MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1280 January 1991

Nonlinear Analog Networks for Image Smoothing and Segmentation

A. Lumsdaine, J.L. Wyatt, Jr., and I.M. Elfadel

Abstract

Image smoothing and segmentation algorithms are frequently formulated as optimization problems. Linear and nonlinear (reciprocal) resistive networks have solutions characterized by an extremum principle. Thus, appropriately designed networks can automatically solve certain smoothing and segmentation problems in robot vision. This paper considers switched linear resistive networks and nonlinear resistive networks for such tasks. The latter network type is derived from the former via an intermediate stochastic formulation, and a new result relating the solution sets of the two is given for the "zero termperature" limit. We then present simulation studies of several continuation methods that can be gracefully implemented in analog VLSI and that seem to give "good" results for these non-convex optimization problems.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124 and under the contract number NSF-MIP 8814612.

1 Introduction

One of the most important, yet most difficult, early vision tasks is that of image smoothing and segmentation. Smoothing is necessary to remove noise from an input image so that reliable processing in subsequent stages is facilitated. However, indiscriminate smoothing will blur the entire image, including edges (e.g., corresponding to object boundaries) which are necessary for later stages of processing. Many researchers are currently seeking to develop algorithms that smooth in a piecewise manner, respecting edges. There are two main approaches taken — stochastic, [1] - [6], and deterministic [7] - [9]. The former relies on such methods as simulated annealing to accomplish the minimization. The deterministic approach, on the other hand, often relies on the application of continuation methods [2], [10] to certain nonlinear systems, or in the case of [11], on using a neural network similar to that of Tank and Hopfield [12].

Although efficient computation techniques exist for numerically computing the solutions to vision problems [13], even the fastest algorithms running on a parallel supercomputer (such as the Connection Machine ® system[1] [14]) do not approach real-time performance. The motivation of this work is to produce solutions to the smoothing and segmentation problem that are amenable to analog VLSI network implementation, an area that has been explored in [15] - [18]. See also [11], [19], [20].

Section 2 presents the smoothing and segmentation task as a minimization problem. Section 3 presents methods for solving the minimization problem and discusses network implementations of these methods. Simulation results are provided in Section 4. Finally, conclusions and suggestions for further research are given in Section 5.

[1] Connection Machine is a registered trademark of Thinking Machines Corporation

2 Image Restoration as a Minimization Problem

The difficulty with using a *linear* network for image smoothing is that noise and signal are equally smoothed so that edges become blurred. We therefore seek a method for segmenting the signal into regions which can be smoothed separately. One technique for doing this is to introduce a *line process* (i.e., a set of binary variables) which selectively breaks the smoothness constraint at given locations. This method appears widely in the literature, e.g., [1] - [4], [6], [11].

For simplicity of notation, all equations in this paper are formulated for the one-dimensional case. The results generalize trivially to two dimensions, and the simulation results are for the two-dimensional case.

The smoothing and segmentation problem with the line process can be treated as a minimization problem. Let $\mathbf{u} \in \Re^N$ be the input image, $\mathbf{y} \in \Re^N$ be the output image, and $\mathbf{l} \in \Re^{N-1}$ be the line process, where the binary line process variable l_i assumes the values $\{0, 1\}$ depending on whether the smoothness penalty between nodes i and $i+1$ is enforced or not. Consider the following cost function:

$$J_{\mathbf{u}}(\mathbf{y}, \mathbf{l}) \triangleq \frac{1}{2}\left[F_{\mathbf{u}}(\mathbf{y}) + S(\mathbf{y}, \mathbf{l}) + H(\mathbf{l})\right] \tag{1}$$

where F, S, and H are the "fidelity," "smoothness," and "line" penalty terms, respectively, i.e.,

$$F_{\mathbf{u}}(\mathbf{y}) \triangleq \lambda_f \sum_{i=1}^{N} (y_i - u_i)^2 \tag{2}$$

$$S(\mathbf{y}, \mathbf{l}) \triangleq \lambda_s \sum_{i=1}^{N-1} (y_i - y_{i+1})^2 (1 - l_i) \tag{3}$$

$$H(\mathbf{l}) \triangleq \lambda_h \sum_{i=1}^{N-1} l_i. \tag{4}$$

This formulation assumes that the optimal reconstructed image and edges $(\mathbf{y}_{opt}, \mathbf{l}_{opt})$ satisfies as well as possible the generally conflicting requirements of agreeing with the data $\mathbf{u}$, being smooth between edges, and containing as few edges as possible. The parameters λ_f, λ_s, and λ_h determine the weights given to each of these criteria.

The expression (1) can be minimized with respect to $\mathbf{y}$ for fixed $\mathbf{l}$ by differentiating with respect to each y_i and setting the derivatives to zero. This produces the following system of equations:

$$\lambda_f(y_i - u_i) + \lambda_s(y_i - y_{i-1})(1 - l_{i-1}) + \lambda_s(y_i - y_{i+1})(1 - l_i) = 0, \tag{5}$$

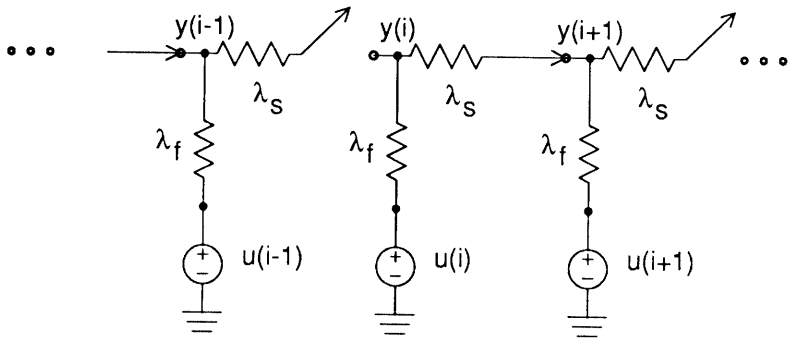


Figure 1: A simple smoothing network with switches. The vertical and horizontal resistors have conductances λ_f and λ_s, respectively.

with appropriate modifications at the boundaries $i = 1$ and $i = N$.

The new results in Section 2.1 all follow from the observation that (5), as well as the related equation (9) below, describe the solution to certain resistive electrical networks. Notice that (5) can be viewed as the Kirchoff's current law (KCL) relation at every node of a resistive ladder in which the horizontal resistive elements have switches (corresponding to a line process element) associated with them. A network for computing $\mathbf{y}$ given $\mathbf{l}$ is shown in Figure 1. Note that $(F + S)$ is the electrical power dissipated in the resistors. Similar networks have appeared in [6] and [11]. This type of network will be referred to as a resistor-with-switch (or RWS) network. For any setting of the switches, the network solution automatically minimizes the cost function with respect to $\mathbf{y}$. The difficulty is in minimizing with respect to $\mathbf{l}$.

Much has been said in the literature in regard to finding a global minimum to (1) by stochastic and deterministic methods. These techniques are necessary to find the minimizing $\mathbf{l}$ — minimizing with respect to $\mathbf{y}$ *given* $\mathbf{l}$ only requires the solution of a linear system. The deterministic approaches rely on the fact that the minimization problem can be recast into one in which the line process variables have been eliminated. The latter will be studied here since they appear to lead to practical VLSI implementations.

2.1 Discontinuous Resistive Fuse Elements

The line process variables can be removed from (1) by straightforward algebraic manipulations. In fact, Blake and Zisserman [7] demonstrated that the original cost function $J_{\mathrm{u}}(\mathbf{y}, \mathbf{l})$ containing real and boolean variables is intimately related to

the following cost function containing only real variables:

$$K_{\mathbf{u}}(\mathbf{y}) \triangleq \frac{1}{2}\left[\lambda_f \sum_{i=1}^{N}(y_i - u_i)^2 + \sum_{i=1}^{N-1} G(y_i - y_{i+1})\right], \tag{6}$$

where

$$G(v) \triangleq \min_{l\in\{0,1\}}\left\{\lambda_s v^2(1-l) + \lambda_h l\right\} = \begin{cases} \lambda_s v^2, & |v| < \sqrt{\frac{\lambda_h}{\lambda_s}} \\ \lambda_h, & \text{otherwise} \end{cases}. \tag{7}$$

The line process is found *a posteriori* according to:

$$l_i = \begin{cases} 0, & |y_i - y_{i+1}| < \sqrt{\frac{\lambda_h}{\lambda_s}} \\ 1, & \text{otherwise} \end{cases}. \tag{8}$$

Note that $K_{\mathbf{u}}$ is a *non-convex* cost function with respect to $\mathbf{y}$.

Apart from instances in which solutions occur at points where G is not differentiable, the minimum of $K_{\mathbf{u}}$ is to be found among those points where $\nabla K_{\mathbf{u}}(\mathbf{y}) = 0$, i.e.,

$$\lambda_f(y_i - u_i) + g(y_i - y_{i-1}) + g(y_i - y_{i+1}) = 0, \tag{9}$$

where $g(v) = \frac{1}{2}\frac{d}{dv}G(v)$.

Equation (9) can also be viewed as the KCL relation at each node of a nonlinear resistive network with the topology illustrated in Figure 2. The nonlinear resistor characteristic, $g(v)$, is that of a linear resistor that reversibly becomes an open circuit when the voltage across it exceeds a certain threshold, as shown in Figure 3. Then, in electrical terms, G is twice the *co-content* function for this nonlinear resistor [16],[21], i.e., $G(v) = 2\int_0^v g(u)\,du$. We refer to an element of this type as a *discontinuous resistive fuse* and to a network incorporating resistive fuses as an RWF network, i.e., a *resistor-with-fuse* network.

For a given cost function, one can construct corresponding RWS and RWF networks. For every solution of an RWF network, there exists a similar solution to the corresponding RWS network, but there are switch configurations of an RWS network for which there is *no* corresponding solution in a corresponding RWF network. The question then arises whether restricting attention to the RWF network might cause one to overlook a solution to the RWS network that is in fact a local minimum and therefore of potential use in an optimization procedure. The answer is no, by the following proposition:

Proposition 1 Consider the cost function $J_{\mathbf{u}}(\mathbf{y},\mathbf{l})$ as specified in (1) for a one- or two-dimensional network and the corresponding RWS and RWF networks specified by (5) and (9). If the switches are set so that the solution $\mathbf{y}^*$ for the RWS network

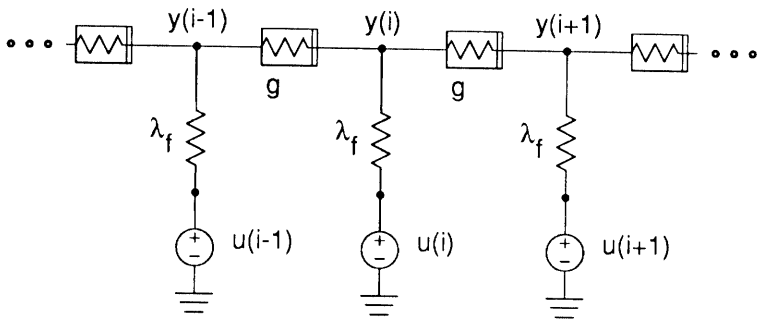


Figure 2: Resistor-With-Fuse (RWF) network topology.

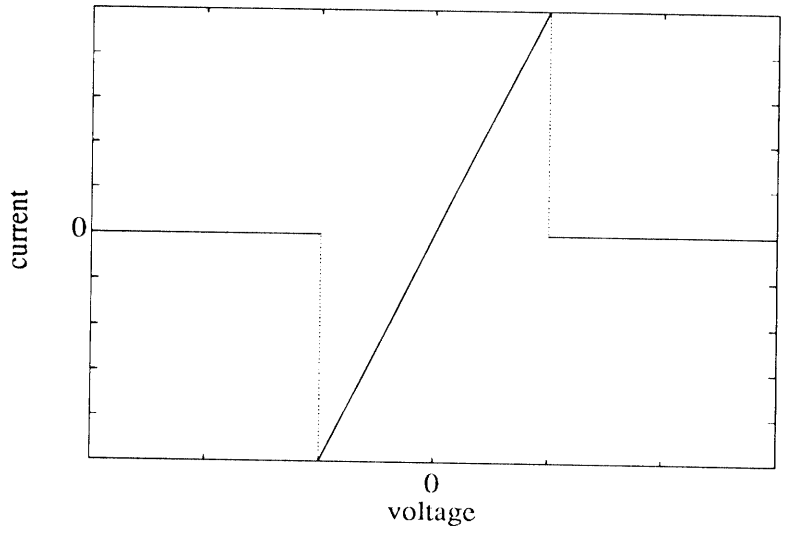


Figure 3: Characteristic of the discontinuous nonlinear resistor known as a "resistive fuse." The dotted vertical lines are *not* part of the constitutive relation.

is not also a solution to the RWF network, then $\mathbf{y}^*$ is *not* a local minimum to $J_{\mathbf{u}}$, meaning that changing the setting of a *single* (appropriately selected) switch in the RWS network will produce a new solution (with a new value of $\mathbf{y}$) for which the value of $J_{\mathbf{u}}$ is strictly lower.

Remark Proposition 1 differs from the result in [7] (pp. 112-113) in that it concerns *local* minima and applies *after* the network has settled to a new $\mathbf{y}$ following the closing of the switch.

In order to complete the proof, we need the following lemma:

Lemma 1 (Local Measurement Principle) Consider a one-dimensional or two-dimensional network of the type shown in Figure 1, in which an arbitrary number of switches (≥ 1) are open. The *power dissipated* in the network is given by

$$\begin{aligned} P &= \lambda_f \sum_{i=1}^{N} (y_i - u_i)^2 + \lambda_s \sum_{i=1}^{N-1} (y_i - y_{i+1})^2 (1 - l_i) \\ &= (F + S). \end{aligned} \tag{10}$$

For a given switch, let P^- be the value of P with the switch open, P^+ be the value of P with the switch closed, and define the change in the dissipated power as $\Delta P = P^+ - P^-$. Let v_{oc} be the voltage across the switch when it is open and let i_{sc} be the current through the switch when it is closed (after the network has settled). Then the increase in dissipation which results from closing the switch is

$$\Delta P = v_{oc} i_{sc} > 0. \tag{11}$$

Remark Lemma 1 is a startling result. The local measurement principle states that one can measure the *global* change in the network cost function due to a switch change (after the network settles to a new solution) merely by taking two measurements *at the switch*. Both proofs below use circuit theory techniques, but can also be carried out, albeit laboriously, by mathematical arguments divorced from a network realization, e.g., the proof of Lemma 1 via a rank one perturbation method in [22]. Related work appears in [23] and [24].

