Copyright © 1991, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

A GENERALIZED CELLULAR NEURAL NETWORK IMPLEMENTATION OF A NOVEL EDGE DETECTION ALGORITHM

by

Bertram E. Shi and Leon O. Chua

Memorandum No. UCB/ERL M91/25

9 April 1991

A GENERALIZED CELLULAR NEURAL NETWORK IMPLEMENTATION OF A NOVEL EDGE DETECTION ALGORITHM

by

Bertram E. Shi and Leon O. Chua

Memorandum No. UCB/ERL M91/25

9 April 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A GENERALIZED CELLULAR NEURAL NETWORK IMPLEMENTATION OF A NOVEL EDGE DETECTION ALGORITHM

by

Bertram E. Shi and Leon O. Chua

Memorandum No. UCB/ERL M91/25

9 April 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A Generalized Cellular Neural Network Implementation of a Novel Edge Detection Algorithm*

Bertram E. Shi Leon O. Chua†

April 9, 1991

Abstract

Chua and Yang have proposed an analog VLSI circuit architecture for image processing they have dubbed Cellular Neural Networks (CNNs). To date most of the proposed applications for CNNs have been for processing of binary (black or white) images. We present here a new edge detection algorithm for gray scale images which is extremely well suited for VLSI implementation under the recently proposed framework of Generalized CNNs. Although several authors have already proposed various analog circuit networks for edge detection, the VLSI implementation of the algorithm presented here overcomes some of the major shortcomings of previously proposed analog circuit networks for edge detection, while maintaining many of their advantages.

1 Introduction

Edge detection is one of the primary problems in research into early visual processing. It is an important preprocessing step for many computer vision algorithms. The location of intensity discontinuities is one of the primary components of the *raw primal sketch* proposed by Marr[19]. Marr asserts that the importance of the raw primal sketch "is that it is the first representation derived from an image whose primitives have a high probability of reflecting physical reality directly." Canny[5] states that "The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about object boundaries." For example, the stereo depth detection algorithm of Grimson utilizes edges detected in a preprocessing

*This research is supported in part by a National Science Foundation Graduate Research Fellowship and by the Office of Naval Research under Grant N00014-89-J1402.

†The authors are with the Electronics Research Laboratory of the University of California at Berkeley.

stage[10]. Ullman's approach to the interpretation of visual motion begins with establishing the correspondence between low level *correspondence tokens* such as edges[30]. Indeed, the applications for edge detection are quite diverse [13, 1, 4].

Recently, researchers have been paying increased attention to the possibility of using analog circuit arrays to perform early vision processing[17, 21, 22]. This has lead to the introduction of several analog circuit networks for edge detection. Poggio, Torre and Koch[27] note that many early vision processes, including edge detection, are ill posed in the sense defined by Hadamard. They propose to overcome this ill posedness by using *a priori* knowledge to restrict the class of admissible solutions. The *a priori* knowledge they use is essentially an assumption on the smoothness of visual surfaces. They exploit variational regularization methods developed mainly by Tikhonov to recast early visual problems as functional minimizations. It turns out that the minimization required by regularization involving quadratic functionals is ideally suited for implementation using analog circuit networks consisting of voltage sources and linear resistors[26]. Recently, several researchers have discovered that some algorithms for edge detection map naturally onto *non-linear* resistive networks. However, the analog VLSI implementations of these networks suffer from several disadvantages which we will discuss in more detail below.

In a related effort, motivated by the resurgence in interest in the potential of real time processing using the massive parallelism of neural networks, Chua and Yang have proposed an analog circuit architecture called Cellular Neural Networks[7, 8] (CNNs) which has many applications in image processing[15]. The primary advantage of the cellular neural network architecture is each basic processing unit, called a "cell", is restricted to interconnect only with cells within a small neighborhood of itself. This simplifies their implementation in analog VLSI. In fact, a chip implementing a "connected component detector"[20] has been successfully fabricated and tested. Unfortunately, the original CNN architecture is primarily useful for binary (black or white) image processing, although the half-toning CNN is a notable exception. To overcome this difficulty, the CNN architecture has recently been generalized while maintaining the features which make it easily implementable in VLSI.

In this paper, we present a new edge detection algorithm for gray scale images which can be implemented in analog VLSI under the framework of Generalized Cellular Neural Networks (GCNNs). The GCNN implementation of this algorithm overcomes many of the

major shortcomings of previous analog VLSI architectures for edge detection. The paper is organized as follows. In section 2, we discuss three previously proposed analog networks for edge detection and their advantages and disadvantages. Then in section 3 we present the new algorithm. We defer the rigorous analysis of the algorithm to a later paper, but highlight some of the key results pertaining to parameter choice and give examples of the algorithm operating on real images in section 4. In section 5, we propose two possible implementations of the algorithm in the GCNN architecture and discuss their advantages over previous approaches. Finally, we conclude in section 6 by summarizing the results we have presented here.

2 Previous Work

We begin by briefly reviewing some of the important issues in edge detection, before introducing the three main edge detection algorithms which have been proposed for implementation in analog VLSI. The algorithms we will discuss were developed by Perona and Malik[24], Blake and Zisserman[3] and Nordstrom[23]. Although Koch, Marroquin and Yuille[17] have also proposed an analog network edge detection, we have chosen not to discuss this network as it appears to be too complicated for effective implementation in analog VLSI. We conclude this section by discussing the major shortcomings of the VLSI implementations of these algorithms which our algorithm overcomes.

Up until the early 1980's almost all of the algorithms for edge detection were essentially based upon searching for the local maxima of the first derivative or the zeros of the second derivative of a linearly low pass filtered version of the image, [5, 9, 18]. The low pass filtering is typically accomplished by convolving the image intensity data with a Gaussian or Gaussian-like convolution kernel. Low pass filtering serves two purposes. Because the computation of the derivative is extremely sensitive to high frequency noise, filtering the image helps prevent spurious responses to noise. In addition, convolving the image with convolution kernels of varying widths gives rise to edge based representations of the image at different levels of resolution or "scales"[19, 31]. Many researchers acknowledge the utility of multiple representations of an image at varying resolutions. For example, Grimson's stereo depth detection algorithm begins by establishing correspondence between edges in the image at very coarse scales to avoid the problem of "false targets" encountered at very fine resolutions.

Unfortunately, although the linear filtering methods are simple and relatively easy to analyze, edge detection using this approach entails a fundamental uncertainty principle between the accurate localization of an edge and the scale of the linear filter[5].

More recently, several non-linear algorithms to edge detection have been proposed. Being non-linear, they are not subject to the above uncertainty principle. In fact, one of the primary advantages of these algorithms is that the edges remain well localized over varying scale. Unfortunately, due to the complexity of these algorithms, they tend to be quite computationally intensive on a traditional digital computer. In order to overcome this, several researchers have proposed to implement some of these algorithms using analog VLSI resistive grids. Perona and Malik have proposed an anisotropic diffusion algorithm for edge detection and an associated analog VLSI implementation[24]. Nordstrom modifies Perona and Malik's algorithm and derives a biased anisotropic diffusion algorithm which can also be implemented in analog VLSI[23]. Harris, Koch, Luo and Wyatt discuss an implementation of the Graduated Non Convexity (GNC) algorithm of Blake and Zisserman[3] using "resistive fuses"[11]. In fact, two dimensional resistive fuse network has already been successfully built and tested[12]. It should be noted the GNC algorithm is actually a method of visual surface reconstruction which automatically detects the locations of surface discontinuities. However, it can be used as an edge detection algorithm by using the image intensity data as input to the algorithm.

Perona and Malik's anisotropic diffusion algorithm is based upon the relationship between convolution with a Gaussian and the solution of the diffusion equation[14]. Finding the solution at time t of the diffusion equation with a constant coefficient of diffusion and initial distribution equal to the image intensity is equivalent to convolving the image intensity with a Gaussian distribution of variance proportional to $1/t$. Perona and Malik essentially allow for a spatially varying diffusion coefficient and develop an algorithm which filters the image using a non-linear diffusion process they call anisotropic diffusion. They show that this process has the desirable property that edges are enhanced and show by example that the edges detected by thresholding the gradient of the filtered image are thin and remain well localized at all scales.

The algorithms of Blake and Zisserman and Nordstrom have their roots in the Terzopoulos' regularization approach to surface reconstruction with discontinuities. Terzopou-

los approaches the problem of reconstructing a surface from noisy or sparse data by finding functions v and c defined on the image plane which minimize a cost function of the following form:

$$\mathcal{F}(v,c) = \mathcal{P}(v) + \mathcal{S}(v,c) + \mathcal{D}(c)$$

In the above equation, v is the reconstructed surface and c is a "continuity control function" which controls the smoothness of the reconstruction. $\mathcal{P}$ is a functional measuring the distance between the reconstructed surface and the data. $\mathcal{S}$ is a functional penalizing excessive variation of the reconstruction, except at points of discontinuity specified by c. $\mathcal{D}$ is a functional penalizing the introduction of discontinuities. When the input to the reconstruction algorithm is the image intensity, the points of discontinuity specified by c correspond to the edges of the image.

While Terzopoulos uses relatively complex functionals and performs minimization with respect to v and c alternatively, Blake and Zisserman and Nordstrom choose simpler functionals and show that for fixed v the c which minimizes the functional $\mathcal{F}$ is unique. Minimizing first with respect to c reduces the problem to the minimization of a non-convex functional depending only upon v. Once this minimization is performed, the c function can be recovered from the v data. To overcome the problem of multiple local minima of this non-convex functional, Blake and Zisserman's Graduated Non-Convexity algorithm begins by minimizing a strictly convex approximation to the non-convex functional and successively minimizes closer and closer approximations using the result of the previous step as the initial conditions to the next. Blake and Zisserman argue that the solution found in this way will be nearly optimal. Nordstrom, on the other hand, uses the Euler-Lagrange equations to show that the minimum of the functional he obtains must satisfy a non-linear partial differential equation. His approach is to solve this PDE using relaxation methods. Due to the presence of multiple solutions to this PDE, the edges his algorithm obtains vary with the initial conditions of the relaxation algorithm.

The proposed analog VLSI implementations of these algorithms are quite similar. They all use a two dimensional grid of non-linear resistors, where each node of the grid corresponds to one pixel of the image. This grid architecture has the advantage that the interconnections are purely local. Perona and Malik propose an implementation of their algorithm using a non-linear resistive grid where each node is grounded through a capacitor (Figure 1). To process

Figure 1: The array of Perona and Malik settles to a steady state where the voltages across the capacitors are all equal.

Figure 2: The nonlinear resistive arrays of Harris and Nordstrom are similar to the array of Perona and Malik.

an image, the voltage across each capacitor is initialized to the intensity of the associated image pixel. As the circuit settles, the node voltages evolve according to a continuous time discrete space approximation to anisotropic diffusion. However, this implementation has the disadvantage that the desired result occurs at an intermediate stage of the circuit's transient response. In fact, the steady state of the array contains no useful information as all the capacitors will have the same voltage. Thus, the capacitors of the array must be large enough so that the readout of the voltages can be done quickly relative to the settling time of the array. Even if the transient could be stopped at some point by somehow disconnecting the capacitors from the resistive array, current leakage from the capacitors would still be a problem in practice.

Unlike the Perona and Malik's circuit, in Harris et al.'s and Nordstrom's circuits the desired result of the computation occurs at steady state. Harris et al.'s analog circuit implementation of Blake and Zisserman's algorithm consists of a two dimensional grid of voltage controlled non-linear resistors where the nodes are grounded through branches consisting of a linear resistor in series with an independent voltage source whose output is equal to the intensity of the corresponding image pixel (Figure 2). The node voltages correspond to the approximation, v, to the image data and the total resistive co-content of the array corresponds to the non-convex cost functional of the GNC algorithm. The non-linearity of the grid resistors can be controlled by an external voltage. The sequence of function approximations of the GNC algorithm is approximated by adjusting the grid resistances slowly from

a saturating resistance to the the final target non-linearity. Thus, this implementation has the disadvantage that manual adjustment of the circuit is necessary for correct processing of the image.

Nordstrom's algorithm maps onto to a similar array where the non-linear resistors are replaced by voltage controlled conductances controlled by the voltages of six of the surrounding nodes. Because of the practical difficulty in implementing these voltage controlled conductances, he proposes using an approximation to his algorithm which results in a two dimensional grid of voltage controlled non-linear resistors. The I-V characteristic of his non-linear resistors is similar to that of the target non-linearity in the implementation of Harris et al. Instead of manually adjusting the resistive co-content of the network, Nordstrom proposes to load the voltages and allow the circuit to settle to steady state, arguing that by design the circuit will settle to a "meaningful" steady state. In this case no manual adjustment is necessary. However, the non-convexity of the resistive co-content of the array leads to many possible stable steady state voltage configurations of this array. Because of this non-uniqueness, the final voltage distribution will depend upon the sequence with which the voltages were loaded and on the values and initial conditions of various parasitic elements. Therefore, unless circuitry to initialize the node voltages and capacitors large enough to dominate the effects of parasitic elements are also incorporated into the circuit array, the final steady state of the array cannot be reliably predicted. This would be especially undesirable in applications requiring matching edges between two images such as stereo depth detection.

