Computer Science Department Technical Report
Machine Perception Laboratory
University of California
Los Angeles, CA 90024-1596

BOUNDARIES AS UNPREDICTABLE DISCONTINUITIES

Richard J. Gleeson
Josef Skrzypek

December 1989
CSD-890073

Boundaries as Unpredictable Discontinuities

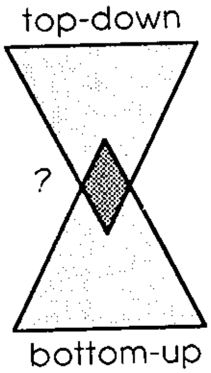
Richard J. Gleeson
Josef Skrzypek

MPL

Machine
Perception
Lab

UCLA
Computer Science
Department

mpl report 0014



Boundaries as unpredictable discontinuities

Richard J. Gleeson, Josef Skrzypek

Machine Perception Laboratory
Computer Science Department
University of California
Los Angeles, CA 90024

1. INTRODUCTION

A large part of the information content of an image is conveyed by the location of pixels that determine the boundaries between different segments. In absence of prior knowledge about objects in the scene, it is impossible to predict where the edges will occur. The gray levels at locations away from boundaries are often highly correlated. They can be estimated from the values of nearby locations using assumptions based on our knowledge of the physical laws governing reflections from surfaces. Boundaries occur at points of essentially infinite variation. At these points corresponding gray levels result from summing energy reflected by two different regions and cannot be easily estimated from the values at nearby pixels. In this sense gray level values at discontinuities are less predictable.

We attempt to approximate the image data with a nearly second order function. The presence of discontinuities makes this approximation inadequate. As a result in the process of deriving the approximating function a statistic is generated which can be used to predict the occurrence of boundaries. This differs from most edge detection algorithms which are based on local differentiation. Our approach avoids the use of derivatives and assumes minimum knowledge about the structure of gray levels at edge locations. It is a bottom up approach and does not use high level expert knowledge to complete edge segments.

Our method is similar to variational techniques used in other low level image processing algorithms[1]. However, since we do not compute extrema, our algorithm can not be classified as a strict variational method. The statistics used for predicting boundaries are collected at variaous iterations and always before the convergence to an extremum. In this sense our algorithm resembles a dynamic system before it reaches an equilibrium state.

The algorithm's computational complexity is justified if most of the useful information within an image is encoded in surface discontinuities. For an nxn image and 2^8 gray levels there are 256^{n*n} different possibilities. These large numbers create an inherent computational complexity in the process of extracting information from images[2] which can only be reduced if there is correlation among pixel values. The algorithm proposed here is designed to exploit this correlation.

2. ALGORITHM

The algorithm can be described as a two level process. The first level process drives a functional of the form

$$Q = \Sigma_i \Sigma_j E_{ij} + w_a V_{ij} + w_m M_{ij} \tag{1}$$

towards a minimum. This minimization problem can be mapped onto a locally connected Hopfield[3] like network which might have some biological validity; however, it is not our intent to model any known biological system. The conversion to a Hopfield net will be described below.

The initial computation is based on the assumption that two dimensional gray level surfaces are nearly second order. A local surface approximation is created at each pixel location. These approximations are repeatedly adjusted so as to minimize the squared error and a following measure of the higher order variation: For each pixel in 3x3 neighborhood we us local surface approximations to compute the gradient and magnitude. The gradient and magnitude at a given location are predicted in two different ways. One set of predictions is based on the approximating surface anchored at the central pixel of the kernel. The second set of predictions generates gradient and magnitude at peripheral points based on local surfaces at those points. Differences in these predictions are used as the measure of higher order variation. This measure does not constrain surface curvature in any way. It is only required that the surface approximation at each location be consistent with neighboring approximations. However, because the algorithm uses second order surfaces and limits the variation in these surfaces from point to point it is equivalent to minimizing higher order variation.