Proof of Lemma 1 Define v_k^- and i_k^- to be the network branch voltages and currents when the switch is open. Define v_k and i_k to be the network branch voltages and branch currents when the switch is closed. Define $\Delta v_k = v_k - v_k^-$ and $\Delta i_k = i_k - i_k^-$. By Tellegen's theorem [25],[26],

$$\sum_{\substack{\text{all} \\ \text{branches}}} [v_k \Delta i_k - i_k \Delta v_k] = 0. \tag{12}$$

Group the terms in (12) according to branch element, and note that

$$\sum_{\substack{\text{voltage}\\ \text{sources}}} [v_k \Delta i_k - i_k \Delta v_k] \quad + \quad \sum_{\text{resistors}} [v_k \Delta i_k - i_k \Delta v_k] \tag{13}$$
$$+ \sum_{\substack{\text{fixed}\\ \text{switches}}} [v_k \Delta i_k - i_k \Delta v_k] \quad + \quad v_{sw} \Delta i_{sw} - i_{sw} \Delta v_{sw} = 0,$$

where the subscript "sw" refers to the switch that is being closed and "fixed switches" to all others. To simplify (13), note that $\Delta v_k = 0$ for the voltage sources, v_k and Δv_k vanish for closed switches, i_k and Δi_k vanish for open switches, and for the resistors:

$$v_k \Delta i_k - i_k \Delta v_k = R_k i_k \Delta i_k - i_k R_k \Delta i_k = 0. \tag{14}$$

Equation (13) then becomes

$$\begin{aligned} 0 &= \sum_{\substack{\text{voltage}\\ \text{sources}}} [v_k \Delta i_k] + v_{sw} \Delta i_{sw} - i_{sw} \Delta v_{sw} \\ &= \sum_{\substack{\text{voltage}\\ \text{sources}}} [v_k \Delta i_k] + v_{sw}(i_{sw} - i_{sw}^-) - i_{sw}(v_{sw} - v_{sw}^-) \\ &= \sum_{\substack{\text{voltage}\\ \text{sources}}} [v_k \Delta i_k] + v_{sw}^- i_{sw} \end{aligned} \tag{15}$$

The summation term in (15) is just the change in power delivered to the network, i.e., $-\Delta P$, and $v_{sw}^- i_{sw} = v_{oc} i_{sc}$. Therefore,

$$\Delta P = v_{oc} i_{sc}. \tag{16}$$

∎

Proof of Proposition 1 Consider any RWS network with any input $\mathbf{u}$, switch configuration $\mathbf{l}$, and corresponding network solution $\mathbf{y}^*$, such that $\mathbf{y}^*$ is not a solution of the corresponding RWF network. Then there must exist *some* resistor-switch composite element (element q, say), such that $\mathbf{y}^*$ is no longer a network solution if a resistive fuse is substituted in its place. Make such a substitution and then consider the load-line describing the remainder of the linear RWS network as seen from this location. The two possible cases marked in Figure 4 are Case X, in which switch q was open in the original RWS network, and Case W, in which switch q was closed. Note that the area in the first quadrant under the triangle is $\frac{1}{2}\lambda_h$. In Case X, closing the switch in the original RWS network would have caused the solution to move to the circled point on line A. By Lemma 1 the change would be

$$J_{\mathbf{u},closed} - J_{\mathbf{u},open} = \frac{1}{2}[v_{oc} i_{sc} - \lambda_h] < 0, \tag{17}$$

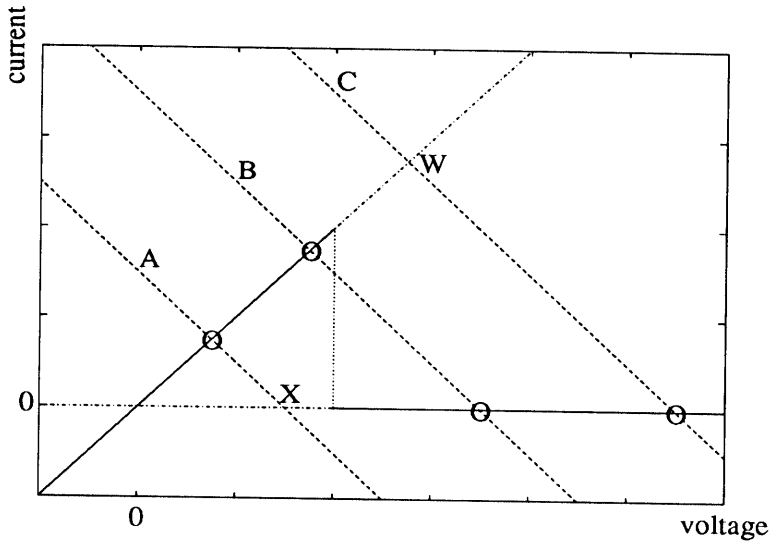


Figure 4: Load-Line diagram for RWS network and for RWS network with resistive fuse substituted for one resistor-switch composite element. The dashed lines A, B, and C are possible load-lines representing the behavior of an RWS network as seen by one resistor-switch pair. The six marked points indicate possible solutions, depending on switch position. If a resistive fuse element (with characteristic $i = g(v)$, shown with a solid line) is substituted for the resistor-switch pair, the four circled solutions remain, while the solutions marked X and W disappear.

where the inequality follows from the fact that $\frac{1}{2}v_{oc}i_{sc}$ (the area under the line connecting the origin to (v_{oc}, i_{sc})) is less than $\frac{1}{2}\lambda_h$ (the area under the triangle). For Case W, similar reasoning shows $J_{\mathbf{u},open} - J_{\mathbf{u},closed} < 0$ if opening the switch causes the network solution to move from point W to the circled point on line C. Thus points X and W in Figure 4 are not local minima of $J_{\mathbf{u}}$. ∎

Remark The converse of the proposition is not true. If the network solution lies on load-line B, one intersection point or the other will generally have lower cost for the RWS network, yet both are valid solutions to the RWF network.

2.2 Continuous Resistive Fuse Elements

We will show that for the purposes of numerical optimization and physical (VLSI) implementation it is advantageous to replace the discontinuous resistive fuse element in Figure 3 by a controllable element with a single-parameter family of i-v

curves, such as those drawn in dotted lines in Figure 7. Elements of this general type have also been used in analog VLSI circuits for image enhancement. To the best of our knowledge, the first related circuit appeared in [15] and had a *monotone*, saturating characteristic of the general form $i = a\tanh(bv)$. John Harris at Caltech has invented the first *nonmonotone* circuit element of this type, named it a *resistive fuse*, and built image processing networks using it in analog VLSI [16] - [18]. More recently, Steve Decker, Hae-Seung Lee, and John Wyatt at MIT have developed more compact nonmonotone continuous resistive fuse circuits using fewer transistors.

The behavior of the nonmonotone fuses in the network in Figure 2 is intuitively easy to understand. In a smooth region of the image where the input $\mathbf{u}$ is nearly constant, only the linear portion of the fuse curve near the origin is excited, i.e., the fuse acts essentially as a linear smoothing element. But at any point where a discontinuity in the input occurs, i.e., where $|u_i - u_{i-1}|$ is sufficiently large, the fuse current becomes quite small and little smoothing results. An extremum formulation of this behavior is given in [16].

A fundamental question is whether any rigorous relationship can be found that connects the continuous nonmonotone fuse curves in Figure 7 with the discontinuous fuses in Figure 3 or the switches in Figure 2. The surprising answer is *yes*, due to a remarkable result of Geiger and Girosi, based on a stochastic formulation of the problem [1]. Section 2.3 gives the necessary background and Section 2.4 gives a variant on their approach, based on a formulation in terms of a marginal probability distribution function.

2.3 Stochastic Formulation of the Image Smoothing and Segmentation Problem

This section shows that the deterministic minimization problem in (1) - (4) is in fact the optimum image reconstruction procedure in a particular *probabilistic* formulation of the problem (see Figure 5). In this formulation, the original image brightness and discontinuities are modeled as a pair of random vectors $(\mathbf{B}, \mathbf{D})$. We cannot directly measure $(\mathbf{B}, \mathbf{D})$ but have to work instead with a noisy observation $\mathbf{U}$ of the brightness values alone.

Remark The notation in this section follows the one classically used in probability theory, where the random variables or vectors are denoted by uppercase letters $(\mathbf{B}, \mathbf{D}, \mathbf{U})$ while the values taken by them are denoted by lowercase letters $(\mathbf{b}, \mathbf{d}, \mathbf{u})$. In (18), $p(\mathbf{X} = \mathbf{x})$ is denoted by $p(\mathbf{x})$ and $p(\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z})$ is denoted by $p(\mathbf{x}|\mathbf{z})$ to simplify notation.

For this analysis, it is assumed that the probability distribution for $(\mathbf{B}, \mathbf{D})$ is

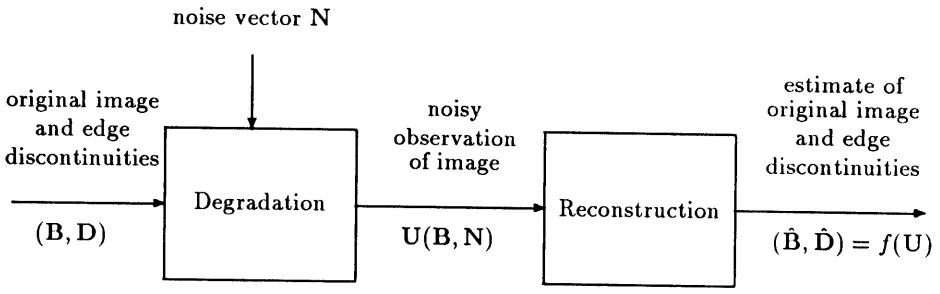


Figure 5: The original image one wishes to reconstruct is modeled as a random brightness vector $\mathbf{B}$ and a random binary edge discontinuity vector $\mathbf{D}$, where $D_i = 1$ if there is an edge between pixels i and $(i+1)$. The observation $\mathbf{U}$ is a version of $\mathbf{B}$ degraded by observation noise $\mathbf{N}$, and the reconstruction algorithm produces an estimate $(\hat{\mathbf{B}}, \hat{\mathbf{D}})$ of $(\mathbf{B}, \mathbf{D})$. In our particular example, the function f is the optimization procedure specified in (23) - (25).

given by

$$\begin{aligned} p(\mathbf{b},\mathbf{d}) &= p(\mathbf{b}|\mathbf{d})p(\mathbf{d}) \\ &= \left\{c_s(\beta\lambda_s)e^{-\beta S(\mathbf{b},\mathbf{d})}\right\}\left\{c_h(\beta\lambda_h)e^{-\beta H(\mathbf{d})}\right\}, \end{aligned} \tag{18}$$

where the functions S and H are given in (3) and (4), respectively, and where c_s and c_h insure that $p(\mathbf{b},\mathbf{d})$ and $p(\mathbf{d})$ have unit area. The vector $\mathbf{D}$ represents a finite Bernoulli sequence. More specifically, the components of the binary random vector $\mathbf{D}$ are independent, identically distributed binary random variables with a probability mass function

$$p(D_i = d_i) = \frac{e^{-\beta\lambda_h d_i}}{1 + e^{-\beta\lambda_h}}, \tag{19}$$

so that the joint probability mass function of $\mathbf{d}$ is given by

$$\begin{aligned} p(\mathbf{D} = \mathbf{d}) &= \prod_{i=1}^{N-1} \frac{e^{-\beta\lambda_h d_i}}{1 + e^{-\beta\lambda_h}} \\ &= \frac{e^{(-\beta\lambda_h \sum_{i=1}^{N-1} d_i)}}{(1 + e^{-\beta\lambda_h})^{N-1}} = c_h(\beta\lambda_h)e^{-\beta H(\mathbf{d})}, \end{aligned} \tag{20}$$

which decreases with the total number of discontinuities in the scene. The brightness vector $\mathbf{B}$ is a Gaussian random vector dependent on $\mathbf{D}$: specifically, the elements of the vector taken as a sequence represent, in the one-dimensional case, discrete Brownian motion with a uniformly distributed initial value at $i = 1$ and at the right of each discontinuity location. In the absence of a discontinuity between pixels i and $(i+1)$, the standard deviation of $B_{i+1} - B_i$ is $\sigma = 1/\sqrt{2\beta\lambda_s}$.

We also assume the observations $\mathbf{U}$ are distributed as

$$p(\mathbf{u}|\mathbf{b}) = c_f(\beta\lambda_f)e^{-\beta F_{\mathbf{u}}(\mathbf{b})}, \tag{21}$$

where the function $F_{\mathbf{u}}$ is given in (2) and where c_f normalizes $p(\mathbf{u}|\mathbf{b})$ to unit area. Equation (21) is equivalent to assuming that the observations are corrupted by additive, independent Gaussian noise, i.e., $U_i = B_i + N_i$, where N_i is a Gaussian random variable with zero mean and standard deviation $\sigma = 1/\sqrt{2\beta\lambda_f}$, and $\mathbf{N}$ is independent of $(\mathbf{B}, \mathbf{D})$. The variable β is actually redundant because scaling β is equivalent to scaling λ_f, λ_s, and λ_h. Since increasing β will reduce all the variances, $\frac{1}{\beta}$ is analogous to temperature in statistical mechanics.

Remark More precisely, $\mathbf{B}$ is Brownian motion and $\mathbf{N}$ is Gaussian if the allowed values for each B_i and each N_i are continuous and unconstrained. But in that case the description of $\mathbf{B}$ is flawed because a uniformly distributed initial value over the whole real line is not a normalizable probability distribution. This problem vanishes if the allowed brightness values are discrete and finite in number or continuous and bounded, e.g., $0 \leq B_i \leq B_{max}$.

How should we best attempt to reconstruct $(\mathbf{B}, \mathbf{D})$ given $\mathbf{U}$? Given both the *a priori* (i.e., *prior* to a particular observation) probability distribution $p(\mathbf{b}, \mathbf{d})$ of the image with discontinuities, and the noise-produced conditional distribution $p(\mathbf{u}|\mathbf{b}, \mathbf{d})$ of the observation, Bayes rule [27] gives the *a posteriori* distribution (i.e., the conditional distribution *after* noisy observation) of the image and discontinuities by the formula

$$p(\mathbf{b}, \mathbf{d}|\mathbf{u}) = \frac{p(\mathbf{u}|\mathbf{b}, \mathbf{d})p(\mathbf{b}, \mathbf{d})}{p(\mathbf{u})}. \tag{22}$$

Maximum a posteriori estimation is a reconstruction technique that chooses $(\hat{\mathbf{b}}, \hat{\mathbf{d}})$ as the value of $(\mathbf{b}, \mathbf{d})$ that maximizes $p(\mathbf{b}, \mathbf{d}|\mathbf{u})$, i.e.,

$$\begin{aligned}(\hat{\mathbf{b}}, \hat{\mathbf{d}}) = f(\mathbf{u}) &= \arg\max_{\mathbf{b},\mathbf{d}} p(\mathbf{b}, \mathbf{d}|\mathbf{u}) \\ &= \arg\max_{\mathbf{b},\mathbf{d}} \left[\frac{p(\mathbf{u}|\mathbf{b}, \mathbf{d})p(\mathbf{b}, \mathbf{d})}{p(\mathbf{u})}\right] \\ &= \arg\max_{\mathbf{b},\mathbf{d}} [p(\mathbf{u}|\mathbf{b})p(\mathbf{b}, \mathbf{d})],\end{aligned} \tag{23}$$

where the last equality holds because the denominator is independent of $(\mathbf{b},\mathbf{d})$ and the observation $\mathbf{u}$ depends directly on the brightness levels alone — $\mathbf{d}$ is not measured but only statistically inferred. Using (18) and (21) in the last line of (23), we have

$$(\hat{\mathbf{b}},\hat{\mathbf{d}}) = \arg\max_{\mathbf{b},\mathbf{d}} \left\{ c_f c_s c_h e^{-\beta[F_{\mathbf{u}}(\mathbf{b})+S(\mathbf{b},\mathbf{d})+H(\mathbf{d})]} \right\}. \tag{24}$$

Remark There is a useful distinction between the $(\mathbf{y},\mathbf{l})$ notation and the $(\mathbf{b},\mathbf{d})$ notation. In the deterministic picture, $(\mathbf{y},\mathbf{l})$ represents algorithm or circuit variables over which we are attempting to optimize. A particular network solution $(\mathbf{y},\mathbf{l})$ *does* describe the circuit behavior but may or may not bear any simple relation to $(\mathbf{b},\mathbf{d})$. The stochastic picture adds a new quantity not present in the earlier deterministic story: the original uncorrupted random image-discontinuity pair $(\mathbf{B},\mathbf{D})$. The variables $(\mathbf{b},\mathbf{d})$ always refer to possible values of $(\mathbf{B},\mathbf{D})$ and may or may not relate directly to circuit behavior. Without this dual notation the variables would misleadingly be used to describe both original image-discontinuity pairs and also node voltages and switch positions inside an electrical network.

