The Semantics of Edges

by

William Brent Lander

Thesis submitted to the Graduate Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science and Applications

Approved:

Dr.George Gorsline, Chairman

Dr. Roger Ehrich
Research Advisor

Dr. Richard A. Arndt

December, 1977
Blacksburg, Virginia

I wish to thank all of those people who have made this thesis possible. This is the report of over a year of trials, and errors. Finally, it is finished.

I would like to thank my committee members for the help given me. They have been of great aid in dealing with the rules and regulations involved with getting an advanced degree.

I would like to thank the people of room 118 for their constant companionship. The long hours and almost endless nights of work were made lighter by their assistance. I want to especially thank for his help with the HP 2100 computer.

I would like to thank my friends for putting up with all of the whining that I have uttered in the black moments of my dispair. Their words of wisdom were a balm to my pains.

Most of the typing in this thesis was done originially by . I would like to thank her for the use of her dancing fingers which often worked long into the night.

I would like to thank my parents for always being there. Their devotion through all the months of writing and re-writing has been a sign of their true love of me.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

CHAPTER 1

Introduction

The concepts involved in the totality of vision seem simple before examination; after all, even the lowest of insects seem to have some form of vision. However, closer examination reveals an unexpected complexity in this daily experience.

From one's environment we visually perceive a real set of objects around us; tables, chairs, cups, saucers, and so on. Light from some source falls on this physical scene and is reflected. In all known natural vision systems, this light is then focused by a set of lenses so that a two-dimensional representation of the real world falls on some light sensitive material in the eye. This representation is called an image. While this portion of the total phenomenon can be explained rather straightforwardly by physics, the next step is yet to be understood. By some mechanism, from this pattern of light and dark, along with variations in color, is constructed a coherent grouping, called a segmentation, in which objects, such as books, coffee cups, pens and paper, emerge. This mechanism is called perception. One of the goals of machine vision is the understanding of this recognition process and the algorthmic duplication of this process.

For machine studies, an image is represented by a real-valued function $g(x,y)$, called a picture function, which describes the reflected intensity from a scene at point (x,y). The value of the picture function is called the brightness or gray-level. As picture functions seldom have any analytic form, a real-world image is sampled mechanically and represented in the computer as a 2-dimensional matrix of points. For the remainder of this thesis, it will be assumed that a picture function can be represented by this matrix. The entries in this matrix give a somewhat satisfactory approximation to the picture function.

Often, the function $g(i,j)$ will be called the digital picture function, where each (i,j) pair is associated with one picture element or pixel. The digital picture function is formed by sampling the original image at predetermined regular intervals. For the purposes of this thesis, the sampling is done in a rectilinear manner. Any other sampling method could be just as useful.

Image analysis involves determining a description of the essential, usually simpler, perceptual features in the picture function. A single picture function may consist of thousands of picture elements, each of which convey some minutia of information about the image. What is desired is a algorithm leading from the picture function to a description of the image, which will convey the original essentials of the picture. For example, in figure 1, instead of the picture as perceived, we would hope for some simpler description in terms of figure and background. One possible step would be to describe the original picture function in terms of a line drawing (Fig. 2).

A line drawing, such as described here, preserves the original spatial relationships among the objects, as well as, the shape of the important regions of the original picture. A line drawing can reduce the computational complexity of future steps in the image analysis. Of course, a reduced picture such as this will contain less information than the original. There is no reason for believing that all of the information discarded is useless. The fact that we can recognize a line drawing as a representation of the original image implies that information lost in line drawing reductions is generally of lesser importance. We are looking for a new picture

FIGURE 1

A Typical Real-World Scene

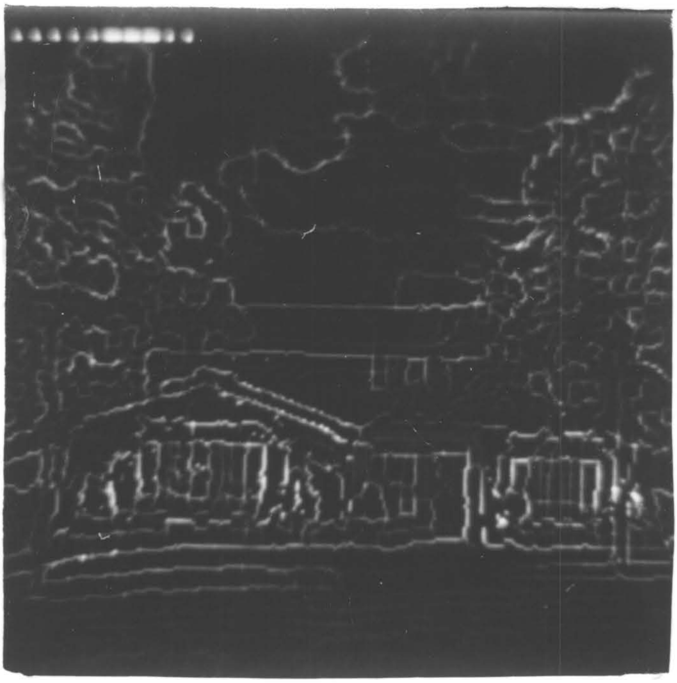


FIGURE 2

A Line Drawing

function G(i,j), which will delineate the various regions of the original picture function. With this G(i,j), like properties and inter-regional relationships may be deduced (Waltz,1970).

Line drawings of pictures are used to facilitate the separation of regions of differing intensity. This could be the first step in determining the relationships between large numbers of pixels. Knowing these relationships it is hoped the amount of information explicitly stored could be reduced and the image simplified (Waltz,1975).

The algorithms for deriving line drawings from images have generally used some form of gradient edge detector such as those due to Roberts, Sobel, and Kirsch. A great number of gradient detectors have been described over the past few years; as would be suggested by the abudance of such edge detectors, none of the reported detectors produces a consistent, good quality line drawing of a natural scene. Examples of the use of several of these edge detectors are shown in several pictures in Appendix C. The problem with the previously reported edge detectors is two-fold. First, they each seem to be overly sensitive to particular classes of intensity transitions. Second, these same detectors are insensitive to other classes of intensity transitions which would, to the eye, appear just as importaint.

Most gradient edge detectors work in the following way. A value representative of the intensity differences in the local neighborhood of a hypothesized edge point is obtained. This representative value is obtained by computing a weighted sum of picture elemnts over some area, called the filter width. The two dimensional derivative is then either estimated directly or computed by combining information about one-dimensional derivatives taken in a number of directions. The new value for that pixel in the gradient image is set at the value of the two-dimensional derivative.

As mentioned above, current edge gradient detectors have certain characteristics and properties. Among these properties are:

1) The broader the filter width (larger region sampled for the gradient aproximation), the broader will be the response of the detector in the filter output image, f(i,j). This smoothing is proportional to the filter width.

2) The broader the image gradient (the longer the period it takes to reach some local maximum) the larger the filter mask must be to optimally detect the gradient.

3) Broader filters are more noise immune (less sensitive to the effects of points that might be noise) while small filters show good transient response with poor noise immunity.

4) The maximum value of the filter output is a function of the difference in image intensity between two regions. The relationship varies from filter to filter, and is a function of the size of

the filter as well as the picture function being examined.

5) The determination of a "representative value" for the one-directional differencing requires some degree of interpretation of the pixels which may not be warranted. Some of the pixels contain noise, and to weigh these equally with noise-free pixel values would be a mistake.

Since analytically derived edge detectors have failed to yield consistent results, other methods of edge detection must be examined. An alternate method of analysis is semantic evaluation. In this technique, the initial digital picture function is described in terms of a set of smaller primitives which are then combined to form a more concise, higher level description of the original image.