In some sense, our method is more restrictive than minimizing the third derivative. For example, a "roof" edge can be modeled as an intersection of two flat surfaces. The third derivative at such discontinuities is zero. The restrictions imposed by our method are desirable because they maximize the measure of higher order variation at discontinuities. At the same time they do not exclude naturally occurring surface variation. Errors computed in the process of approximating the surface are collected as statistics that indicate a need for higher order terms. These statistics are then used to create an error image which is passed onto a second level process. Here the information is used for edge detection. The second level process will be described in a future paper.

Let p and q be integer values in the interval [-1,1] and let I_{ij} be the measured gray levels at (i,j). At each pixel (i,j) a two dimensional second order surface is described using the coefficients $a_{ij}, b_{ij}, c_{ij}, d_{ij}, e_{ij},$ and f_{ij}, and the equation

$$F(p,q)_{ij} = a_{ij}p^2 + b_{ij}pq + c_{ij}q^2 + d_{ij}p + e_{ij}q + f_{ij}. \quad (2)$$

This description is used to predict the local nature of the surface. For an n x n image there will be n^2 such polynomials involving $6n^2$ coefficients. The values of the coefficients are determined by driving the functional (1) towards a minimum. E_{ij}, V_{ij}, and M_{ij} are defined as follows.

$$E_{ij} = \Sigma_p \Sigma_q \, (F(p,q)_{ij} - I_{i+p,j+q})^2 \, . \quad (3)$$

This measures how well each surface fits the local data. If $w_a = w_m = 0$, minimizing Q would correspond to making second order least squares fits to the 9 gray level values in a 3 X 3 neighborhood of each pixel. Let the gradient function $G(p,q)_{ij}$ be defined by

$$G(p,q)_{ij} = (2a_{ij}p + b_{ij}q + d_{ij} \, , b_{ij}p + 2c_{ij}q + e_{ij}). \quad (4)$$

$G(p,q)_{ij}$ is the gradient at (p,q) relative to (i,j) as predicted by the fit centered at (i,j).

$$V_{ij} = \Sigma_p \Sigma_q \, |G(0,0)_{i+p,j+q} - G(p,q)_{ij}|^2. \quad (5)$$

$$M_{ij} = \Sigma_p \Sigma_q \, (F(0,0)_{i+p,j+q} - F(p,q)_{ij})^2. \quad (6)$$

V_{ij} and M_{ij} penalize variation greater than second order. They do not limit the magnitude of the gradient. Some surfaces such as those resulting from specular reflection contain higher order variation. It is shown in section 3 that both the algorithm and a human observer detect edges in this case. w_a and w_m must be large enough to force final surfaces with small higher order variation but the exact values don't seem critical. Values of $w_a = w_m = 10^5$ are being used. Q is driven toward a minimum by repeatedly solving the six simultaneous equations $\partial Q/\partial a_{ij}=0$, $\partial Q/\partial b_{ij}=0$, $\partial Q/\partial c_{ij}=0$, $\partial Q/\partial d_{ij}=0$, $\partial Q/\partial e_{ij}=0$, $\partial Q/\partial f_{ij}=0$, for all i,j with all other coefficients held fixed. The surface obtained after a few iterations is smooth and has the statistical properties needed for edge detection. Convergence to a minimum value of Q is slow. Convergence is guaranteed since every solution to the equations represents a decrease in Q. As will be seen in section 3 complete convergence is neither necessary nor desired. In order to detect edges, statistics about higher order terms are collected. The statistics obtain significant values after a few iterations and remain stable for several iterations thereafter. These statistics become insignificant indicators of edges as a final minimum value for Q is approached.

If w_a and w_m are sufficiently large and the minimum of Q reached; the surface described would be a least square fit of a single second order polynomial to the entire image. For smaller values of w_a and w_m of the order of 10^5 ; an extremum surface would result in small values of V_{ij} distributed over large areas of the image. This is not desired, because the V_{ij} could not be use to indicate edges in this case. This is a dynamic system and edge information is being extracted from intermediate pseudo stable values computed in the middle of convergence.