3 A New Edge Detection Algorithm

The analog circuit implementation of the edge detection algorithm we propose below does not suffer from the disadvantages of the circuit arrays we discussed above. In this section, we present the heuristic motivation behind our approach and the mathematical derivation of the algorithm. Since we are primarily interested in the analog circuit implementation of the algorithm we embed our derivation in a circuit theoretic framework. A brief discussion of the implementation of the algorithm using traditional digital computers concludes this section.

We first give an overview of the algorithm to provide a framework for our subsequent

derivation. Following the regularization approach, the algorithm begins by constructing an approximation to the image which represents a compromise between fidelity to the image data and the smoothness of the approximation. This is accomplished by finding an approximation which minimizes a cost functional penalizing both the distance between the approximation and the data and the variation of the waveform at each point. Then, the algorithm locates the points of the image where decreasing the penalty on excessive variation in the approximation results in the greatest decrease in the error between the image and the approximation. Since smoothing across sharp discontinuities in the image data leads to the greatest error, the points located in this step should correspond to locations of sharp discontinuities. These points are labeled as edge locations.

For mathematical simplicity, we develop the algorithm assuming a one dimensional discrete image space consisting of n points, before generalizing to two dimensions. We consider a sequence of m scalar values $x_i, i = 1, \ldots m$ as both a discrete waveform and as an m dimensional vector and will use the two terms interchangeably. Both the waveform and vector will be denoted by by the unsubscripted letter x. The notation $x > 0$ will denote that every component of x is greater than zero. In the two dimensional case, with a slight abuse of notation we will also use the unsubscripted letter to denote the two dimensional discrete waveform.

Let $d_i, i = 1, \ldots n$ be the image intensity, and $v_i, i = 1, \ldots n$ be a discrete waveform. Let ω and $c_i, i = 1, \ldots n-1$ be positive constants. Consider the following functional:

$$\mathcal{F}(v) = \omega^2 \sum_{i=1}^{n} (v_i - d_i)^2 + \sum_{i=1}^{n-1} c_i (v_i - v_{i+1})^2$$

The waveform v which minimizes $\mathcal{F}(v)$ is the desired approximation. In the continuous limit, $\mathcal{F}$ is similar to the functionals used by Blake and Zisserman and Nordstrom, except that the term corresponding to the penalization of the introduction of discontinuities is missing. The first term measures the distance between v and the image data. The second measures the smoothness of v. It is easy to see that this functional is exactly the total resistive co-content of a one dimensional grid of linear resistors with conductance c_i whose nodes are grounded through resistances with conductance ω and independent voltages sources which output d_i volts. The nodal voltages correspond to the v_i variables (Figure 3).

The conductivity of the resistances linking the voltage sources to the resistive grid, ω, gives global control of the smoothness of the approximation. For small ω relative to c_i,

Figure 3: The functional $\mathcal{F}$ is exactly the total resistive co-content of a one dimensional linear resistive grid.

smoothness is favored in the reconstruction as the fidelity cost is weighted less in the co-content function. Thus, the resulting approximation v will be smoother than for large ω. On the other hand, c gives local control to the smoothness of the approximation. The sequence c is analogous to the continuity control function of Terzopoulos. For large c_i, the difference between v_i and v_{i+1} is heavily penalized in the co-content function. Smoothness is favored in the approximation v at those points. In the limit of $c_i = \infty$, the resistive link between nodes v_i and v_{i+1} is a short circuit and $v_i = v_{i+1}$. Conversely for small c_i, the difference $v_i - v_{i+1}$ is lightly penalized. In the limit $c_i = 0$, the resistive link is an open circuit.

Consider the case of a uniform resistive grid where all the c_i equal to one. Sharp discontinuities in the approximating waveform v are penalized equally at all points and the overall smoothness of the approximation is controlled by the value of ω. Assume that for all image data d the discrete waveform v is a well defined function of c for $c > 0$ and consider the following error functional

$$\mathcal{E}(c) = \frac{1}{2}\sum_{i=1}^{n}(v_i(c) - d_i)^2$$

between the approximation and the data. The functional $\mathcal{E}$ depends only *implicitly* on the vector c through the dependence of the approximation on c. The ith component of the gradient of this functional, $\nabla\mathcal{E}$, is the partial derivative of the error with respect to the c_i, the conductance of the ith resistor of the resistive grid. The local maxima of the gradient considered as a discrete $n-1$ point waveform correspond to those conductances of the resistive grid which when decreased will result in the greatest decrease in the value of $\mathcal{E}$ in the limit as the amount of decrease tends to zero. The locations of resistors corresponding to local maxima above a certain threshold value will be detected by our algorithm as edge locations.

Before computing the gradient, we verify the validity of the assumption that v is a well defined function of c for all $c > 0$. By the stationary co-content theorem[6], each local

minimum of the co-content corresponds to a steady state nodal voltage distribution of the resistive grid. In addition, any steady state must satisfy Kirchoff's current law:

$$\begin{aligned} \omega^2(v_1 - d_1) + c_1(v_1 - v_2) &= 0 \\ \omega^2(v_i - d_i) + c_{i-1}(v_i - v_{i-1}) + c_i(v_i - v_{i+1}) &= 0 \qquad i = 2, \ldots n-1 \\ \omega^2(v_n - d_n) + c_{n-1}(v_n - v_{n-1}) &= 0 \end{aligned}$$

Using matrix notation, we can simplify this to the following:

$$\mathbf{C}(c)v = \omega^2 d \tag{1}$$

where $\mathbf{C}(c)$ is an n by n tridiagonal matrix

$$\mathbf{C}(c) = \begin{bmatrix} \omega^2+c_1 & -c_1 & & & & & & \\ -c_1 & c_1+\omega^2+c_2 & -c_2 & & & & & \\ & -c_2 & \ddots & & & & & \\ & & & -c_i & & & & \\ & & & -c_i & c_i+\omega^2+c_{i+1} & -c_{i+1} & & \\ & & & & -c_{i+1} & \ddots & & \\ & & & & & & -c_{n-2} & \\ & & & & & -c_{n-2} & c_{n-2}+\omega^2+c_{n-1} & -c_{n-1} \\ & & & & & & -c_{n-1} & c_{n-1}+\omega^2 \end{bmatrix}$$

By symmetry and the Gerschgorin circle criterion[25], the matrix $\mathbf{C}(c)$ is positive definite for all $c > 0$. Thus, the steady state of the array (and the v which minimizes $\mathcal{F}$) is unique for all $c > 0$ and all d. The inverse of a matrix is a continuous function of its entries as long as the matrix is non-singular. Thus, v is a well defined and continuous function of c. and the functional $\mathcal{E}(c)$ is well defined.

To compute an expression for gradient, first apply the chain rule to $\mathcal{E}$.

$$\nabla\mathcal{E} = \frac{dv}{dc}^T \frac{\partial\mathcal{E}}{\partial v}^T$$

Clearly, $\frac{d\mathcal{E}}{dv}^T = v - d$. To find $\frac{dv}{dc}$, implicitly differentiate (1) with respect to c to obtain

$$\mathbf{C}(c)\frac{dv}{dc} + \mathbf{D} = 0$$

where $\mathbf{D}$ is an n by $n-1$ dimensional matrix whose ith column, $\mathbf{D}_i$, is

$$\mathbf{D}_i = \begin{bmatrix} 0 \\ \vdots \\ v_{i+1} - v_i \\ v_i - v_{i+1} \\ 0 \\ \vdots \end{bmatrix}$$

The first $i-1$ elements and the last $n-i-1$ elements of $\mathbf{D}_i$ are zero. Since $\mathbf{C}(c)^{-1}$ is well defined for $c > 0$, $\frac{dv}{dc}$ is unique and well defined for all $\mathbf{D}$ and $c > 0$. Finally, exploiting the fact that $\mathbf{C}$ is symmetric we obtain

$$\nabla \mathcal{E} = \mathbf{D}^T \mathbf{C}^{-1}(v - d)$$

At first glance it might appear that the gradient computation would be very difficult to accomplish in analog VLSI, as it involves a matrix inverse. However, it turns out that this computation can be accomplished very simply using another resistive grid similar to the one used to compute the approximation. Define $p = \mathbf{C}^{-1}(v - d)$. Then

$$\nabla \mathcal{E} = \mathbf{D}^T p \tag{2}$$

where p is the solution of

$$\mathbf{C}(c)p = v - d \tag{3}$$

Note the similarity between this equation and (1). The vector p is the unique minimum of the functional

$$\mathcal{G}(p) = \omega^2 \sum_{i=1}^{n} (p_i - \frac{v_i - d_i}{\omega^2})^2 + \sum_{i=1}^{n-1} c_i (p_i - p_{i+1})^2$$

which is the resistive co-content of the linear resistive array in figure 3 where d_i is replaced by $\frac{v_i - d_i}{\omega^2}$ and the nodal voltages correspond to p_i rather than v_i. By (2) and the form of the matrix $\mathbf{D}$, the ith component of the gradient, $\nabla \mathcal{E}_i$, is the negative of the product of the voltages across the ith resistor of each grid

$$\nabla \mathcal{E}_i = \frac{\partial \mathcal{E}}{\partial c_i} = -(v_{i+1} - v_i)(p_{i+1} - p_i)$$

Finally, the locations of the edges of the image are determined by thresholding the local maxima of $\nabla \mathcal{E}$. Since $\nabla \mathcal{E}$ is a one-dimensional waveform, the detection of local maxima

can be accomplished by comparing the value of each $\nabla\mathcal{E}_i$ to its neighbors in some neighborhood to either side. The location of the local maxima and thresholding can be effectively accomplished using a post processing step implemented on a traditional digital computer. As mentioned above, the edges detected are actually located between two adjacent pixels of the image.

The generalization to two dimensions is quite straight forward. Assume a discrete m by n pixel image space. We split the continuity control function into two parts, c^h and c^v, corresponding to coefficients weighting variations in the horizontal and vertical directions. The functionals $\mathcal{F}$ and $\mathcal{E}$ are modified to

$$\begin{aligned}\mathcal{F}(v) &= \omega^2\sum_{i=1}^{m}\sum_{j=1}^{n}(v_{i,j}-d_{i,j})^2+\sum_{i=1}^{m}\sum_{j=1}^{n-1}c^h_{i,j}(v_{i,j}-v_{i,j+1})^2 \\ &\quad+\sum_{i=1}^{m-1}\sum_{j=1}^{n}c^v_{i,j}(v_{i,j}-v_{i+1,j})^2 \\ \mathcal{E}(c) &= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(v_{i,j}-d_{i,j})^2\end{aligned}$$

The minimizing v and p can be computed using a two-dimensional resistive grid where the c^v correspond to the conductances of the vertical grid resistances and the c^h correspond to the conductances of the horizontal grid resistances. As in the one-dimensional case, the derivative of $\mathcal{E}$ with respect to each resistance in the first resistive array is the negative of the product of the voltages across the corresponding resistances in both arrays.

$$\begin{aligned}\frac{\partial\mathcal{E}}{\partial c^h_{i,j}} &= -(v_{i,j+1}-v_{i,j})(p_{i,j+1}-p_{i.j})\; i\in\{1,\ldots m\}j\in\{1,\ldots n-1\} \\ \frac{\partial\mathcal{E}}{\partial c^v_{i,j}} &= -(v_{i+1,j}-v_{i,j})(p_{i+1,j}-p_{i.j})\; i\in\{1,\ldots m-1\}j\in\{1,\ldots n\}\end{aligned}$$

The problem of searching for the local maxima in two dimensions is simplified by the fact that the continuity control function has been split into two parts, which control to the horizontal and vertical smoothness of the approximation. Although it is possible to use more complex methods, for the sake of simplicity we have exploited this partition to reduce the problem to searching for local maxima along one dimension. For each $i\in\{1,\ldots m\}$, we define the one-dimensional waveform

$$\nabla\mathcal{E}^h_i(j)=\frac{\partial\mathcal{E}}{\partial c^h_{i,j}}\qquad j=1,\ldots n-1$$

Similarly, define for each $j \in \{1,\ldots n\}$,

$$\nabla\mathcal{E}^v_j(i) = \frac{\partial \mathcal{E}}{\partial c^h_{i,j}} \qquad i = 1,\ldots m-1$$

The detection of local maxima can be done by comparing the value at each point of each one-dimensional waveform to the values in a neighborhood to either side as in the one-dimensional algorithm. However, in the two dimensional case, instead of simply thresholding these values to find the edge locations, we use a thresholding with hysteresis process which is similar to that used by Canny's edge detection algorithm[5]. In this process, two threshold values t_1 and t_2 are chosed where $t_1 < t_2$. "Strong" edges are defined by the local maxima which exceed t_2. Similarly, "weak" edges correspond to local maxima which lie between t_1 and t_2. In thresholding with hysteresis, all strong edges are labeled as edge locations. In addition, all weak edge contours which are connected to the strong edges are also labeled as edges. However, weak edges which are not connected to strong edges are not labeled. This process leads to long connected edges while also reducing the number of isolated false edge points which would arise if the image were simply thresholded at the lower value. Table 1 summarizes the two-dimensional algorithm.