Substituting our previous deterministic notation $(\mathbf{y},\mathbf{l})$ for $(\mathbf{b},\mathbf{d})$, we recover

$$(\hat{\mathbf{b}},\hat{\mathbf{d}}) = (\mathbf{y}_{opt},\mathbf{l}_{opt}) = \arg\min_{\mathbf{y},\mathbf{l}} \{F_{\mathbf{u}}(\mathbf{y}) + S(\mathbf{y},\mathbf{l}) + H(\mathbf{l})\} \tag{25}$$

as in (1) - (4).

In conclusion, the optimization problem in (1) - (4) yields as its solution the *maximum a posteriori estimator* of the original image brightness and discontinuity vectors $(\mathbf{B},\mathbf{D})$, assuming the *a priori* distribution (18) and the observation noise model (21).

2.4 Derivation of the Continuous Fuse from the Resistor-with-Switch Network in a Probabilistic Formulation

In the Bayesian formulation, one first calculated the *a posteriori* distribution

$$p(\mathbf{b},\mathbf{d}|\mathbf{u}) = c_f c_s c_h e^{-\beta[F_{\mathbf{u}}(\mathbf{b})+S(\mathbf{b},\mathbf{d})+H(\mathbf{d})]}, \tag{26}$$

and then attempted to maximize it over $(\mathbf{b},\mathbf{d})$. If one wishes to reconstruct only the intensities but not the discontinuity locations, it is appropriate to maximize the simpler *marginal a posteriori* distribution $p(\mathbf{b}|\mathbf{u})$ over $\mathbf{b}$, where

$$p(\mathbf{b}|\mathbf{u}) \triangleq \sum_{\mathbf{d}\in\mathcal{C}} p(\mathbf{b},\mathbf{d}|\mathbf{u}), \tag{27}$$

and $\mathcal{C}$ is the hypercube of all 2^{N-1} possible binary $\mathbf{d}$-vectors. The form of this density will specify the nonlinear continuous fuse characteristics. The sum above was first calculated by Geiger and Girosi in [1], and a somewhat more detailed derivation is given below.

Lemma 2 The marginal *a posteriori* distribution $p(\mathbf{b}|\mathbf{u})$ over $\mathbf{b}$ is given by:

$$p(\mathbf{b}|\mathbf{u}) = c_1 e^{-\beta[F_{\mathbf{u}}(\mathbf{b})+J_2(\mathbf{b})]} \tag{28}$$

where c_1 is a normalizing constant, $F_{\mathbf{u}}(\mathbf{b})$ is given in (2), and

$$J_2(\mathbf{b}) = \frac{1}{\beta}\sum_{i=1}^{N-1} \ln\left(\frac{1+e^{\beta\lambda_h}}{1+e^{\beta[\lambda_h-\lambda_s(b_i-b_{i+1})^2]}}\right). \tag{29}$$

The proof of the lemma requires the following fact, which can easily be verified.

Fact Let $\mathbf{a} = (a_1, \ldots, a_n)$ be a vector of n binary variables, $a_i \in \{0, 1\}$, and let $\mathcal{A}$ be the set of all such vectors. Then for any $\mathbf{r} \in \Re^n$,

$$\sum_{\mathbf{a}\in\mathcal{A}} e^{\mathbf{a}\cdot\mathbf{r}} = \prod_{i=1}^{n}(1+e^{r_i}), \tag{30}$$

where $\mathbf{a}\cdot\mathbf{r}$ is the standard inner product.

Proof of Lemma 2

Substituting (26) into (27) yields

$$\begin{aligned} p(\mathbf{b}|\mathbf{u}) &= \sum_{\mathbf{d}\in\mathcal{C}} p(\mathbf{b},\mathbf{d}|\mathbf{u}) \\ &= c_f c_s c_h \sum_{\mathbf{d}\in\mathcal{C}} e^{-\beta[F_{\mathbf{u}}(\mathbf{b})+S(\mathbf{b},\mathbf{d})+H(\mathbf{d})]}. \end{aligned} \tag{31}$$

The terms being summed in (31) can be decomposed as follows:

$$\exp\left(-\beta\left[\lambda_f\sum_{i=1}^{N}(b_i-u_i)^2+\lambda_s\sum_{i=1}^{N-1}(b_i-b_{i+1})^2\right]\right)\exp\left(-\beta\sum_{i=1}^{N-1}d_i\left[\lambda_h-\lambda_s(b_i-b_{i+1})^2\right]\right). \tag{32}$$

Using (30), the second exponential term in (32) sums over $\mathbf{d}\in\mathcal{C}$ to

$$\exp\left(-\beta\left[-\frac{1}{\beta}\sum_{i=1}^{N-1}\ln(1+e^{-\beta[\lambda_h-\lambda_s(b_i-b_{i+1})^2]})\right]\right) \tag{33}$$

and further algebraic manipulation shows that

$$p(\mathbf{b}|\mathbf{u}) = c_f c_s c_h e^{-\beta[F_{\mathbf{u}}(\mathbf{b})+\tilde{J}_2(\mathbf{b})]}, \tag{34}$$

where

$$\tilde{J}_2(\mathbf{b}) = \frac{1}{\beta}\sum_{i=1}^{N-1} \ln\left(\frac{1}{1+e^{\beta[\lambda_h-\lambda_s(b_i-b_{i+1})^2]}}\right) + (N-1)\lambda_h. \tag{35}$$

Absorbing an additive term into the normalizing constant $c_f c_s c_h$, the lemma was stated in (29) in terms of

$$J_2(\mathbf{b}) \triangleq \tilde{J}_2(\mathbf{b}) + \frac{N-1}{\beta}\left[\ln\left(1+e^{\beta\lambda_h}\right) - \beta\lambda_h\right], \tag{36}$$

which is constructed so that $J_2(\mathbf{0}) = 0$. This is a necessary step if we are to later interpret $J_2(\mathbf{b})$ as the co-content function of a set of nonlinear resistors. ∎

The marginal distribution in (28) suggests a new cost function

$$K_{\mathbf{u}}^{\beta}(\mathbf{y}) = \frac{1}{2}\left[F_{\mathbf{u}}(\mathbf{y}) + J_2(\mathbf{y})\right], \tag{37}$$

in which the line process variables have been eliminated. The local minima of $K_{\mathbf{u}}^{\beta}(\mathbf{y})$ are obtained from the set of points satisfying

$$\nabla K_{\mathbf{u}}^{\beta}(\mathbf{y}) = 0. \tag{38}$$

Taking the i-th component of (38) gives:

$$\lambda_f(y_i - u_i) + g_\beta(y_i - y_{i-1}) + g_\beta(y_i - y_{i+1}) = 0, \tag{39}$$

where

$$g_\beta(v) = \frac{\lambda_s v}{1+e^{-\beta(\lambda_h-\lambda_s v^2)}}. \tag{40}$$

Equation (39) can be considered the KCL relation at each node of a nonlinear network having vertical linear resistive elements with conductance λ_f and horizontal nonlinear elements with constitutive relation $i = g_\beta(v)$. In this case, $K_{\mathbf{u}}^{\beta}(\mathbf{y})$ is the *total co-content* of the network. Notice that as $\beta \to \infty$, we recover the RWF network, i.e., $K_{\mathbf{u}}^{\beta}(\mathbf{y}) \to K_{\mathbf{u}}(\mathbf{y})$. Moreover, we have defined a family of β-dependent resistive elements, illustrated in Figure 7, that can be used in continuation methods.

3 Solution Methods

The resistive fuse and marginal distribution approaches produced switch-free nonlinear networks with identical topologies (see Figure 2) but with different constitutive relations for the nonlinear elements. For either network, multiple solutions generally exist. On the theoretical side this is a difficulty because we are trying to find the global minimum of a specific cost function. On the practical side this is a difficulty because the solution that is obtained by a physical network realization will depend strongly on such things as parasitic capacitances and other characteristics of the network over which we have little control. We therefore seek some modification of the network that will allow us to exercise some control over the solution it finds. In this section, we apply continuation methods to the nonlinear smoothing and segmentation networks.

3.1 Example — A Special Case

The simplest special case that nonetheless provides insight into the phenomenon of multiple solutions is the response of a one-dimensional network to a step edge input, i.e.,

$$u_i = \begin{cases} u_{hi} & i \leq k \\ u_{lo} & i > k \end{cases}, \tag{41}$$

for some $k < N$. This corresponds to a step of $u_{hi} - u_{lo} > 0$ between nodes k and $k+1$ and serves as a model for the simplest two-dimensional edge, i.e., a step that extends across the entire network and is parallel to one of the network "axes."

For the step input described above, the one-dimensional network has a simple circuit equivalent, shown in Figure 6. The simplification proceeds as follows. First, we *assume* that the signal is "well-smoothed" on either side of the step so that each nonlinear element can be replaced by an equivalent linear resistance whose value is the incremental resistance of the nonlinear element about zero volts. The network elements on either side of the step are then replaced by their Thevenin equivalents, which are combined into a single linear element and voltage source. The simplified network will be referred to as the zero-dimensional case. Analysis of the behavior of the network to a step input is reduced to solving the KCL equation at one node: some insight into the circuit behavior can be gained by using load-line techniques (see Figures 6 – 8).

This "linear load-line assumption" holds *exactly* only for the RWS network with fixed switch positions and for the marginal distribution network with $\beta = 0$. For the RWF network and for the marginal distribution network with $\beta \to \infty$, it is exact over the limited voltage range in which no new discontinuities are introduced

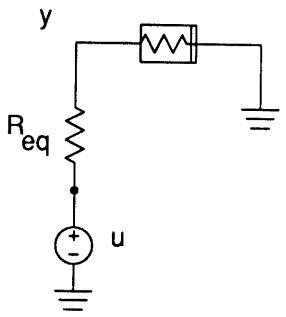


Figure 6: Thevenin equivalent circuit for one-dimensional nonlinear smoothing and segmentation network with step input.

into **y**. Otherwise, it is only an approximation and its applicability to other cases of interest must be individually determined.

3.2 Continuation Methods

We seek a modification to the networks so that the solution will be repeatable and also be visually and quantitatively "good." One technique that works well within the context of smoothing and segmentation is to apply a continuation method to the network [2], [10].

A continuation (sometimes called "deterministic annealing") can be realized in network form by the simultaneous application of a given homotopy (continuous deformation) to some or all of the circuit elements. Two types of continuations are particularly appropriate for our class of nonlinear networks. Assume we have a network with horizontal nonlinear resistors whose constitutive relation is described by $i = g(v)$, and vertical linear resistors with conductance λ_f. Consider the following two homotopies for the horizontal and vertical elements, respectively:

CH: Replace g with $g^{(p)}$, $p \in [a, b]$, such that $g^{(a)}$ constrains the network to have a unique solution and that $g^{(b)} = g$;

CV: Replace λ_f with $\lambda_f^{(p)}$, $p \in [a, b]$, such that $\lambda_f^{(a)}$ constrains the network to have a unique solution and that $\lambda_f^{(b)} = \lambda_f$.

Note that **CH** and **CV** define *where* the homotopies are applied in the network to produce a continuation; we are still free to decide the specific form of the homotopy.

3.3 β-Continuation

Blake and Zisserman suggest a **CH** continuation method — the so-called "graduated non-convexity" algorithm, or GNC [7]. There are some apparent weaknesses to using the GNC algorithm in network form, however. First, there is no reason to expect that, for an arbitrary image, the specific continuation used by GNC will produce the global minimum or that it will even produce a "good" minimum. Second, the nonlinear resistive element in a network realization of GNC will have a discontinuous first order derivative which can cause convergence difficulties in numerical simulation.

On the other hand, the marginal distribution derivation of our nonlinear network provides a natural homotopy for realization of the **CH** continuation. For $\beta = 0$, the network with elements described by (40) is linear, whereas for $\beta \to \infty$, the elements become identical to those in Figure 3 and will (locally) solve our minimization problem. This suggests using β directly as the continuation parameter for a **CH** continuation for solving (39) and hence (9). Furthermore, because of the way this continuation was derived, one might expect that it would do a good job of seeking the global cost minimum.

Some insight into the behavior of this type of network can be gained by examining the zero-dimensional case. Figure 7a shows the marginal distribution nonlinear resistor characteristic for various values of β, along with two load-lines representing two different values of the input. As β is taken from 0 to ∞, the solution will follow the continuous path represented by the intersection of the resistor curve and the load-line. In this example, the smaller step will be smoothed, and the larger step will be segmented.

Interestingly, discontinuous behavior can occur with this type of continuation, as is shown in Figure 7b. In this example, the initial solution point will be the intersection of the load-line and the marginal distribution resistor characteristic for $\beta = 0$. As β is increased, the "hump" of the nonlinear resistor curve will at one point pass completely beneath the load-line, at which point the solution will jump from being a smoothing solution to being a segmenting solution.

3.4 λ_f-Continuation

The **CV** continuation can be realized in a straightforward manner by varying the vertical resistors in the network. In particular, we begin with the resistors having infinite (or sufficiently large) conductance so that the network has only one solution, namely $\mathbf{y} = \mathbf{u}$ (or, for large conductance, $\mathbf{y} \approx \mathbf{u}$). Then, we continuously decrease the value of the conductance to λ_f.

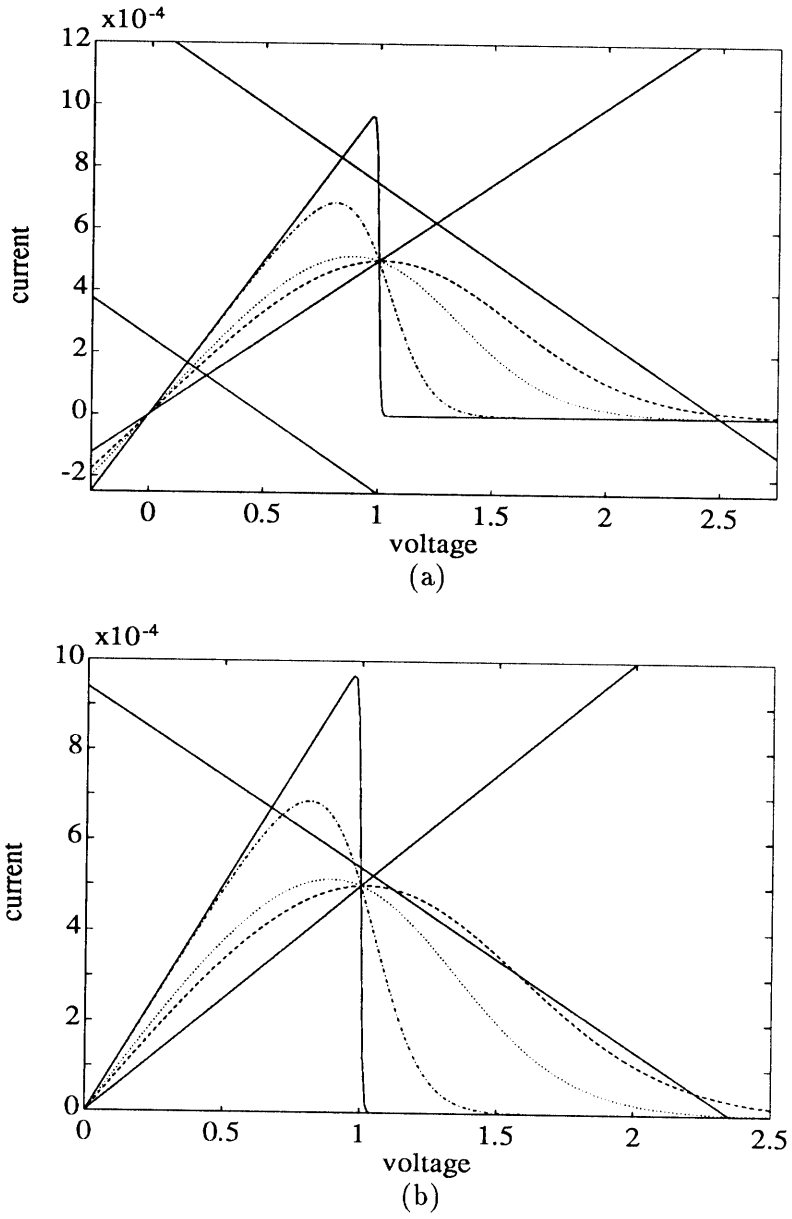


Figure 7: Approximate load-line plot for marginal distribution network with β-continuation. In part (a), the solid lines with negative slope represent the load-lines for two different input values ($0.5\,V$ and $2.5\,V$). The nonlinear resistor is shown for various values of β. For $\beta = 0$, the nonlinear resistor acts as a linear resistor. As $\beta \to \infty$, the nonlinear resistor characteristic becomes that of the discontinuous resistive fuse. In part (b), the solution exhibits a discontinuous jump, from a smoothing solution to a segmenting solution.