The minimization problem can be described as a autonomous system similar to a Hopfield network. This can be shown by mapping the $6n^2$ coefficients { a_{ij}, b_{ij}, c_{ij}, d_{ij}, e_{ij}, f_{ij} } to a new set of variables { v_i }. Let

$$k = 6(i-1) + 6n(j-1) +1$$

where n is the dimension of the image and let

$$v_{k+0} = a_{ij},\ v_{k+1} = b_{ij},\ v_{k+2} = c_{ij},\ v_{k+3} = d_{ij},\ v_{k+4} = e_{ij},\ v_{k+5} = f_{ij}.$$

Then Q can be written as

$$Q = -1/2\Sigma_i\Sigma_j T_{ij}v_iv_j - \Sigma_i L_iv_i\ , \tag{7}$$

where T_{ij} is a symmetric matrix of constants and L_i is a linear combination of I_{ij} both derived algebraically from the original expression for Q. When the terms of the original expression for Q are multiplied out, those involving products of the I_{ij}'s are constant and can be dropped. T_{ij} is a sparse matrix since the coefficients for each pixel appear only in the terms for that pixel and the terms for it's nearest neighbors. This is the form of the energy function minimized by Hopfield. The minimum is found by determining the stable points of a set of coupled differential equations. The main differences between our minimization problem and those considered by Hopfield is that he analyses fully connected networks and problems with solutions at the boundaries of the solution space. The application considered here has a locally connected network, is concerned with intermediate values of the independent variables, and has stable final states that occur at interior locations of the solution space. The coefficients are the independent variables of the autonomous system

$$dv_i/dt = -\Sigma_j T_{ij}v_j - L_i. \tag{8}$$

In section 3, we show the statistics collected from intermediate states of the system. They are determined by the initial conditions and the final stable state.

3. DATA AND ERROR ANALYSIS

To perform data and error analysis, we recorded gray level images of geometrically simple scenes. The scenes contained surfaces of various curvature and examples of edges. Using this data we investigated (1) if a surface fit to the gray level values has a characterisable statistic which is significantly different at boundary locations, (2) if the algorithm convergences in a well behaved manner, and (3) if the curvature of the surface has a large effect on the results.

The statistic must not be unduly affected by the initial conditions and must have properties useful for edge detection. The following statistics were considered. In each case the statistic used was $\mu_{ij}/<\mu_{ij}>$ where $<\mu_{ij}>$ is the average in a neighborhood of i,j.

For statistic (a), $\mu_{ij} = (f_{ij}-I_{ij})^2$.

For statistic (b), $\mu_{ij} = E_{ij}$.

For statistic (c), $\mu_{ij} = |G(0,0)_{i+1,j} - G(1,0)_{ij}|^2 + |G(0,0)_{i-1,j} - G(-1,0)_{ij}|^2$.

For statistic (d), $\mu_{ij} = |F(0,0)_{i+1,j} - F(1,0)_{ij}|^2 + |F(0,0)_{i-1,j} - F(-1,0)_{ij}|^2$.

There are three other directional statistics similar to (c) and (d) oriented at 45 degree intervals. In the examples shown below only (c) and (d) were used because all of the selected images had edges oriented in one direction.

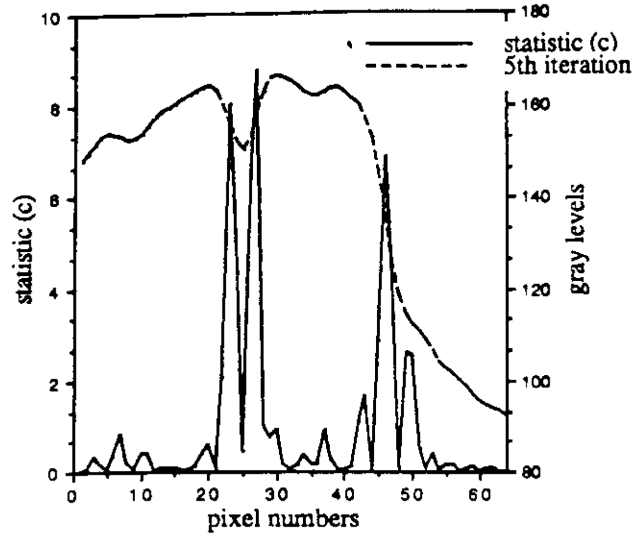


fig 1. The response of statistic (c) to a line and step edge. The dashed line is the approximating surface after 5 iterations. Statistic (c) displays the properties needed by an edge detector.