Before concluding this section, we briefly digress to discuss the implementation of this algorithm using a traditional digital computer. Since the post-processing phase consisting of the detection of local maxima and thresholding is easily computed using conventional digital hardware, we discuss only the digital computer solution of the resistive grid equations. In the one-dimensional case, $\mathbf{C}(c)$ is a non-singular tridiagonal matrix so (1) and (3) can be solved in $O(n)$ operations using the LU decomposition[28]. For general matrices, the LU decomposition is not as efficient as in the the tridiagonal case. Fortunately, the linear sets of equations resulting from Kirchoff's current law for the two uniform dimensional grid correspond exactly to the equations of the 5 point discrete approximation to the modified Helmholtz equation with constant coefficients and Neumann boundary conditions of vanishing normal derivative at all points of the boundary. These can be solved on a serial computer using efficient relaxation algorithms such as the alternating direction implicit method of operating splitting[28] or standard routines for solving discrete approximations to elliptic partial differential equations[2].

Table 1: Summary of the two-dimensional algorithm

Given: $d_{i,j}$, $i \in \{1,\ldots m\}$ $j \in \{1,\ldots n\}$

Step 0: Choose $\omega > 0$, $t_2 > t_1 > 0$, $\delta > 0$

Step 1: Compute v minimizing

$$\begin{aligned}\mathcal{F}(v) &= \omega^2 \sum_{i=1}^{m}\sum_{j=1}^{n}(v_{i,j} - d_{i,j})^2 + \sum_{i=1}^{m-1}\sum_{j=1}^{n}(v_{i,j} - v_{i,j+1})^2 \\ &\quad + \sum_{i=1}^{m}\sum_{j=1}^{n-1}(v_{i,j} - v_{i+1,j})^2\end{aligned}$$

Step 2: Compute p minimizing

$$\begin{aligned}\mathcal{G}(p) &= \omega^2 \sum_{i=1}^{m}\sum_{j=1}^{n}(p_{i,j} - \frac{v_{i,j} - d_{i,j}}{\omega^2})^2 + \sum_{i=1}^{m-1}\sum_{j=1}^{n}(v_{i,j} - v_{i,j+1})^2 \\ &\quad + \sum_{i=1}^{m}\sum_{j=1}^{n-1}(v_{i,j} - v_{i+1,j})^2\end{aligned}$$

Step 3: Compute for $i \in \{1,\ldots m\}$

$$\nabla\mathcal{E}_i^h(j) = -(v_{i,j+1} - v_{i,j})(p_{i,j+1} - p_{i.j}) \qquad j \in \{1,\ldots n-1\}$$

and for $j \in \{1,\ldots m\}$

$$\nabla\mathcal{E}_j^v(i) = -(v_{i+1,j} - v_{i,j})(p_{i+1,j} - p_{i.j}) \qquad i \in \{1,\ldots n-1\}$$

Step 4: Find local maxima by comparing the value of each element of the one dimensional waveforms $\nabla\mathcal{E}_i^h(\cdot)$ and $\nabla\mathcal{E}_j^v(\cdot)$ to the value of each element in a neighborhood of size δ around it.

Step 5: Threshold local maxima using hysteresis with t_1 and t_2 to determine edge locations.

4 Parameter Choice and Examples

In a forthcoming paper, we will give the details of a quantitative analysis of the algorithm's performance. For now, we will highlight some of the key results pertaining to the choice of parameter values and give some examples of the algorithms performance on real images.

For simplicity we assume an infinite one-dimensional discrete image space. This is equivalent to restricting our attention to two-dimensional images with variation in only one dimension. This is a good approximation in the case of long linear edges, but breaks down in the case of corners and junctions in the image. In this case, the solution to steps 1 and 2 of the algorithm can be solved using a discrete convolution.

$$v_i = \sum_{k=-\infty}^{\infty} G(i-k)d_k \tag{4}$$

where

$$\begin{aligned} G(i-k) &= \frac{\omega}{\sqrt{4+\omega^2}} e^{-\alpha|i-k|} \\ \alpha &= \cosh^{-1}\left(\frac{2+\omega^2}{2}\right) \end{aligned}$$

The linearity of the convolution and the fact that the only non-linearity in the algorithm is a multiplicative non-linearity performed at the end of the algorithm considerably simplifies the analysis of the algorithm. It is straightforward to show that the gradient waveform resulting from an image consisting of a single step edge of height h between pixels 0 and 1 is

$$\nabla \mathcal{E}_i = \frac{h^2}{4+\omega^2}\left(1 - \frac{2+\omega^2}{4+\omega^2}\right) e^{-2\alpha|i|} - \frac{h^2 w}{(4+\omega^2)^{\frac{3}{2}}}|i| e^{-2\alpha|i|}.$$

This expression attains its maximum at $i = 0$. The algorithm interprets an above threshold local maximum of the gradient waveform at $i = 0$ as an edge between pixels 0 and 1, since the 0th component of the gradient is the derivative of the error functional with respect to the conductance of the resistance linking the nodes corresponding to the pixels 0 and 1. Therefore given ω, a threshold value of

$$t = \frac{h^2}{4+\omega^2}\left(1 - \frac{2+\omega^2}{4+\omega^2}\right) \tag{5}$$

will ensure that all isolated step edges of height greater than h will be detected.

Similar to Blake and Zisserman, we demonstrate that the value of ω determines the characteristic distance for the interaction of discontinuities by considering the response of

the algorithm to an image waveform consisting of two step discontinuities of height h_1 and h_2 at pixels 0 and ϵ respectively. The resulting gradient waveform is

$$\nabla \mathcal{E}_i = \left(1 + \frac{h_2 e^{-\alpha|i-\epsilon|}}{h_1 e^{-\alpha|i|}}\right) h_1^2 \chi_i + \left(1 + \frac{h_1 e^{-\alpha|i|}}{h_2 e^{-\alpha|i-\epsilon|}}\right) h_2^2 \chi_{i-\epsilon}$$

where χ_i is the step response to an isolated step of height 1. Thus, the response is equal to the superposition of the responses of isolated steps except for a space-varying coefficient parameterized by α. Consider a "top hat" input consisting of two step discontinuities of heights h and $-h$ at pixel 0 and pixel ϵ respectively. Then $\frac{1}{\alpha}$ is a characteristic length of discontinuity interaction. If $\epsilon > \frac{1}{\alpha}$ the response of each step is nearly the same as the response to an isolated step edge. If $\epsilon < \frac{1}{\alpha}$, then the response to each step decreases to zero as ϵ tends to zero. Since $\alpha = \cosh^{-1}(\frac{2+\omega^2}{2}) \approx \omega$ for small ω, the choice of ω determines the "scale" at which edges are detected.

Unfortunately, in the case of general discontinuities this superposition-type property may lead to spurious responses to small edges. Equation (4) indicates that for coarse scales (small ω) the response to an edge is broader than for large ω. At coarse scales, this may increase the response to smaller edges around a large edge, leading to a spurious responses to a small edges near large edges. This is especially true in the presence of noise. Experimentally, we have found that setting the size of the comparison neighborhood for detection of the local maximum detection to be a fixed fraction ($\frac{1}{4}$) of the value of $\frac{1}{\omega}$ provides qualitatively "good" results. However, this is still an area of active research.

We now present several examples of our algorithm operating on real images. We give examples of the edges detected over varying scales and over varying ranges of noise added to the image. All of the images we use have a resolution of 8 bits, corresponding to 256 gray levels.

Figure 4 illustrates that the algorithm detects edges which are well localized over varying scale. We have run our algorithm on the image shown in figure 4(a). For comparison purposes, note that this image was also used as input to the anisotropic diffusion algorithm of Perona and Malik[24]. The algorithm was run with values of ω of 0.505, 0.25 and 0.0625, corresponding to characteristic discontinuity interaction lengths of 2, 4 and 16 pixels. Using (5), the threshold value t_1 was chosen so that all isolated step edges of height greater than 25 grey levels would be labeled as weak edges. The value of t_2 was chosen so that all isolated step edges of height greater than 50 would be labeled as strong edges. Using our

Figure 4: The algorithm detects edges well localized over varying scales. (a) The image. (b) Edges detected with $\omega = 0.505$. (c) Edges detected with $\omega = 0.25$. (d) Edges detected with $\omega = 0.0625$. (e) Edges detected by Canny algorithm with $\sigma = 1$ pixel. (f) Edges detected by Canny algorithm with $\sigma = 2$ pixels. (g) Edges detected by Canny algorithm with $\sigma = 4$ pixels.

Figure 5: The algorithm is quite robust in the presence of noise. (a) The noise free image. (b) The edges detected in (a). (c) The noisy image with $\sigma = 15$. (d) The edges detected in (c). (e) The noisy image with $\sigma = 30$. (f) the edges detected in (e).

experimental rule of thumb, the sizes of the neighborhoods for comparison for the detection of local maxima were chosen at 1, 1 and 4 pixels respectively. Note that the edges detected do remain well localized and that as ω decreases, the edges detected correspond to coarser and coarser scales. Contrast these results with the results of the Canny edge algorithm, also shown in figure 4.

To show the performance of the algorithm in the presence of noise, ω was fixed at 0.25 and the same values of t_1 and t_2 as above were used. The algorithm detected the edges of the image in figure 5(a) corrupted by additive Gaussian noise of standard deviation σ equal to 0, 15, and 30 grey levels. Even at this fairly fine scale, the algorithm is quite robust in the presence of noise.

Finally, figure 6 shows the effect of the thresholding with hysteresis. Our algorithm was run on the image in figure 6a with $\omega = 0.5$ and a maximum comparison neighborhood of size 1. The thresholds t_1 and t_2 were chosen so that isolated step edges of heights 15 and 30 would be detected as weak and strong edges respectively. Figures 6b and 6c show the edges detected by simple thresholding at the two threshold levels. Notice that thresholding at t_1 leads most of the edges to be detected, but also contains many spurious responses. On the other hand, thresholding at t_2 eliminates these spurious responses but leads to many broken

Figure 6: Thresholding with hysteresis gives better edges than simple thresholding. (a) The input image. (b) The edges detected by simple thresholding t_1. (c) The edges detected by simple thresholding at t_2. (d) The edges detected using thresholding with hysteresis.

or missed edges. Thresholding with hysteresis as shown in figure 6d effectively combines the two sets of edges, leading to long connected edges while avoiding spurious responses.

5 Implementation using GCNNs

In this section we describe two possible analog circuit implementations of this algorithm under the framework of Generalized Cellular Neural Networks. Our strategy is to use a hybrid analog-digital architecture. As we indicated at the end of section 3, the digital computer implementation of the functional minimizations can be quite computationally intensive on a traditional digital computer. However, as we indicated in that section, the solutions of the functional minimizations can be realized using linear resistive networks. The post processing of the gradient waveform to find the local maxima and thresholding can effectively be solved using a digital post processing step. There seems little reason to incorporate this stage into the analog network, as it would limit the number of nodes possible to put on a single chip,

One possible hybrid architecture is to use a chip consisting of a single resistive grid and compute steps 1 and 2 in two separate steps. In other words, the independent voltage sources are first initialized to the image data d. Once the array settles, the steady state node voltages corresponding to v are read off the chip and the independent voltage sources reset to the difference $\frac{v-d}{\omega^2}$. After the array settles again, the node voltages which now correspond to p are read off and the multiplication done in a digital post processing step. The disadvantage of this implementation is that it requires reading voltages onto and off of the chip twice. Not only is this fairly slow, too much noise may be introduced into the signals, due to the limited precision entailed with dealing with analog circuitry. A preferable architecture may be to build two identical resistive grids on a single chip and linking them using linear voltage controlled voltage sources. The first resistive grid calculates the minimization of step 1 of the

Figure 7: A double grid implementation of the edge detection algorithm.

algorithm. The voltage controlled voltage sources take the place of the independent voltage sources in the second resistive grid. The output of these voltage sources is the voltage across the resistors between the independent voltage sources and nodes of the first resistive grid multiplied by $\frac{1}{\omega^2}$. For small ω, the result of this multiplication may exceed the dynamic range of the circuit. Fortunately, however, this multiplication is actually unnecessary, as it only leads to a scaling of the gradient waveform. In this architecture, it is also desirable to implement the multiplication of step 3 on chip to reduce the amount of data which must be read from the chip (figure 7). This architecture has the disadvantage of high complexity.