Examination of the zero-dimensional case provides some insight into the behavior of this type of network. Figure 8a shows the marginal distribution nonlinear resistor characteristic for large β, along with two sequences of load-lines representing two different values of the input. In this example, the solution for the larger input will remain at the initial intersection point of the load-line and the resistor curve as $\lambda_f^{(p)}$ is taken from $\lambda_f^{(a)} = \lambda_0$ to $\lambda_f^{(b)} = \lambda_f$. On the other hand, the solution for the smaller input will follow the continuous path represented by the intersection of the resistor curve and the load-line. Hence, the larger step will be segmented and the smaller step will be smoothed.

Discontinuous behavior can also occur with this type of continuation, when the continuation is used with non-linear resistors of finite β, as is shown in Figure 8b. In this example, the initial solution point will be a segmenting solution in the lower right-hand corner of the figure. As λ_f is decreased, the load-line will at some point pass completely beneath the nonlinear resistor characteristic, at which point the solution will jump from being a segmenting solution to being a smoothing solution.

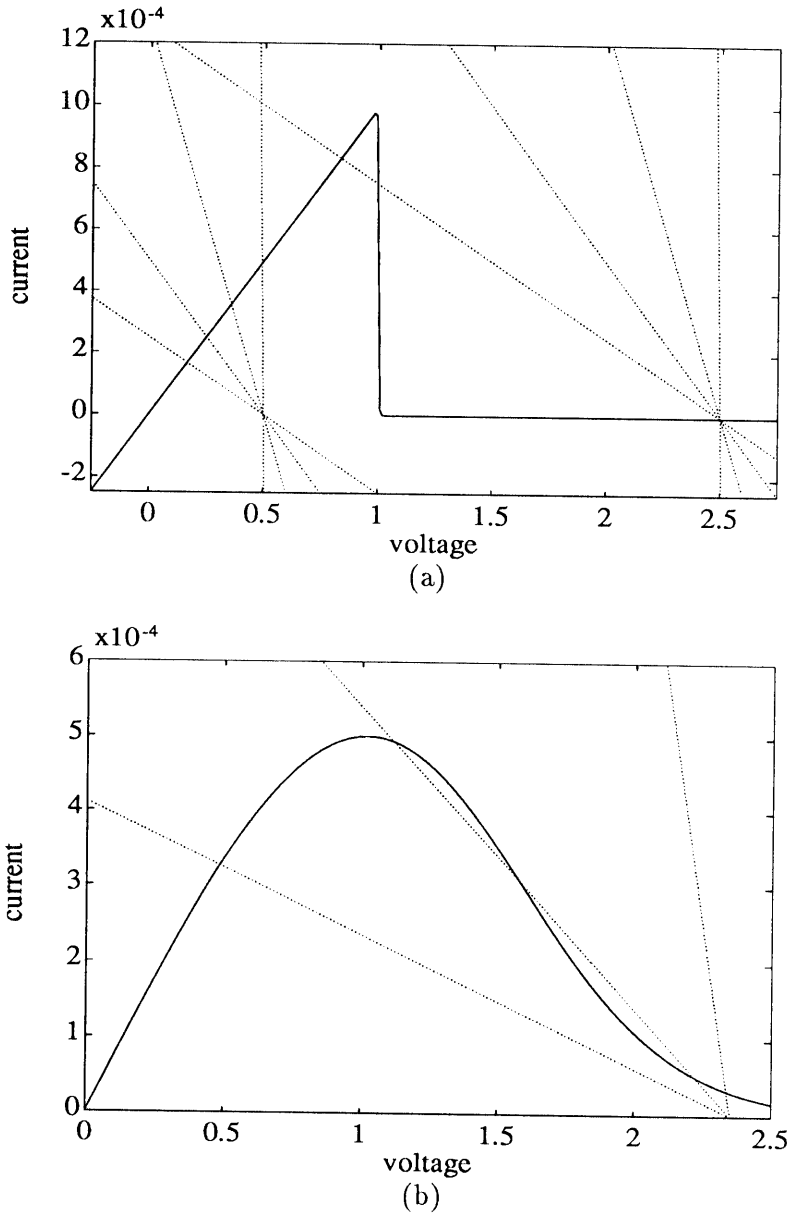


Figure 8: Approximate load-line plot for λ_f-continuation. In part (a), the nonlinear resistor characteristic $g_\beta(v)$ is shown for large β along with two sets of load-lines, each set for a different value of the input (the load-lines intersect the $g_\beta(v) = 0$ line at the value of the input voltage: $0.5\,V$ and $2.5\,V$). As λ_f is decreased, the load-lines rotate counter-clockwise. In part (b), the non-linear resistor characteristic is shown for finite β. In this case, the solution exhibits a discontinuous jump, from a segmenting solution to a smoothing solution, as λ_f is decreased.

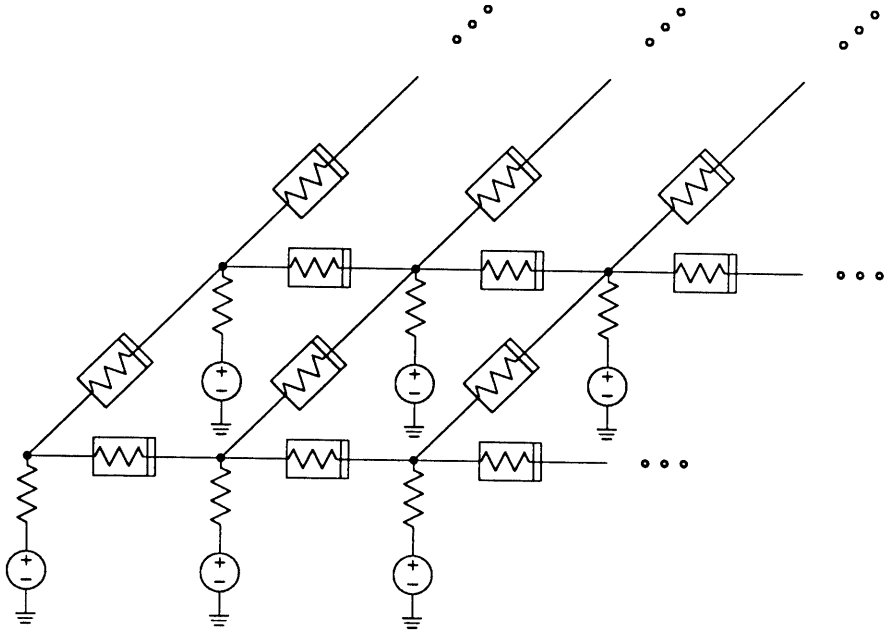
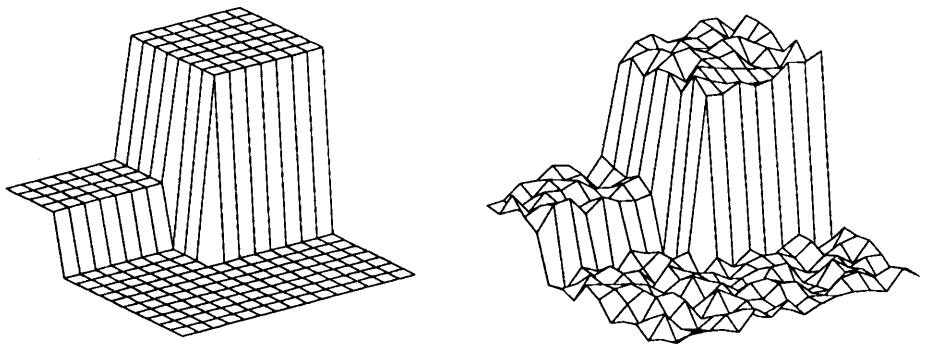


Figure 9: Two-Dimensional network topology.

4 Numerical Experiments

In order to quantitatively and qualitatively demonstrate the behavior of the β-continuation and λ_f-continuation networks, the results of several numerical experiments are presented. The experiments were all conducted with two-dimensional networks, the topology of which is shown in Figure 9. The experiments were conducted using serial and parallel versions of a special purpose circuit simulator developed specifically for vision circuits [28] - [30].

The continuations were simulated by performing dynamic simulations of the networks. In order to add dynamics to the networks, a small parasitic capacitance to ground was added at each node such that the time constants of the network were much faster than the rate at which the circuit elements were varied to perform the continuation. Dynamic simulation of the networks in this way has several advantages. First, the presence of parasitic capacitances is somewhat more physical and will allow the system to perform a gradient descent which will thereby guarantee that the network does not settle on a solution which statically satisfies KCL but is actually a local *maximum* of the network cost function [16]. Second, the dynamics will insure that the network behavior is well-defined at points where solutions in the static case would disappear, as in Figures 7b and 8b. (Our experience has been that



(a) (b)

Figure 10: (a) Original image. (b) Original image corrupted with noise. The noisy image was used as input for all experiments in Section 4.1.

discontinuous circuit behavior is much more common in the λ_f-continuation network than in the β-continuation network, which causes simulation of a λ_f-continuation network to take much more time.)

4.1 Experiments with a Synthetic Image

A series of seven experiments conducted on a 16 x 16 circuit grid with a synthetic input image was conducted. Figure 10a shows the 16 x 16 synthetic image used for the experiments. The small step is $1\,V$ in height and the large step is $3\,V$. The original image was then corrupted by the addition of $0.5\,V$ of uniformly distributed noise and is shown in Figure 10b. The noisy signal was used as input for this series of experiments.

For each experiment, a cost function was determined and the corresponding β-continuation and λ_f-continuation networks constructed. Then, the networks were each simulated using the input image shown in Figure 10b. For each experiment, the value of λ_s was fixed at 1.0×10^{-3} and the value of λ_h was changed. For the β-continuation, the value of λ_f was fixed at 1×10^{-4} and the value of β was increased from 0 to $20/\lambda_h$. For the λ_f-continuation, the value of β was set to $20/\lambda_h$ and the value of λ_f was varied from 1 to 1×10^{-4}. Thus, for each experiment, the *final* states of the β-continuation and λ_f-continuation networks were the same.

Expt	λ_h	Cost	
		β-cont.	λ_f-cont.
1	1.0×10^{-3}	1.775×10^{-2}	1.770×10^{-2}
2	5.0×10^{-4}	9.699×10^{-3}	1.254×10^{-2}
3	1.0×10^{-4}	3.299×10^{-3}	2.940×10^{-3}
4	5.0×10^{-5}	1.740×10^{-3}	1.740×10^{-3}
5	1.0×10^{-5}	7.800×10^{-4}	2.641×10^{-3}
6	5.0×10^{-6}	6.600×10^{-4}	1.650×10^{-3}
7	1.0×10^{-6}	5.518×10^{-4}	4.246×10^{-4}

Table 1: Experimental results showing the values of the cost function of the solutions produced by the β- and λ_f-continuation networks for different values of λ_h.

Solutions obtained by the two nonlinear networks were compared as follows:

1. Given a cost function, construct the corresponding nonlinear networks, and in addition, construct a corresponding RWS network;
2. Provide each network with the same input and allow each network to attain its solution;
3. For each nonlinear network, transfer the line process solution obtained to the RWS network by setting the switches according to equation (8);
4. Allow the RWS to attain its voltage solution and compute the resulting cost — it is this cost that is used for comparison.

The results of the seven experiments are shown in Table 1. For the particular values of parameters used, each network computed a lower cost in roughly half the experiments. This set of experiments was actually taken from a larger set of 49. Of those, the β-continuation found the lower cost 35 times, the λ_f-continuation found the lower cost eight times, and there were six ties. Thus, in these experiments, the β-continuation performs its task of minimizing the cost function (1) extremely well.

If the cost function were the last word on image smoothing and segmentation, we could immediately recommend the β-continuation. However, remember that the ultimate goal for a smoothing and segmentation network is essentially to recover an original image minus any noise, and the cost function was introduced to give us a quantitative means for doing this. Now consider Figure 11, which corresponds qualitatively to the solutions produced by the two nonlinear networks in experiments 2 and 3. Note that whereas 11a is the qualitatively correct solution, it corresponds to

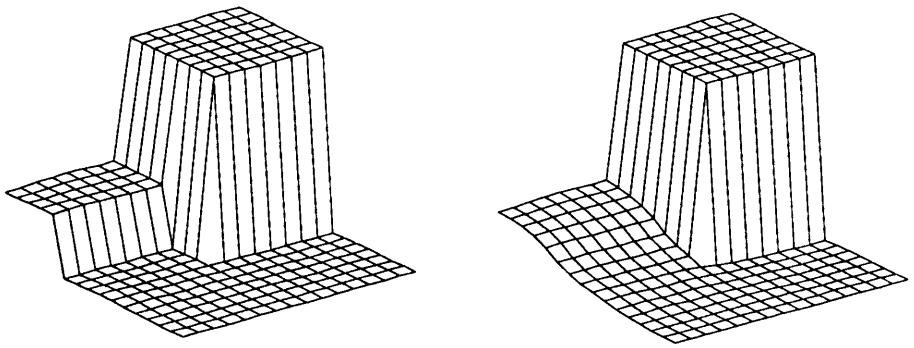


Figure 11: (a) Network solution produced by λ_f-continuation in experiments 2 and 3. (b) Network solution produced by β-continuation in experiments 2 and 3.

the higher cost in experiment two, but corresponds to the lower cost in experiment three.

Naturally, this calls into question the entire cost function methodology used for smoothing and segmentation. The difficulty arises because our efforts were concentrated only on finding an optimal solution rather than the larger issue of determining the best cost function and parameter values. See however [31].

4.2 Experiments with a Real Image

The networks were then tested with a real image. Figure 12 shows the 256×256 input image — a portion of the San Francisco sky line. The output images shown in Figures 13 – 16 were produced using a recently developed circuit simulation program on the Connection Machine.