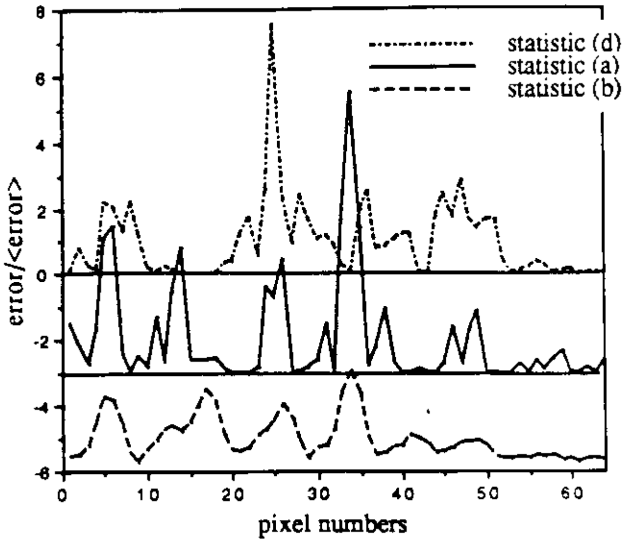


fig 2. The response of statistics (a), (b), and (d) to the same line and step edges shown in fig 1. Statistic (a) and (b) were offset for clarity of presentation. Comparing these to statistic (c) in fig 1, we see that they either have poor signal to noise or don't correctly locate edges.

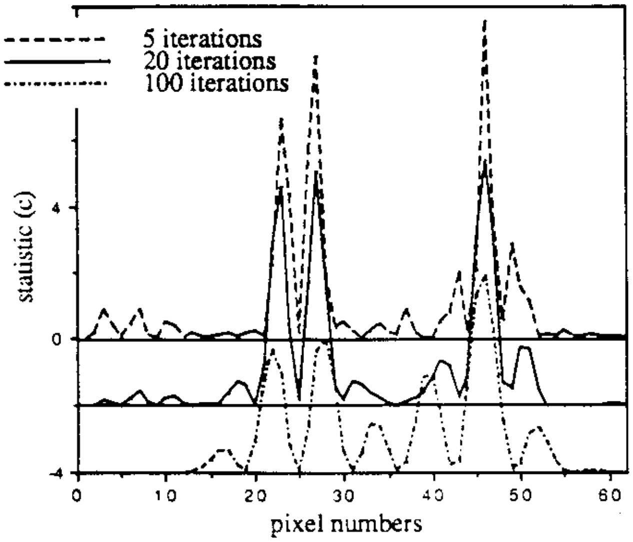


fig 3. Statistic (c) after 5, 20, and 100 iterations. The means ($<\mu_{ij}>$) are 13357, 6350, and 2597 respectively. The values have been offset for clarity of presentation. Initial conditions: $a_{ij} = b_{ij} = c_{ij} = d_{ij} = e_{ij} = 0$ and $f_{ij} = I_{ij}$.