In the absence of parasitic elements, the stability of the analog components of both hybrid architectures is trivially guaranteed. Since all of the elements of a single resistive array are strictly passive, the resistive grid itself is strictly passive and thus stable. In interconnecting the two resistive arrays, although the input to the second resistive array connection is time varying, stability is not compromised since there is no feedback between the two arrays. For stability results in the case of parasitic elements, see [29].

Like the circuits proposed by Harris et al. and Nordstrom, our implementation is designed to settle to the desired result of the computation. Therefore, unlike Perona and Malik's analog implementation, no mechanism to stop the transient response or large space consuming capacitors are required. On the other hand, unlike the resistive grids of Nordstrom and Harris, the resistive links of our resistive grids are *linear*. This linearity and the associated strict convexity of the resistive co-contents of our arrays is the key to the advantages of our implementation. The problem of non-uniqueness is eliminated. Therefore, unlike Harris' implementation, no manual adjusting of the circuit is necessary during processing. Unlike Nordstrom's implementation, the edges detected using our architecture are completely predictable, independent of the sequence the data is loaded onto the chip and the values of parasitic elements. In addition, Karplus has shown that linear resistive grids are robust against random errors in the values of the resistances[16].

Another consideration overlooked by previous analog implementations is the fact that the amount of nodes possible to place on a chip is limited. The chip built by Harris et al. contains a resistive network of 20 by 20 nodes. However, typical image processing applications involve images of 200 by 200 pixels and up. It is currently impossible to put that many nodes of any of the proposed resistive circuits onto a single chip, although it may soon be possible using wafer scale technology. Because resistive links between nodes are purely linear, it should be much easier to effectively link chips together using appropriate drive circuitry. Since the steady state nodal voltage distribution is independent of the time history of the nodal voltages, any increased capacitance incurred by going off chip is irrelevant to the final result. In contrast, the performance of Nordstrom's and Perona and Malik's proposed analog circuit arrays depends critically upon the time history of the circuit, making the result of the calculation much more sensitive to variations in the drive circuitry to link chips.

6 Conclusion

In this paper, we have described some of the problems in proposed analog implementations of current edge detection algorithms. To solve these problems we have proposed a new edge detection algorithm. We have provided some results regarding the choice of parameter values and have shown by example that the algorithm detects edges which are well localized over varying scales and fairly robust in the presence of noise.

This edge detection algorithm can be effectively implemented in a hybrid analog-digital architecture under the framework of generalized CNNs. The analog circuit implementation of this network overcomes many shortcomings of the previous architectures while preserving their advantages of purely local interconnections and highly parallel computation.

7 Acknowledgements

The authors would like to thank Prof. Jitendra Malik for his invaluable advice and assistance in our efforts in computer vision, and his generosity in allowing us access to his computers to run the Canny edge detection algorithm.

References

[1] R. J. Beattie. *Edge Detection for Semantically Based Early Visual Processing.* PhD thesis, Univ. Edinburgh, 1984.

[2] Garrett Birkhoff and Robert E. Lynch. *Numerical Solution of Elliptic Problems.* SIAM, Philadelphia, PA, 1984.

[3] Andrew Blake and Andrew Zisserman. *Visual Reconstruction.* The MIT Press, Cambridge, MA, 1987.

[4] R. A. Brooks. Symbolic reasoning among 3-d models and 2-d images. Technical Report AIM-343, Stanford Univ., 1981.

[5] John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-8:679–697, November 1986.

[6] L. O. Chua. Stationary principles and potential functions for nonlinear networks. *Journal of the Franklin Institute*, pages 91–114, August 1973.

[7] Leon O. Chua and Lin Yang. Cellular Neural Networks: Applications. *IEEE Transactions on Circuits and Systems*, 32, October 1988.

[8] Leon O. Chua and Lin Yang. Cellular Neural Networks: Theory. *IEEE Transactions on Circuits and Systems*, 32, October 1988.

[9] Larry S. Davis. A survey of edge detection techniques. *Computer Graphics and Image Processing*, 4:248–270, 1975.

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

[11] J. Harris, C. Koch, J. Luo, and J. Wyatt. Resistive fuses: Analog hardware for detecting discontinuities in early vision. In C. Mead and M. Ismail, editors, *Analog VLSI Implementation of Neural Systems*, chapter 2, pages 27–56. Kluwer Academic Publishers, Boston, 1989.

[12] J. G. Harris, C. Koch, and J. Luo. A two-dimensional analog vlsi circuit for detecting discontinuities in early vision. *Science*, 248:1209–1211, June 1990.

[13] B. K. P. Horn. The Binford-Horn line-finder. Technical Report AI Memo 285, M.I.T. AI Lab., Cambridge, MA, 1971.

[14] R. A. Hummel. Representations based on zero-crossings in scale-space. In *Proc. IEEE Computer Vision and Pattern Recognition Conf.*, pages 204–209, June 1986.

[15] IEEE. *Intl. Wrkshp. on CNNs and Their Applications*, Budapest, Hungary, 1990. IEEE.

[16] W. J. Karplus. *Analog Simulation: Solution of Field Problems.* McGraw-Hill, New York, 1958.

[17] Christof Koch, Jose Marroquin, and Alan Yuille. Analog "neuronal" networks in early vision. *Proc. Natl. Acad. Sci. USA*, 83:4263–4267, June 1986.

[18] D. Marr and E. Hildreth. Theory of edge detection. *Proc. R. Soc. Lond. B*, 207:187–217, 1980.

[19] David Marr. *Vision.* W. H. Freeman and Company, New York, 1982.

[20] T. Matsumoto, L. O. Chua, and H. Suzuki. CNN cloning template: Connected component detector. *IEEE Transactions on Circuits and Systems*, 37:633–635, May 1990.

[21] C. Mead. *Analog VLSI and Neural Systems.* Addison Wesley, Reading, MA, 1989.

[22] C. Mead and M. Ismail, editors. *Analog VLSI Implementation of Neural Systems.* Kluwer Academic Publishers, Boston, 1989.

[23] Niklas Karl Nordstrom. *Variational Edge Detection.* PhD thesis, UC Berkeley, 1990.

[24] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, July 1990.

[25] Sergio Pissanetzky. *Sparse Matrix Technology.* Academic Press, Inc., New York, 1984.

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

[27] Tomaso Poggio, Vincent Torre, and Cristof Koch. Computational vision and regularization theory. *Nature*, 317:314–319, September 1985.

[28] W. H. Press, B. P.Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipies in C: The Art of Scientific Computing.* Cambridge University Press, New York, 1988.

[29] D. L. Standley and Jr. J. L. Wyatt. Stability criterion for lateral inhibition and related networks that is robust in the presence of integrated circuit parasitics. *IEEE Trans. on Circuits and Systems*, 36(5):675–681, May 1989.

[30] S. Ullman. *The Interpretaton of Visual Motion.* The MIT Press, Cambridge, MA, 1985.

[31] A. Witkin. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, pages 1019–1021, Karlsruhe, 1983.