At the lowest level, the semantic approach to machine vision might involve the description of the real world in terms of primitives contained in the intensity profiles of the digitized images. These descriptions would be used by various algorithms to build higher level descriptions of the image directly from the primitives. The need for performing filtering or determining early interpretations of individual pixels would be obviated. A useful semantic definition language will also be required to handle the problem of noise reduction in the picture functions.

In the following chapters, the details of semantic edge detection are defined. A system of descriptive primitives for pattern matching is covered and the results of a study of the use of these techniques is presented.

CHAPTER 2

The Description Language

Herskovitz and Binford (1970) found that most common intensity changes in images of polyhedral objects were step changes, bumps, and roof shaped profiles. This is a generalization which can only be made in the blocks-world universe. Real world images are not that simple. Not only must the descriptive language be able to describe any simple intensity profile, noise and texture in an image compounds the problem. The language should report not only the fluctuations in intensity, but it should convey enough information to permit the differentiation between signal and noise at a later date.

To define a competent edge finder a set of descriptive primitives is needed to represent changes in intensity, the position of intensity changes, and the rate at which changes occur. The human eye is an excellent edge finder due to the higher order effects, like Mach banding (Duda, 1973). It would seem reasonable to make use of these higher order effects in determining the position of edge points. Because of small local disturbances, it is rather difficult to find regions of maximum intensity gradient which correspond to the noise free image. It is, however, relatively easy to find the critical points of the picture function.

A critical point is a point of local maximum, local minimum, or the boundary of a plateau. A local maximum, or peak point, is a point which has the greatest intensity value in some neighborhood. A local minimum, or valley point, is similar to a local maximum, except that the local minimum has the smallest intensity value in some neighborhood. A plateau is an extended region over which the intensity does not vary, that is, a region which can be approximated by a horizontal line segment in the intensity profile. The boundary points on a plateau are called plateau points.

Detection of these peak points, valley points, and plateau points should be a relatively simple matter, involving only a small neighborhood. For example, in one dimension, the the classification of the pixels at position j with intensity I(j) requires only examination of the pixels at positions j-1, j, and j+1. If $I(j)<I(j-1)$ and $I(j)<I(j+1)$ then j is the position of a valley. If $I(j)>I(j-1)$ and $I(j)>I(j+1)$ then j is the position of a peak. If $I(j)=I(j+1)$ or $I(j)=I(j-1)$ then j is on a a plateau, and if either of the equalities is false, the position j is a plateau point. All other cases are called transition points.

This simple technique for marking points of interest has several important properties. First, no attempt to eliminate noise has been made. Peaks, valleys, and plateaus can all be generated by noise as well as by actual trends in the picture function. Second, the marking of points of interest is unique and invariant in right-left direction along any single scan line. Of course, rotating the axis along which the scan is being done could result in a re-evaluation of the marking, but the marking in this new direction would be independent of previous direction of scan. Third, this technique determines regions of greater interest but does not rule out any of the other points in the original picture function from future interpretation. Indeed, the original function is not changed. All that has been accomplished is the identification of the basic structures of the picture.

As shown in Table 1, there are nine ways to produce ordered pairs of transitions. Transition types 1 and 5 can never occur because of the way in which peak points and valley points are defined. Also note that there are two

TABLE 1

Available Critical Point Transitions

Pair Number	Transition (Leading-Trailing)
1	Peak-Peak
2	Peak-Valley
3	Peak-Plateau
4	Valley-Peak
5	Valley-Valley
6	Valley-Plateau
7	Plateau-Peak
8	Plateau-Valley
9	Plateau-Plateau
	a.Constant Intensity
	b.Varying Intensity

types of behavior associated with transitions of the ninth type. Transitions of type 9a are exhibited in regions of constant intensity where an image has no edges. Transitions of type 9b are found in places were two regions, each of constant intensity, are separated by some intensity variation which contains no critical points. Transitions of type 9b are discussed further in Chapter 6.

CHAPTER 3

Intensity Patterns in Real World Scenes

It has been observed that the most common intensity profiles found in an image composed of polyhedra are plateaus, roof-like ramps, and sharp discontinuities (Herskovitz and Binford, 1970). Each of these patterns is associated with some well-defined properties of the overall image. Regions are associated with the plateaus, and edges are associated with the ramps and discontinuities. This permits simple characterization of what is happening in the scene.

In real world images, analysis is not so simple; the problem is compounded by several phenomena. The first problem is the irregularity of the shapes found in the real world. This irregularity of shape leads to irregularity in intensity profile variations. These irregularities are due to (1) the play of light and shadow, (2) the variations in optical texture of objects, and (3) small holes in objects allowing portions of the background to be seen through them. The second problem in real world analysis is noise. Small errors in the intensity measurements can turn a plateau region into a murky set of peaks and valleys, not recognizable as the original plateau. The presence of noise in an image need not have anything to do with irregularities in the sampling equipment, but in the sampling theory itself. Any time a rigid grid is imposed as the guide for measurements, there is some interference between the sampling pattern and the sampled item. A third problem is diffuse lighting. Most blocks world images use a single source of illumination and hence, avoid this problem.

Three types of critical points have so far been defined; peak, valley, and plateau. Additionally, possible pairing of these points has been discussed. In this chapter, the discussion centers on transition modes. Table 1 lists the types of transitions which can occur. and in this section, we will examine the regions delineated by these points.

There are two modes of intensity transitions: linear,or ramp, and non-linear, or shading. Statistics describing the occurrences of these two transition modes can be found in Appendix A. The ramp-mode transitions include the Binford-Horn(1976) classification, step. Both of the Binford-Horn transitions, step and ramp, are linear, the difference being one of degree. For this reason, they can be treated in a similar manner and are both grouped together in this report.

Because of the critical point marking techniques used in this analysis, all transitions are monotonic. This makes the ramp-mode marking simple. Knowing the endpoints which define a transition, a prediction of the behavior of a ideal linear transition can be made. Knowing the behavior of an ideal linear transition, it is easy to determine whether the transition is in agreement with the ideal, to within some local neighborhood. If the two are in agreement, then the transition is linear.

The other mode of critical point transition is called shading-mode transition. These shading-mode transitions need not be one of the basic forms of intensity variation as noted in the block universe. Figure 3 illustrates how a shading transition can arise. The illumination is greatest on the flat surface of the curved object. As the curved object bends away from the light, the surface gets darker. Coming from the opposite direction, the intensity profile dips sharply as it enters the shadow of the curved object and slowly fades into the darker shadow near the base of the object. This type of real-world behavior is what, in many cases, gives rise to the shading-mode transition and is how the name came to be applied. It is this type of intensity profile which is called a shading-mode transition. These are particularly difficult to mark for edges.

