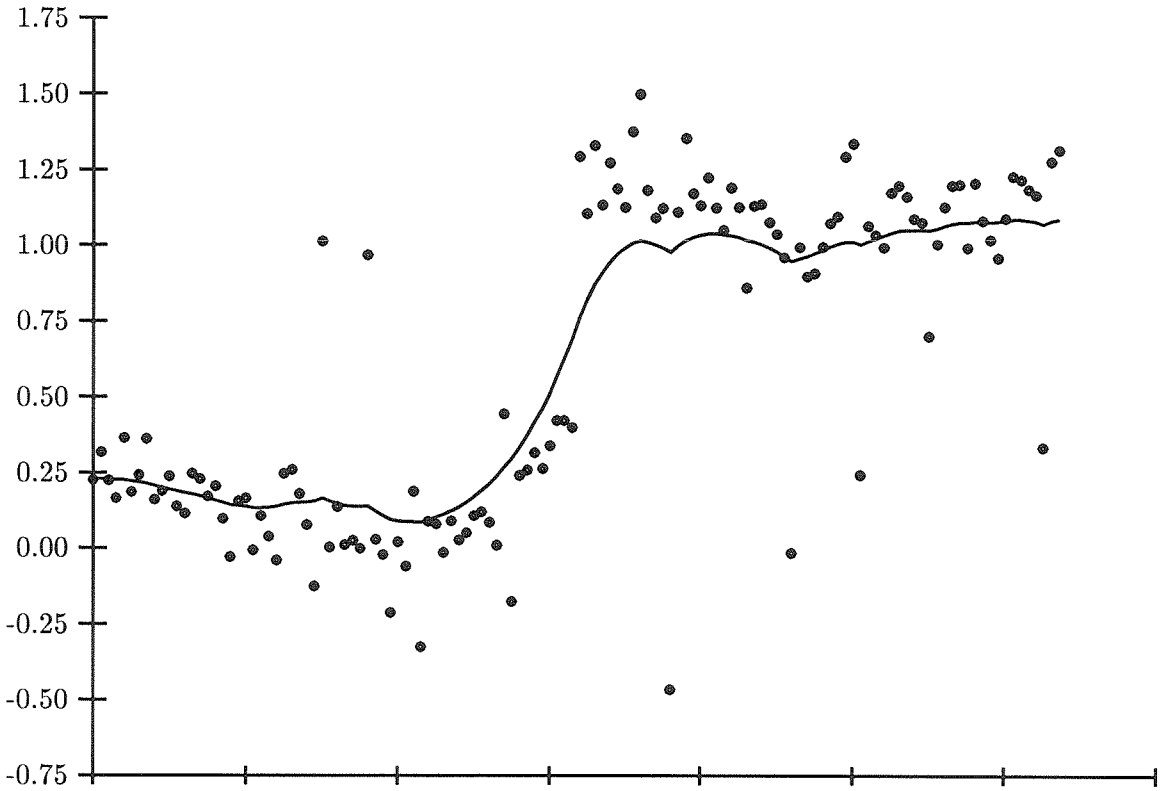


Figure 4.20: Standard regularization solution. Solid circles show the input data. Line is solution for $\lambda = 80$.

robust methods for stereo, motion, surface interpolation, edge detection and shape from shading.

Robust estimation was pioneered and formalized by Huber [Huber, 1981], however, outlier rejection techniques have been used in an ad hoc fashion by many in the past centuries.[5] Perhaps, one of the first was Daniel Bernoulli in 1769 [Hampel et al., 1986]. Statistical methods fail when prior assumptions about a situation are incorrect. Huber defines robust methods as "insensitivity to small deviations from the assumptions." Figure 4.20 shows an example of noisy outlier data. The data consists of two rounded segments separated by a step jump of one unit. Gaussian distributed noise with $\sigma = .1$ was added and 10% of the data points were replaced by outlier points, evenly distributed between -0.5 and 1.5. A standard regularization network with $\lambda = 80$ was simulated on the data. The shape of

[5]These ideas are also related to theories of outliers in networks for approximation and learning [Girosi et al., 1990].

resulting curve in Figure 4.20 clearly reflects the presence of the outliers as well as the step discontinuity.

To deal with outliers and discontinuities, we minimize:

$$E(u,\ell,m)=\sum_i\left[(d_i-u_i)^2(1-m_i)+\lambda(u_{i+1}-u_i)^2(1-\ell_i)+\alpha^2\ell_i+\beta^2 m_i\right] \tag{4.18}$$

where λ sets the length constant of the network, α^2 is the cost of breaking a line discontinuity, and β^2 is the cost of breaking an outlier. The first term forces the surface u to be close to the measured data d. The second term enforces the piecewise smoothness constraint. Assuming that $m_i = 0$ everywhere reduces Equation 4.18 to the familiar line process energy given in Equation 4.1. If all variables, with the exception of u_i, u_{i+1}, and ℓ_i, are held fixed and $\lambda(u_{i+1}-u_i)^2 < \alpha^2$, then it is cost effective to pay the price $\lambda(u_{i+1}-u_i)^2$ and to set $\ell_i = 0$ instead of paying the larger price α^2. If the gradient becomes too steep, $\ell_i = 1$, and the surface is segmented at that location. ℓ_i represents the line process variable in the functional above. When $\ell_i = 1$ a discontinuity has been detected at the ith pixel ($\ell_i = 0$ otherwise).

The novel contribution of this algorithm is to provide the m_i terms which allow for rejection of outliers. Suppose $(u_i - d_i)^2 < \beta^2$. It is "cheaper" to pay the price $(u_i - d_i)^2$ and to set $m_i = 0$ then to pay the larger price β^2. If on the other hand $(u_i - d_i)^2 > \beta^2$ then it will be cheaper to pay the price β^2 and set $m_i = 1$. When this occurs, d_i is isolated from the network and plays no part in the final solution.

We solve the nonlinear energy minimization problem given above by mapping the functional onto the resistive network shown in Figure 4.21. A voltage source of d_i is applied to every node and the stationary voltage is the solution u_i. The nominal conductance between the voltage source and the grid is assumed to be 1. In the absence of any discontinuities (all $\ell_i = 0$) or outliers (all $m_i = 0$), smoothness is implemented via a conductance of value λ connecting neighboring grid points; that is, the nonlinear resistors in Figure 4.21 can simply be considered linear resistors. The cost functional E can then be interpreted as the

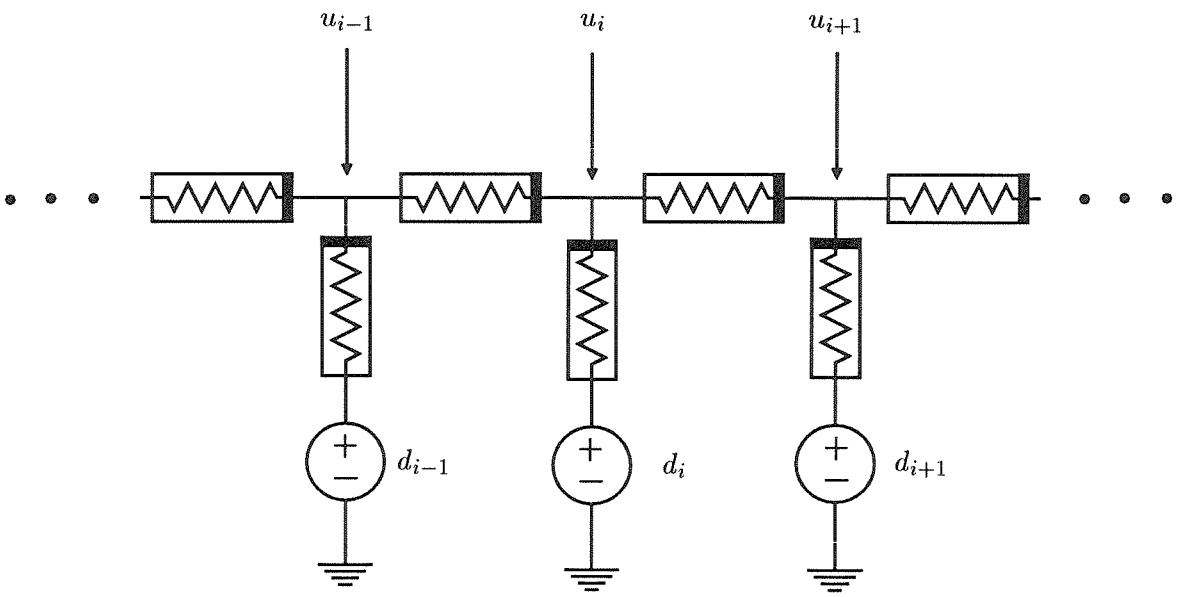


Figure 4.21: Nonlinear resistor network for outlier detection.

power dissipated by the circuit. If parasitic capacitances are added to the circuit, E acts as a Lyapunov function of the system and the stationary voltage distribution corresponds to the smooth surface. Intuitively, the idea is to break one of the horizontal resistors ($\ell_i = 1$) wherever a discontinuity occurs and to break one of the vertical resistors ($m_i = 1$) wherever an outlier occurs. The linear resistors are replaced by the resistive fuse, to implement piecewise smoothness and outlier detection. A simulation of this network on noisy outlier data is shown in Figure 4.22.

Images collected from man-made sensors like imaging arrays, millimeter wave radar and laser radar suffer from problems such as "glint" and missing data points. This algorithm has been successfully demonstrated on laser radar images for segmentation and outlier detection [Harris et al., 1991]. Alternative noise reduction methods would require either averaging over multiple frames or substantial spatial smoothing. Temporal integration results in blurring of moving targets and spatial smoothing techniques result in blurring of

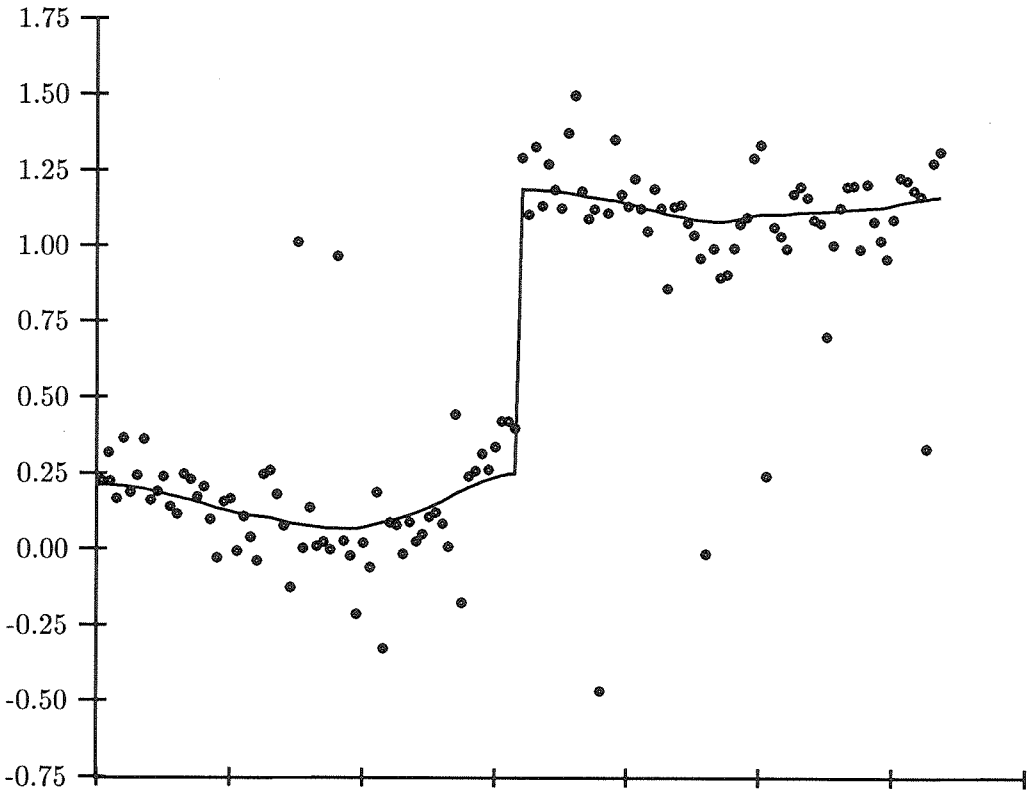


Figure 4.22: Solution for outlier network simulation. Solid circles show the input data. Line is solution for $\lambda = 80$, $\alpha^2 = 0.05$, and $\beta^2 = 0.02$.

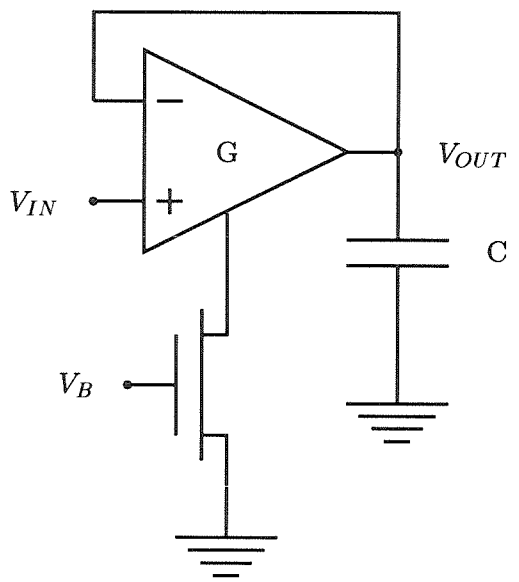


Figure 4.23: Standard follower-integrator circuit from [Mead, 1989].

edges.

Robust statistical methods are useful in computer vision but they are extremely computational intensive. We have demonstrated a network that incorporates robustness in its computation but can potentially converge to a solution in a very small time constant. The network is a modification of a resistive network to provide for detection and removal of outliers in image segmentation. We expect this method to extend to most other vision problems which can be solved using a resistive network such as surface interpolation and Horn and Schunck's [Horn and Schunck, 1981] motion computation.