List of Figures

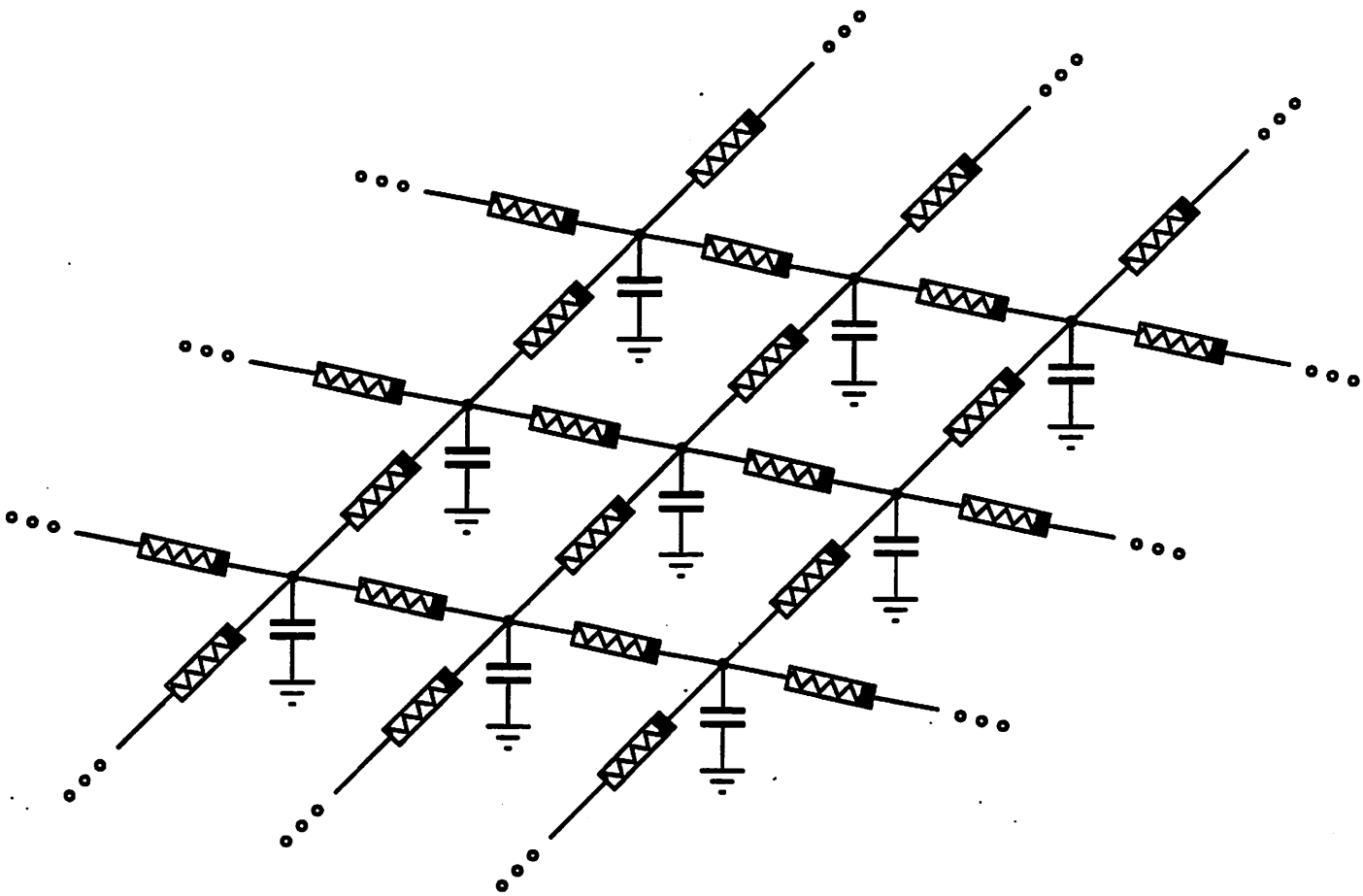

FIG 1

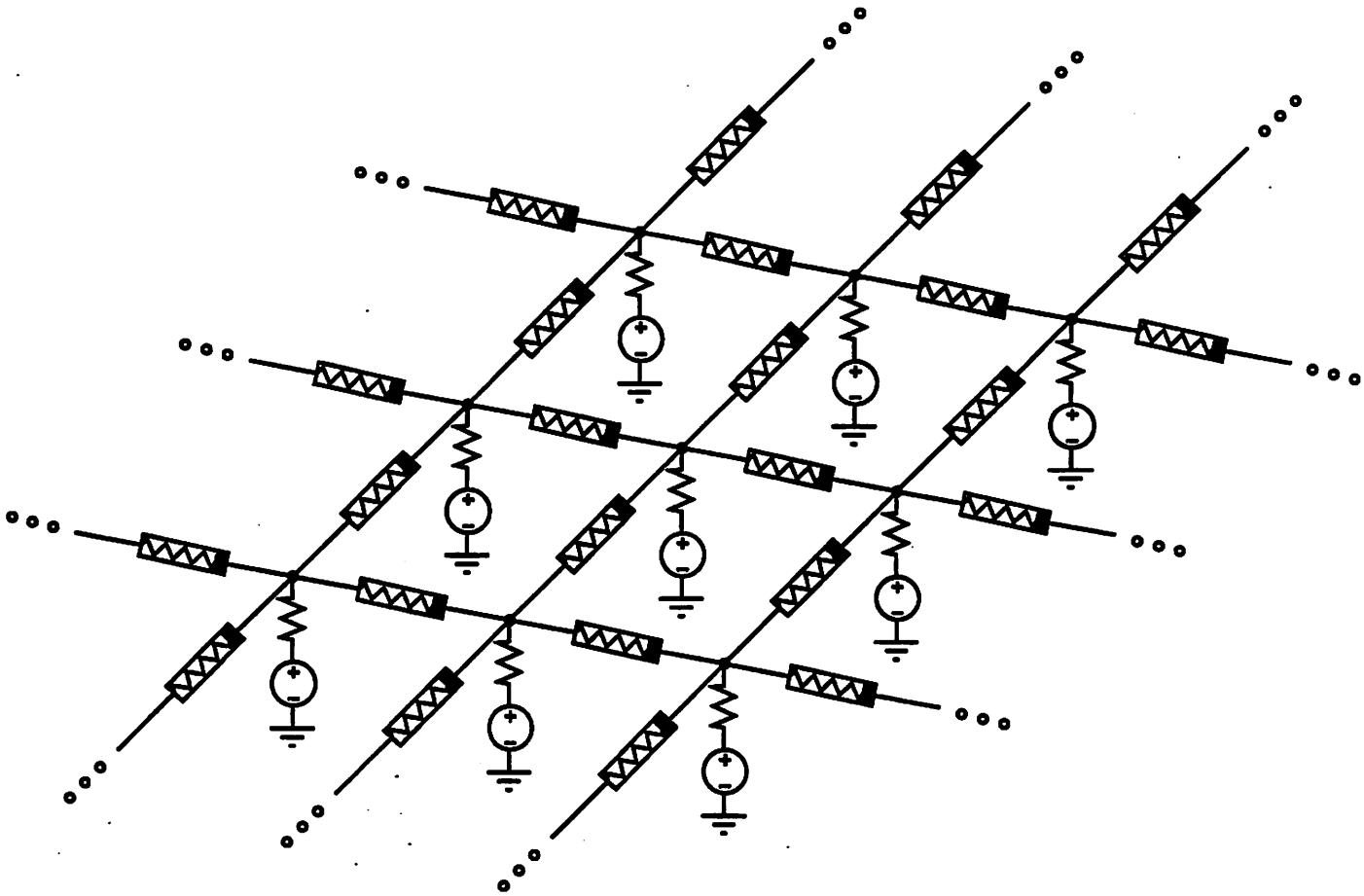

FIG 2

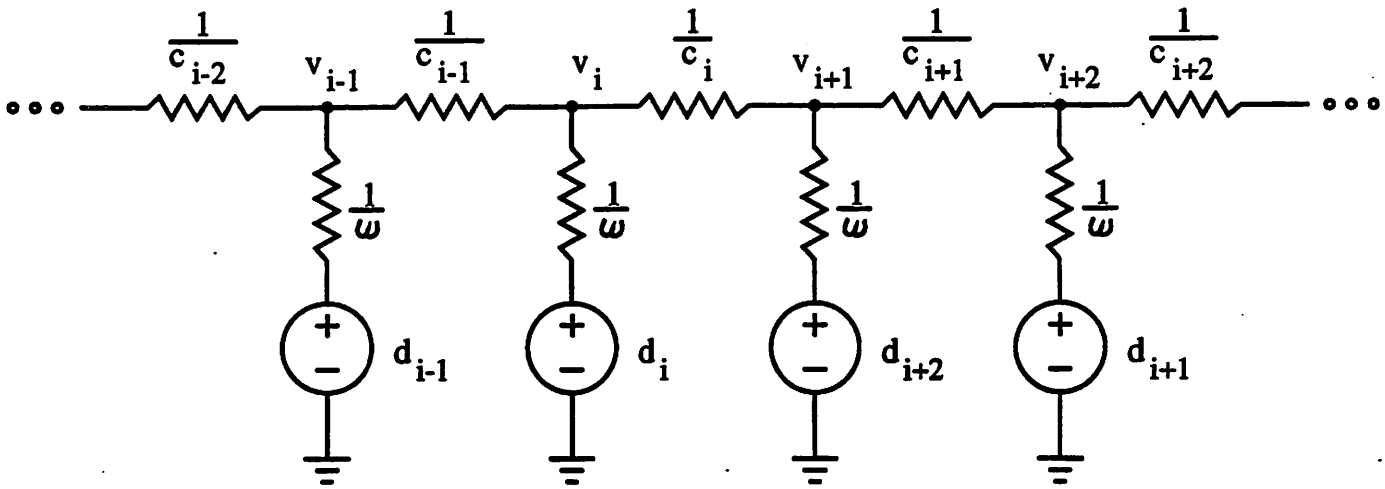

Fig 3

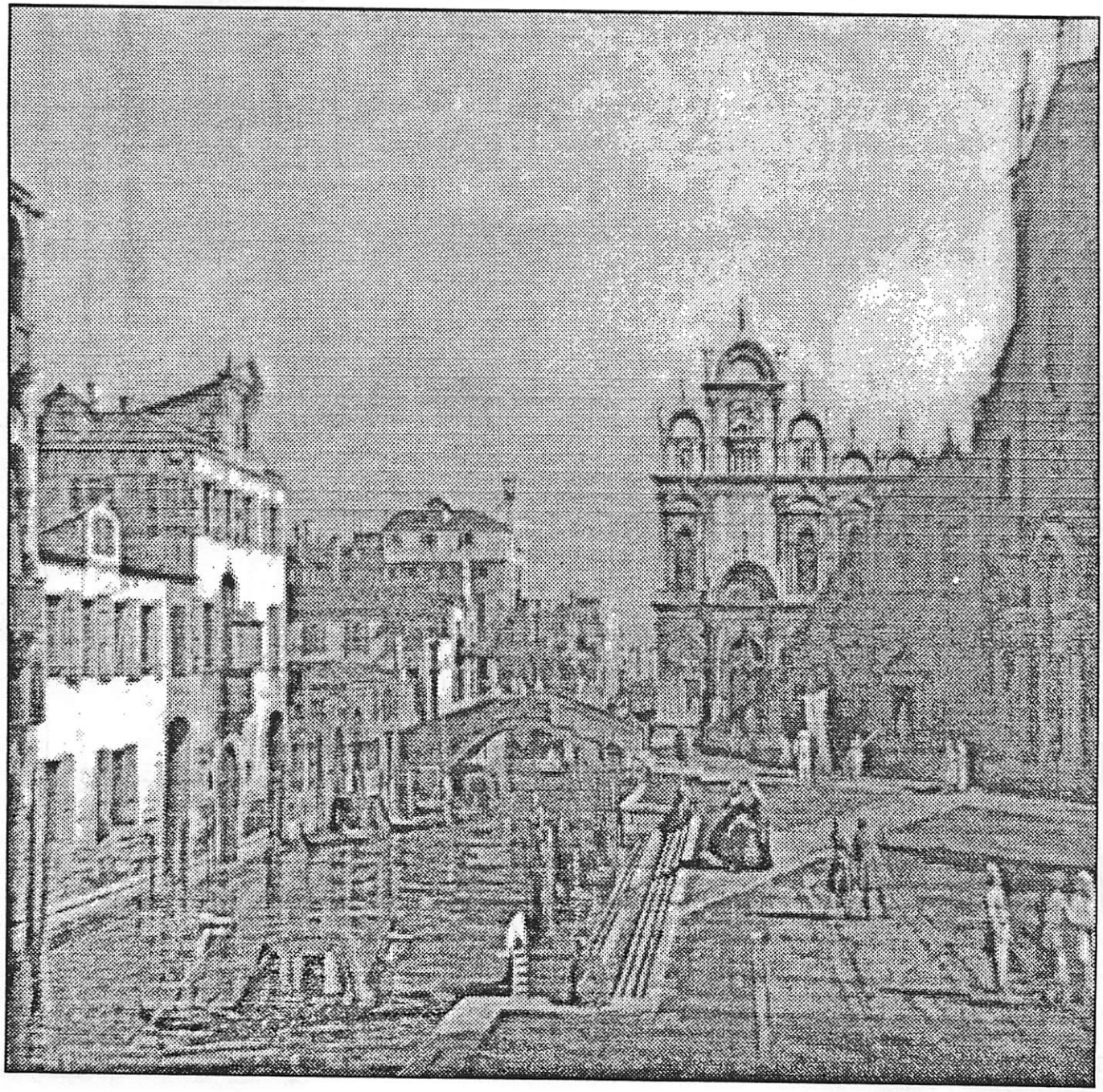

Fig 4a

Fig 4b

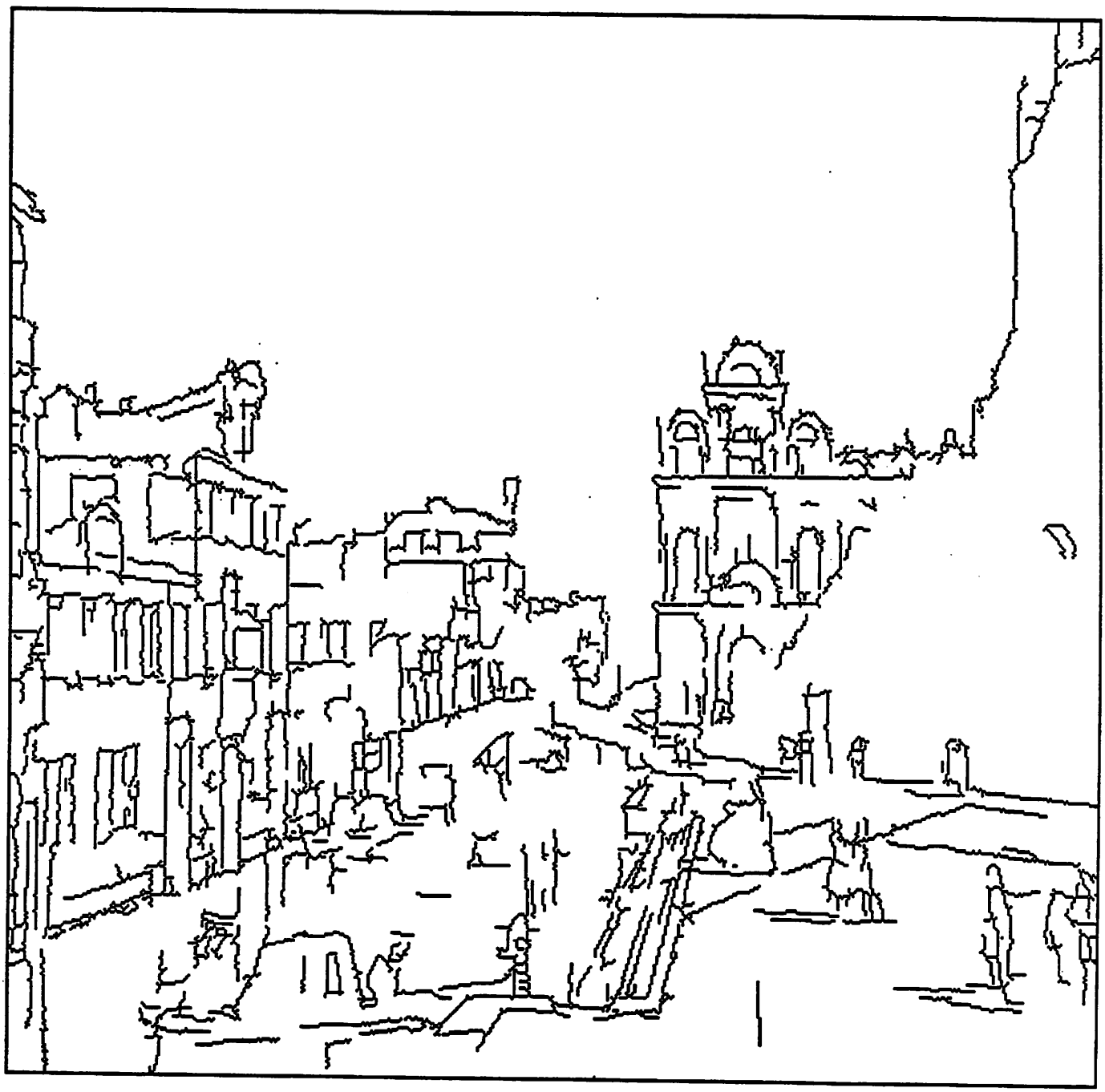

Fig 4c