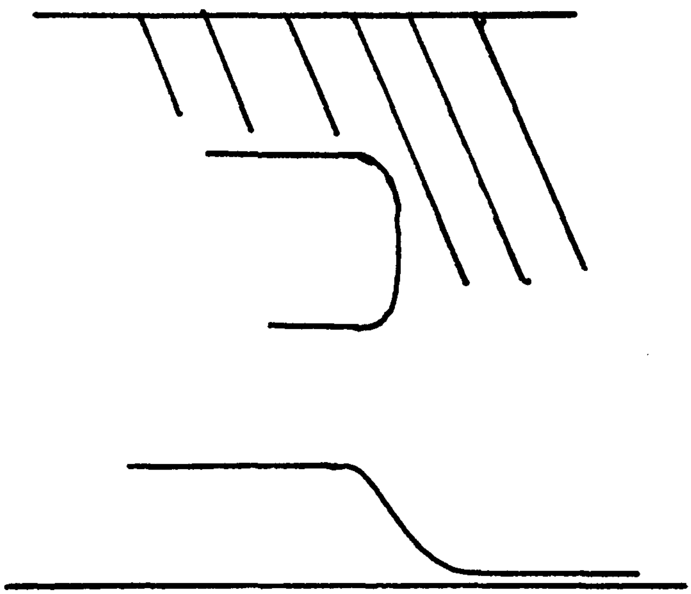


FIGURE 3

Schematic Drawing of A Shading Edge Arising

Shading-mode transitions compose a large portion of the various intensity changes observed in an image. Table 2 shows a breakdown of the number of shading-mode transitions observed in a (typical) image. The transitions were allowed to vary by up to 40 intensity levels from the behavior of the ideal linear edge before the transition was labled shading-mode. In this case, the image is named CAPS. The objects shown are electrical components, called capacitors, shown on a light background. Table 3 shows the percentage of transitions which were found to have shading characteristics. As would be expected, the results derived for peak-valley and valley-peak transitions are in very close agreement. Similarly, those results for the peak-plateau, valley-plateau, plateau-peak, and plateau-valley transitions are all similar. Shading edges are impossible in two remaining cases. Appendix A gives tables of the properties of the transitions found in several typical real-world images. As is demonstrated by those results, the average width of an edge is found to be approximately 2.8 pixels. This means that a gradient-type edge detector passing over an edge would divide the total variation in intensity by this number of pixels and spread the intensity over an equal number of pixels. This gives rise to a band, as opposed to a line, on the resulting picture function.

TABLE 2

Count of Observed Shading Edges

In A Typical Scene

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading	Valley	--	2582	192
Inflection	Peak	2672	--	254
Point	Plateau	241	206	41

This edge distribution accounts for a large portion of the poor behavior of gradient edge detectors. Appendix C contains photographs of several real-world images and the resultant images after application of some of the most common gradient edge detectors.

The problems of noise also contribute to poor edge detection in shaded regions. Noise is any fluctuation in the intensity profile which does not represent the actual intensity variations in the image. Noise may be due to poor choice of sampling interval, or electrical problems in the encoding of the image. The noise appears to be uniformly distributed over the entire image, and cannot be ignored.

Noise and texture reduction in the picture function is not easy. Several techniques exist for smoothing out rapid, minor variations in the image. Duda and Hart(1976) mention several methods for noise reduction. The problem is that these techniques may reduce all fluctuations in an image which are below a certain level. Important textural information can be lost and the image detail impaired. The effects of noise and various methods of reducing noise effects can be found in Chapters 5 and 6.

TABLE 3

Percentages of All Edges Found to be Shaded

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading Inflection Point	Valley	--	18.0	1.9
	Peak	2.2	--	13.8
	Plateau	13.1	11.0	9.4

CHAPTER 4

Semantic Edge Detection

As was stated in the introduction, one of the goals of image processing is to to produce a line drawing of a real world scene by studying the pixels which compose the edges of the objects represented in the picture function. The most useful line drawing would delineate regions of similar properties. The job of an edge detector is finding and marking the exact position where regions meet.

The first step in defining edges is determining what the properties of edges are. Edges are associated with gray-level transitions. Evidence points to three modes of simple transition: constant, roof-mode (or ramp), and shading-mode. Examples are shown in figure 24.

Constant transitions occur when the intensity is uniform over some neighborhood. Areas of constant transition are related to regions in images in which objects are uniformly illuminated with fixed color and hue. These transitions are of type 9a as listed in Table 1.

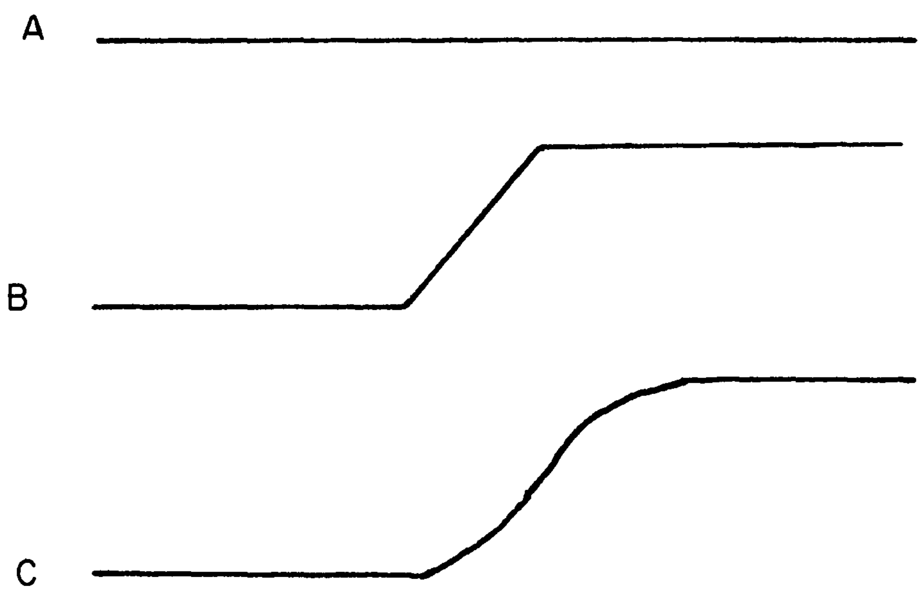


FIGURE 4

Examples of Various Transition Modes in Intensity Profiles

Ramp-mode transitions occur when the intensity of an image changes value quasi-linearly between adjacent critical points. These quasi-linear variations occur in various ways; these include the shadow of a three-dimensional object due to a point light source, the gradual curve of a surface, or an actual color or hue change in the object itself. Since the intensity variation in this mode of transition is quasi-linear, the actual edge point can be computed by a linear interpolation between the critical points (relative maximum and minimum) of the segment once they are known.

Shading-mode transitions show marked variations from linear behavior between critical points. The technique used in finding edges with ramp-mode

transitions is not suitable. Figure 5 shows how a slight variation in the shape of a transition can result in enormous discrepancies in the location of edge points.

It is this interesting property of shading-mode transitions which distinguishes them from ramp-mode transitions; shading transitions are transitions of relativly long duration which experience changes in intensity that can not be modeled linearly Given the span of the proposed edge, it can be determined whether or not the local variations in intensity follow a linear pattern. If a

FIGURE 5

Example of Shading Effects

linear pattern is observed then the transition is ramp-mode, otherwise, the transition is shading-mode.

The actual edge point to mark in a shading-mode transition is difficult to determine. Several marking algorithms are possible; among these are: (1) marking the center of the region of longest constant slope, (2) marking the center of the region of greatest slope, (3) marking the center of the region of greatest product of slope and length, (4) marking the point of greatest slope, (5) marking the point of greatest intensity and (6) marking the point of faintest intensity. Each of these techniques has certain advantages. Experiments were run with these various algorithms, and the results are reported below. It must be remembered that, due to the length of shading edges, many marking algorithms might mark the same point.

For shading-mode transitions, marking the center of longest constant slope resulted in a set of edge points which formed a smooth edge. The markings varied from one edge to the next. This resulted in a shading edge which sometimes delineated the totality of a given object as observed by a person, and sometimes did not. Even more importantly, edges that contained both modes of transition were not smooth.