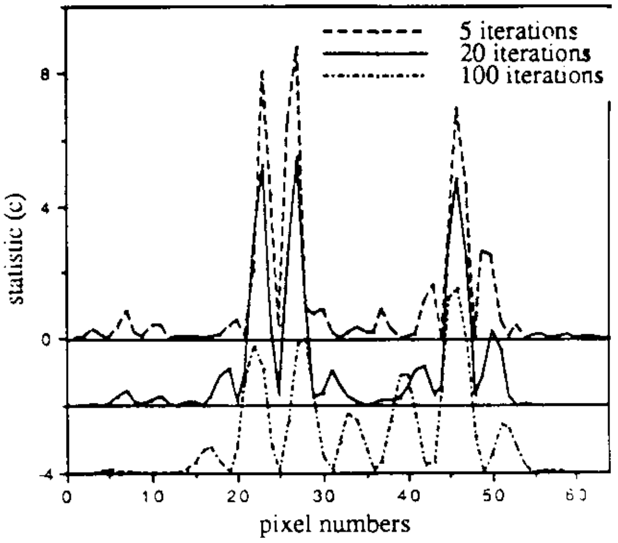


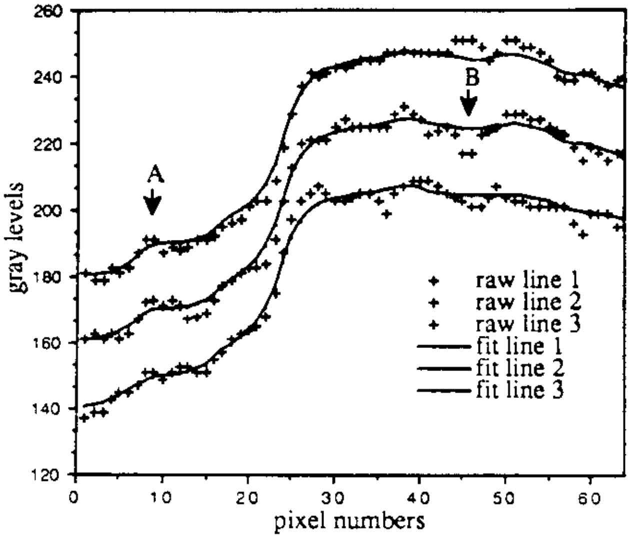
fig 4. Statistic (c) after 5, 20, and 100 iterations. The means ($<\mu_{ij}>$) are 31886, 13039, and 4291 respectively. The values have been offset for clarity of presentation. Initial conditions: coefficients determined by fit to a 5X5 neighborhood center at (i,j).

The results shown in figures 1-9 were obtained by analyzing single scan lines from two dimensional images. The results displayed in figures 1 and 2 were collected after five iterations of the algorithm. Statistics (a), (b), and (d) either don't have good signal to noise characteristics or respond in non-edge areas (fig. 2). Statistic (c) has the properties needed by an edge detector (fig. 1) and is the only statistic considered in the rest of this section.

The initial conditions are important because the statistics are collected as the algorithm converges from the initial to a final state. To observe the effect of different initial conditions, the coefficients were initialized in two ways. In one case the initial coefficients were determined by setting $a_{ij}=0$, $b_{ij}=0$, $c_{ij}=0$, $d_{ij}=0$, $e_{ij}=0$, and $f_{ij}=I_{ij}$. In the other case the initial coefficients were determined by a least squares fit to a 5X5 neighborhood of each pixel. Figures 3 and 4 show that this change in initial conditions didn't have a significant effect on the results. Figure 3 and 4 also illustrate that the statistic serves as a stable edge detector over several iterations.

A noise spike caused by a pixel whose gray level value is significantly different from all of it's neighbors is smoothed out in a couple of iterations. A ripple that has correlated support in one dimension (i.e. has the characteristics of and edge) requires several iterations before being smoothed out. This is illustrated in figures 5 and 6 where three adjacent lines of image data are shown. The small ripple to the left of the step (A) persists after the larger noise variation on the center line to the right of the step (B) has been smoothed.

The reflection from points along a homogeneous surface varies as a function of the curvature of the surface, the variation in the light source, and the variation of the reflection properties of the surface versus incident and reflected angle. Consequently, the resulting gray level surfaces are only nearly second order. The need for higher order terms adds to the gradient error term being used for detection. The data displayed in figures 7 and 8 indicates that this additional error will not reduce the edge detection ability of the statistic by any significant amount.