4.8 Time Fuse

This section discusses a nonlinear follower-integrator circuit for segmenting temporal sequences in time. This is useful for smoothing the noise at a given pixel over time without reducing the spatial resolution of an image by smoothing in space. This can be accomplished somewhat with the standard follower-integrator circuit shown in Figure 4.23 [Mead, 1989]. The voltage V_B sets the conductance of the amplifier. When the follower is in its linear region of operation, the follower integrator operates as a simple RC circuit. The differential

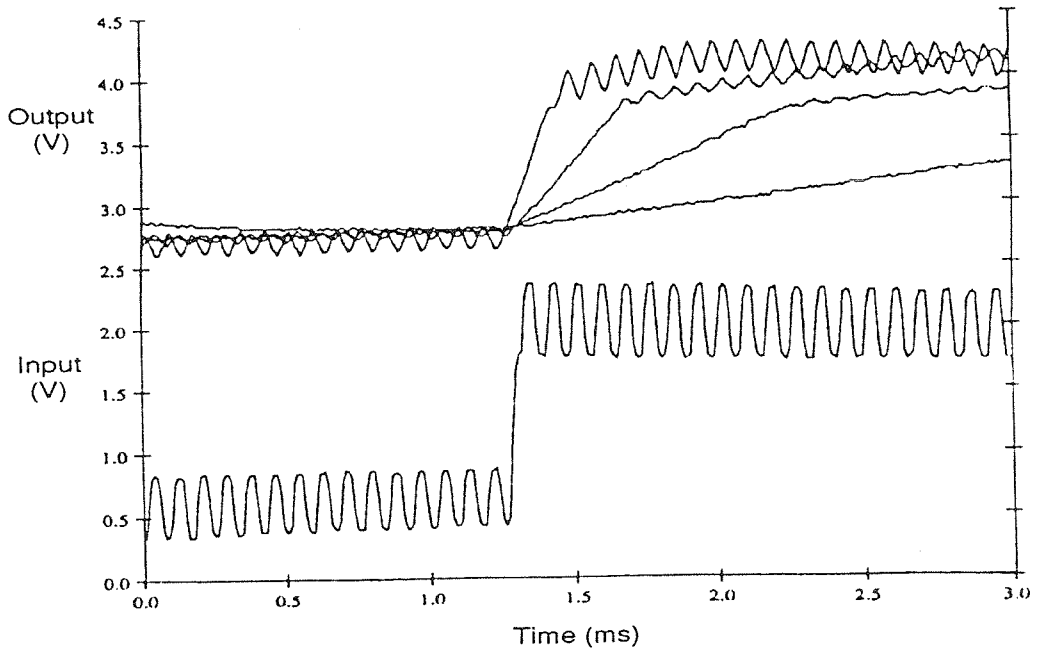


Figure 4.24: Measured follower-integrator results. Bottom plot shows step input with added high frequency sinusoid. Bottom shows output of the follower integrator for V_G = 0.6V, 0.65V, 0.7V, and 0.75V. No single setting of V_G satisfies the conflicting constraints of smoothing the sinusoid while preserving the step edge.

equation for node V_{OUT} is:

$$C\frac{dV_{OUT}}{dt} = G(V_{IN} - V_{OUT}) \tag{4.19}$$

If we assume $V_{IN} = h$ for $t < 0$ and $V_{IN} = 0$ for $t \geq 0$ the output is a simple exponential decay:

$$V_{OUT} = he^{-t/\tau} \tag{4.20}$$

where $\tau = C/G$.

This circuit integrates the the voltage input signal in time and can successfully filter out noise in the signal by the setting the τ of the circuit to be large. A problem arises, however, when either the camera or objects in the image are moving. In this case, instead of blurring the quickly changing input by integrating with a large time constant, we would rather follow the input signal very closely. Suppose we are presented with the signal shown

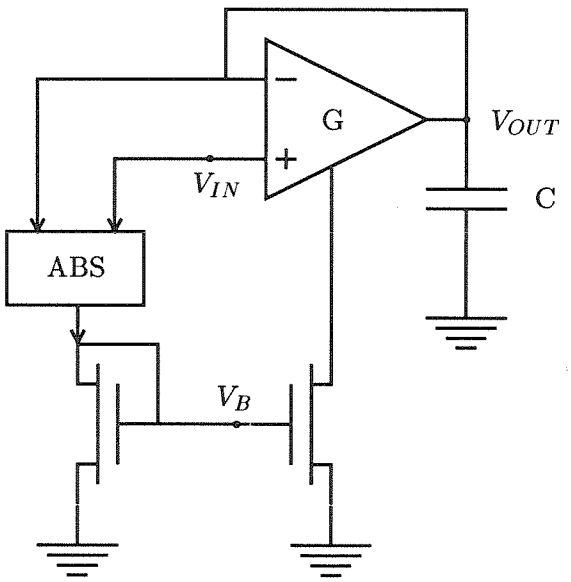


Figure 4.25: The time-fuse circuit. The follower integrator circuit is modified such that the conductance of the follower is proportional to $|V_{OUT} - V_{IN}|$.

in Figure 4.24. The input is a high frequency sinusoid superimposed on a step input. The goal is to smooth out the noise (i.e, the high frequency sinusoid) while preserving the step edge. This is similar to the segmentation problem discussed in this thesis, except that the signal is a function of time instead of space. As shown in Figure 4.24, no setting of G for the follower integrator is satisfactory and we have to trade off between smoothing the step versus keeping some of the noise in the output.

An alternative approach is to use a nonlinear follower-integrator called the "time fuse". The circuit shown in Figure 4.26 uses a follower whose conductance is proportional to the absolute-value of the voltage drop across it. In other words, the conductance of the amplifier is

$$G = G_0 \frac{|V_{IN} - V_{OUT}|}{V_0} \tag{4.21}$$

where G_0/V_0 is the constant of proportionality. The conductance change is in the opposite direction of the change needed for the standard spatial fuse implementation. The conductance of the spatial fuse must decreases as a function of voltage drop across it. The absolute-value circuit used is essentially identical to the design used in Chapter 4 but instead

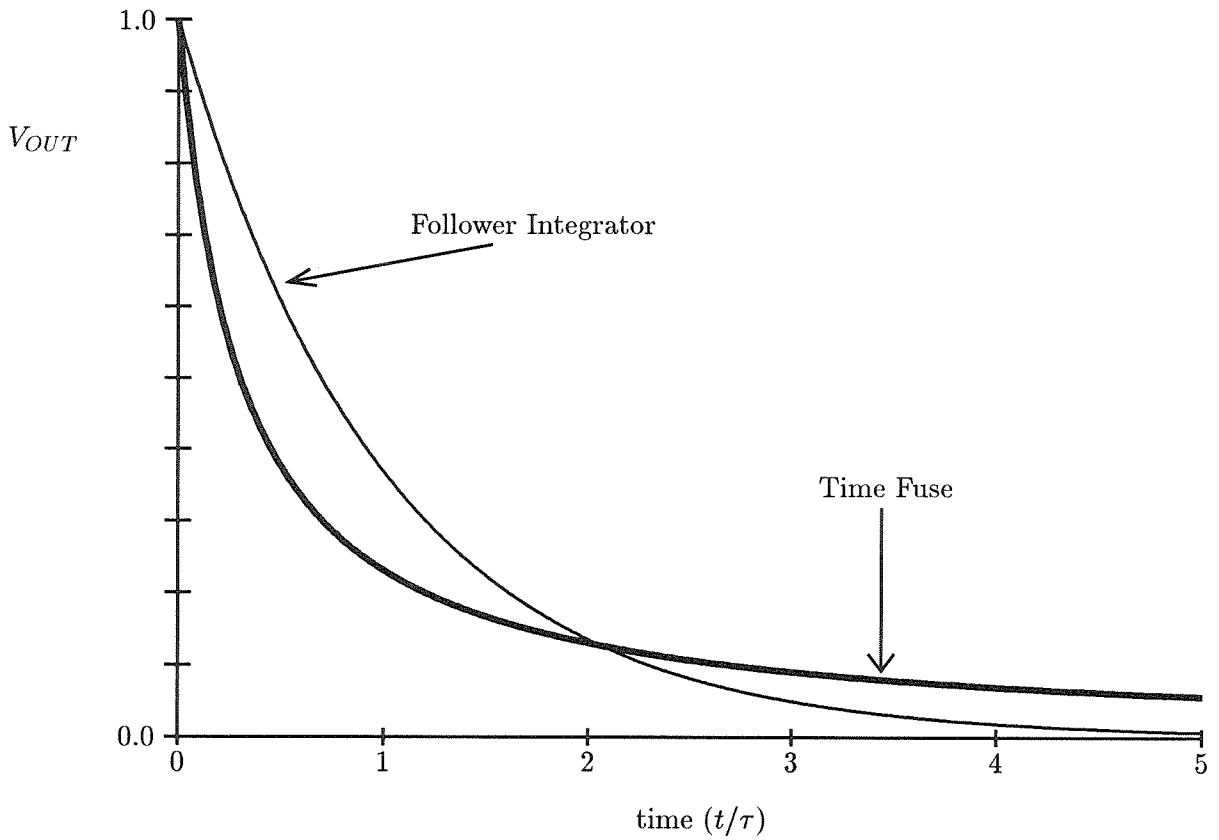


Figure 4.26: Plot of the step response of the follower integrator (Equation 4.20) and the time fuse (Equation 4.23).

of subtracting from the bias current, the absolute-value circuit supplies the bias current. The differential equation for V_{OUT} for the time fuse is:

$$C\frac{dV_{OUT}}{dt} = G(V_{IN} - V_{OUT})\frac{|V_{IN} - V_{OUT}|}{V_0} \tag{4.22}$$

Integrating this equation gives:

$$V_{OUT} = \frac{h}{\frac{t}{\tau}\frac{h}{V_0} + 1} \tag{4.23}$$

Since this is a nonlinear circuit, the shape of the step response depends on the magnitude of the step input. The step responses of the two circuits are compared in Figure 4.26. For large V_{OUT}, i.e. significant changes, the output follows with a fast time constant. For small V_{OUT}, presumably high frequency noise, a long integration time results. The same input given to the standard follower-integrator in Figure 4.24 was given to the time fuse circuit. The results, presented in Figure 4.27, show that the time fuse can preserve temporal edges while suppressing noise better than the follower-integrator circuit. Note that the saturation of the amplifier limits the segmentation performance. For the spatial resistive fuse, on the other hand, saturating devices enhance the segmentation properties.

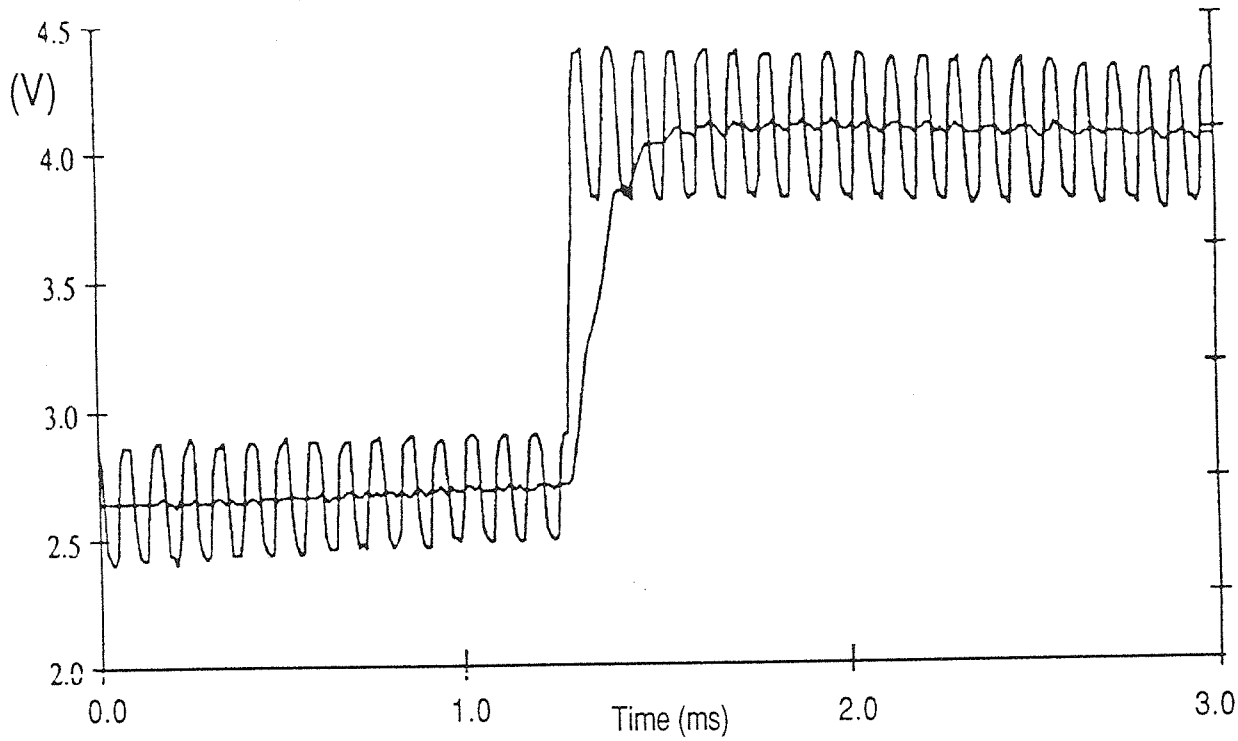


Figure 4.27: Results of processing the noisy step input with the time-fuse circuit. The risetime of the step and the level of noise reduction are both better than that of the follower integrator.