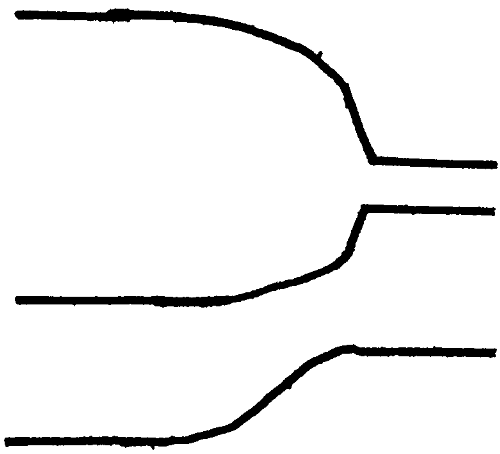


FIGURE 6

Different Types of Shading Edges

Marking the center of the region of greatest slope resulted in extremely wobbly edges. This was due to the occurrence of small textural variations or noise. This marking technique also appeared to mark different edges in different ways and when an edge also contained ramp-mode transitioned, the marked edge was not smooth.

Marking the center of the regions of greatest product of slope and length had the same inadequacies as the marking of the center of the region of greatest slope. This could be predicted from the results shown by the tables in Appendix A. While the length of transition of an edge may be only three or four pixels, the intensity may vary over ten or twenty levels. This allows for the product to be dominated by the slope term. This domination causes the resultant marking to resemble the slope result.

While choosing the point of greatest intensity seems to be a rather ad hoc solution to the problem of marking edge points in transitions, it yielded remarkably good results. The shading edges all followed the edge contour as observed by the human eye, and the marked points produced similar results from edge to edge. The major problem was that the points marked for shading-mode transitions were not alligned well with the marked for linear transitions.

The best results were obtained by the technique of marking the lowest intensity point of each transition as being the edge. This technique produces smooth and consistent edges even were an edge contained both shading-mode and ramp-mode transitions.

So far, the discussion has centered on the question of where to mark the position of an edge. Next, the question of assigning amplitude for each marked edge point must be examined. Gradient-type edge detectors use the value obtained from the derivative approximation to mark the edge intensity, and this technique has yielded inadequate results. The mask size used in determining the gradient is insensitive to the actual transition size and length of any variations.

In semantic edge detection techniques, the presence and location of an edge have been limited by the critical points which define it. Thus the total extent of each transition is known. This allows the amplitude assigned to an edge to be based only on the transition region. The amplitude assigned to the edge should be based on the total scope of intensity variations which are present in the related transition. Since all transitions are monotonic, it is necessary only to find the magnitude of the intensity difference between the transition end points and assign this value to the edge.

CHAPTER 5

Noise Effects

Several times in the preceeding sections, the problems of noise have been mentioned. The presence of noise in image processing is a fact which must be coped with at the most basic level. Noise reduction takes place in the biological systems via homeostatic processes which seem to be more complex and refined than any understood.

An example of the types of degradation due to noise which an image can undergo is shown in the following figures. Figure 7 shows a simple real world image of a set of capacitors on a flat surface. This image has been smoothed by using a symmetric hysteresis technique with the width of the curser set at 40 within a 256 level gray scale. The picture is rather crisp due to the removal of small shading edges by the technique. The image has been logarithmically digitized so that changes in the numbers are proportional to changes in perceptual strength.

In figure 8, the image has had a normally distributed noise signal with mean 0 and variance 1.56 added to it.

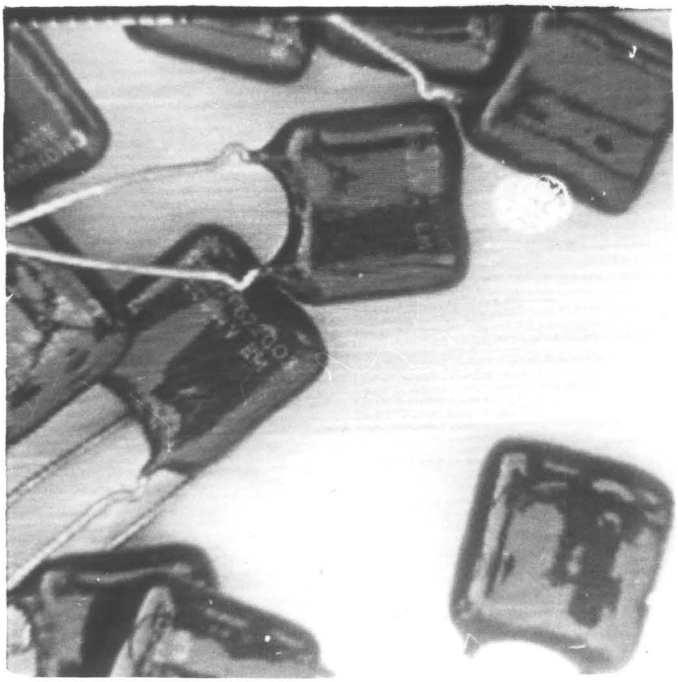


FIGURE 7

Picture CAPS with Hysteresis Smoothing of 40

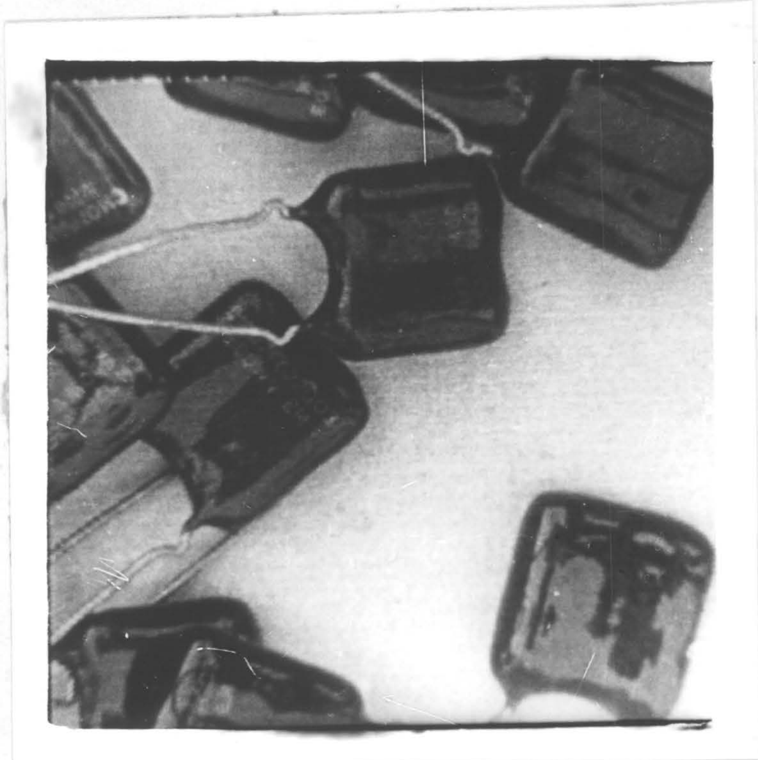


FIGURE 8

Smoothed CAPS with Noise of Variance 1.56 Added

Figure 9 through 13 show the same image as in figure 8, with variance increasing from 1.56 to 51.56. In Figure 13, normally distributed noise with variance from 1.56 to 51.56 was added to the original image. Some degradation of the image is visible in this image. Considering that over 20% of the total intensity range could have been added to a point as noise, the fact that the picture is recognizable at all is suprising.

The decision to use a normally distributed signal was based upon an experimental observation. In several picture functions, regions of minimal texture and low intensity variation showed intensity variations which were close to normally distributed. This distribution was noted in several unrelated pictures which had vastly different texture properties. This finding prompted the choice of a normally distributed noise to be added.

It is interesting to examine the effects of noise upon the number of peaks, valleys, and plateaus observed in a picture function. For the image found in the photographs related to this chapter, the statistics gathered in a row and column evaluation are shown in Tables 4 and 5.