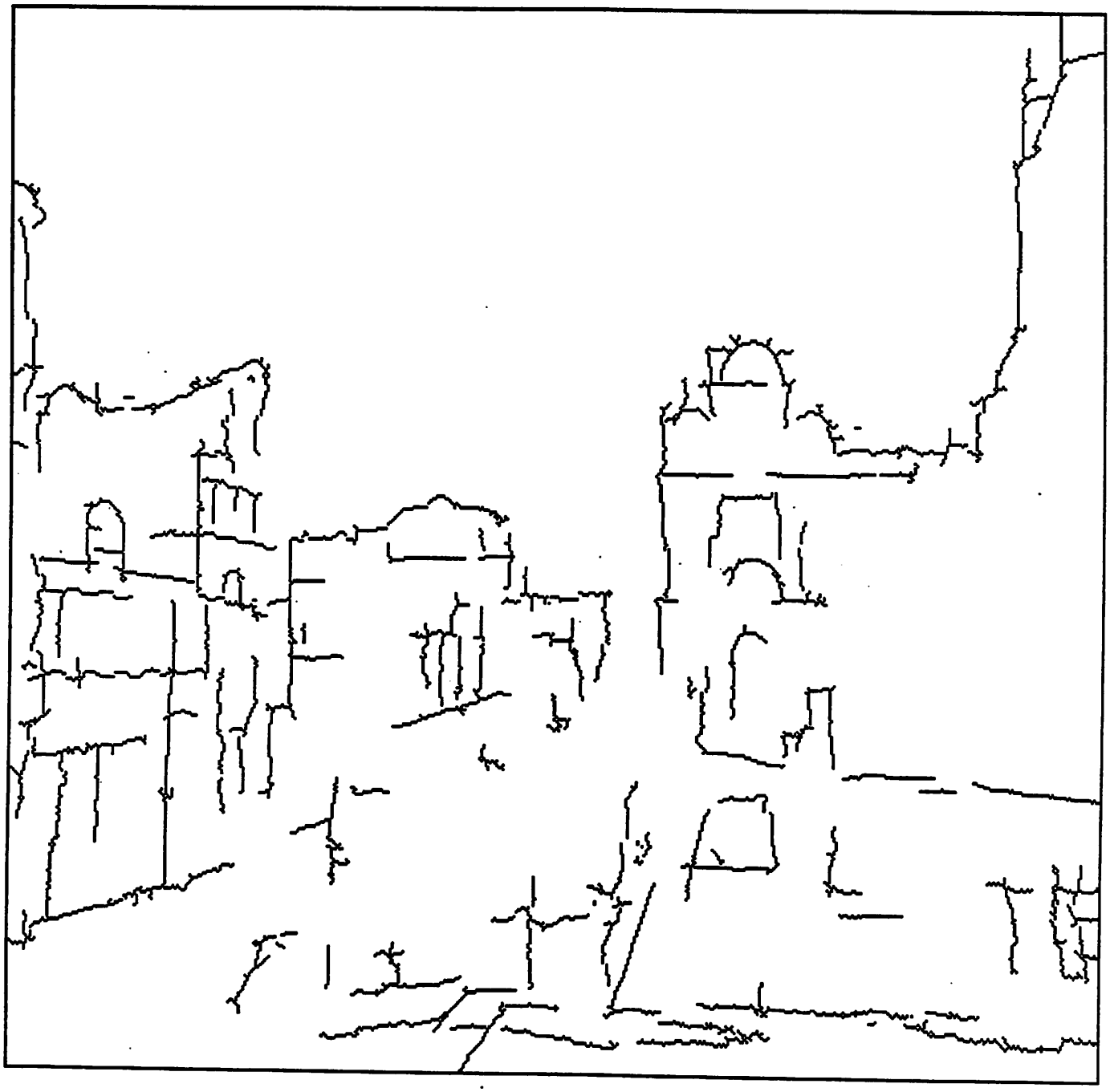

Fig 4d

Fig 4e

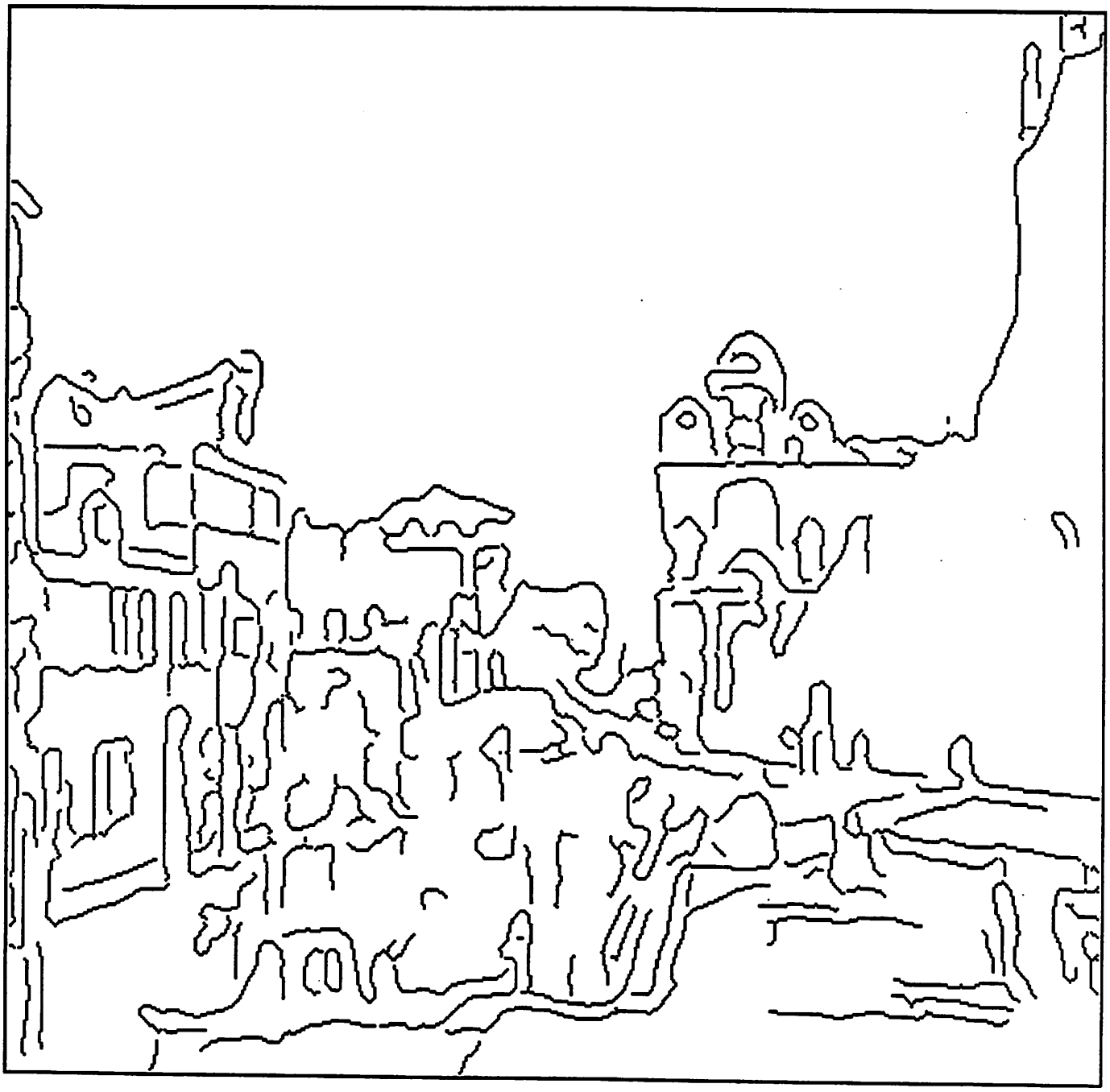

Fig 4f

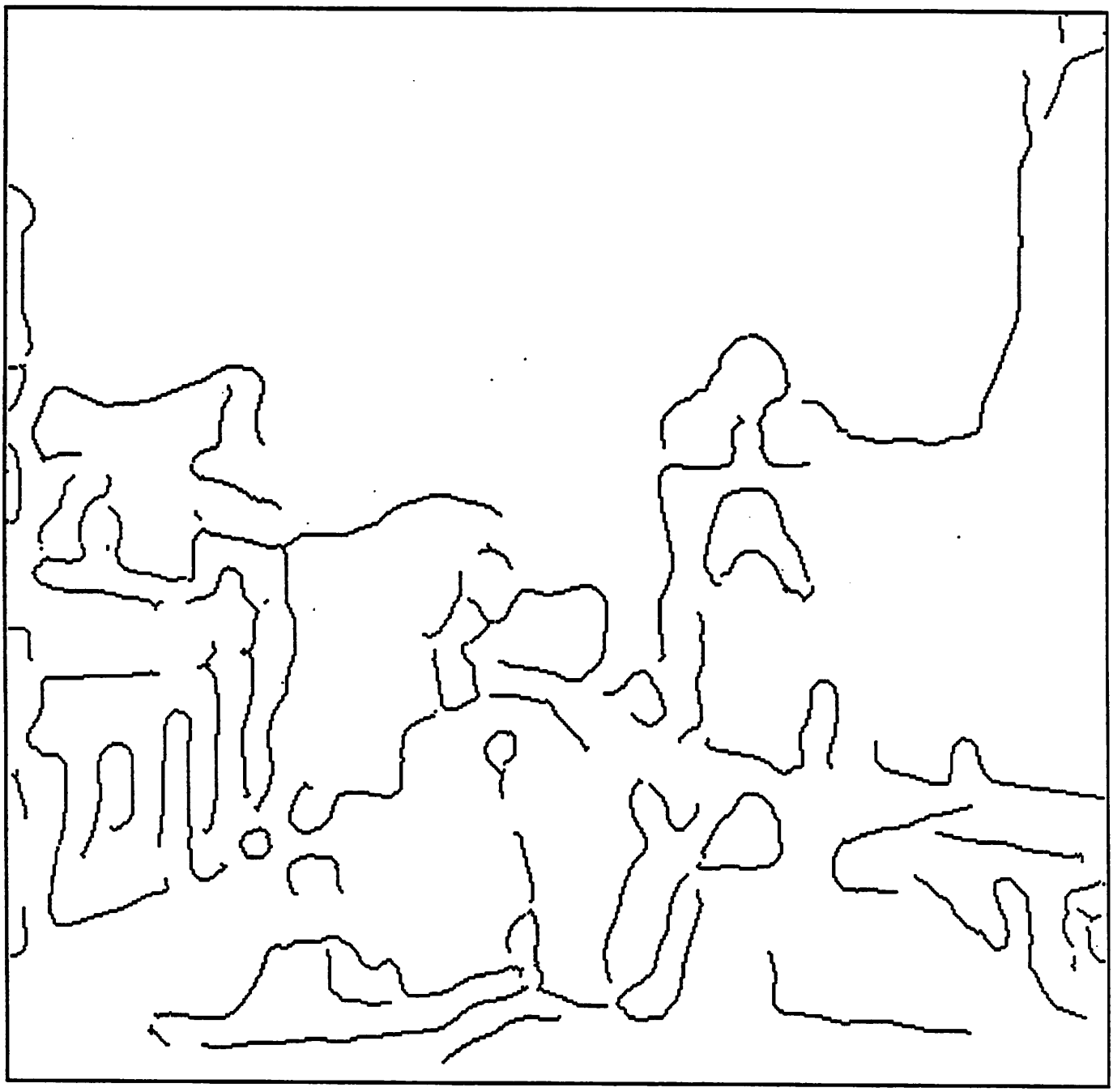

Fig 4g

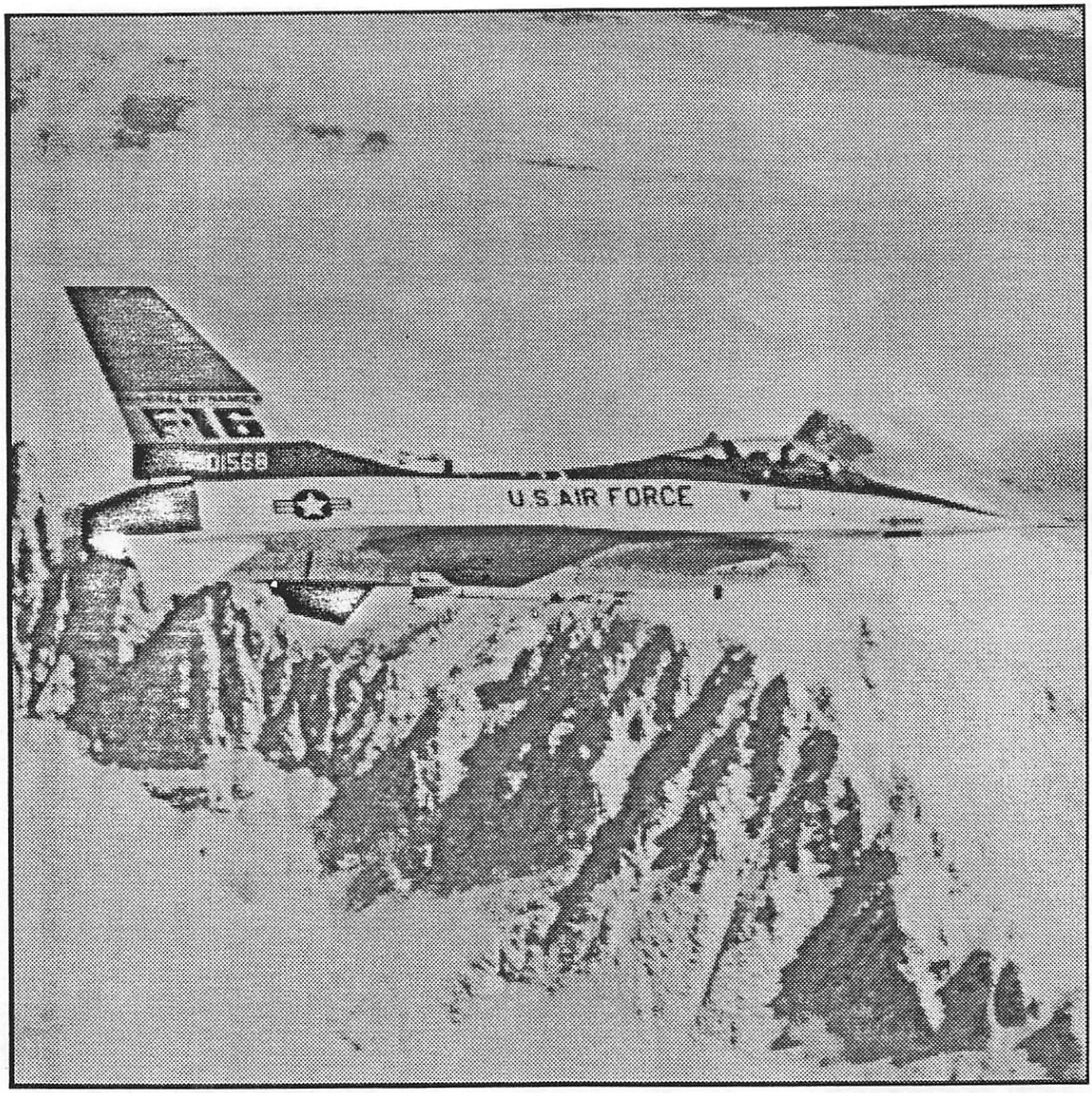

Fig 5a

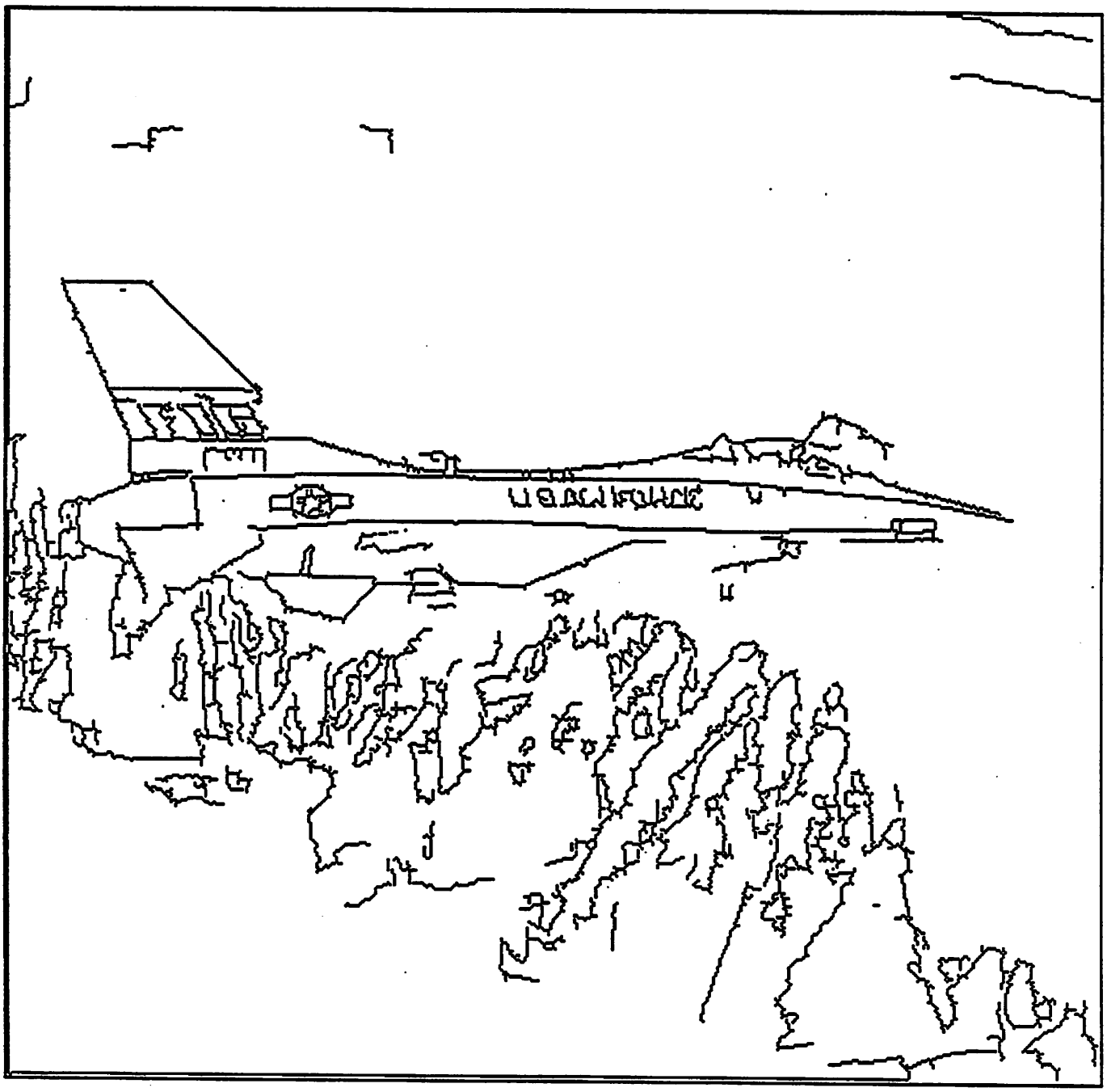

Fig 5b

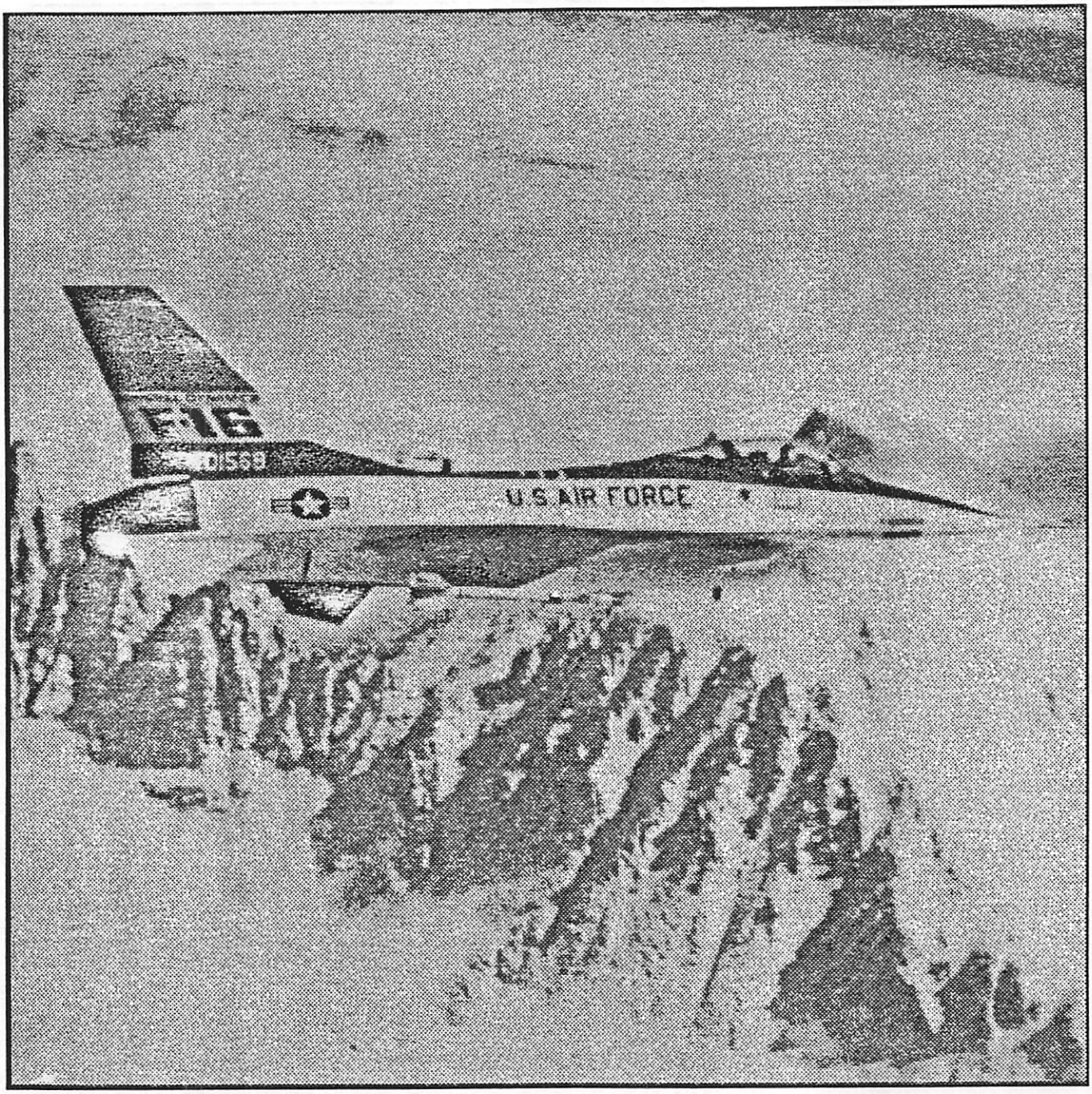