Chapter 5

The Tiny Tanh Network

This chapter introduces the *tiny-tanh* network, a continuous-time network that performs piece-wise constant segmentation. A convex Lyapunov energy is utilized so that the system does not get stuck in local minima. No annealing algorithms of any kind are necessary—gradient descent finds the unique minimum energy solution. The relationship of the tiny-tanh network co-content with constrained optimization methods is discussed. Finally, the coupled depth/slope and tiny-tanh networks are combined to perform piece-wise linear segmentation[1]

5.1 Convex Energy for Segmentation

Rather than deal with networks that have many possible stable states, a network that has a single unique stable state will be used. Consider a network that minimizes

$$E(u) = \frac{1}{2}\sum_i (d_i - u_i)^2 + \lambda \sum_i |u_{i+1} - u_i| \tag{5.1}$$

where λ is a free parameter. The absolute-value function is used for the smoothness penalty instead of the more familiar quadratic. These two functions are compared in Figure 5.1.

[1] Portions of this chapter have already been published [Harris, 1991].

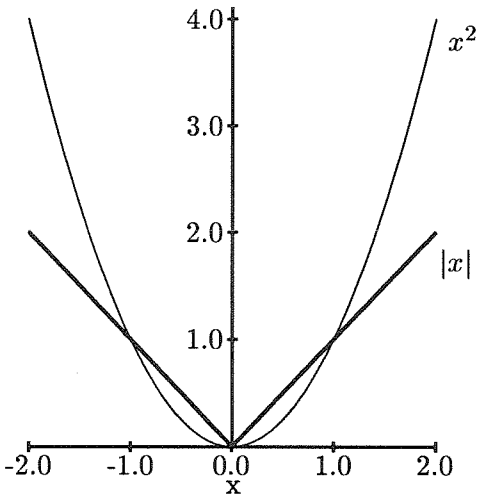


Figure 5.1: Comparison of the absolute-value and the square function.

There are two reasons why the absolute value penalty is better than the quadratic penalty for piece-wise constant segmentation. First, for the large values of $|u_i - u_{i+1}|$, the penalty is not as severe, which means that edges will not be smoothed as much. Second, small values of $|u_i - u_{i+1}|$ are penalized more than they are in the quadratic case, resulting in a flatter surface between edges.

This method is very similar to the absolute-value constrained minimization methods discussed in Chapter 1. With this interpretation, we are minimizing $(d_i - u_i)^2$ with the constraint that $u_{i+1} = u_i$. If λ is set to a large enough finite value, the constraints will be satisfied exactly and a flat surface will result. Unlike constrained optimization, however, we want the constraint to fail at discontinuities, so λ must be chosen carefully.

To build this network, take a derivative of Equation 5.1 to yield Kirchoff's current equation at that node

$$(u_i - d_i) + \lambda \text{sgn}(u_i - u_{i+1}) + \lambda \text{sgn}(u_i - u_{i-1}) = 0 \tag{5.2}$$

where the sgn function is defined as

$$\begin{aligned} \mathrm{sgn}(x) &= +1 \quad \text{if} \quad x \geq 0 \\ &= -1 \quad \text{if} \quad x < 0 \end{aligned}$$

The nonlinear resistor I-V characteristic should have a sharp step function between negative and positive saturating currents. Of course a zero transition region is impossible to realize in any real physical device. Instead, we choose to implement a tanh function with an extremely narrow linear region. For this reason, this element is called the tiny-tanh resistor. This saturating resistor is used as the smoothing resistor in the nonlinear resistive network (i.e., the one shown in Figure 1.3). Its I-V characteristic is

$$I = \lambda \tanh(V/\delta) \tag{5.3}$$

where the linear region of the tanh functions can be made arbitrarily small as δ goes to zero. If we take the limit as δ goes to zero, the tanh becomes the sgn curve. The co-content of this network is

$$\frac{1}{2}\sum_i (u_i - d_i)^2 + \sum_i \lambda \int_0^{u_{i+1}-u_i} \tanh(v/\delta)dv \tag{5.4}$$

The co-content becomes the absolute-value cost function in Equation 5.1 in the limiting case as $\delta \rightarrow 0$.

We can generalize Equation 5.3 by adding an extra factor:

$$I = \lambda(1+\delta)\tanh(V/\delta) \tag{5.5}$$

In the limit, a linear resistor of conductance λ results as $\delta \rightarrow \infty$. One equation describes a linear resistor, Mead's saturating resistor, and the tiny-tanh resistor. The value of δ continuously varies the I-V curve between the three different forms.

As was discussed in Chapter 1, any circuit made of independent voltage sources and

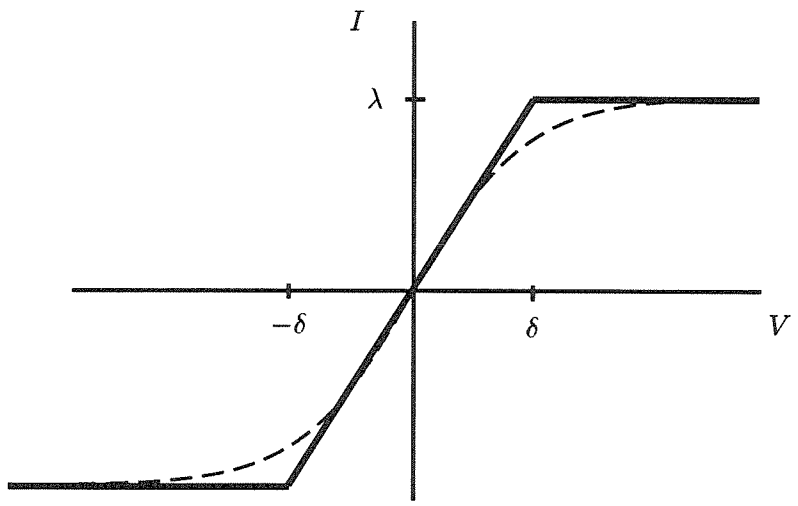


Figure 5.2: The desired saturating I-V characteristic is shown as a dotted line. Its piece-wise linear approximation (solid line) was used for simulation and analysis. The slope of the linear region (λ) and the saturation threshold (δ) are adjustable parameters.

two-terminal resistors with strictly increasing I-V characteristics has a single unique stable state. Since the tiny-tanh element has a strictly increasing slope, a convex co-content energy is being minimized, and gradient descent algorithms are guaranteed to find this solution. Figure 5.2 shows a plot of our proposed element's I-V characteristic, a simple hyperbolic tangent relationship with an adjustable width and slope. The slope in the linear region is defined to be λ/δ, and the curve saturates for voltage drops greater than δ. The piece-wise linear approximation of this curve was used in simulations and analysis.

5.2 Simulation

To evaluate how this saturating resistor performs, a fourth-order Runge-Kutta simulation was developed for a two-dimensional network of saturating resistors. A unit step edge was generated for a 20 x 20 pixel image, and Gaussian noise with $\sigma = .2$ was added. The filled circles in Figure 5.3 represent a 1-D cross section through this array. Figure 5.3a shows the typical segmentation result for the saturating resistor, $\delta = .1$ and $\lambda = 0.4$. As Mead has observed, a network of saturating resistors performs an edge enhancement. Unfortunately, the noise is still evident in the output, and the curves on either side of the step have started

to slope toward one another. As λ is increased to smooth out the noise, the two sides of the step will blend together into one homogeneous region. These same results are observed with hardware resistive networks using Mead's saturating resistor. These saturating resistors cannot segment data into regions of roughly uniform voltage. The width of the linear region of the saturating resistor (100mV for subthreshold circuits) is too large and cannot be easily decreased.[2]

The new idea is that the saturating resistor can be run in a very different region of operation where network segmentation properties are greatly enhanced. If we decrease the width of the linear region significantly ($\delta = .005$) while simultaneously increasing the conductance ($\lambda = 1$), the result shown in Figure 5.3b is produced. The height of the detected step is 200 times larger than δ. If the signal height was 100mV, for example, a resistor with a linear region with a total width of 1mV would be required to provide the same performance. Mead's saturating resistor has a minimum linear region width of about 100mV. Figure 5.4 shows a simulation of the tiny-tanh network segmenting the famous mandrill image for various values of δ. The network segments the image into blocks of approximately uniform brightness with λ and δ determining the scale of the computation. No continuation method or annealing strategy of any sort was necessary to produce these simulation results.

The network does not recover the exact heights of input edges. Rather it subtracts a constant from the height of each input. This effect is noticeable in Figure 5.3 where the unit step input has decreased slightly in magnitude. In two dimensions, this effect becomes more interesting. This can be understood by considering an input tower of n x n pixels at height h and computing the height of the result. It is assumed that the background is very large in area and grounded and that as $\delta \rightarrow 0$, the correct segmentation of the network is computed. Using the 2-D equivalent of Equation 5.1 and assuming that the network returns

[2]Shulman and Hervé also use the equivalent of a saturating resistor network for segmenting the optical flow field [Shulman and Hervé, 1989].

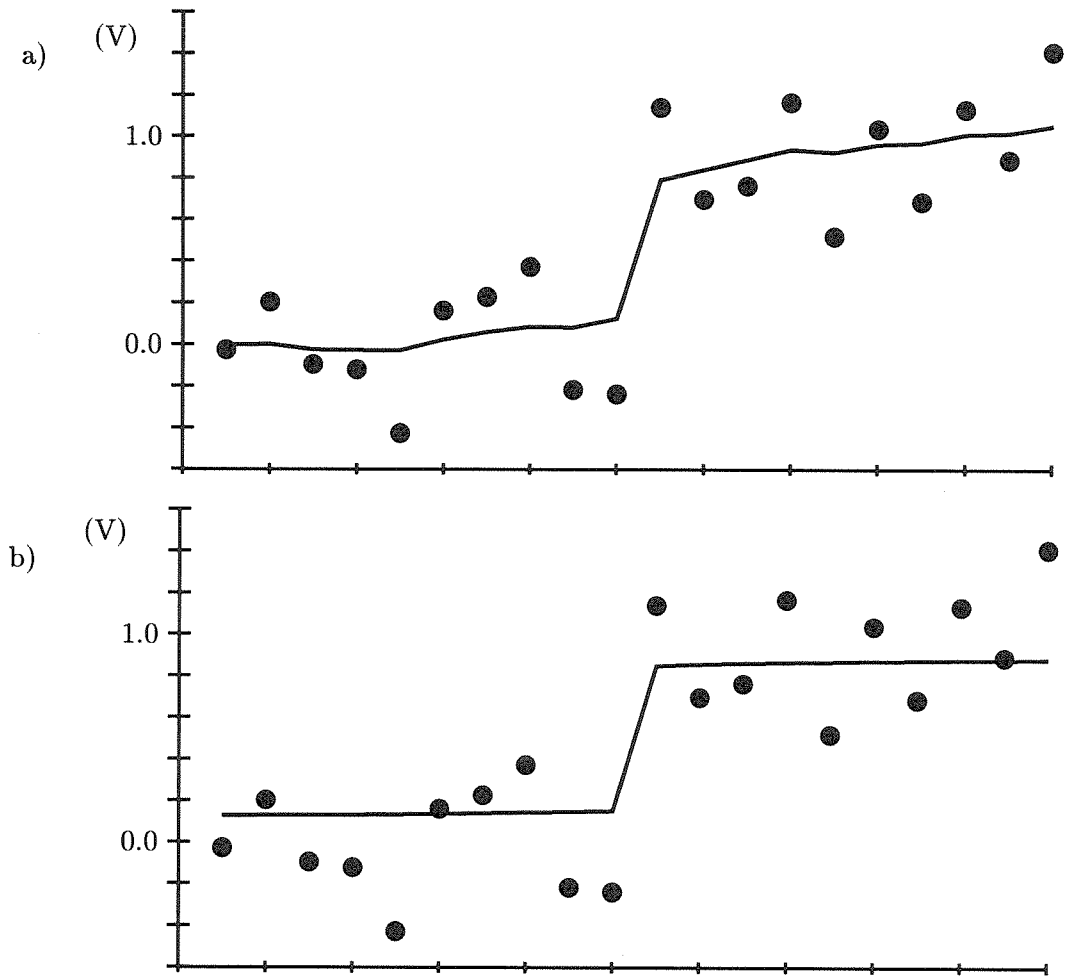


Figure 5.3: A 20x20 pixel unit step edge was generated, and Gaussian noise with $\sigma = .2$ was added. The filled circles indicate a 1-D cross section through this input array. **(a)** Shows the segmentation result for the saturating resistor in the usual setting, $\delta = .1$ and $\lambda = 0.4$. For these parameter settings, the noise is still visible in the output and the curves on either side of the step have started to slope toward one another. **(b)** The enhanced region of operation ($\delta = .005$ and $\lambda = 1$) segments the input into two distinct uniform regions.