Predictions of the number of intensity peaks, valleys, and plateaus observed when a given intensity distribution is

FIGURE 9

Smoothed CAPS with Noise of Variance of 6.25

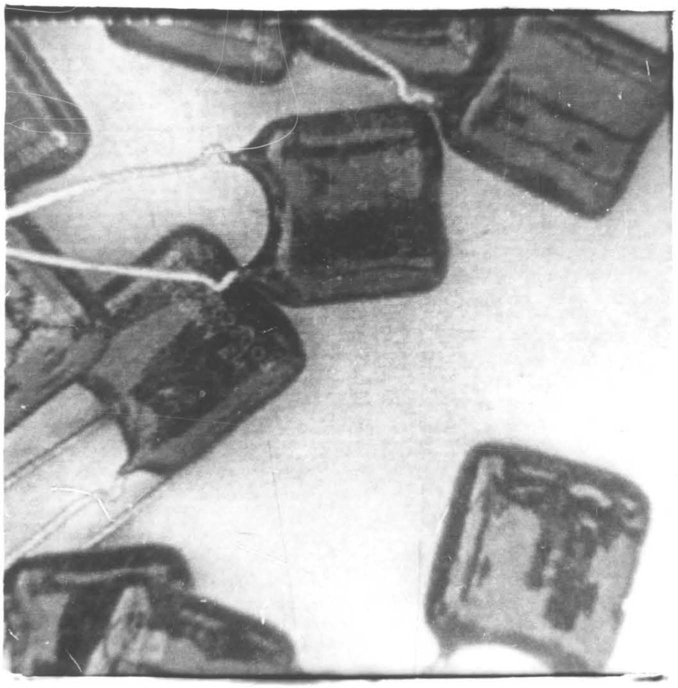


FIGURE 10

Smoothed CAPS with Noise of Variance of 14.0

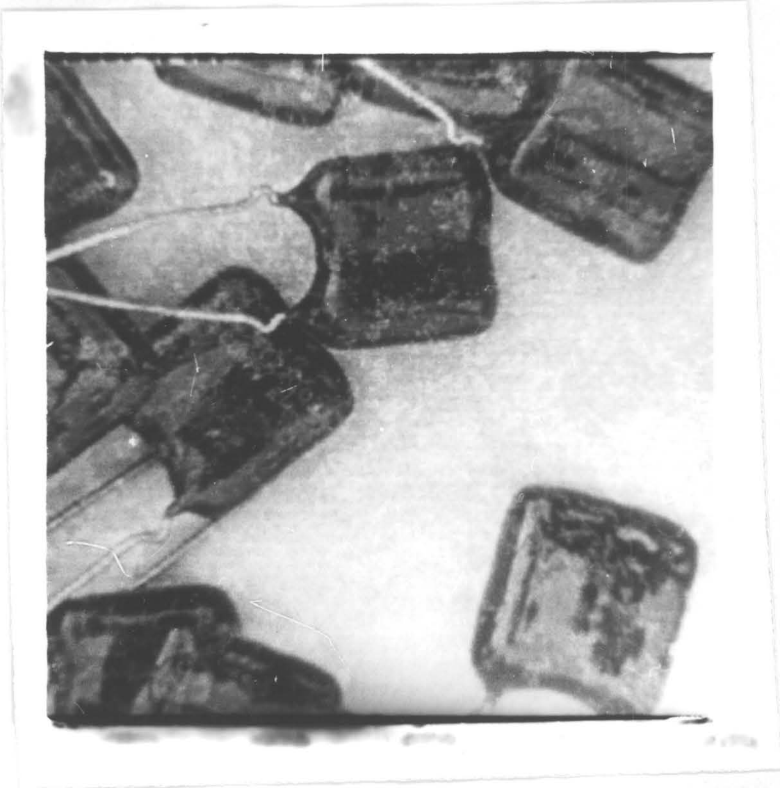


FIGURE 11

Smoothed CAPS with Noise of Variance of 25.0

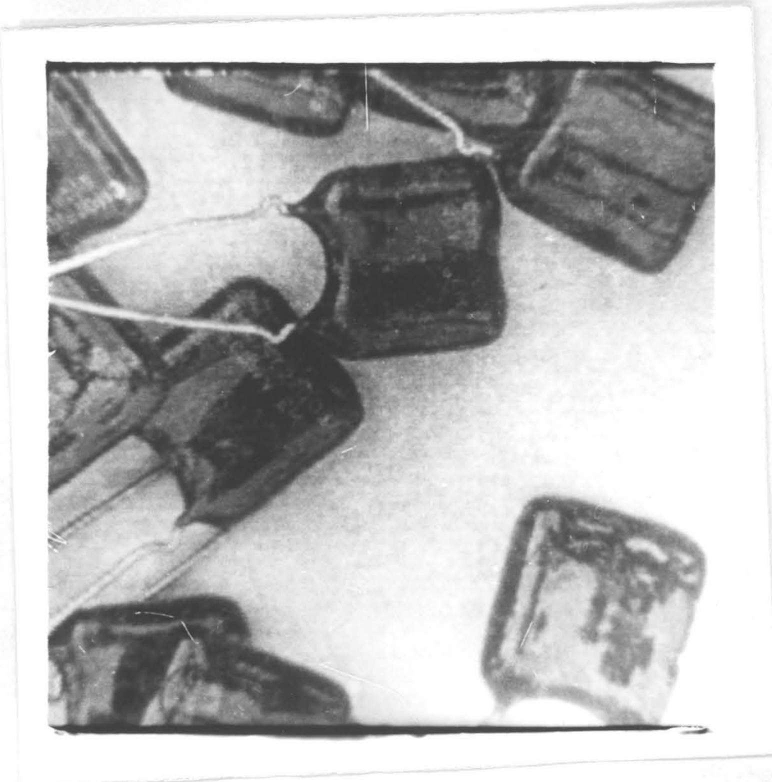


FIGURE 12

Smoothed CAPS with Noise of Variance of 39.0

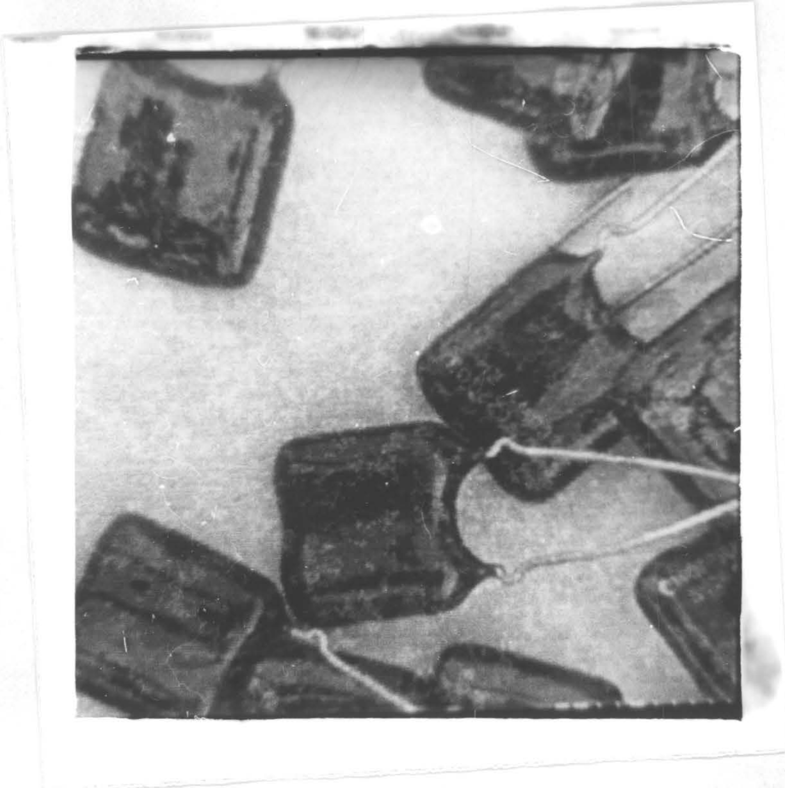


FIGURE 13

Smoothed CAPS with Noise of Variance of 56.23

added to some known signal are beyond the scope of this thesis. However, some observations based on the experimental results can be made.