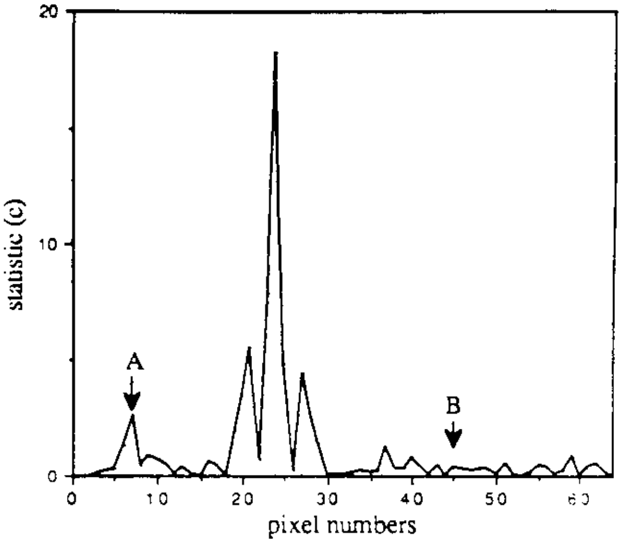


fig 5. The uncorrelated noise to the right of the step (B) is smoothed over after 5 iterations. The lower amplitude correlated noise (low amplitude edge) to the left of the step (A) takes longer to smooth out. (see fig 6.)

fig 6. Small peak to the left of the step (A) corresponds to correlated noise. It would be significant if step was not present.

A specular surface is an example of a more dramatic departure from the second order assumptions. The algorithm still has approximately the same response when applied to these surfaces and is able to locate the edges. For the smooth specular surface, it took ten times as many iterations for the peaks to develop (fig. 9). The peaks in the statistic localize the edges in approximately the same position as the human would perceive them.

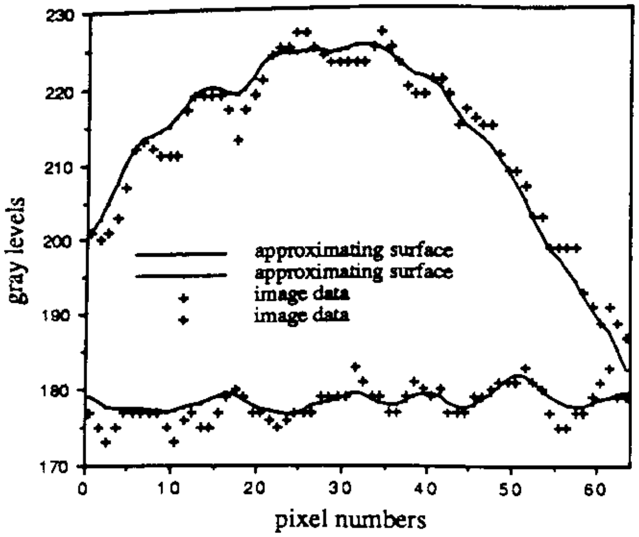


fig 7. Surface approximations for curved and flat gray level surfaces.

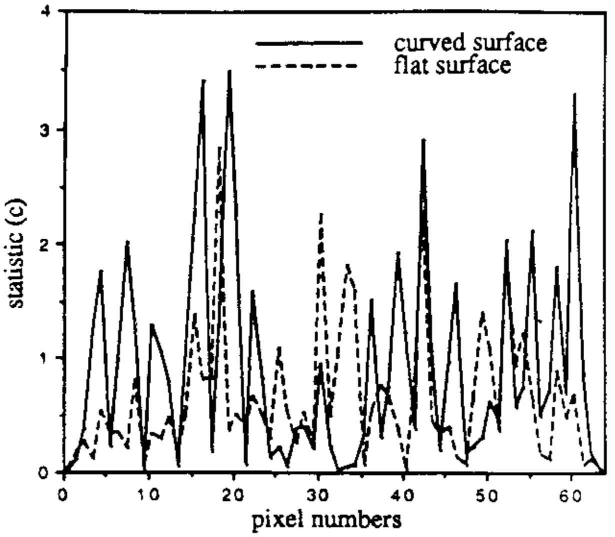


fig 8. The need for higher order terms as measured by statistic (c) for the two edge free surfaces shown in fig 7. The peaks are less than 4 times the surrounding noise. $<\mu_{ij}>$ equals 3893 and 2989 respectively for the curved and flat surfaces.