Figure 13 shows the output produced by the β-continuation with fixed parameter values $\lambda_s = 1\times10^{-3}$, $\lambda_h = 2\times10^{-5}$, and $\lambda_f = 1\times10^{-4}$. Figure 13a shows the output of the network at the beginning of the continuation when $\beta = 0$; Figure 13b shows the output of the network at an intermediate point of the continuation when $\beta = 5\times10^{3}$; Figure 13c shows the output of the network at the end of the continuation when $\beta = 5 \times 10^{5}$. Figure 14 shows the output produced by the β-continuation with fixed parameter values $\lambda_s = 1 \times 10^{-3}$, $\lambda_h = 1 \times 10^{-5}$, and $\lambda_f = 3 \times 10^{-5}$. Figure

14a shows the output of the network at the beginning of the continuation when $\beta = 0$; Figure 14b shows the output of the network at an intermediate point of the continuation when $\beta = 2\times10^4$; Figure 14c shows the output of the network at the end of the continuation when $\beta = 1\times10^6$. Figure 15 shows the output produced by the λ_f-continuation with fixed parameter values $\lambda_s = 1\times10^{-3}$, $\lambda_h = 1\times10^{-5}$, and $\beta = 1\times10^6$. Figure 15a shows the output of the network at the beginning of the continuation when $\lambda_f = 1$; Figure 15b shows the output of the network at an intermediate point of the continuation when $\lambda_f = 1\times10^{-3}$; Figure 15c shows the output of the network at the end of the continuation when $\lambda_f = 3\times10^{-5}$. Note that the final parameter values of this network are identical to those for the network of Figure 14. Figure 16 shows the output produced by the λ_f-continuation with parameter values $\lambda_s = 1\times10^{-3}$, $\lambda_h = 2\times10^{-5}$, $\beta = 5\times10^4$. Figure 16a shows the output of the network at the beginning of the continuation when $\lambda_f = 1$; Figure 16b shows the output of the network at an intermediate point of the continuation when $\lambda_f = 5\times10^{-4}$; Figure 16c shows the output of the network at the end of the continuation when $\lambda_f = 1\times10^{-6}$.

Discussion

As can be seen from the experiments with the real image, not only does the selection of parameter values affect the behavior of the networks, but the continuation used also has a profound effect on the network behavior. The differences between cost functions for a particular continuation can be seen by comparing Figures 13c and 14c, and by comparing Figures 15c and 16c. The differences between continuations methods for a given cost function can be seen by comparing Figures 14c and 15c.

One can understand the differences in the continuation methods quite readily. At the beginning of the β-continuation, the output of the β-continuation network is rather smooth, since initially the network is equivalent to a linear resistive network (see Figures 13a and 14a). The edges are then added during the course of the continuation (see Figures 13b and 14b). This is a difficulty because without any initial edge information, some of the edges might be misplaced or even completely lost. Notice that in Figures 13c and 14c, edges tend to line up along the network axes.

On the other hand, the initial output image of the λ_f-continuation network is very close (or identical) to the input one (see Figures 15a and 16a). All the edges are initially present and only the spurious edges are smoothed during the course of the continuation. Since all the edge information is present at the start of the continuation, one would expect that the λ_f-continuation would more properly

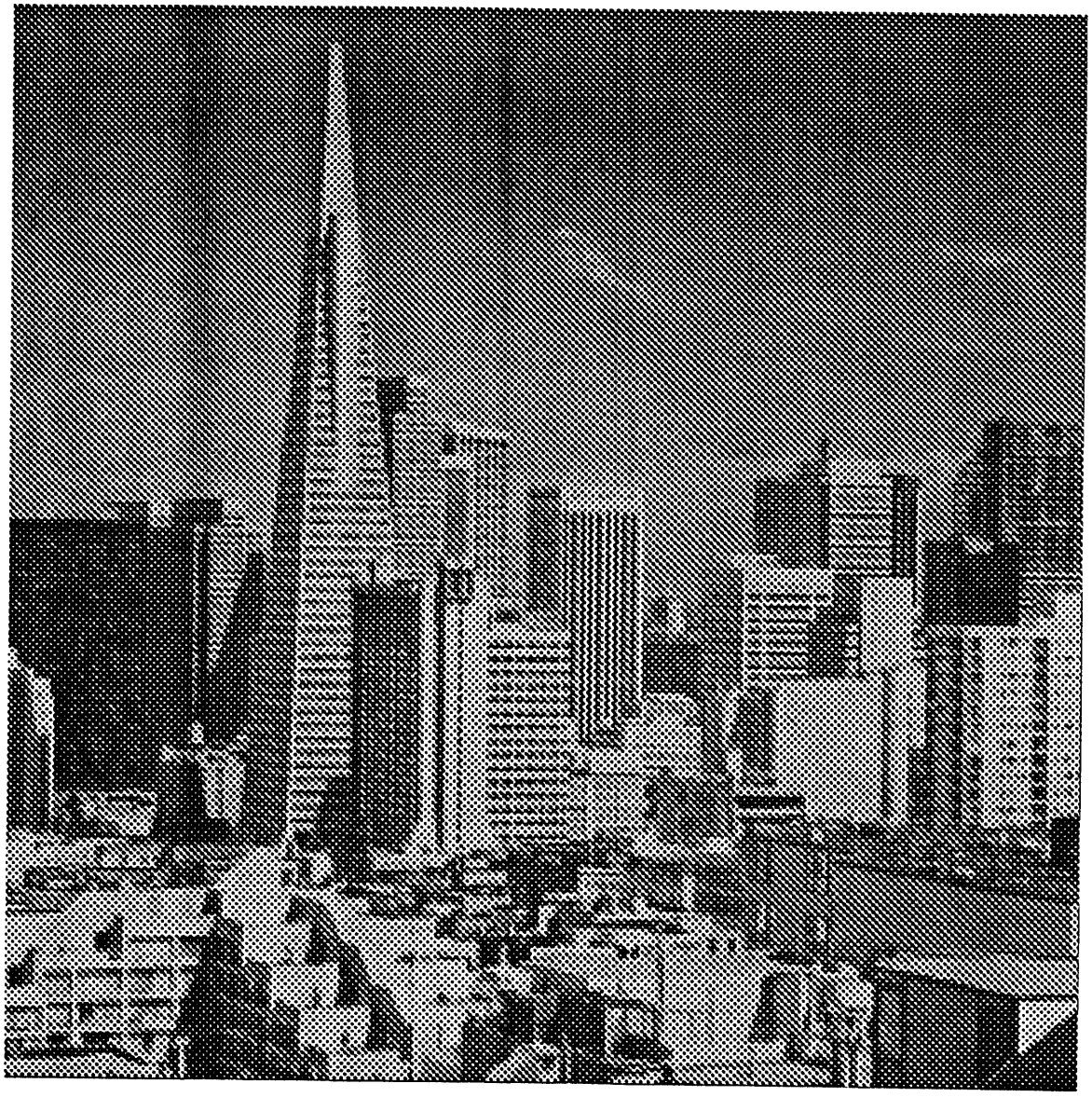


Figure 12: 256 x 256 image of the San Francisco sky line.

Figure 13: (a) Output produced by β-continuation network. Here, the parameter values are $\lambda_s = 1\times10^{-3}$, $\lambda_h = 2\times10^{-5}$, $\lambda_f = 1\times10^{-4}$, and $\beta = 0, 5\times10^{3}$, and 5×10^{5} for Figures 13a, 13b, and 13c, respectively.

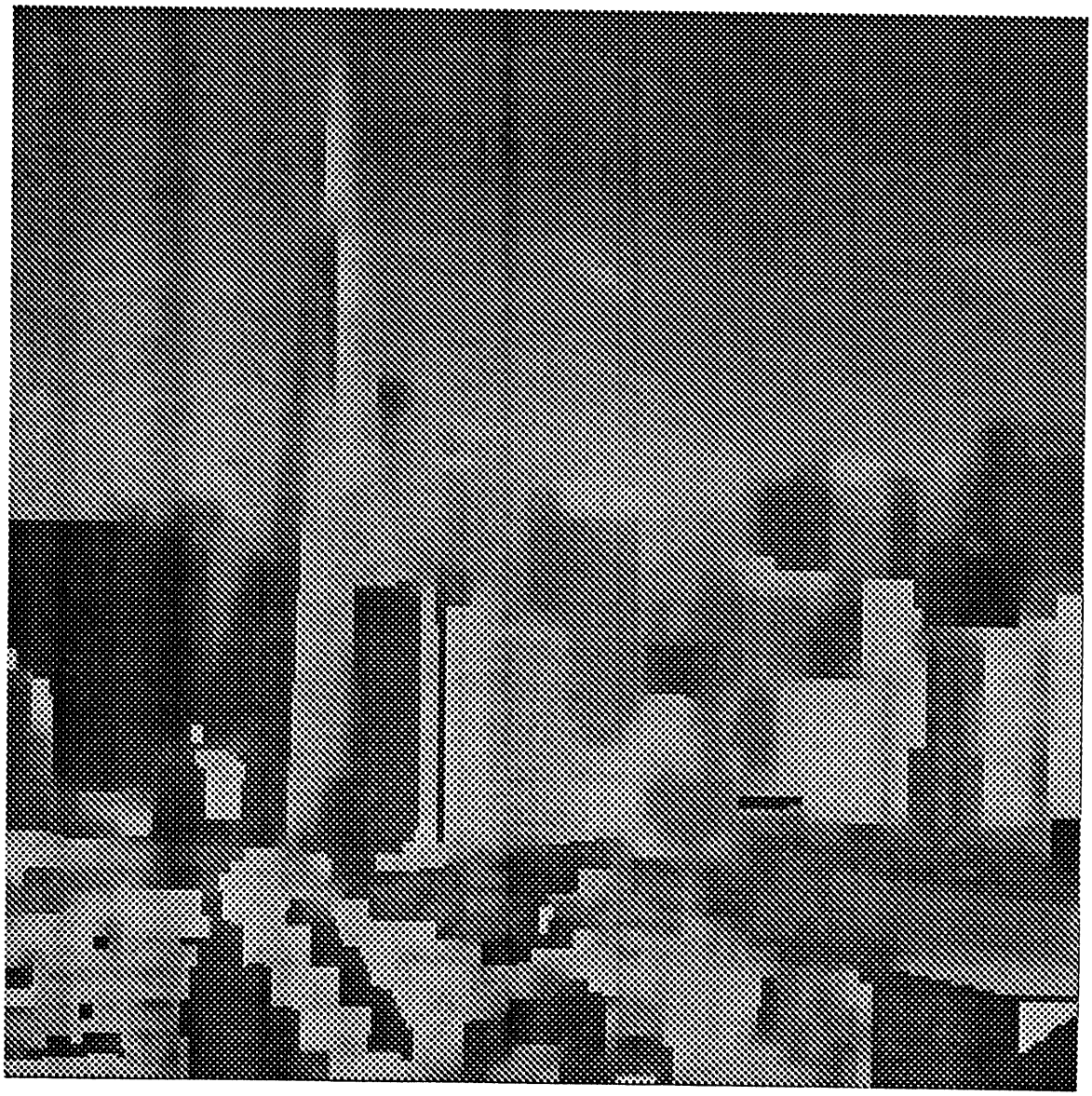


Figure 13: (b) Output produced by β-continuation network. Here, the parameter values are $\lambda_s = 1\times10^{-3}$, $\lambda_h = 2\times10^{-5}$, $\lambda_f = 1\times10^{-4}$, and $\beta = 0, 5\times10^{3}$, and 5×10^{5} for Figures 13a, 13b, and 13c, respectively.

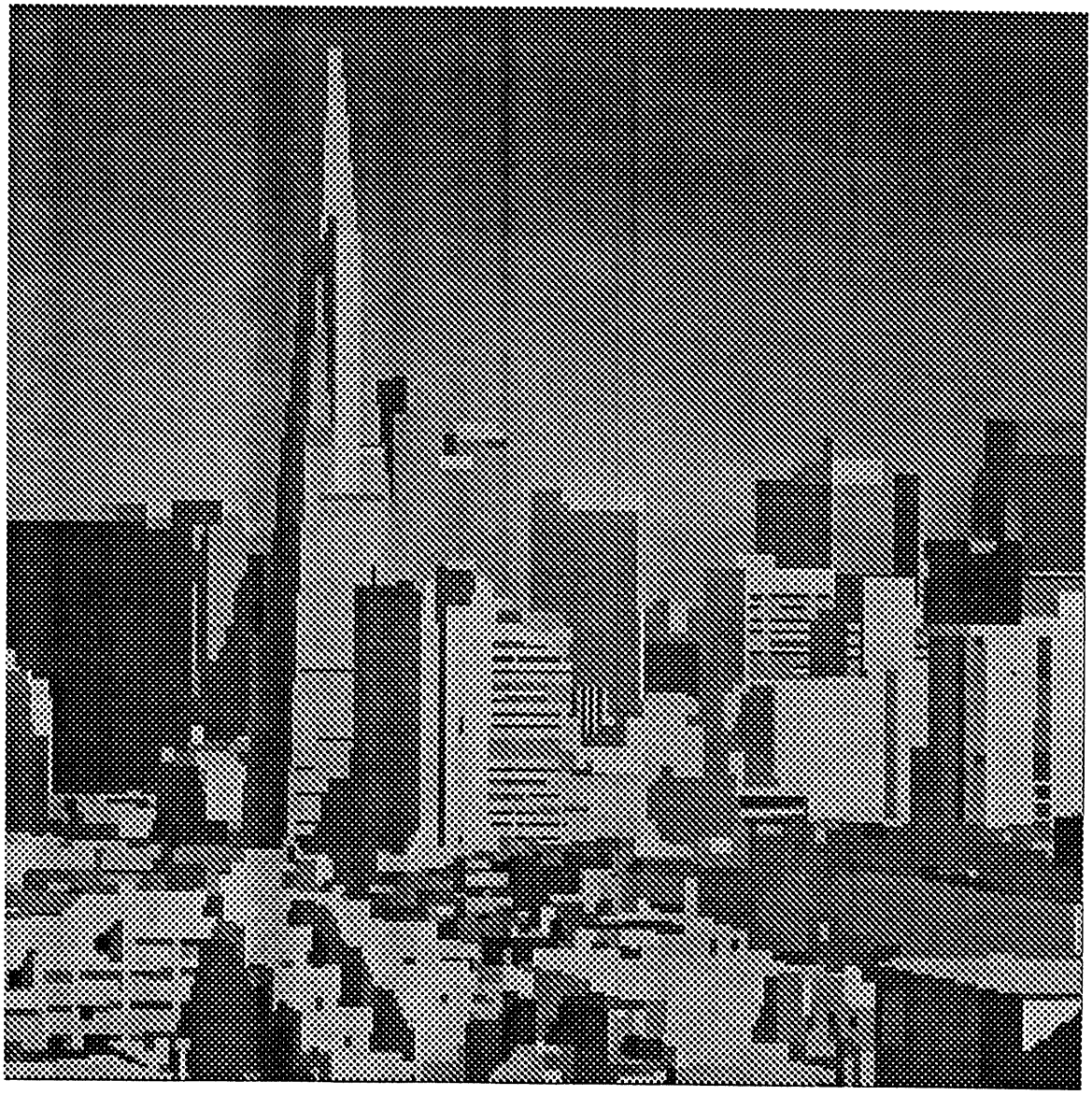


Figure 13: (c) Output produced by β-continuation network. Here, the parameter values are $\lambda_s = 1\times10^{-3}$, $\lambda_h = 2\times10^{-5}$, $\lambda_f = 1\times10^{-4}$, and $\beta = 0$, 5×10^{3}, and 5×10^{5} for Figures 13a, 13b, and 13c, respectively.

Figure 14: (a) Output produced by β-continuation network with smaller fidelity and line penalty weights and larger final β value than for the network in Figure 13. Here, the parameter values are $\lambda_s = 1 \times 10^{-3}$, $\lambda_h = 1 \times 10^{-5}$, $\lambda_f = 3 \times 10^{-5}$, and $\beta = 0, 2 \times 10^4$, and 1×10^6 for Figures 14a, 14b, and 14c, respectively.