1) The number of plateau points decreased at a rate of aproximately one half the rate of increase in the number of peaks or valleys.

2) The number of peaks and valleys increased at aproximately the same rate.

3) The number of plateau points remained high for noise with a much higher variance than was expected.

TABLE 4

Noise Effects in Rows: Critical Points vs. Noise

Noise Level	5	10	15	20	25	30
Observed						
Peaks	4603	4544	4680	5677	7783	7830
Valleys	4725	5583	5851	6790	7440	7493
Plateaus	7752	7427	7325	6850	6724	6803

TABLE 5

Noise Effects in Columns: Critical Points vs. Noise

Noise Level	5	10	15	20	25	30
Observed						
Peaks	4578	3606	4540	5429	6318	7174
Valleys	4894	5371	5953	6167	7165	7988
Plateaus	7398	7915	7418	7087	6836	6180

CHAPTER 6

Effects of Various Noise Removal Techniques

Noise reduction introduces an interesting problem into the analysis of intensity fluctuations. While slight variations in the actual intensities are often due to texture in the original scene, these variations can also be due to noise which distracts and provides misinformation about the scene.

One interesting property of the critical points defined by this semantic edge detection is that there are often trends in adjacent peaks, valleys, and plateaus within a profile. A large number of sequential peaks will be monotonically changing in intensity, while the dividing valleys and plateaus are also varying similarly. This effect can be caused either by noise or by texture in the picture function. An example is shown in figure 14. Whereas the initial curve, a, has only two inflection points, with the addition of even a small amount of noise, the number of critical points can be increased (b) until every intervening pixel is a critical point. Techniques such as hysteresis smoothing can be used to reduce the effects of noise however, these techniques are by no means optimal.

Hysteresis smoothing reduces all variations which are below a certain level. Techniques based on hysteresis smoothing are insensitive to any trends in peaks and valleys, or even fluctuations in the picture function which might be important variations describing details in the image.

As an example, consider Figure 14. An intensity profile that originally showed only a small intensity fluctuation is completely smoothed out to a constant value after filtering with a hysteresis gap of only 20. As typical filtering widths are on the order of one-sixth the total intensity range, it is quite easy to see how loss of valuable information might occur.

Additionally, the use of hysteresis smoothing may not reduce the number of critical points at all. Consider the example shown in figure 17. In part A, a small secondary peak is located on the side of a larger peak. The size of the smaller peak is less than the hysteresis gap used in smoothing. The results of this smoothing are shown in B. The critical points which defined the secondary bump are gone. But they have been replaced by a pair of plateau points. The point believed to be noise has not been eliminated, only changed in form.

What would be more desirable, in terms of image filtering is some method of removing spurious points which do not follow general trends of the overall intensity patterns.

This elimination of extraneous signals can be achieved by a method called colinearity smoothing. Colinearity smoothing is a method of combining critical points to reflect overall trends in the intensity profile. As in the original marking of critical points, colinearity smoothing is a mechanism by which points of interest can be distingushed.

Colinearity smoothing first locates a cluster of critical values. An example is shown in figure 15. The slope of the line segment AB is measured and compared with the slope of the line segment CD. If these are found to be sufficiently close, then the two slopes are compared to the slope of a new corrected line segment AD. If agreement is found to be sufficiently good, it can be hypothesized that the valley peak pair (B,C) is result of some noise in the image. The marking of the critical points B and C can now be removed. Subsequent computations involving these points can be reduced.

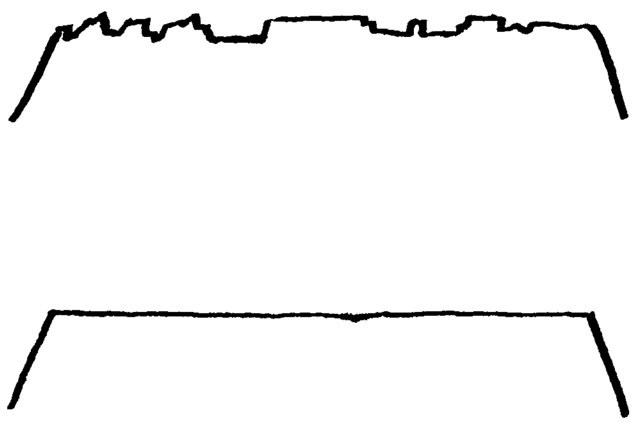


FIGURE 14

Comparison of Smoothed and Non-Smoothed Profiles

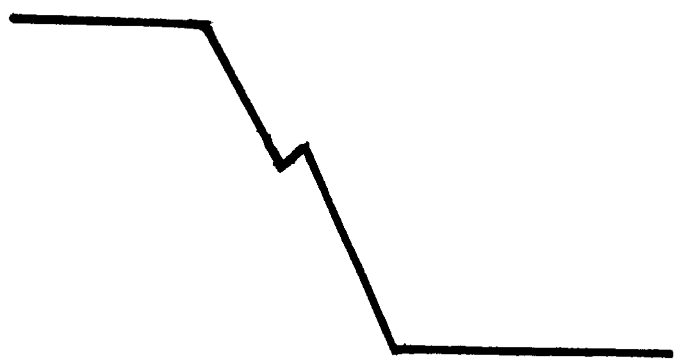


FIGURE 15

Intensity Profile Where Colinearity Smoothing Could be Used

The effectiveness of colinearity smoothing can be measured by counting the number of points that are removed by the algorithm. In the typical real-world images used in this study, there were, on the average, 20,000 local peaks and valleys in the intensity profiles in any single direction. Colinearity smoothing reduced the number of points marked by up to 8,000 points per image. This 33% reduction is strong evidence for the effectiveness of colinearity smoothing.

Colinearity smoothing is subject to 'skating'. Skating refers to a phenomenon by which the slope of a transition can be "adversely" changed through a sequence of small "allowable" changes. The resultant change in slope can be so great that the final degraded transition no longer represents the original transition.

Figure 16 shows an example where skating could be a problem. The line segments AB, CD, EF, GH, IJ, and KL are all parallel, within the measurement limitations allowed. If the distance between points B and D is small, then the angle <BAD will be small, due to the greater length of the line segment AB; because of this small angle, segment AD will be accepted as the true transition. This introduces a deviation from line segment AB by the angle <BAD, which is

within the limit of error prescribed. However, this new line segment will be merged with segment FF if the distance from D to F is small. The process increases the deviation from line segment AB by an amount proportional to <DAF where <DAF is within experimental error. Now the total deviation from line segment AB is proportional to <BAF which could well be beyond the allowed error. By successive applications of this process, the deviation from one segment can be made large.

One method of controlling skating is to limit the number of transition points which can be merged in one operation; this restricts the spatial extent of any single merging to some predefined limit. The problem with this concept is that the variations in transition size make any attempt to limit the scope by a constant unreasonable. A large upper limit will allow for proper considerations in large transitions, while allowing skating to occur on the smaller transitions. Similarly, a smaller upper limit will allow for proper consideration in the smaller transitions, but will give poorer results in the longer transitions. A floating limit, whose value would change with the syntax might be effective, however no investigation of the algorithm defining this limit was done.