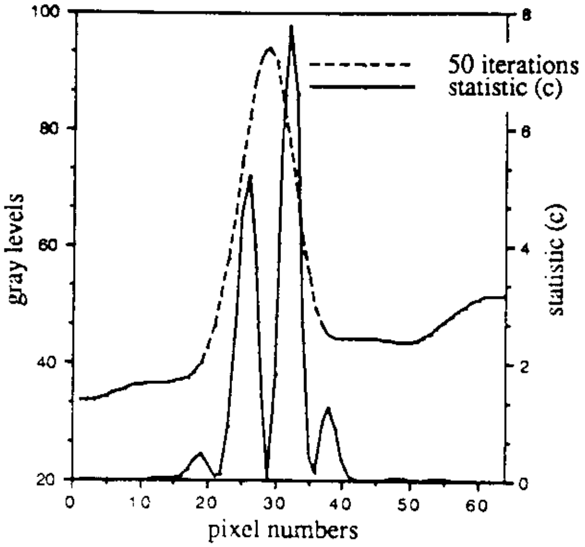


fig 9. Statistic (c) and surface fit for a specular surface. The left hand peak is one or two pixel positions to the right of where a human would locate the edge. The right peak is located where a human observer would perceive the edge. $<\mu_{ij}> = 10312$.

4. CONCLUSIONS

We introduce an algorithm for edge detection which does not rely on differentiating image data but predicts the location of discontinuities based on the reflection properties of the continuous surfaces adjacent to the boundaries. The simple assumption of nearly second order appears sufficient to embody these reflection properties. The statistic generated by the algorithm has a large signal to noise ratio and very sharp peaks. These are the signal properties needed for detection and localization of edges. Some aspects of the algorithm need further work. One problem we don't understand is the irregular accumulation of errors at discontinuities during convergence of the algorithm. It is possible that simplifications in the mathematical formulation and an investigation of the trajectories determined by the differential equations will lead to a better understanding. The parameter space (w_a, w_m) needs to be investigated to determine the relative affect of the terms in Q. In addition the results present are based on only nearest neighbor interactions. It is possible that the use of more distant interactions will affect the convergence rate.

The algorithm for simulating the second level detection process has not been finalized. It will be based on analysis of the data obtained from the first level process. It will use local statistics to detect candidate edge locations and it will incorporate the fact that most edge pixels lie along smooth curves.

5. ACKNOWLEDGEMENTS

We would like to thank David Gungner and Edmond Mesrobian from the UCLA Machine Perception Lab. for their unselfish help in this project. Support for the UCLA Machine Perception Lab. environment is provided in part by ARCO-UCLA Grant #1, MICRO-TI grant #541122-19907, ONR grant #Nooo14-86-K-0395, and ARO grant DAAL03-88-K-0052. Richard Gleeson is supported by an NWC, China Lake fellowship.

6. REFERENCES

1. Horn, B.K.P. , Schunck, B.G., Determining optical flow., Artif. Intell. 17, 141-184 (1981).
2. Abu-Mostafa, Yaser S., The complexity of information extraction. IEEE Trans. Inform. Theory, Vol IT-32, No. 4, 513-525 July 1986
3. Hopfield, J.J., Neurons with graded response have collective computational properties like those of two-state neurons., Proc. Natl. Acad. Sci. USA 81, 3088-3092 1984.

address correspondence to:
Josef Skrzypek, 3532D Boelter Hall, UCLA, Los Angeles, CA 90024 electronic mail: (arpanet) skrzypek@cs.ucla.edu
Richard Gleeson, Machine Perception Laboratory, Computer Science Department, University of California, Los Angeles, CA 90024 electronic mail: (arpanet) rjg@cs.ucla.edu