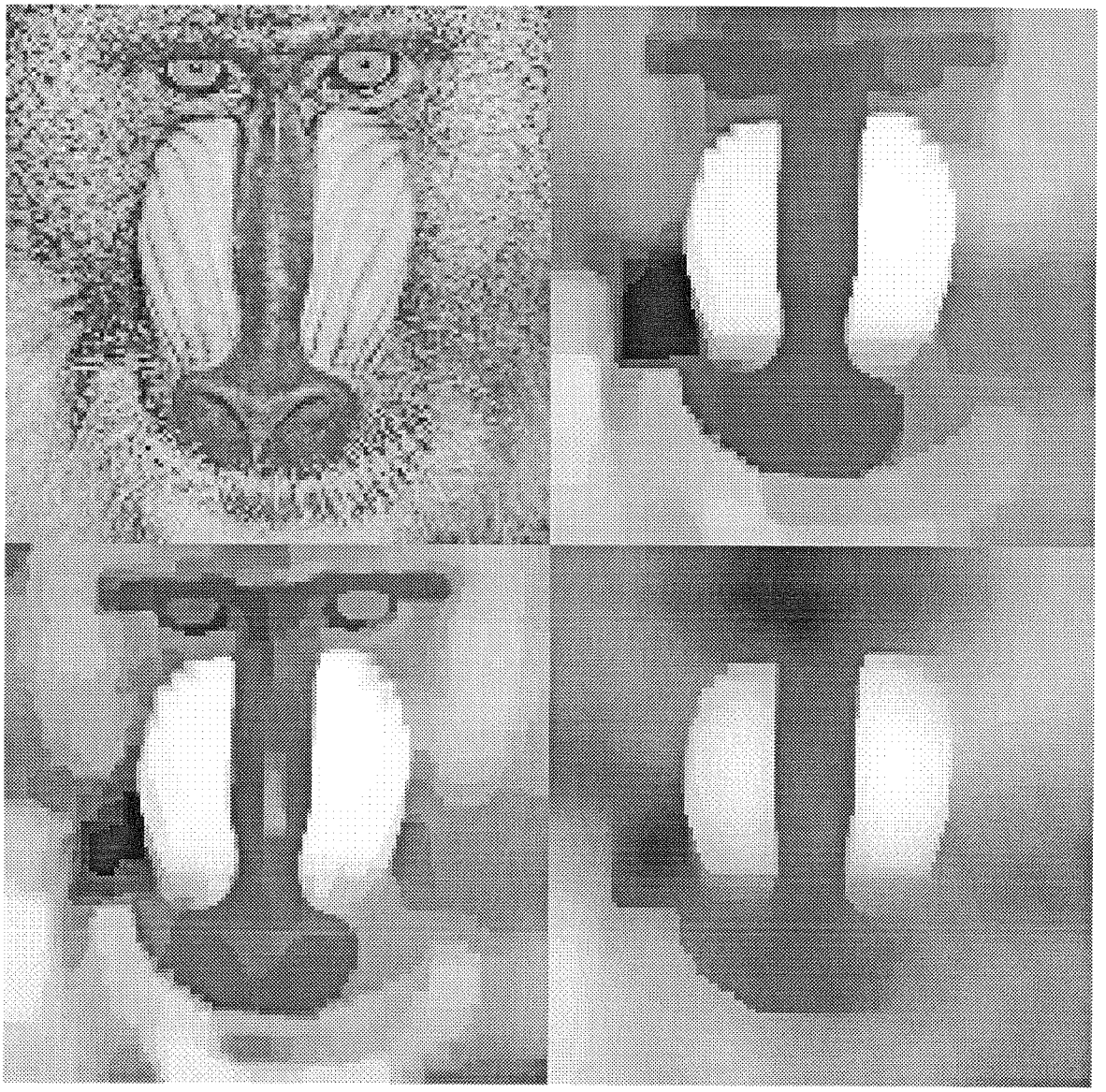


Figure 5.4: Simulations of tiny-tanh network on the mandrill image. Top left is the original, top right $\delta = 0.0064$, bottom left $\delta = 0.0128$ and bottom right $\delta = 0.0512$. $\lambda/\delta = 32$ for the three processed images. The original image consisted of 256x256 8-bit pixels. Each pixel ranged in value from 0 to 255. The network segmented the image into blocks of approximately uniform brightness with λ and δ determining the scale of the computation.

a tower of height x, the co-content at the unique stationary point of the system is

$$E(x) = \frac{1}{2}n^2(x-h)^2 + 4n\lambda|x| \tag{5.6}$$

since the first term was nonzero over the area of the tower (n^2 pixels) and the second term was nonzero over the perimeter (4n pixels). Assuming that $x > 0$, find the stable point by differentiating Equation 5.6 and setting it equal to zero:

$$n^2(x-h) + 4n\lambda = 0 \tag{5.7}$$

Solving for x, the recovered height of the step:

$$x = \lfloor h - \frac{\lambda}{4n} \rfloor \tag{5.8}$$

or more generally

$$x = \lfloor h - \lambda \frac{\text{perimeter}}{\text{area}} \rfloor \tag{5.9}$$

The $\lfloor x \rfloor$ function ensures that x never be pulled below 0. Important features with large area/perimeter ratios will remain very close to their original height h. Noise points have small area/perimeter ratios and therefore will be pulled towards the background. Thus, this network will reduce the height of noise points that are larger or smaller than the background value.

Typically, the exact values of the heights are less important than the location of the discontinuities. Furthermore, it would not be difficult to construct a two-stage network to recover the exact values of the step heights if desired. In this scheme a tiny-tanh network would control the switches on a second fuse network. Unlike typical line-process energy minimization algorithms, the whole segmentation computation is accomplished without defining a binary threshold for edges. The height of the output steps encode a continuous function of the original step height and the perimeter/area ratio.

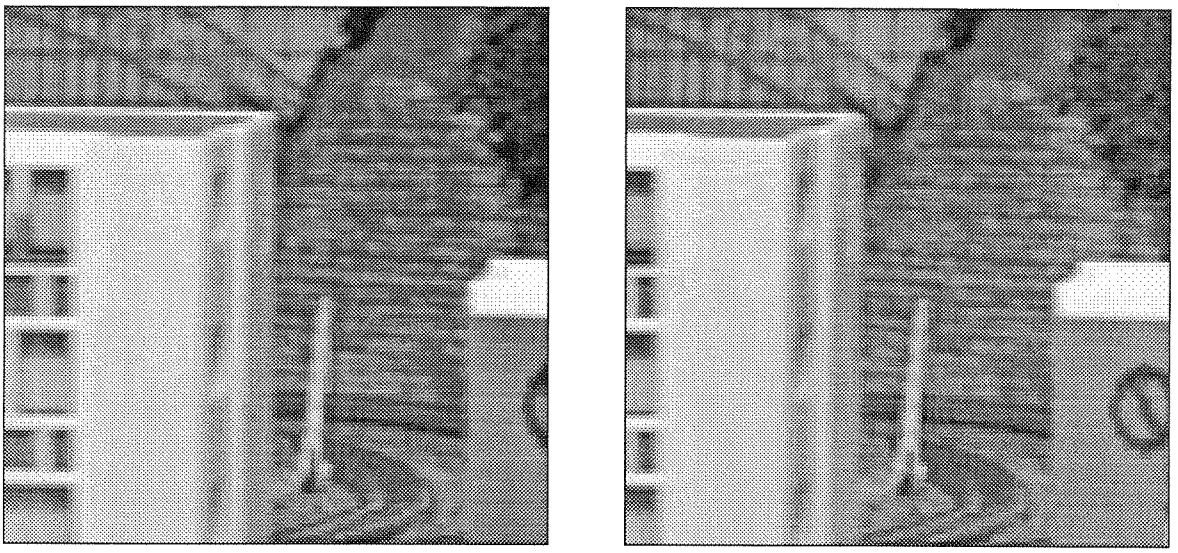


Figure 5.5: The two 128x128 images used in the experiments. The slight shift between the two images is apparent by studying the circle on the box on the right side of each image.

In retrospect, we have changed the nonlinear line-process computation to a simpler problem that can be solved by minimizing a convex energy. Notice that the network does not have an explicit representation of discontinuities. It merely subtracts a fixed voltage from the height of input step edges.

5.3 An Application to Stereo

We have simulated the performance of a correlation-based stereo algorithm with a tiny-tanh postprocessing step. The two images shown in Figure 5.5 were taken sequentially with a single camera which was translated horizontally.[3] Using an extremely simple matching method, we generated the noisy motion estimates given in Figure 5.6a. A patch of 5x5 pixels was correlated with 5x5 windows in the second image. The shift with the minimum sum of absolute value differences was marked as the winner. Disparities of up to 12 pixels were searched. The algorithm is a simplified version of that used by [Little et al., 1988].

The noisy data in Figure 5.6a was passed through a simulated tiny-tanh network to produce the segmented image shown in Figure 5.6b. In the simulation, λ was set to 25 and

[3]These images were supplied by Larry Mathies at CMU.

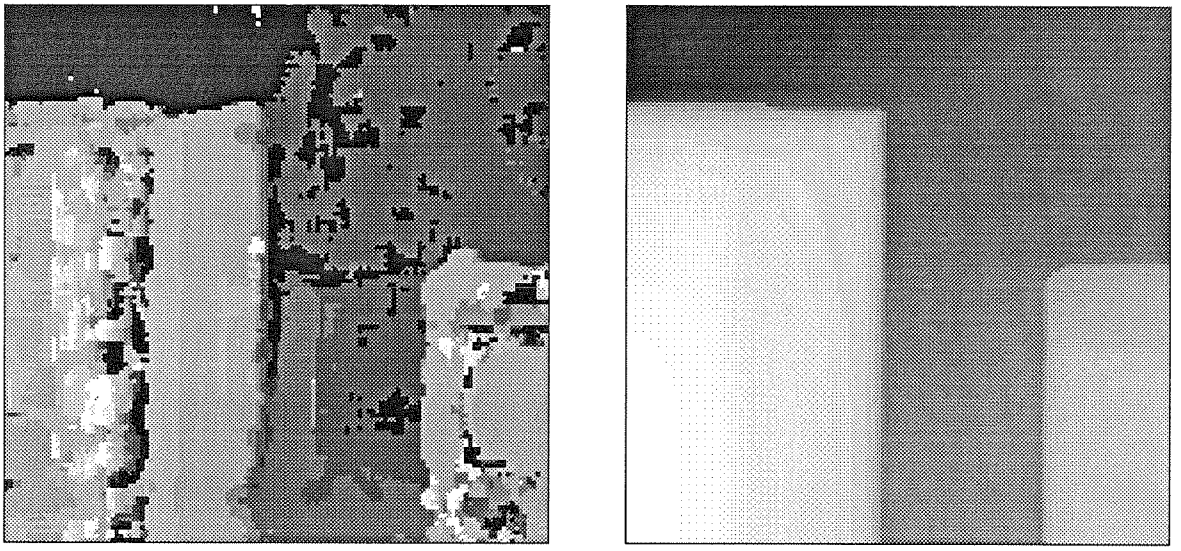


Figure 5.6: Left image is the motion correlation output. Right shows the smoothed and segmented motion output using the tiny-tanh network.

δ was set to 0.01. The network effectively segmented the image into three objects: the left and right boxes and the background.

5.4 Step Response

In order to better understand tiny-tanh segmentation, we will study the response of the 1D network to an ideal step edge.

Linear Case

First, consider the linear case. A 1D linear resistive network is shown in Figure 5.7a. The inputs to the left half of the network are grounded and the right half inputs are set to a voltage h. Rather than solve for the full step response, we will derive the voltage drop across the central resistor $V_H - V_L$. If we assume that $\lambda > 1$, we can replace the two semi-infinite networks on either side of the step with their approximate equivalent networks. [Mead, 1989], shown in Figure 5.7b. This continuum limit approximation for a discrete network becomes exact as $\lambda \rightarrow \infty$. Solving this equivalent network gives:

$$V_L = h\frac{\lambda}{2\lambda + \sqrt{\lambda} + 1}$$

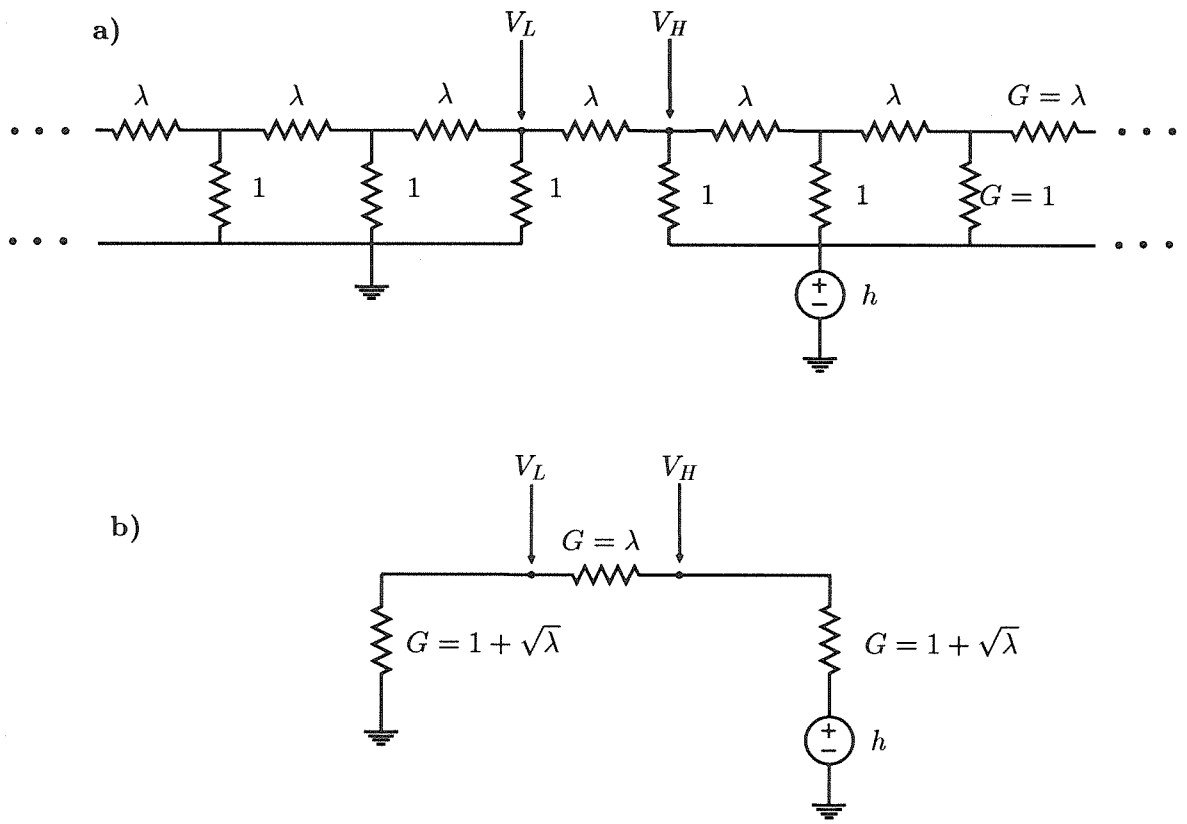


Figure 5.7: **(a)** shows a step input of voltage h to a linear resistor network. **(b)** shows the equivalent network using the continuum limit approximation.

$$\begin{aligned} V_H &= h\frac{\lambda+\sqrt{\lambda}+1}{2\lambda+\sqrt{\lambda}+1} \\ V_H - V_L &= h\frac{\sqrt{\lambda}+1}{2\lambda+\sqrt{\lambda}+1} \end{aligned} \tag{5.10}$$

For large λ,

$$V_H - V_L \approx \frac{h}{2\sqrt{\lambda}} \tag{5.11}$$

This is the maximum voltage drop across any resistor in the network as a function of h and λ. Equation 5.11 is useful for setting the threshold for breaking in resistive fuse networks.