Fig 5c

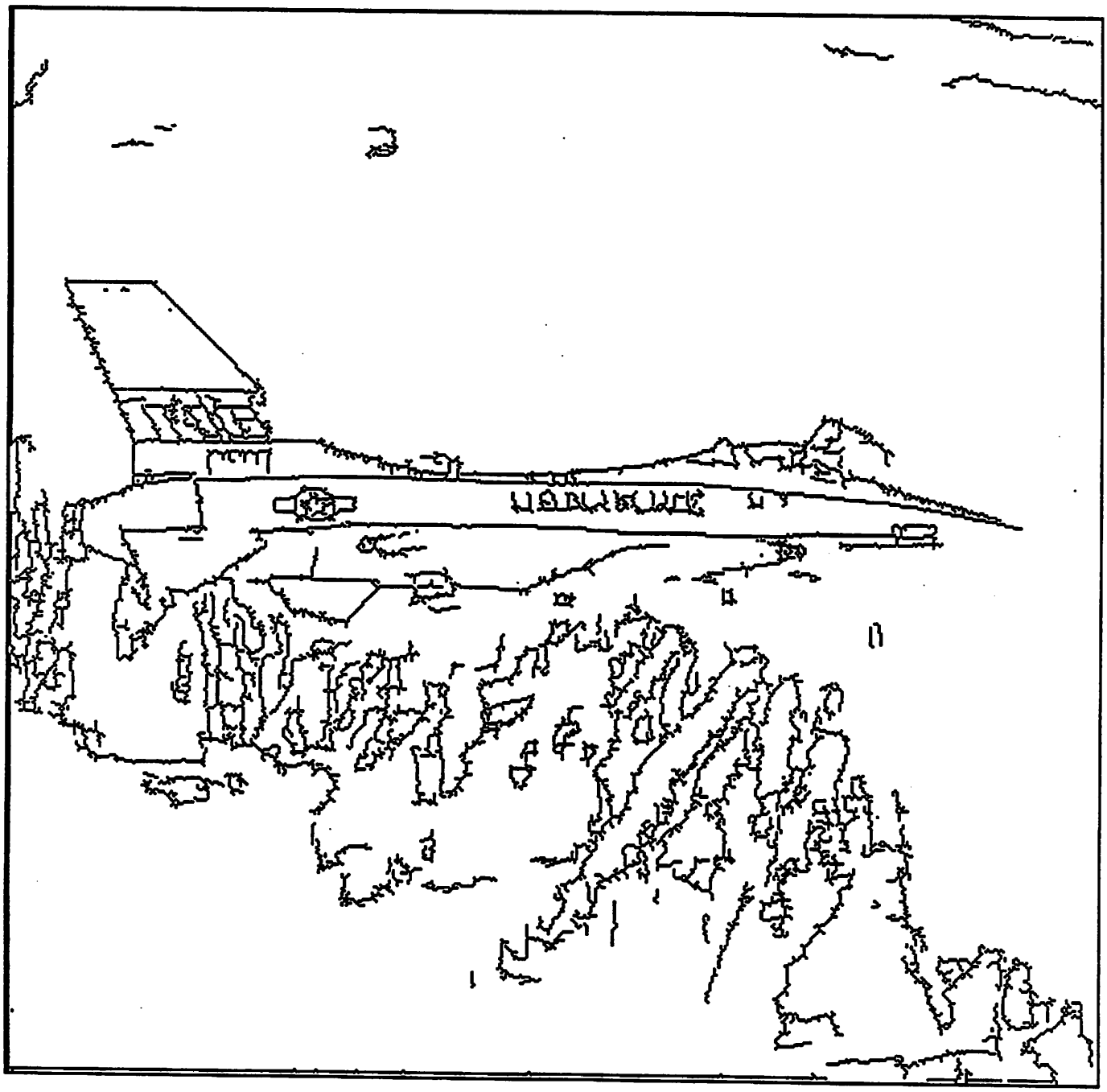

Fig 5d

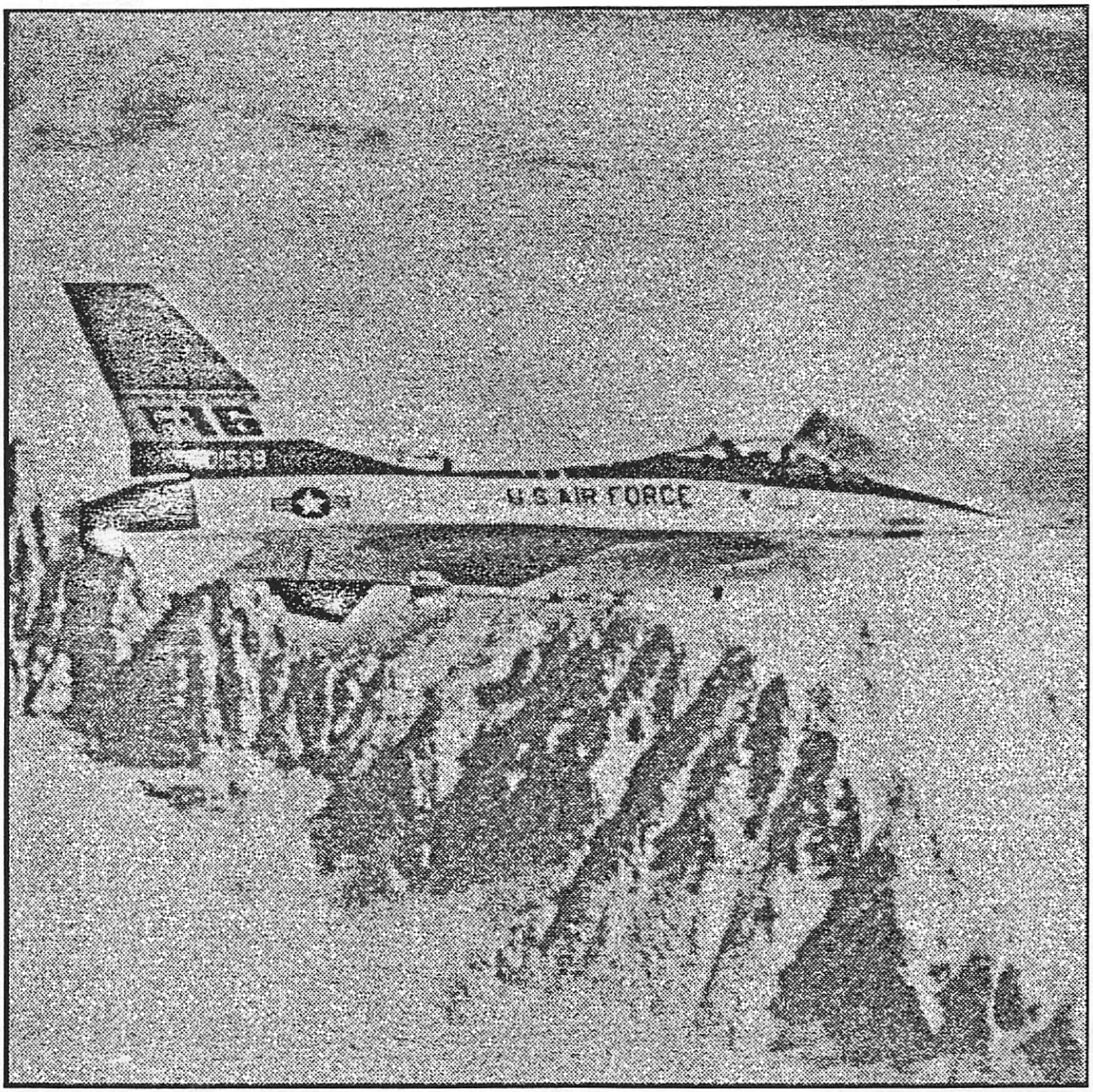

Fig 5e

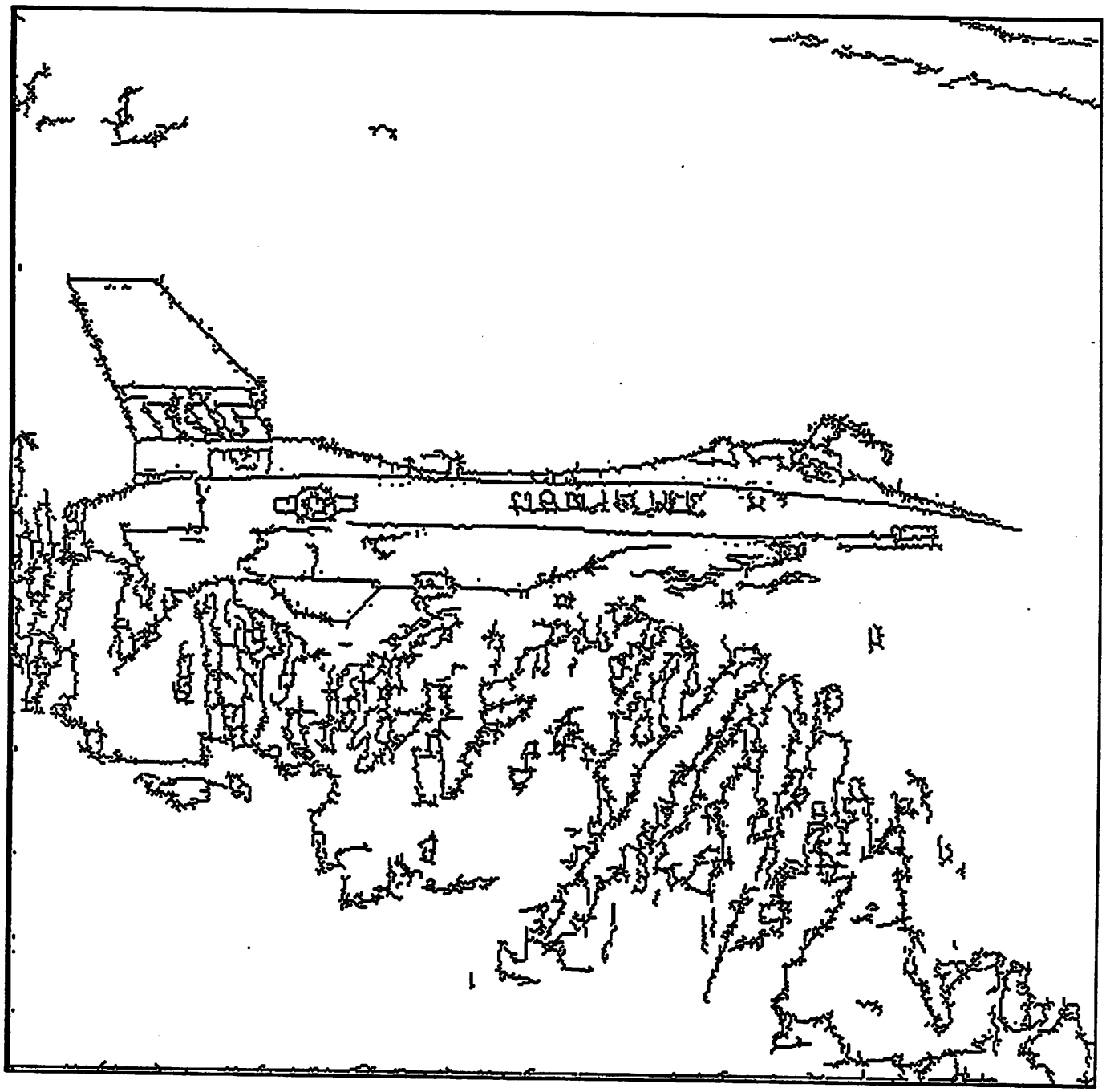

Fig 5f

Fig 6a

Fig 6b

Fig 6c

Fig 6d

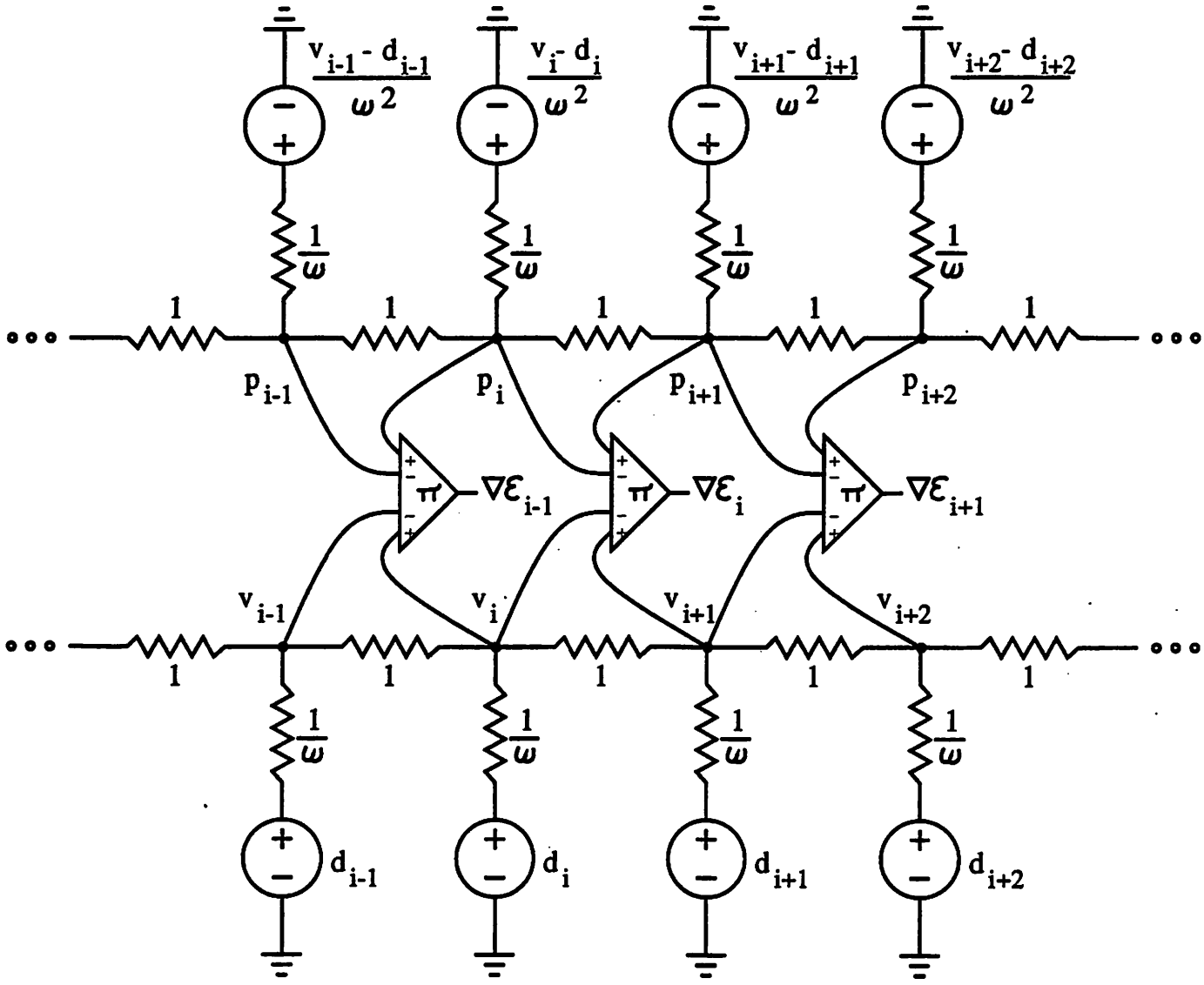
$\frac{v_{i-1} - d_{i-1}}{\omega^2}$
$\frac{v_i - d_i}{\omega^2}$
$\frac{v_{i+1} - d_{i+1}}{\omega^2}$
$\frac{v_{i+2} - d_{i+2}}{\omega^2}$
$\frac{1}{\omega}$
1
p_{i-1}
p_i
p_{i+1}
p_{i+2}
π
$\nabla\mathcal{E}_{i-1}$
$\nabla\mathcal{E}_i$
$\nabla\mathcal{E}_{i+1}$
v_{i-1}
v_i
v_{i+1}
v_{i+2}
d_{i-1}
d_i
d_{i+1}
d_{i+2}

Fig 7