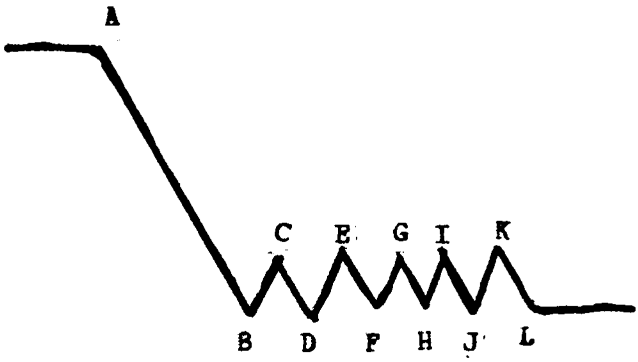


FIGURE 16

An Intensity Profile in Which Skating Can Occur

Another method of controlling skating is to limit the number of colinearity smoothing approximations that can be applied to a profile. In this way the number of times that merging can be done and, hence, the number of times the allowed deviation can be applied could not increase without bound. This method has the major disadvantage that regions which should be merged may not be merged due to the number of operations required. For example, in Figure 17, if only one merging was allowed, then the segment could not be fully adjusted to eliminate all noise peaks, as shown in Figure 16.

A method of avoiding the skating problem is to measure the deviation allowed for colinearity smoothing with respect to the original line segment slope rather than the merged line segment slope. In figure 16 this would mean that the segment AB would be merged with line segment CD to produce a new line segment AD. That is where the skating would be completed, since the deviation from line segment AB to include line segment EF is too large to allow for further merging.

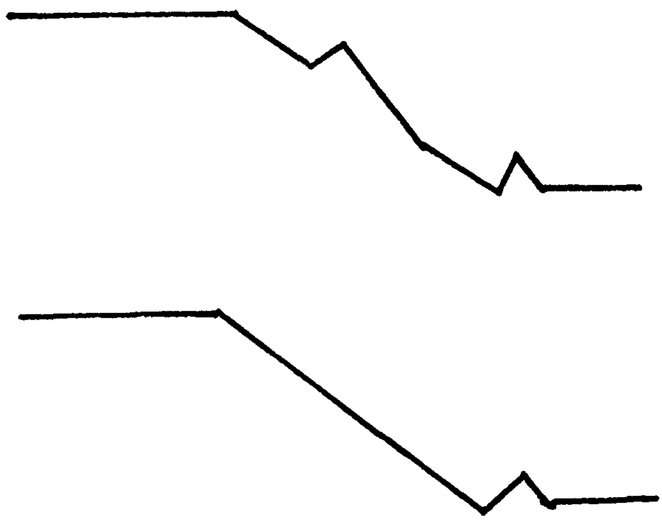


FIGURE 17

Example of the Use of Colinearity with Anti-Skating

CHAPTER 7

A Semantic Edge Detector

The preceeding chapters have all discussed the nature of edges, where they occur, what modes of transition occur, what the effects of noise are, and how to overcome noise effects. This section reports on an attempt to apply these findings in the development of an edge detector which is able to give better results than those now existing.

For reference, Appendix D contains a source listing of the program developed. The program was run on the HP-2100 owned by the department of Computer Science of Virginia Polytechnic Institute and State University. The processor has 32768 words of memory. The systems monitor was DOS III-B with many local modifications made by Mr. Glen Thompson. The FORTRAN compiler was an HP release dated 24177B-JULY 1971. Images were stored on a dual, single platter disk drive model 7900A. The intensity range of the images was 0 to 255. This limit on the intensity range (8 bits) made it possible to store the information in a packed format with two pixels in one 16-bit word.

The program logic is as follows. Several scan lines of the image are read into a buffer to minimize the number of disk accesses. As needed, a single scan line is unpacked. Subroutine MINMX is then used to mark the critical points. If requested, subroutine SIMPL is called to do a colinearity reduction on the critical points.

After reduction is complete, the first transition is classified as to leading and trailing critical points. Plateau-plateau transitions of constant height are disregarded since this mode of transition contains no edge point. Valley-valley and peak-peak transitions are both ignored, since they do not occur. The mode of all other transitions is then examined to see if they are quasi-linear. Depending on the mode of transition found, the edge is marked according to the techniques discussed in chapter 3. The output of line-by-line processing is then stored on disk.

The results of this program are shown in the following images. Appendix C contains photographs of the results of several edge detectors for comparison. The detector output shown in figure 18 shows a reduction of the picture called CAPS. The reduction was done in the horizonatal direction only. A deviation of 40 gray-levels was allowed from a line aproximation between the critical points before classifying a transition as shading-mode. As can be seen, the image is clear and crisp. Some jittering of the lines is present due to the fact that all rows are treated independently. Figure 19 shows the result of performing the same operation on the image, but in the vertical direction. In this second image, the points appear to be colinear, but in the other direction.

Both of these one-dimensional techniques have yielded good results. To get some idea of the results of a two-dimensional result, the picture functions shown in figures 18 and 19 were added together. The results are shown in figure 20 . No attempt was made to eliminate exact point-for-point co-incidence, which has lead to some points being abnormally bright.

To add some degree of sensitivity to the surroundings, the picture function was blurred by a rectangular filter of five by five pixels. The final result after detection is shown in figure 21. As can be seen, all of the jittering and irregularities of the image have been eleminated. Detail in the interor of the capacitors has been improved.

An interesting phenomena occurs when blurring is done. Due to the local averaging process, linear transitions can