Saturating Case

We can perform a similar analysis for the nonlinear case. A step is input to the saturating resistor network in Figure 5.8a. If the form of the saturating resistors is: $I = \lambda \tanh(V/\delta)$, the network cannot be solved in closed form. We therefore assume a piece-wise linear approximation to the I-V curve of:

$$I(V) = \begin{cases} \lambda & \text{if } V > \delta \\ \lambda V/\delta & \text{if } -\delta \leq V \leq \delta \\ -\lambda & \text{if } V < -\delta \end{cases} \tag{5.12}$$

Since all resistors except the center one are guaranteed to be operating in the linear region of operation, we can replace them by linear resistors of conductance λ/δ. Replacing the left and right sides of the network by their equivalents leads to the equivalent network shown in Figure 5.8b.

Assuming $|V_H - V_L| \leq \delta$,

$$\begin{aligned} V_L &= h\frac{\lambda/\delta}{2\lambda/\delta+\sqrt{\lambda/\delta}+1} \\ V_H &= h\frac{\lambda/\delta+\sqrt{\lambda/\delta}+1}{2\lambda/\delta+\sqrt{\lambda/\delta}+1} \\ V_H - V_L &= h\frac{\sqrt{\lambda/\delta}+1}{2\lambda/\delta+\sqrt{\lambda/\delta}+1} \end{aligned} \tag{5.13}$$

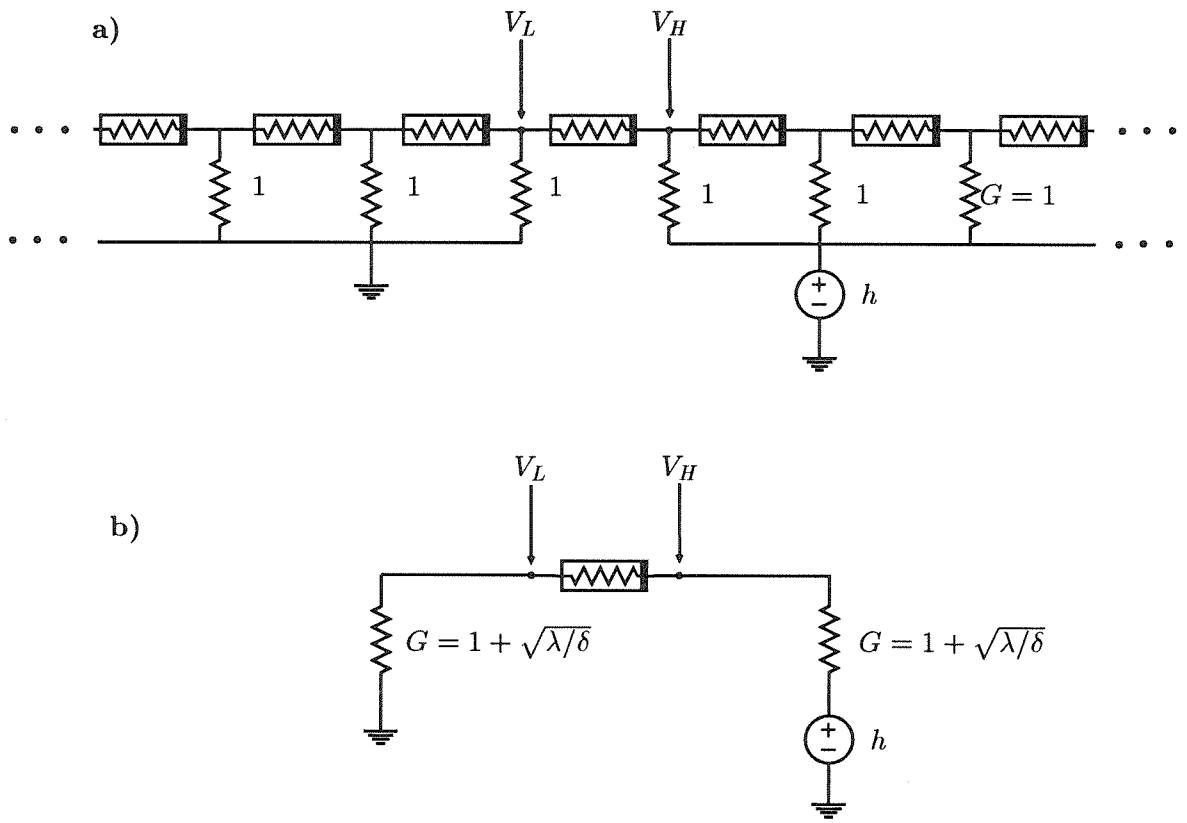


Figure 5.8: **(a)** shows a step input of voltage h to a saturating resistor network. **(b)** shows the equivalent network using the continuum limit approximation.

For large λ/δ, Equation 5.13 can be approximated by:

$$V_H - V_L \approx \frac{h}{2\sqrt{\lambda/\delta}} \tag{5.14}$$

Equation 5.14 is true when all of the resistors are operating in their linear region, in other words: $|V_H - V_L| > \delta$. Substituting this relation into Equation 5.14 gives the condition that: $h < 2\sqrt{\delta\lambda}$.

For $|V_H - V_L| > \delta$, the resistor is saturated and can be replaced by a constant current source. Solving for V_H and V_L gives:

$$V_L = \frac{\lambda}{1+\sqrt{\lambda/\delta}} \tag{5.15}$$

$$V_H = h - \frac{\lambda}{1+\sqrt{\lambda/\delta}} \tag{5.16}$$

$$V_H - V_L = h - \frac{2\lambda}{1+\sqrt{\lambda/\delta}} \tag{5.17}$$

$$\tag{5.18}$$

For large λ/δ,

$$V_H - V_L \approx h - 2\sqrt{\delta\lambda} \tag{5.19}$$

which is true when $h > 2\sqrt{\delta\lambda}$.

In summary, the full range of the step response of the saturating resistor network is:

$$V_H - V_L \approx \begin{cases} \frac{h}{2\sqrt{\lambda}} & \text{if } h \leq 2\sqrt{\delta\lambda} \\ h - 2\sqrt{\delta\lambda} & \text{if } h > 2\sqrt{\delta\lambda} \end{cases} \tag{5.20}$$

A plot of these results is shown in Figure 5.9. In the limiting case, as $\delta \to 0$, $V_H - V_L = h$ and the full step height is preserved.

5.5 Hardware Implementation

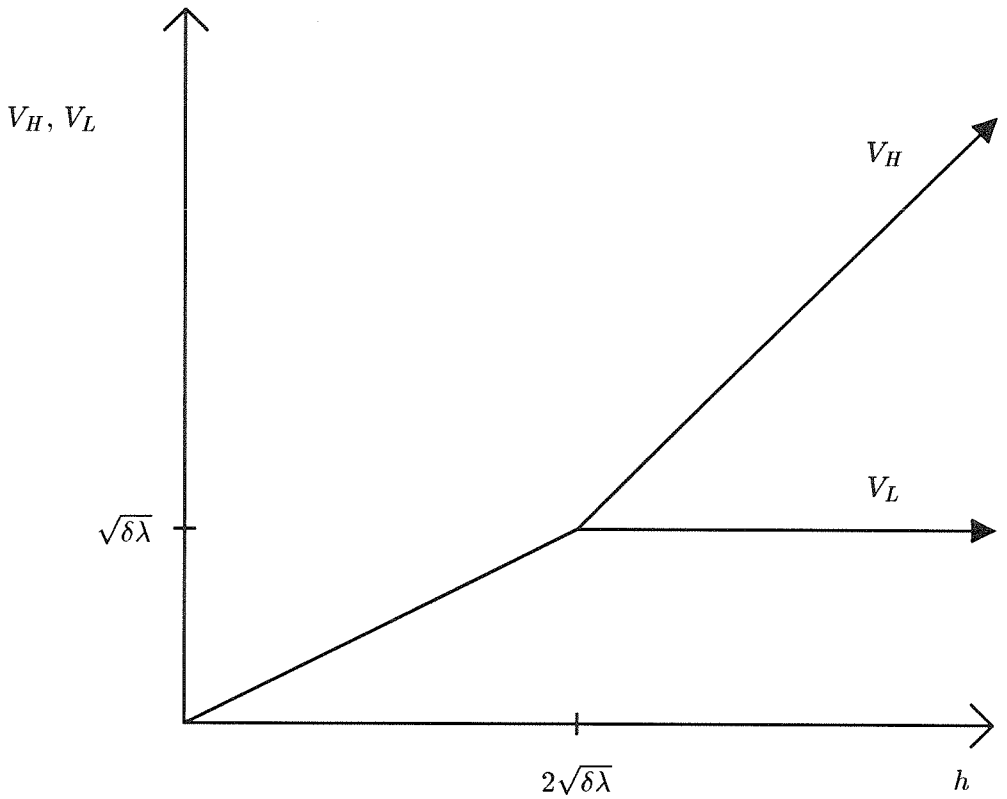


Figure 5.9: Plot of V_H and V_L (defined in Figure 5.8) versus the input step height h. For $h < \sqrt{\delta\lambda}$, $V_H = V_L$ and the step is smoothed out. For $h > 2\sqrt{\delta\lambda}$, V_H and V_L diverge and the step is preserved, although reduced in height by $\sqrt{\delta\lambda}$.

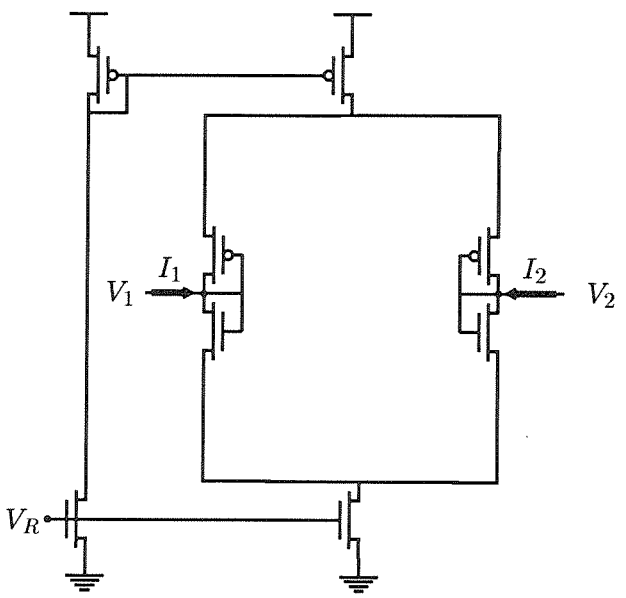


Figure 5.10: Old saturating resistor circuit. The resistance is seen between nodes V_1 and V_2.

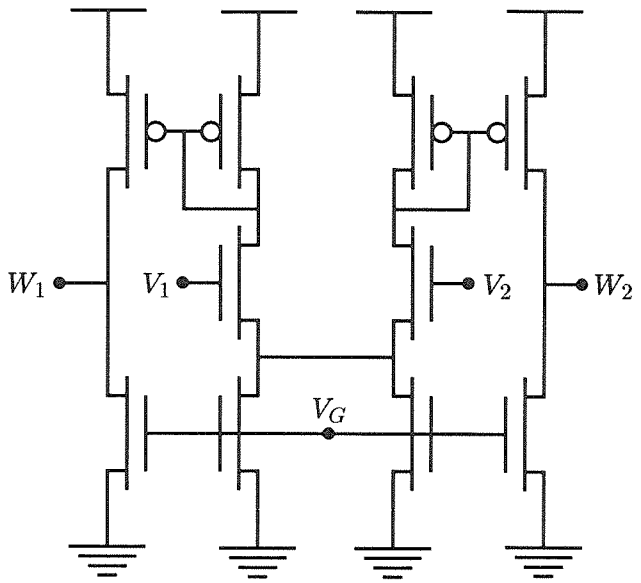


Figure 5.11: Simple gain stage.

There are numerous difficulties with building a resistor that saturates in a few millivolts. A circuit such as Mead's saturating resistor saturates when the voltage difference is a few kt/q (about 50mV). There is no obvious way of shrinking this linear region short of reducing kT/q by cooling the circuitry. The tiny-tanh element builds on the "old" version of the saturating resistor shown in Figure 5.10 [Sivilotti et al., 1987]. This resistor also saturates at about 50mV below threshold. Mead's new saturating resistor with its guaranteed zero offset voltage and small size has made the old resistor obsolete. However, the old resistor has the advantage that additional circuitry can decrease the width of the linear region of its I-V characteristic.

The simple gain stage shown in Figure 5.11 is used. In subthreshold operation, this amplifier can easily have a gain of several hundred. This gain is reduced as the amplifier is driven above threshold.