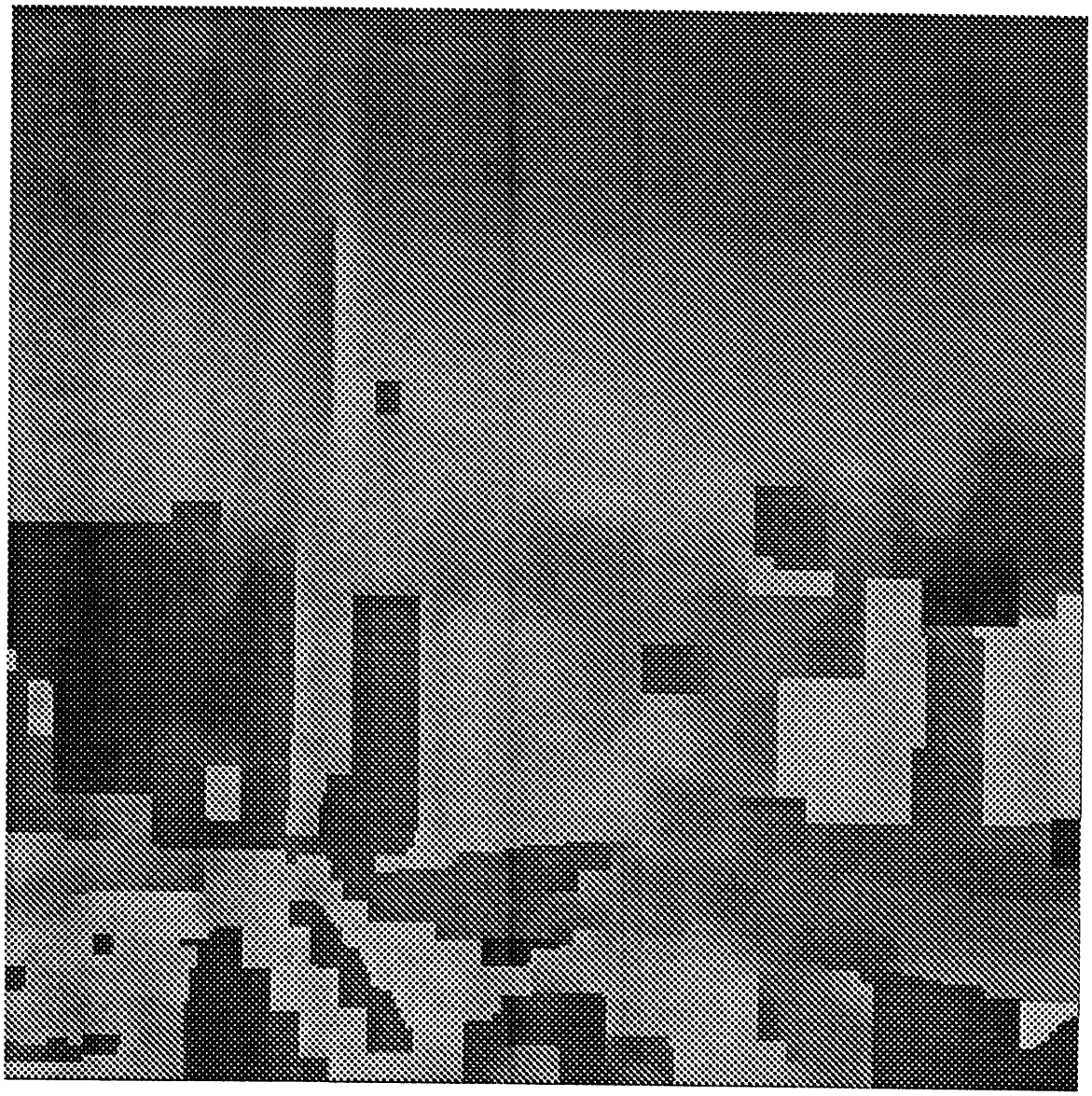


Figure 14: (b) Output produced by β-continuation network with smaller fidelity and line penalty weights and larger final β value than for the network in Figure 13. Here, the parameter values are $\lambda_s = 1 \times 10^{-3}$, $\lambda_h = 1 \times 10^{-5}$, $\lambda_f = 3 \times 10^{-5}$, and $\beta = 0, 2 \times 10^4$, and 1×10^6 for Figures 14a, 14b, and 14c, respectively.

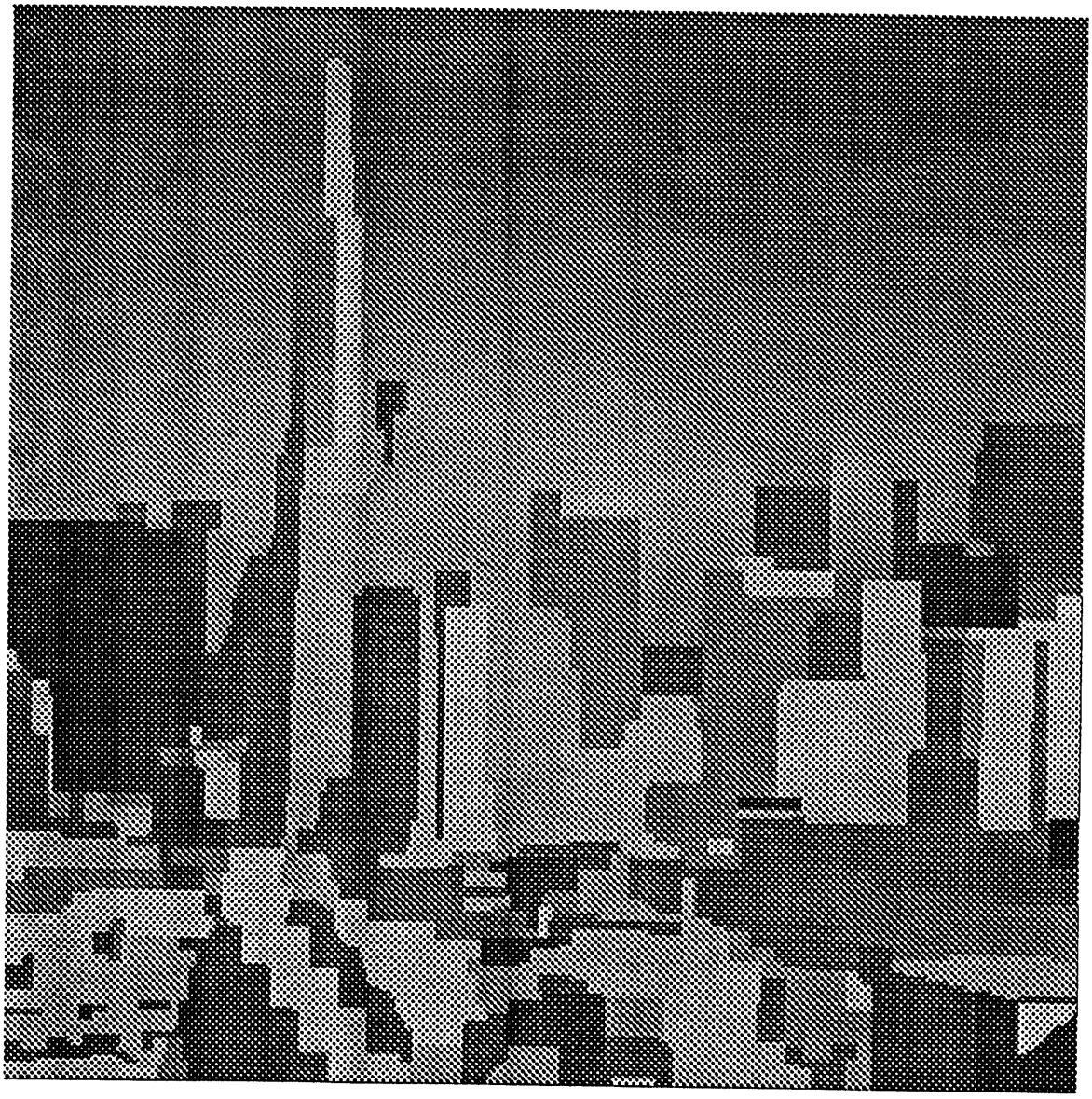


Figure 14: (c) Output produced by β-continuation network with smaller fidelity and line penalty weights and larger final β value than for the network in Figure 13. Here, the parameter values are $\lambda_s = 1 \times 10^{-3}$, $\lambda_h = 1 \times 10^{-5}$, $\lambda_f = 3 \times 10^{-5}$, and $\beta = 0$, 2×10^4, and 1×10^6 for Figures 14a, 14b, and 14c, respectively.

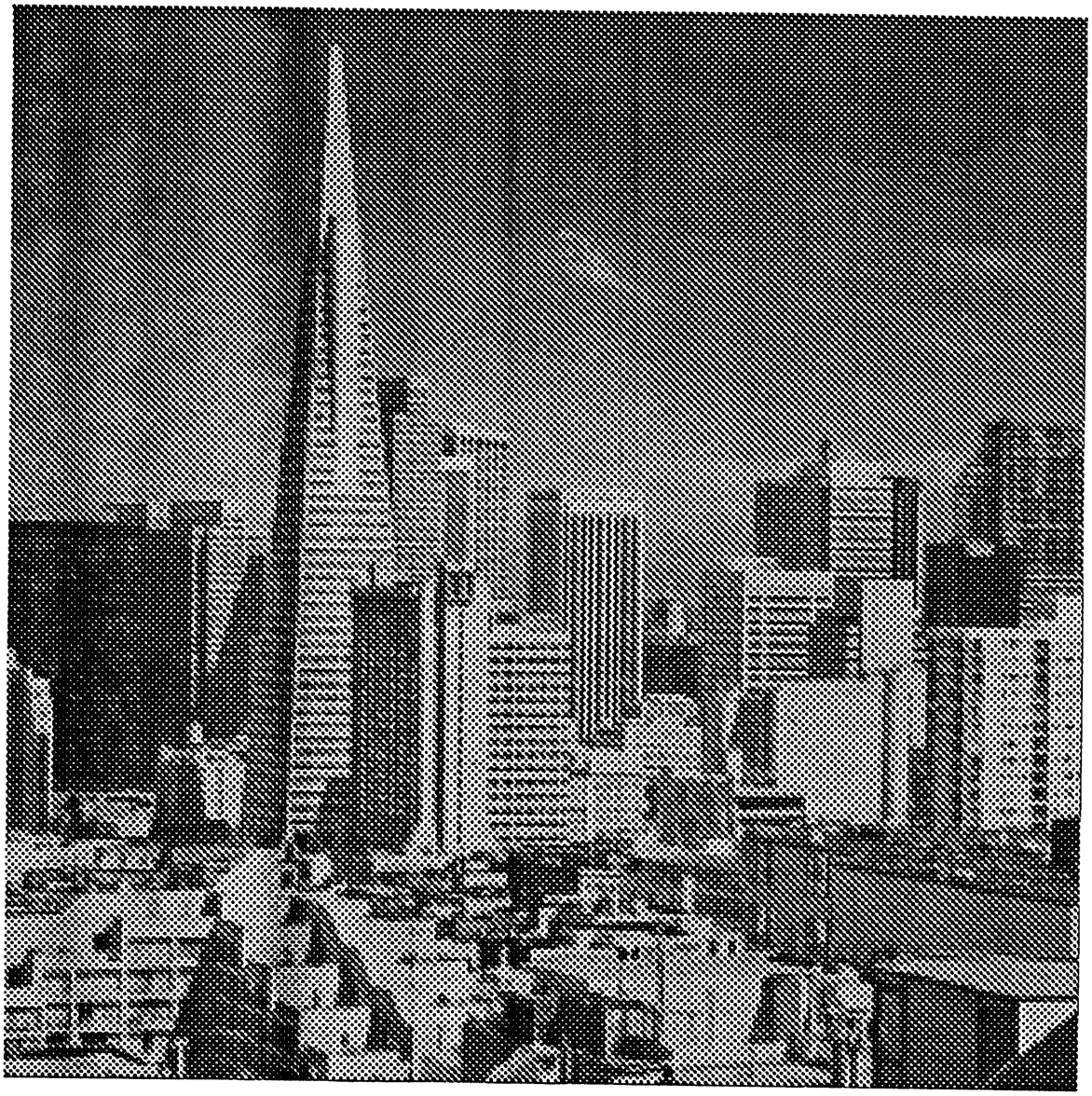


Figure 15: (a) Output produced by λ_f-continuation network. Here, the parameter values are $\lambda_s = 1 \times 10^{-3}$, $\lambda_h = 1 \times 10^{-5}$, $\beta = 1 \times 10^{6}$, and $\lambda_f = 1$, $\lambda_f = 1 \times 10^{-3}$, and $\lambda_f = 3 \times 10^{-5}$ for Figures 15a, 15b, and 15c, respectively. Note that the final parameter values of this network are identical to those for the network of Figure 14, but that the output image is much closer to the input image shown in Figure 12.

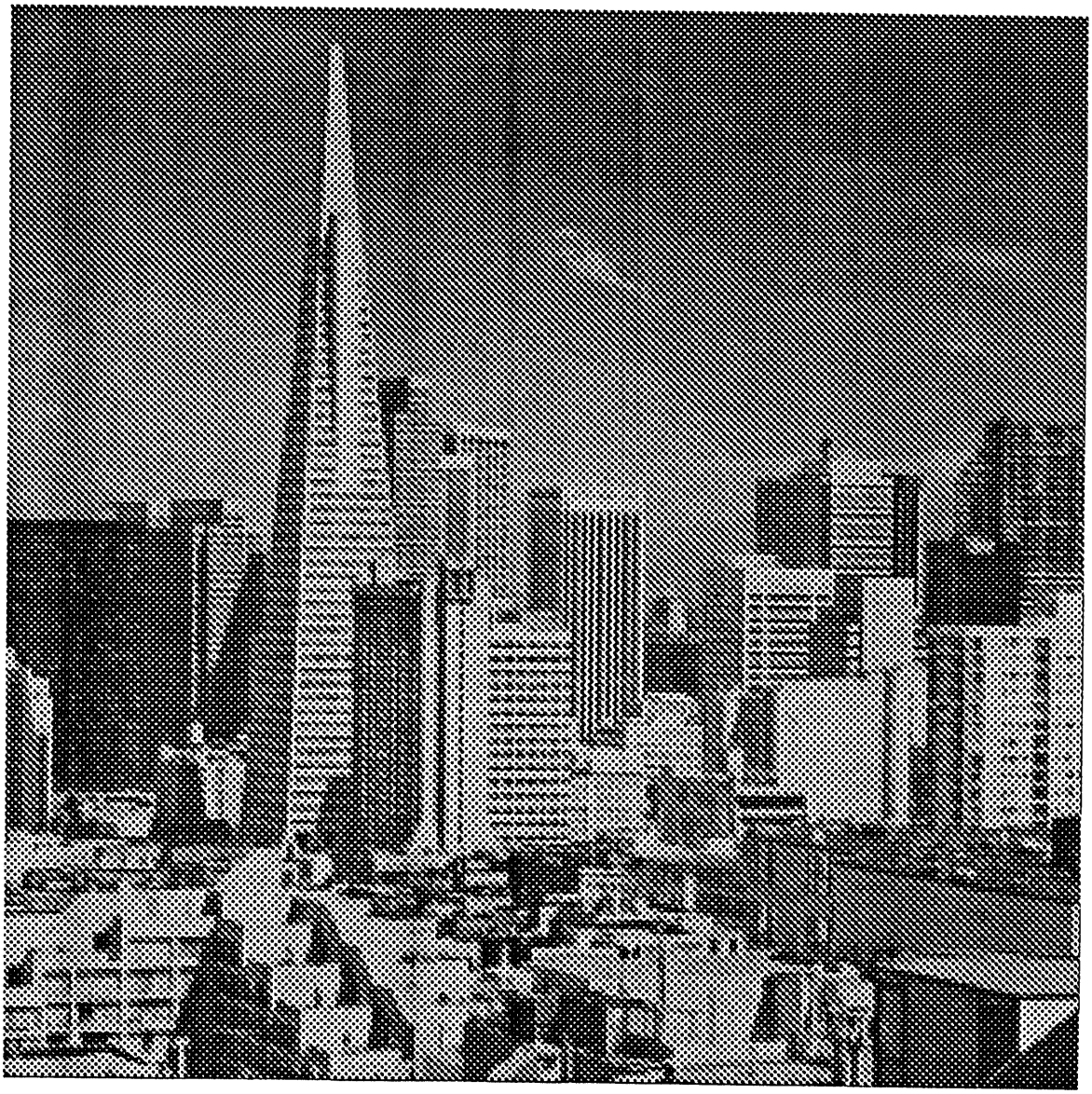


Figure 15: (b) Output produced by λ_f-continuation network. Here, the parameter values are $\lambda_s = 1\times10^{-3}$, $\lambda_h = 1\times10^{-5}$, $\beta = 1\times10^{6}$, and $\lambda_f = 1$, $\lambda_f = 1\times10^{-3}$, and $\lambda_f = 3\times10^{-5}$ for Figures 15a, 15b, and 15c, respectively. Note that the final parameter values of this network are identical to those for the network of Figure 14, but that the output image is much closer to the input image shown in Figure 12.