The combination of the old resistor (Figure 5.10) and the gain-stage (Figure 5.11) gives the tiny-tanh circuit element shown in Figure 5.12. A measured I-V curve from the tiny-tanh circuit for $V_R = 1.2V$ and $V_G = 1.5V$ is shown in Figure 5.13. The linear region in the figure is approximately 4mV with a 1mV offset. Careful design reduced the offsets to

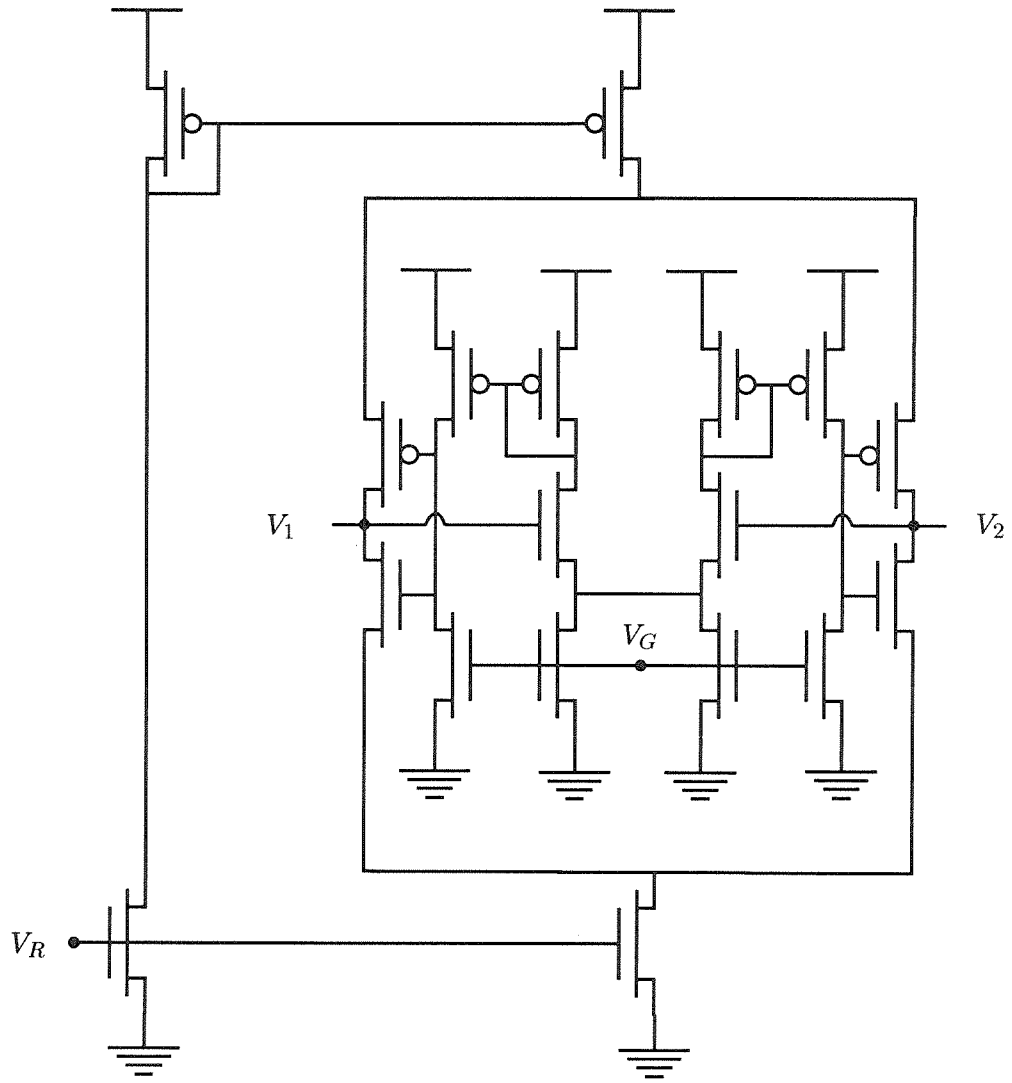


Figure 5.12: Tiny tanh circuit.

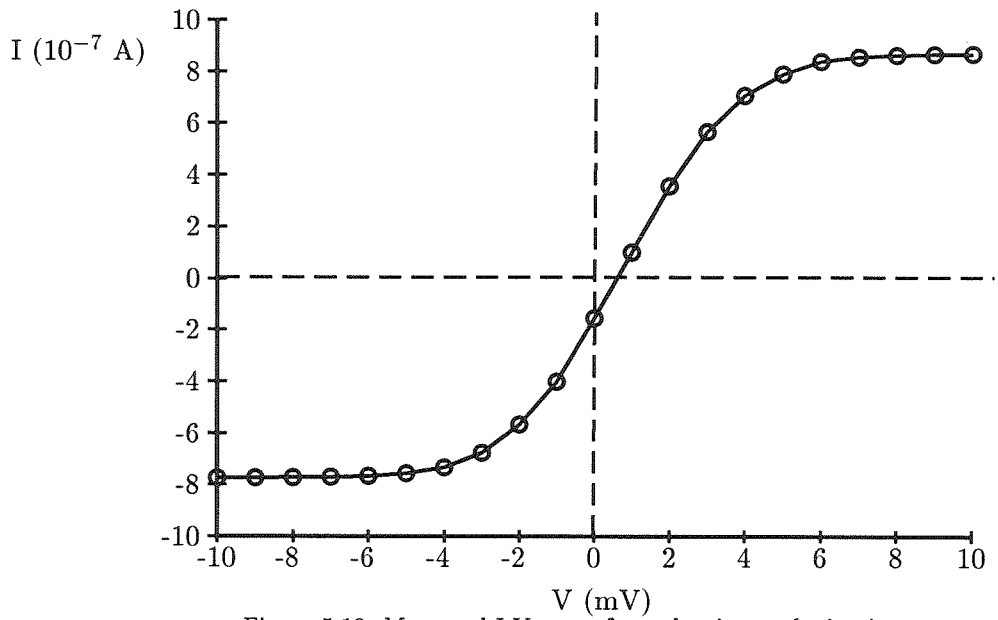


Figure 5.13: Measured I-V curve from the tiny-tanh circuit.

low values.

5.6 Segmentation Results

A 1D tiny-tanh network was successfully designed, fabricated and tested. Figure 5.14 shows the segmentation which resulted when a large step (about 1V) was scanned into the chip. The segmented step has been reduced to about 0.5V. Figure 5.15 shows the same experiment for a for a much smaller step (about 40mV), Note that the small step which is recovered is within the linear region of Mead's saturating resistor (50mV). Since the support of the outlier point is small, it is not detected as a separate object but is smoothed into neighboring values. In both experiments, $V_R = 1V$ and $V_G = 0.7V$. The transconductance amplifier bias (providing the inputs to the network) varied between 0.65V (left plot in each figure) and 0.50V (right plot). This had the effect of varying λ without changing the width of linear region of the tanh curve. No special annealing methods were necessary because a convex energy is being minimized.

Since the tiny-tanh implementation depends on matched transistors, the network, in general, does not obey Kirchhoff's current law. The circuit suffers from both voltage and

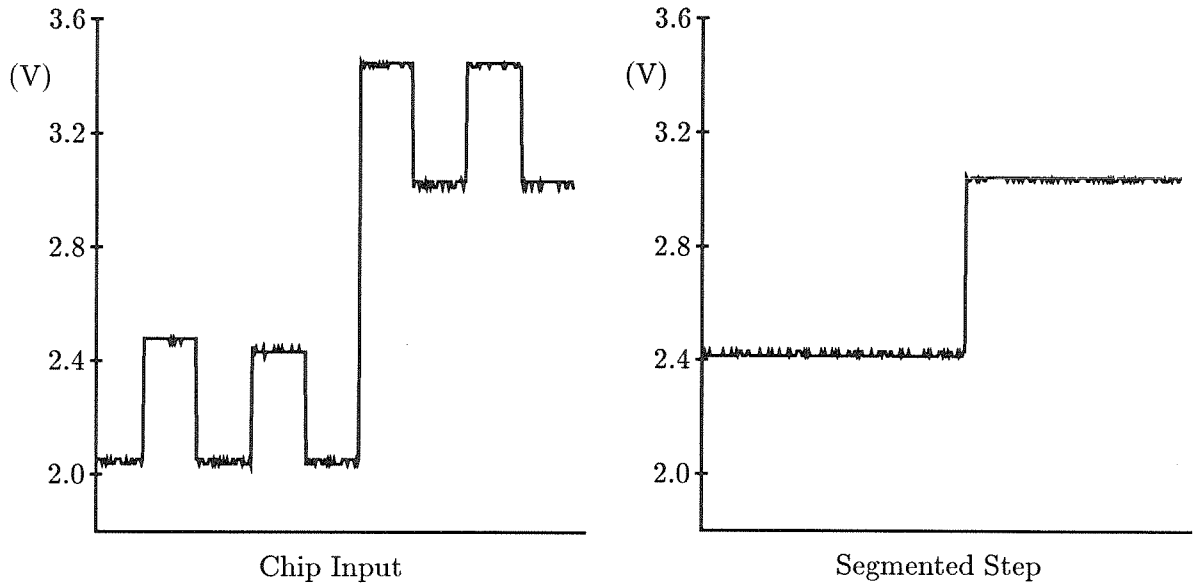


Figure 5.14: Measured segmentation performance of the tiny-tanh network for a large step. The input shown on the left is about a 1V step. The output shown on the right is a segmented step about 0.5V in height.

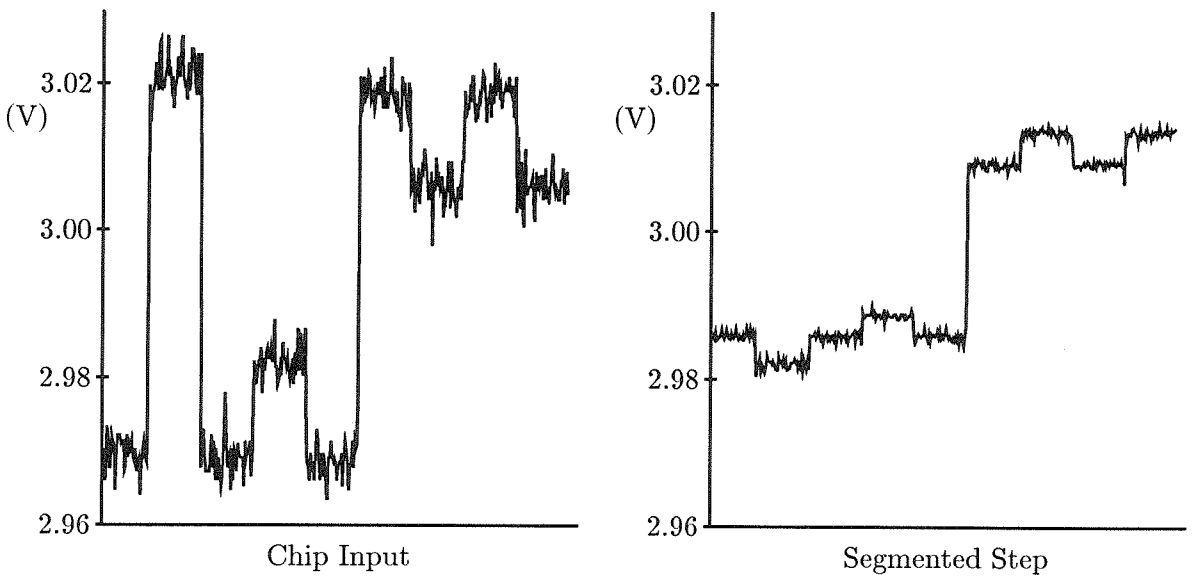


Figure 5.15: Measured segmentation performance of tiny-tanh network for a small step. The input waveform shown on the left is about 40mV in height. The output on the right is the segmented step, also about 40mV in height. The ripple in the output waveform reflects the current and voltage offsets of the tiny-tanh circuit.

current offsets. The voltage offsets can be noticed by the small ripple on either side of the segmented step in Figure 5.15. A network using Mead's saturating resistor, on the other hand, is guaranteed to have zero voltage offset.

5.7 Piece-wise Linear Segmentation

So far only piece-wise constant segmentation has been considered. This work is easily extended to piece-wise linear segmentation using the coupled depth/slope network from Chapter 3. The smoothing resistors in the model (in Figure 3.1) are replaced by tiny-tanh resistors causing segmentation to occur on the derivative and not on the signal itself.[4] The energy minimized is therefore

$$E(u) = \sum_i (u_i - d_i)^2 + G(u_{i+1} - u_i - p_i)^2 + \lambda |p_i - p_{i+1}|^2 \qquad (5.21)$$

As before, the variable G is a measure of the conductance of the subtract constraint box. The input data is d_i and the output and its derivative are u_i and p_i respectively. As the value of G increases, the constraint box accuracy improves.

A triangular pulse of height 1V with additive Gaussian noise of $\sigma = 0.05$ was used as input to the simulated network. Gradient descent was performed on the energy given by Equation 5.21. The results are shown in Figure 5.16 were computed with $G = 10^4$ and $\lambda = 20$. Figure 5.16a shows the noisy input data (d_i) with the segmented result (u_i). Figure 5.16b shows the derivative of the original input along with the network's representation of the derivative of the segmented output (p_i). As before, the tiny tanh does not recover the exact heights and slopes of the line segments. A resistive fuse solution, however, would recover the exact least squares fit of each line segment at the expense of the addition of complex continuation methods [Liu and Harris, 1989].

[4] Previously, resistive fuses and the coupled model have been used in a similar fashion to demonstrate smoothing and discontinuity detection of any arbitrary level of derivative [Liu and Harris, 1989].

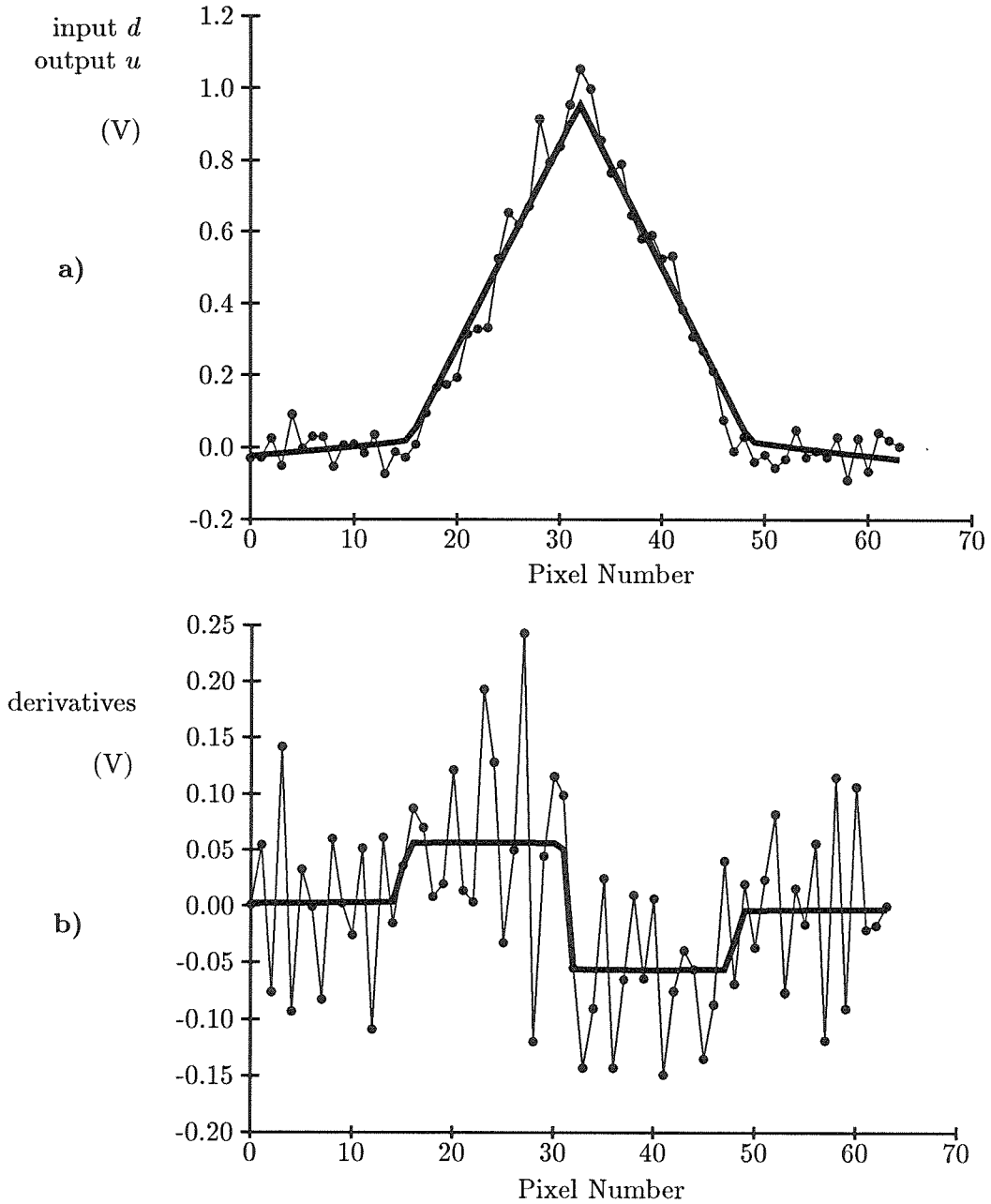


Figure 5.16: Piece-wise linear segmentation of a triangular pulse using the coupled depth/slope methods and the tiny-tanh. a) shows the noisy input data d_i (thin line) with the segmented result u_i (bold line). b) shows the derivative of the original input (thin line) along with the network's representation of the derivative of the segmented output p_i (bold line).

Chapter 6

Conclusions

This thesis has introduced three algorithm/circuit innovations:

Constraint Boxes were introduced as a general methodology for mapping regularization based algorithms to stable analog hardware. Other attempts at mapping these algorithms to hardware required data-dependent and possibly negative resistors as well as voltage sources with values that are complex functions of the data. The constraint box requires only simple voltage sources and positive, data-independent resistors. A biharmonic surface interpolation chip using these principles was discussed.

The Resistive Fuse was shown to be the first hardware circuit that explicitly implements line process discontinuities. A 20x20 pixel fuse network chip was demonstrated. Resistive fuse networks minimize a non-convex energy function that may contain local minima. As in computer vision, complex annealing or continuation methods are necessary for adequate solutions of the problem.

The Tiny-Tanh network smoothes and segments data by minimizing a convex energy functional. It deals with discontinuities in a different way that is not plagued by problems with local minima. Simulations show that this network can perform image and depth segmentation. An experimental 1D hardware circuit was successfully designed and tested.

6.1 Why Analog Models?

1. Solving Difficult Problems

There are many difficult vision problems that are far from being solved satisfactorily. One example is autonomous vehicle navigation. The first demonstration of cross-country navigation was done with the DARPA Autonomous Land Vehicle (ALV) in 1987 by Hughes Aircraft [Daily et al., 1988b] [Daily et al., 1988a]. The ALV successfully avoided ravines, bushes, steep slopes, and rock outcrops, sometimes traversing inclines that rolled and pitched the vehicle up to 15 degrees. As impressive as this feat was, this 10-ton vehicle stocked full of digital computers (and connected via a radio link to more computers in a lab) could only move at about 1 mile/hour through the treacherous terrain. An active laser range sensor was used to produce depth images. A frame was processed every seven seconds. In between frames, the vehicle traveled blind. Vision was not used because processing would have been too "slow."

At present, analog hardware systems are still too primitive to be of much use for something like the ALV. However, it is a good bet that fast, low-power analog hardware will be useful and economical for this and other applications in the coming years. It has already been demonstrated that these analog chips are inherently robust and accurate enough to allow for simple navigation tasks such as following edges or tracking moving light sources when mounted onto small, highly mobile toy cars operating in a laboratory environment [Koch et al., 1990a].

2. Raw Speed of Computation

The raw speed of computation allows for many more experiments, leading towards more intuition and understanding. There are many good stereo algorithms but they require so long to compute that experimentation with them is impossible. Cochran describes one of the best stereo algorithms but it take eight hours to run on a single frame [Cochran and Medioni, 1989].

Continuous-time circuits avoid discrete-time convergence problems. This is especially

evident when simulating nonlinear dynamical systems. The tiny-tanh network simulations require hours of processing time on conventional digital machines. The underlying differential equations are very stiff and extremely small time steps are needed to avoid numerical instabilities. Oscillations which might result would be mere artifacts of the discrete nature of the simulation.

3. Dealing with Time

This new computational medium gives us the ability to deal with time explicitly and rapidly prototype nonlinear systems. The resistive fuse network is a highly nonlinear system that has forced us to think explicitly about dealing with time in our solution strategies. Simulated annealing and other continuation methods are awkward to implement in a continuous-time fashion; alternative methods are desirable. Experimentation with analog hardware has led to the tiny-tanh model which runs in continuous time without the need for any annealing or continuation methods.

4. Thinking Differently

Finally, there are fundamental reasons for studying analog models even if they cannot outperform digital computers. In his 1959 PhD thesis, Dennis wrestled with these same issues [Dennis, 1959]:

> "What then is the value of investigating these electrical models? The answer lies in the fundamental difference in approach of the operations analyst with regard to an optimization problem and the electrical scientist studying a circuit. The analyst nearly always speaks in terms of minimization or maximization subject to constraints. The electrical scientist, however, is merely looking for a distribution of currents and voltages which satisfies the conditions imposed by the circuit—he rarely thinks in terms of minimization, and may not even know that an appropriate extremum principle exists."

There is value in this alternate point of view, that is thinking in terms of local currents and voltages as opposed to global energy minimization. The three major innovations in this

thesis–the constraint box, resistive fuse, and tiny-tanh–were all developed through thinking in the current/voltage domain. It was later that power (or co-content) minimization was brought into the picture. Thinking in the current/voltage domain led to these circuit/algorithm innovations. Afterwards, energy minimization provided further justification and intuition for these methods.

6.2 Future Directions

The goals set out in this thesis are far from complete. The use of analog models and hardware to study and develop computer vision algorithms is still in its infancy. There is much more to be done, more than just optimizing the few existing circuits, or packing more pixels per chip, or making a prettier picture on a screen.

Some future topics we must understand include:

Fusion of multiple sources: Much research is going into the fusion of multiple cues, for example see [Clark and Yuille, 1990] and [Aloimonos and Shulman, 1989]. It is clear that robust, reliable systems must rely on multiple sources of information. Constraints such as forcing depth or motion discontinuities to coincide with intensity edges have proven to be very powerful [Gamble and Poggio, 1987]. This does not mean, however, that we must combine every single sensor output into a single coherent representation of the world. This fusion of information can occur at the sensor level as well as at the behavior level [Brooks, 1986] [Payton, 1986].

Robust Parameter Estimation: Many of the early vision systems researchers are studying are very fragile when is comes to choosing parameters. Robust methods need to be invented for setting values of thresholds, space constants, and time constants. Local adaptation is needed in these networks to make them less brittle. Methods from Bayesian analysis are being studied to set some free parameters for standard regularization-based methods [Szeliski, 1989] [MacKay, 1991].

Transcending pixel-based processing: It is easy to imagine architectures for processing images on a pixel-by-pixel basis. We must move on to high-level vision problems, i.e., recognition or learning. Several architectures that can segment features in a scene based upon intensity or depth have been described in this thesis.. This is just the first step in breaking an image into multiple, labeled objects and subsequently dealing with them in an intelligent fashion.

Analog circuits provide an interesting and powerful computational medium for developing early vision algorithms. As these small, inexpensive, low-power chips become common, researchers will possess real-time prototyping capabilities to address these fundamental issues.

References

[Aloimonos and Shulman, 1989] Aloimonos, J. Y. and Shulman, D. (1989). *Integration of visual modules.* Academic Press, Inc.

[Arrow et al., 1958] Arrow, K., Hurwicz, L., and Uzawa, H. (1958). *Studies in Nonlinear Programming.* Stanford University Press.

[Barman, 1990] Barman, R. A. (1990). A nonlinear switched capacitor network for edge detection in early vision. Master's thesis, Univerisity of British Columbia.

[Bertsekas, 1976] Bertsekas, D. P. (1976). Multiplier methods: a survey. *Automatica*, 12:133–145.

[Birkhoff and Diaz, 1956] Birkhoff, G. and Diaz, J. B. (1956). Nonlinear network problems. *Quart. Appl. Math.*, 13:431–443.

[Blake and Zisserman, 1987] Blake, A. and Zisserman, A. (1987). *Visual Reconstruction.* MIT Press, Cambridge, MA.

[Brooks, 1986] Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2:14–23.

[Chou and Brown, 1986] Chou, P. and Brown, C. (1986). Multimodal reconstruction and segmentation with markov random fields and HCF optimization. In *Proceedings of the DARPA Image Understanding Workshop*, pages 214–221, Cambridge, MA.

[Chua and Lin, 1984] Chua, L. and Lin, G. (1984). Nonlinear programming without computation. *IEEE Tr. on Circuits and Systems*, CAS-31(2). (See also errata, Vol. CAS-32, No. 7, p. 736 July 1985).

[Chua et al., 1987] Chua, L. O., Desoer, C. A., and Kuh, E. S. (1987). *Linear and Nonlinear Circuits.* McGraw-Hill, New York.

[Clark and Yuille, 1990] Clark, J. J. and Yuille, A. L. (1990). *Data fusion for sensory information processing systems.* Kluwer Academic Publishers.

[Cochran and Medioni, 1989] Cochran, S. D. and Medioni, G. (1989). Accurate surface description from binocular stereo. In *IEEE Workshop on the Interpretation of 3D Scenes*, pages 15–23, Austin, TX.

[Daily et al., 1988a] Daily, M., Harris, J., Keirsey, D., Olin, K., Payton, D., Reiser, K., Rosenblatt, J., Tseng, D., and Wong, V. (1988a). Autonomous cross country navigation with the ALV. In *IEEE Robotics and Automation Conference*, Philadelphia, PA.

[Daily et al., 1988b] Daily, M., Harris, J., and Reiser, K. (1988b). An operational perception system for cross-country navigation. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, pages 794–802.

[Decker et al., 1991] Decker, S., John L. Wyatt, J., and Sodini, C. G. (1991). A compact resistive fuse for image segmentation. Unpublished Memo, MIT.

[Delbrück, 1991a] Delbrück, T. (1991a). *Bump* circuits for computing similarity and disimilarity of analog voltages. Computation and Neural Systems (CNS) Memo 10, Caltech, Pasadena, CA.

[Delbrück, 1991b] Delbrück, T. (1991b). *Bump* circuits for computing similarity and disimilarity of analog voltages. In *International Joint Conference on Neural Networks*, Seattle, WA., to appear.

[Dennis, 1959] Dennis, J. (1959). *Mathematical Programming and Electrical Networks.* Wiley, New York.

[Depack, 1947] Depack, D. C. (1947). A resistor network for the approximate solution of the laplace equation. *Rev. Sci. Instr.*, 18:798–799.

[Duffin, 1947] Duffin, R. J. (1947). Nonlinear networks iia. *Bull. Amer. Math. Soc.*, 53:963–971.

[Gamble and Poggio, 1987] Gamble, E. and Poggio, T. (1987). Integration of intensity edges with stereo and motion. Technical report, MIT Artificial Intelligence Laboratory, Cambridge, MA. AI Memo No. 970.

[Geiger and Girosi, 1989] Geiger, D. and Girosi, F. (1989). Parallel and deterministic algorithms for mrfs: surface reconstruction and integration. Technical report, MIT Artificial Intelligence Laboratory, Cambridge, MA. AI Memo No. 646.

[Geman and Geman, 1984] Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741.

[Gill et al., 1981] Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical Optimization.* Academic Press.

[Girosi et al., 1990] Girosi, F., Poggio, T., and Caprile, B. (1990). Extensions of a theory of networks for approximation and learning: outliers and negative examples. Technical Report 9005-12, Istituto pe la Ricerca Scientifica e Tecnologica, Trento, Italy.

[Grimson, 1981] Grimson, W. E. L. (1981). *From Images to Surfaces.* MIT Press, Cambridge, MA.

[Hampel et al., 1986] Hampel, F., Rousseeuw, P., Ronchetti, E., and Stahel, W. A. (1986). *Robust statistics: the approach based on influence functions.* MIT Press, Cambridge, MA.