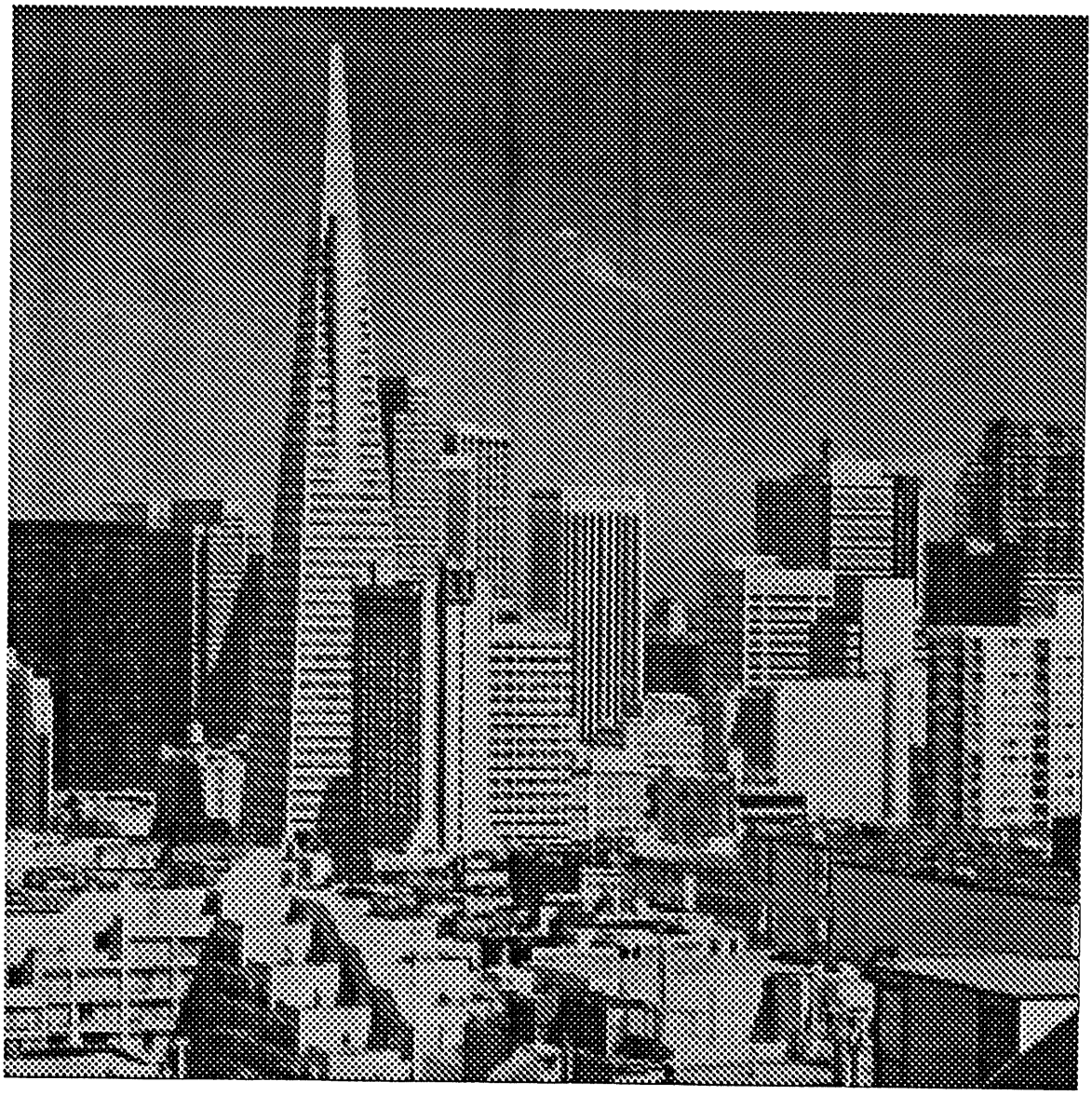


Figure 15: (c) Output produced by λ_f-continuation network. Here, the parameter values are $\lambda_s = 1 \times 10^{-3}$, $\lambda_h = 1 \times 10^{-5}$, $\beta = 1 \times 10^{6}$, and $\lambda_f = 1$, $\lambda_f = 1 \times 10^{-3}$, and $\lambda_f = 3 \times 10^{-5}$ for Figures 15a, 15b, and 15c, respectively. Note that the final parameter values of this network are identical to those for the network of Figure 14, but that the output image is much closer to the input image shown in Figure 12.

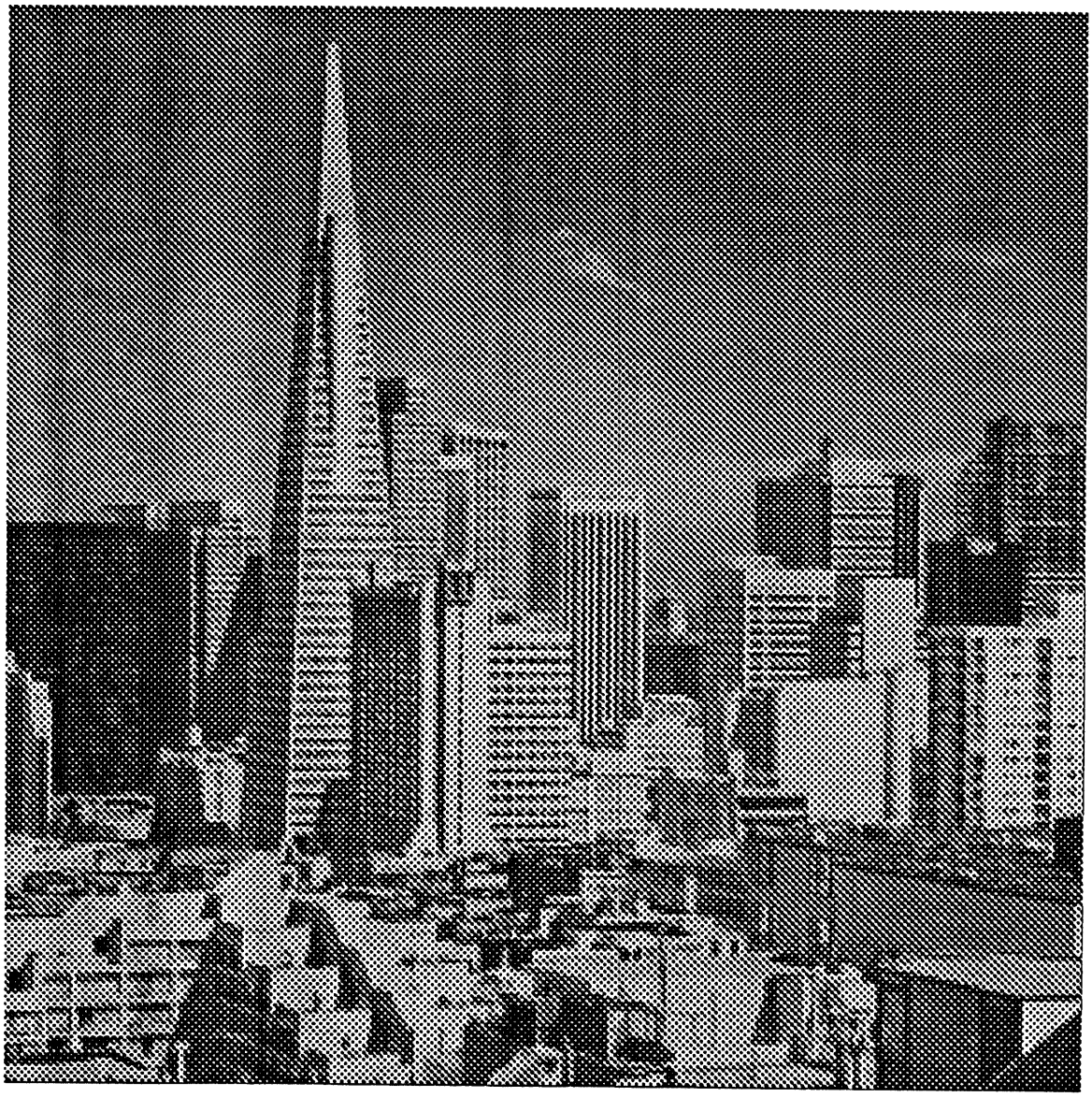


Figure 16: (a) Output produced by λ_f-continuation network. Here, the parameter values are $\lambda_s = 1\times10^{-3}$, $\lambda_h = 2\times10^{-5}$, $\beta = 5\times10^{4}$, and $\lambda_f = 1$, $\lambda_f = 5\times10^{-4}$, and $\lambda_f = 1\times10^{-6}$ for Figures 16a, 16b, and 16c, respectively.

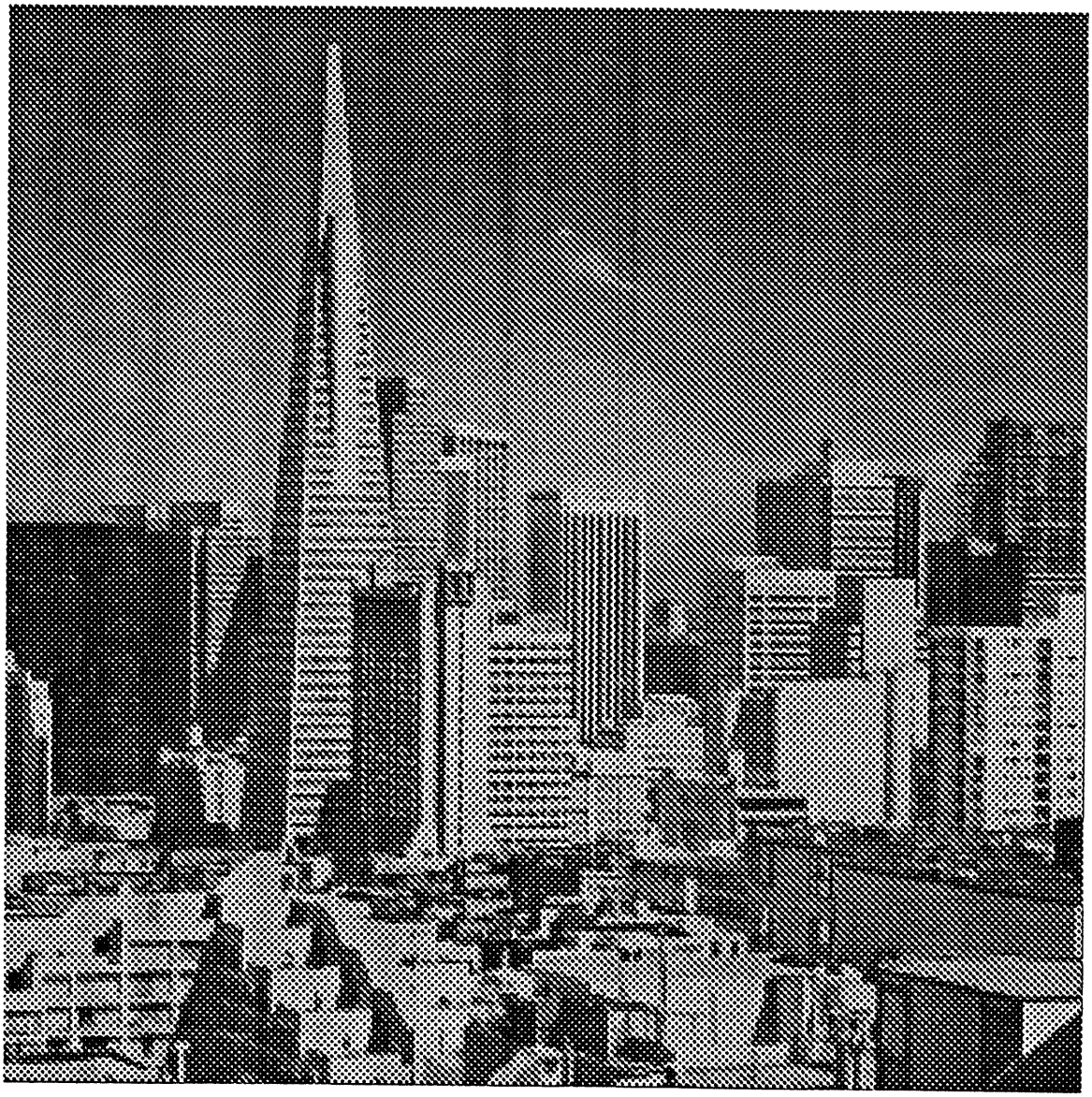


Figure 16: (b) Output produced by λ_f-continuation network. Here, the parameter values are $\lambda_s = 1\times10^{-3}$, $\lambda_h = 2\times10^{-5}$, $\beta = 5\times10^{4}$, and $\lambda_f = 1$, $\lambda_f = 5\times10^{-4}$, and $\lambda_f = 1\times10^{-6}$ for Figures 16a, 16b, and 16c, respectively.