[Harris, 1986] Harris, J. G. (1986). The coupled depth/slope approach to surface reconstruction. Technical report, MIT Artificial Intelligence Laboratory, Cambridge, MA. Technical Report 908.

[Harris, 1987] Harris, J. G. (1987). A new approach to surface reconstruction: The coupled depth / slope model. In *IEEE Proceeding of the First International Conference on Computer Vision*, pages 277–283, London, England.

[Harris, 1988] Harris, J. G. (1988). Solving early vision problems with VLSI constraint networks. In *AAAI Neural Architectures for Computer Vision Workshop*, Minneapolis, Minnesota.

[Harris, 1989] Harris, J. G. (1989). An analog VLSI chip for thin plate surface interpolation. In Touretzky, D., editor, *Advance in Neural Information Processing Systems, Volume 1*, pages 687–694. Morgan Kaufmann, Palo Alto, CA.

[Harris et al., 1989] Harris, J. G., Koch, C., Luo, J., and Wyatt, J. (1989). Resistive fuses: analog hardware for detecting discontinuities in early vision. In Mead, C.and Ismail, M., editor, *Analog VLSI Implementations of Neural Systems.* Kluwer, Norwell, MA.

[Harris and Koch, 1989] Harris, J. G. and Koch, C. (1989). Resistive fuses: circuit implementations of line discontinuities in vision. Snowbird Neural Network Workshop.

[Harris et al., 1990a] Harris, J. G., Koch, C., and Luo, J. (1990a). A two-dimensional analog VLSI circuit for detecting discontinuities in early vision. *Science*, 248:1209–1211.

[Harris et al., 1990b] Harris, J. G., Koch, C., Staats, E., and Luo, J. (1990b). Analog hardware for detecting discontinuities in early vision. *Internat. Journal of Comp. Vision*, 4:211–223.

[Harris, 1991] Harris, J. G. (1991). Continuous-time segmentation networks. In *SPIE Technical Symposia on Optical Engineering and Photonics in Aerospace Sensing*, Orlando, FL. Visual Information Processing: From Neurons to Chips.

[Harris et al., 1991] Harris, J., Liu, S., and Mathur, B. (1991). Discarding outliers in a non-linear resistive network. In *International Joint Conference on Neural Networks*, Seattle, WA., to appear.

[Hasler and Neirynck, 1986] Hasler, M. and Neirynck, J. (1986). *Nonlinear Circuits.* Artech House Inc. pp. 172-173.

[Hildreth, 1984] Hildreth, E. C. (1984). *The Meausurement of Visual Motion.* The MIT Press, Cambridge, MA.

[Hopfield, 1984] Hopfield, J. J. (1984). Neurons with graded response have collective properties like those of two-state neurons. *Proceedings of the National Academy of Sciences USA*, 81:3088–3092.

[Hopfield and Tank, 1985] Hopfield, J. J. and Tank, D. W. (1985). Neural computations of decisions in optimization problems. *Biological Cybernetics*, 52:141–152.

[Horn, 1974] Horn, B. K. P. (1974). Determining lightness from an image. *Computer Graphics and Image Processing*, 3(1):277–299.

[Horn, 1988] Horn, B. K. P. (1988). Parallel networks for machine vision. Technical report, MIT Artificial Intelligence Laboratory, Cambridge, MA. AI Memo No. 1071.

[Horn and Schunck, 1981] Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

[Huang, 1963] Huang, K. (1963). *Statistical Mechanics.* John Wiley, New York.

[Huber, 1981] Huber, P. (1981). *Robust Statistics.* John Wiley & Sons.

[Hutchinson et al., 1988] Hutchinson, J., Koch, C., Luo, J., and Mead, C. (1988). Computing motion using analog and binary resistive networks. *IEEE Computer*, 21:52–63.

[Ikeuchi and Horn, 1981] Ikeuchi, K. and Horn, B. K. P. (1981). Numerical shape from shading and occluding boundaries. *Artif. Intell.*, 17:141–184.

[Karplus, 1958] Karplus, W. J. (1958). *Solution of Field Problems.* McGraw-Hill Book Company, Inc.

[Kayan, 1945] Kayan, C. (1945). An electrical geometrical analog for complex heat flow. *Trans ASME*, 67:713–718.

[Kirchoff, 1845] Kirchoff, G. (1845). Flow of an electric current through a plane particularly through a circular disk. *Ann. Phys. Chemie*, 64:499.

[Knight, 1983] Knight, T. (1983). *Design of an integrated optical sensor with On-Chip preprocessing.* PhD thesis, MIT, Cambridge, MA. Dept. of Electrical Engineering and Computer Science.

[Kobayashi et al., 1991] Kobayashi, H., White, J. L., and Abidi, A. A. (1991). An active resistor network for Gaussian filtering of images. *IEEE Journal of Solid-State Circuits*, 26(5):738–748.

[Koch et al., 1987] Koch, C., Marroquin, J., and Yuille, A. (1987). Analog "neuronal" networks in early vision. *Proc Natl. Acad. Sci. B. USA*, 83:4263–4267.

[Koch, 1989] Koch, C. (1989). Seeing chips: analog VLSI circuits for computer vision. *Neural Computation*, 1:184–200.

[Koch et al., 1990a] Koch, C., Harris, J., Horiuchi, T., Hsu, A., and Luo, J. (1990a). Real-time computer vision and robotics using analog VLSI circuits. In Touretzky, D., editor, *Neural Information Processing Systems.* Morgan Kaufmann, Palo Alto, CA.

[Koch et al., 1990b] Koch, C., Luo, J., and Mead, C. (1990b). Optical flow and surface interpolation in resistive networks:algorithms and analog VLSI chips. In Shriver, B., editor, *Neural Networks.* IEEE Computer Science Press Book. in press.

[Lazzaro and Mead, 1989] Lazzaro, J. and Mead, C. A. (1989). A silicon model of auditory localization. *Neural Computation*, 1:47–57.

[Levine, 1964] Levine (1964). *Methods for Solving Engineering Problems Using Analog Computers.* McGraw-Hill.

[Little et al., 1988] Little, J. J., Bülthoff, H. H., and Poggio, T. (1988). Parallel optical flow using local voting. In *IEEE Proceeding of the Second International Conference on Computer Vision*, Tarpon Springs, FL.

[Liu and Harris, 1989] Liu, S. C. and Harris, J. G. (1989). Generalized smoothing networks in solving early vision problems. In *IEEE Computer Vision and Pattern Recognition Conference*, San Diego, CA.

[Liu and Harris, 1991] Liu, S.-C. and Harris, J. G. (1991). Edge-enhancing resistive fuse. In *SPIE Technical Symposia on Optical Engineering and Photonics in Aerospace Sensing*, Orlando, FL. Visual Information Processing: From Neurons to Chips.

[Lumsdaine and Wyatt, 1991] Lumsdaine, A. and Wyatt, Jr., J. L. (1991). Nonlinear analog networks for image smoothing and segmentation. *Jour. VLSI Signal Processing.* To appear.

[Luo et al., 1989] Luo, J., Koch, C., and Mead, C. (1989). An experimental subthreshold, analog CMOS two-dimensional surface interpolation circuit. Neural Information Processing Systems Conference, Denver.

[Lyon and Mead, 1988] Lyon, R. F. and Mead, C. A. (1988). An analog electronic cochlea. *IEEE Trans. Acoust., Speech, Signal Processing*, 36:1119–1134.

[MacKay, 1991] MacKay, D. J. C. (1991). Bayesian interpolation. Unpublished Memo.

[Marr and Poggio, 1976] Marr, D. and Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, 194:283–287.

[Marr and Hildreth, 1980] Marr, D. and Hildreth, E. C. (1980). Theory of edge detection. *Proceedings of the Royal Society of London*, B(207):187–217.

[Marroquin, 1987] Marroquin, J. (1987). Probabilistic solution of inverse problems. Technical report, MIT Artificial Intelligence Laboratory, Cambridge, MA. AI Memo No. 860.

[Marroquin et al., 1987] Marroquin, J., Mitter, S., and Poggio, T. (1987). Probabilistic solution of ill-posed problems in computational vision. *J. Am. Statistic Assoc*, 82:76–89.

[Maxwell, 1891] Maxwell, J. C. (1891). *A Treatise on Electricity and Magnetism*, volume 1. Republished by Dover Publications, New York 1954, third edition.

[Mead, 1989] Mead, C. (1989). *Analog VLSI and Neural Systems.* Addison-Wesley.

[Mead and Mahowald, 1988] Mead, C. and Mahowald, M. (1988). A silicon model of early visual processing. *Neural Networks*, 1:91–97.

[Millar, 1951] Millar, W. (1951). Some general theorems for non-linear systems possessing resistance. *Phil. Mag.*, 42:1150–1160.

[Nordström, 1990] Nordström, K. N. (1990). Biased anisotropic diffusion: a unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8(4):318–327.

[Ortega and Rheinboldt, 1970] Ortega, J. and Rheinboldt, W. (1970). *Iterative solution of nonlinear equations in several variables.* Academic Press, New York.

[Payton, 1986] Payton, D. W. (1986). An architecture for reflexive autonomous vehicle control. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1838–1845, San Francisco, CA.

[Perona and Malik, 1990] Perona, P. and Malik, J. (1990). Scale-space and edge detection using anistropic diffusion. *IEEE Trans. Pattn. Anal. and Mach. Intell.*, 6:629–638.

[Platt, 1990] Platt, J. (1990). Constraint methods for neural networks and computer graphics. Dept. of Computer Science Technical Report Caltech-CS-TR-89-07, California Institute of Technology, Pasadena, CA.

[Poggio and Koch, 1984] Poggio, T. and Koch, C. (1984). An analog model of computation for the ill-posed problems of early vision. Technical report, MIT Artificial Intelligence Laboratory, Cambridge, MA. AI Memo No. 783.

[Poggio and Koch, 1985] Poggio, T. and Koch, C. (1985). Ill-posed problems in early vision: from computational theory to analogue networks. *Proc. R. Soc. Lond. B*, 226:303–323.

[Poggio et al., 1985] Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature*, 317:314–319.

[Poggio et al., 1986] Poggio, T., Voorhees, H., and Yuille, A. (1986). A regularized solution to edge detection. Technical report, MIT Artificial Intelligence Laboratory, Cambridge, MA. AI Memo No. 833.

[Poggio et al., 1988] Poggio, T., Gamble, E. B., and Little, J. J. (1988). Parallel integration of vision modules. *Science*, 242:436–440.

[Redshaw, 1948] Redshaw, S. (1948). An electric potential analyser. *Proc. Inst. Mech. Engrs. (london)*, 159:55–62.

[Sage, 1984] Sage, J. (1984). Gaussian convolution of images stored in a charge-coupled device. Quart. tech. rep., MIT Lincoln Laboratory, Lexington, MA. 1. August - 31. October 1983.

[Sage and Lattes, 1987] Sage, J. and Lattes, A. (1987). A high-speed two-dimensional CCD gaussian image convolver. Quart. tech. rep., MIT Lincoln Laboratory, Lexington, MA. 1. August - 31. October 1986.

[Schunck, 1989] Schunck, B. (1989). Robust computational vision. In *Robust methods in computer vision workshop.*

[Shulman and Hervé, 1989] Shulman, D. and Hervé, J.-Y. (1989). Regularization of discontinuous flow fields. In *IEEE Workshop on Visual Motion*, pages 81–86.

[Sivilotti et al., 1987] Sivilotti, M. A., Mahowald, M. A., and Mead, C. A. (1987). Real-time visual computation using analog CMOS processing arrays. In *1987 Stanford Conference on Very Large Scale Integration*, Cambridge, MA. MIT Press.

[Suter, 1990a] Suter, D. (1990a). Mixed finite element formulation of problems in visual reconstruction. Technical Report Technical Report No. 2/90,, La Trobe University.

[Suter, 1990b] Suter, D. (1990b). Visual reconstruction and edge detection using coupled deriviatives. Technical Report Technical Report No. 7/90,, La Trobe University.

[Szeliski, 1989] Szeliski, R. (1989). *Bayesian modeling of uncertainty in low-level vision.* Kluwer Academic Publishers, Boston, MA.

[Tanner, 1986] Tanner, J. E. (1986). *Integrated Optical Motion Detection.* PhD thesis, California Institute of Technology. 5223:TR:86.

[Tanner and Mead, 1986] Tanner, J. E. and Mead, C. (1986). An integrated analog optical motion sensor. In Kung, S. Y., editor, *VLSI Signal Processing, II*, pages 59–76, New York. IEEE Press.

[Terzopoulos, 1983] Terzopoulos, D. (1983). Multilevel computational processes for visual surface reconstruction. *Comp. Vision Graph. Image Proc.*, 24:52–96.

[Terzopoulos, 1986] Terzopoulos, D. (1986). Regularization of inverse visual problems involving discontinuities. *PAMI*, 8(4):413–424.

[Vittoz, 1985] Vittoz, E. A. (1985). Micropower techniques. In Tsividis, Y. and Antognetti, P., editors, *Design of MOS VLSI Circuits for Telecommunications*, chapter 4, pages 104–144. Prentice-Hall.

[White and Abidi, 1989] White, J. L. and Abidi, A. A. (1989). Parallel analog circuits for real time signal processing. In *Proceedings of the IEEE Symposium on Circuits and Systems*, pages 70–73.

[Wyatt et al., 1989] Wyatt, J. L., Jr., and Standley, D. L. (1989). Criteria for robust stability in a class of lateral inhibition networks coupled through resistive grids. *Neural Computation*, 1:58–67.

[Yang, 1990] Yang, W. (1990). A charge-coupled device architecture for on focal plane image signal processing. In *Proc. of International Symposium on VLSI Technology, Systems, and Applications*, pages 266–270.

[Zhou and Chellappa, 1988] Zhou, Y. and Chellappa, R. (1988). Computation of optical flow using a neural network. In *Proc.Intl. Conf. on Neural Networks*, pages 71–78, San Diego, CA,. vol 2,.