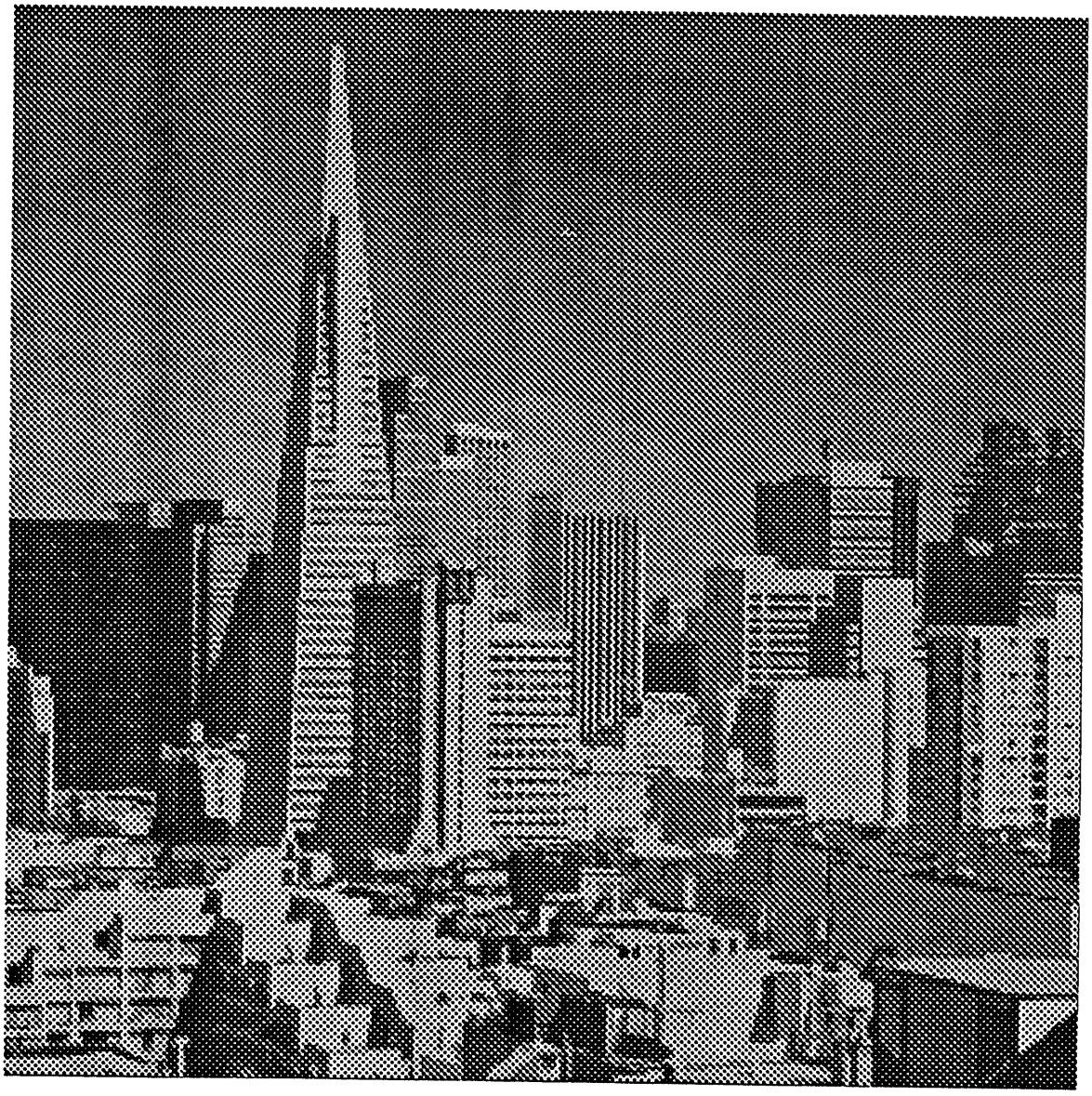


Figure 16: (c) Output produced by λ_f-continuation network. Here, the parameter values are $\lambda_s = 1\times10^{-3}$, $\lambda_h = 2\times10^{-5}$, $\beta = 5\times10^{4}$, and $\lambda_f = 1$, $\lambda_f = 5\times10^{-4}$, and $\lambda_f = 1\times10^{-6}$ for Figures 16a, 16b, and 16c, respectively.

locate and preserve edges. This observation is borne out in Figures 15c and 16c; the edges are much more well preserved than those in Figures 13c and 14c.

Finally, since the λ_f-continuation only requires that a linear resistance be varied, its VLSI implementation should be much more compact than that of the β-continuation, which would require varying the characteristics of a nonlinear resistor.

4.3 Behavior of the λ_f-Continuation

There were some interesting properties exhibited by networks constructed with the elements described in (40) having a fixed $\beta < \infty$. For such a network, it can be shown that there exist a $\lambda_{min} > 0$ and a $\lambda_{max} < \infty$ such that for $\lambda_f > \lambda_{max}$ and for $\lambda_f < \lambda_{min}$, the network has a unique solution. In fact, for $\lambda_f > \lambda_{max}$, the output will essentially match the input (i.e., $\mathbf{y} \approx \mathbf{u}$), whereas for $\lambda_f < \lambda_{min}$, the output will contain no edges. Consider the network behavior as a function of λ_f as λ_f is varied continuously from λ_{max} to λ_{min}. The initial solution of the network will closely match the input. Then, as λ_f is decreased, edges will begin to disappear, first the smaller, then the larger, until all the edges are gone. In other words, λ_f acts as a scale-space parameter. This has important practical applications. The dynamic network of Perona and Malik [9] has the property that *time* acts as a scale-space parameter. In contrast, we can exercise direct control over the scale-space parameter in the λ_f-continuation network. See also [2]. Moreover, this behavior is somewhat reminiscent of the more successful methods used for hierarchical multiscale image representation [32].

Under some mild assumptions, it can be shown that the particular solution path with the endpoints described above is continuous, connected, and can be numerically traced out in $\Re^{N+1}$ ($\lambda_f \times \Re^N$ space) using an arc-length continuation [33]. In such a case, any particular value of λ_f would correspond to an N-dimensional hyperplane parallel to $\Re^N$ given by $x_{N+1} = \lambda_f$ and network solutions for this λ_f would be intersections of the solution path and that hyperplane. An interesting question now arises: why can't we just trace out the path in $\Re^{N+1}$, determine the solutions, and sort them by cost to find the global minimum? The answer, unfortunately, is that there can exist solution *loops* that are disjoint from the main solution path, meaning that the path traced out from the starting point of large λ_f will not necessarily contain all the solutions at any given value of λ_f. The solution loops can occur in as small an example as a three pixel circuit (see Figure 17) and we offer as proof the experimental evidence in Figure 18. To produce the paths, λ_f was parameterized as a function of parameter t according to $\lambda_f = (1-t)\lambda_{\text{large}} + \lambda_{\text{small}}$, and the plots were made in $t \times \Re^2$ space. Notice that, as predicted, there is one solution path with endpoints $\{t = 0, V_1 = V_{in} = 2.5, V_2 = 0\}$

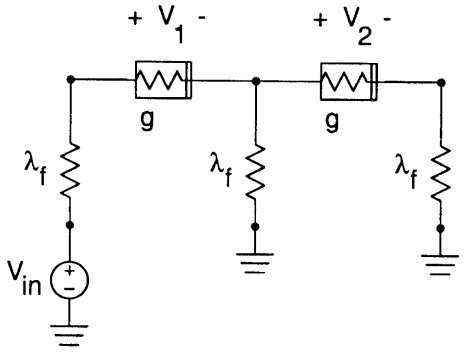


Figure 17: Three pixel example circuit. For the result shown in Figure 18, $V_{in} = 2.5\ V$, $\lambda_s = 0.19$, $\lambda_h = 1.0 \times 10^{-4}$, $\beta = 40$, $\lambda_f = (1-t)\lambda_{\text{large}} + \lambda_{\text{small}}$, $\lambda_{\text{large}} = 0.1$, $\lambda_{\text{small}} = 1.0 \times 10^{-12}$, and t was varied from 0 to 1.

and $\{t = 1, V_1 = 0, V_2 = 0\}$, corresponding to an edge across the first nonlinear resistor. In addition, there is a closed solution loop centered (approximately) at $\{t = 0.9, V_1 = 0.2, V_2 = 1.2\}$, corresponding to a "misplaced" edge, i.e., an edge across the second nonlinear resistor. In general, it can be shown that for a one-dimensional RWS or RWF network with a single step input, solutions corresponding to misplaced edges always have higher cost than solutions corresponding to correctly placed edges.

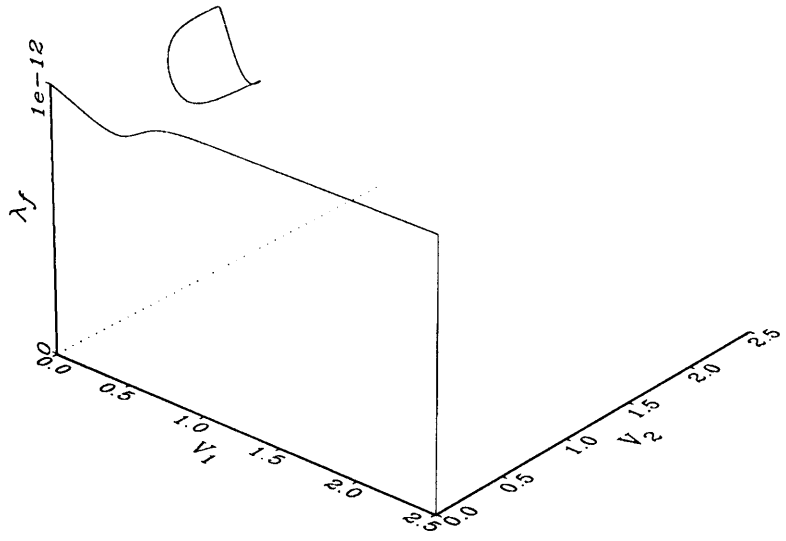


Figure 18: Solution path in $t \times \Re^2$ space for the three pixel example circuit demonstrating the existence of a disconnected solution loop.

5 Conclusion

In this paper, we developed and compared a series of nonlinear networks for image smoothing and segmentation. The results of several experiments indicate that the typical cost (or "energy") function minimization formulation of the smoothing and segmentation problem does not necessarily capture the essence of the task. For the specific parameter values we used, the λ_f-continuation network performed extremely well even though it did not always find the solution with minimum cost. The λ_f-continuation network has several implementation advantages over the β-continuation network. First, in certain cases, it seems to perform the smoothing and segmentation task in a more visually correct fashion. Second, λ_f can be used as a scale-space parameter. Finally, since the λ_f-continuation only requires that a linear resistance be varied, its VLSI implementation should be much more compact than that of the β-continuation.

Several open questions remain. Primary among these is the need for a comprehensive characterization of the natural behavior of these networks. By "natural behavior" we mean a set of quantitative empirical statements that relate the behavior of the network, given certain canonical edge configurations, to the cost function parameters. Furthermore, it is important to know how the networks behave in the presence of varying amounts and types of noise. Finally, Tom Richardson has developed an alternate formulation of the smoothing and segmentation problem based on a rigorous analysis of the continuous case [31]. This leads to a more complex circuit interpretation that might offer better performance than the methods investigated here. Since efficient simulation tools on the Connection Machine are now available, it is hoped that some of these questions can be addressed in the near future.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation and the Defense Advanced Research Projects Agency under Contract No. MIP-88-14612. The first author was also supported by an AEA/Dynatech faculty development fellowship. The authors are grateful to Thinking Machines Corporation, especially Rolf Fiebrich, for providing hardware and software support for the development of the simulator used to produce the experimental results in Section 4. The authors would like to acknowledge helpful discussions with Professor Alan Yuille of Harvard University, Dr. Davi Geiger of Siemens, and Professor Jacob White of MIT.

References

[1] D. Geiger and F. Girosi, "Parallel and Deterministic Algorithms from MRF's: Surface Reconstruction and Integration," to appear in *IEEE Trans. Pattern Analysis and Mach. Intell.*

[2] D. Geiger and A. Yuille, "A Common Framework for Image Segmentation," to appear in *Int. Jour. Comp. Vision.*

[3] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. PAMI-6(6)*, pp. 721 - 741, November 1984.

[4] J.L. Marroquin, "Optimal Bayesian Estimators for Image Segmentation and Surface Reconstruction," MIT AI Laboratory Memo 839, April 1985.

[5] F.S. Cohen and D.B. Cooper, "Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields," *IEEE Trans. PAMI-9(2)*, pp. 195 - 219, March 1987.

[6] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic Solution of Ill-Posed Problems in Computational Vision," *Jour. Amer. Stat. Assoc. (Theory and Methods)*, vol. 82, no. 397, pp. 76-89, March 1987.

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

[8] A. Blake, "Comparison of the Efficiency of Deterministic and Stochastic Algorithms for Visual Reconstruction," *IEEE Trans. PAMI-11(1)*, pp. 2 - 12, January 1989.

[9] P. Perona and J. Malik, "Scale Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. PAMI-12(7)*, pp. 629 - 639, July 1990.

[10] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[11] C. Koch, J. Marroquin, and A. Yuille, "Analog 'Neuronal' Networks in Early Vision," *Proc. Natl. Acad. Sci. USA*, vol. 83, pp. 4263-4267, 1986.

[12] D.W. Tank and J. J. Hopfield, "Simple 'Neural' Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit," *IEEE Trans. CAS-33(5)*, May 1986.

[13] D. Terzopoulos, "Multigrid Relaxation Methods and the Analysis of Lightness, Shading, and Flow," MIT AI Laboratory Memo 803, October 1984.

[14] W.D. Hillis, *The Connection Machine*, MIT Press, Cambridge, MA, 1985.

[15] C. A. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley, Reading, MA, 1988.

[16] J. Harris, C. Koch, J. Luo, and J. Wyatt, "Resistive Fuses: Analog Hardware for Detecting Discontinuities in Early Vision," in *Analog VLSI Implementation of Neural Systems*, C.A. Mead and M. Ismail, eds., Kluwer, 1989.

[17] J. Harris, C. Koch, J. Luo, "A Two-Dimensional Analog VLSI Circuit for Detecting Discontinuities in Early Vision," *Science*, vol. 248, pp. 1209-1211, June 8, 1990.

[18] J. Harris, C. Koch, E. Staats, J. Luo, "Analog Hardware for Detecting Discontinuities in Early Vision," *Int. J. Comp. Vision,* vol.4, pp. 211-223, 1990.

[19] T. Poggio and C. Koch, "Ill-Posed Problems in Early Vision: From Computational Theory to Analogue Networks," *Proc. Roy. Soc. Lond. B* 226:303-323, 1985.

[20] B.K.P. Horn, "Parallel Networks for Machine Vision," MIT AI Laboratory Memo 1071, August 1988.

[21] W. Millar, "Some General Theorems for Non-Linear Systems Possessing Resistance," *Phil. Mag.*, 42:1150-1160, 1951.

[22] I. Elfadel, "Note on a Switching Network for Image Segmentation," unpublished manuscript, October 1988.

[23] P. M. Hart, "Reciprocity, Power Dissipation, and the Thevenin Circuit," *IEEE Trans. CAS-33(7)*, pp.716-718, July 1986.

[24] P. Cristea, F. Spinei, and R. Tuduce, "Comments on 'Reciprocity, Power Dissipation, and the Thevenin Circuit,"' *IEEE Trans. CAS-34(10)*, pp.1255-1257, October 1987.

[25] P. Penfield, Jr., R. Spence, and S. Duinker, *Tellegen's Theorem and Electrical Networks*, MIT Press, Cambridge, MA, 1970.

[26] B.D.H. Tellegen, "A General Network Theorem with Applications," *Philips Res. Rept.* 7, 259-269, August 1952.

[27] A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* 2nd Edition, McGraw-Hill, 1984.

[28] A. Lumsdaine, M. Silveira, and J. White, "Simlab User's Guide," To be published as an MIT memo.

[29] A. Lumsdaine, M. Silveira, and J. White, "CMVSIM User's Guide," To be published as an MIT memo.

[30] L. M. Silveira, A. Lumsdaine, J. White, "Parallel Simulation Algorithms for Grid-Based Analog Signal Processors," to appear in the *Proceedings of the International Conference on Computer Aided Design*, 1990.

[31] T. Richardson, "Scale Independent Piecewise Smooth Segmentation of Images Via Variational Methods," MIT Laboratory for Information and Decision Systems Technical Report LIDS-TH-1940, February 1990.

[32] A. Witkin, "Scale-Space Filtering," *International Joint Conference on Artificial Intelligence,* pp. 1019-1021, Karlsruhe, 1983.

[33] H. Keller, "Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems," in *Applications of Bifurcation Theory*, P. Rabinowitz, ed., Academic Press, New York, 1977.

[34] T. Poggio, E. B. Gamble, and J. J. Little, "Parallel Integration of Vision Modules", *Science*, vol. 242, pp. 436–440, 1988.

[35] T. Poggio, J. Little, E. Gamble, W. Gillett, D. Geiger, D. Weinshall, M. Villalba, N. Larson, T. Cass, Bülthoff, M. Drumheller, P. Oppenheimer, W. Yang, and A. Hurlbert, *The MIT Vision Machine*, Morgan Kaufmann, San Mateo, CA, 1988.