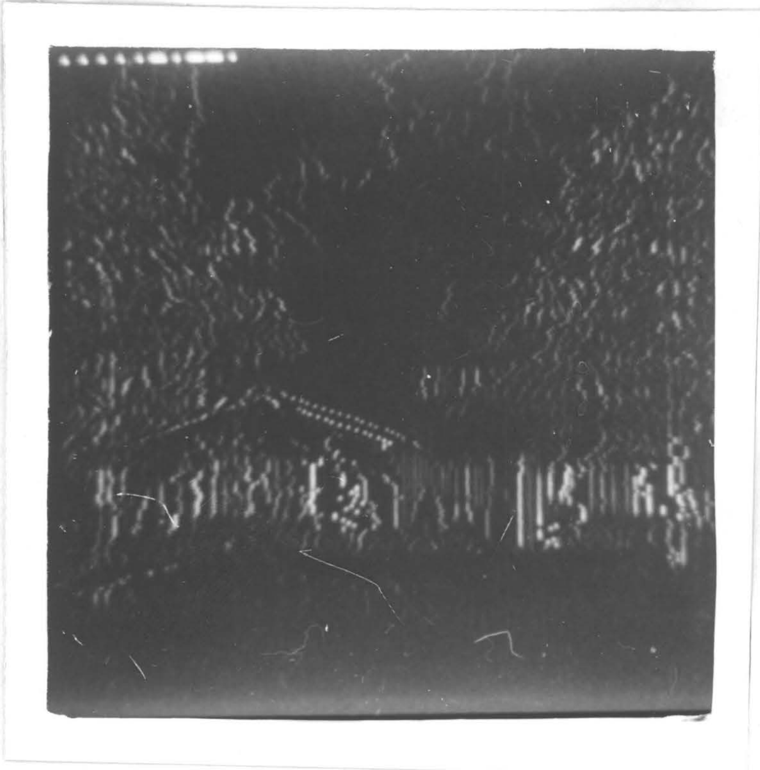


FIGURE 18

Semantic Edge Detection - Horizontal Direction

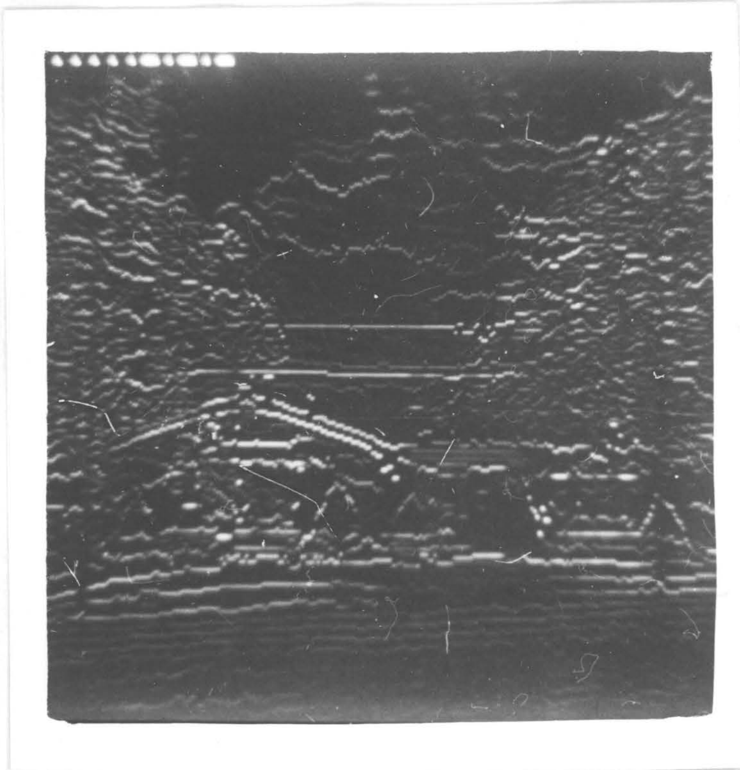


FIGURE 19

Semantic Edge Detection - Vertical Direction

FIGURE 20

Merged Semantic Detector Output

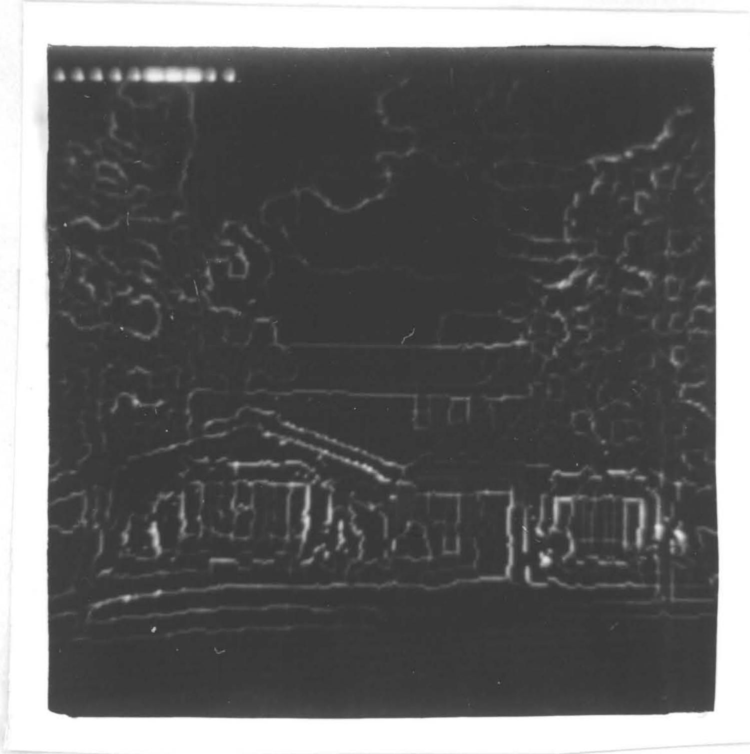


FIGURE 21

Semantic Detection After Blurring by 5 in Both Directions become less linear while not following the same patterns of behavior found in unblurred images. For this reason, a set of points following an edge, but at some distance from it, occurs in an image. This false edge is called the corona. Figure gives an example of the corona effect in the picture CAPS when blurred by a five pixel by five pixel averaging filter. A deviation of 20 intensity levels was allowed before the transition was marked as a shading-mode transition. By increasing the allowed deviation to 40 intensity levels, the corona can be eleminated as shown in figure 23. The corona was eliminated because the larger allowed deviation caused the reshaped ramp-mode transitions to be classified as ramp-mode transitions.

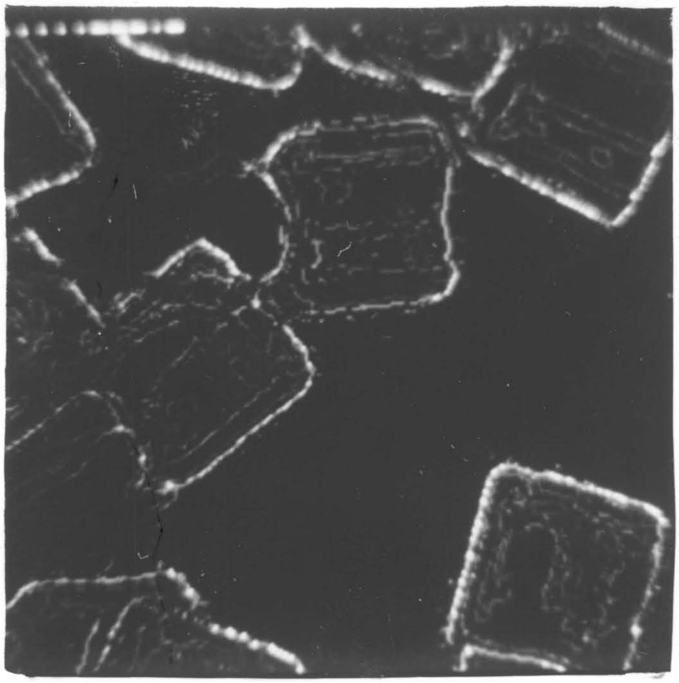


FIGURE 22

Example of the Corona Effect

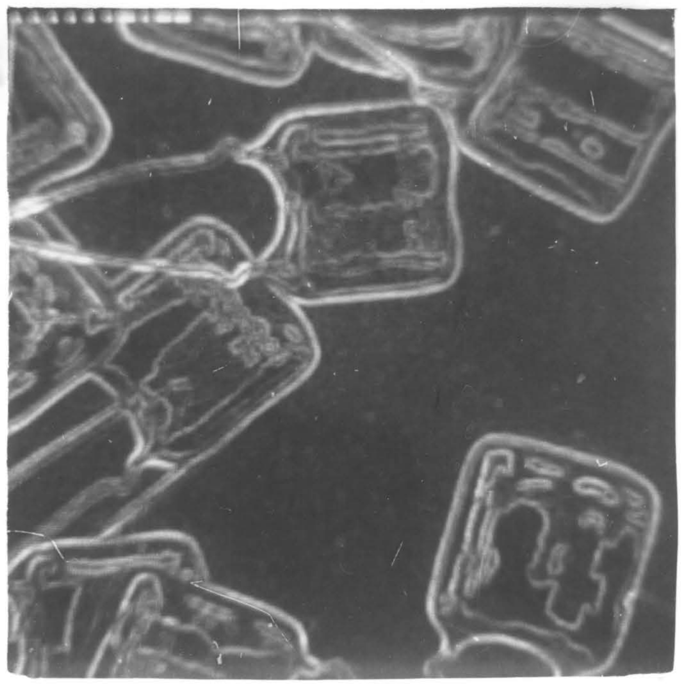


FIGURE 23

Corona-corrected image.

CHAPTER 8

Conclusions

In this thesis, a description language has been defined to describe the behavior of context-independent gray-level transitions in picture functions. The picture semantics developed were then used to describe the necessary conditions for an edge. The description language was tested by the development of a program to do analysis of transitions in picture functions. One of the products of this already existing picture functions. The line drawings produced were compared to those produced by the algorithms commonly in use in image processing. The line drawings produced by semantic edge detection were found to be superior to those produced by the best pre-existing techniques.

APPENDIX A

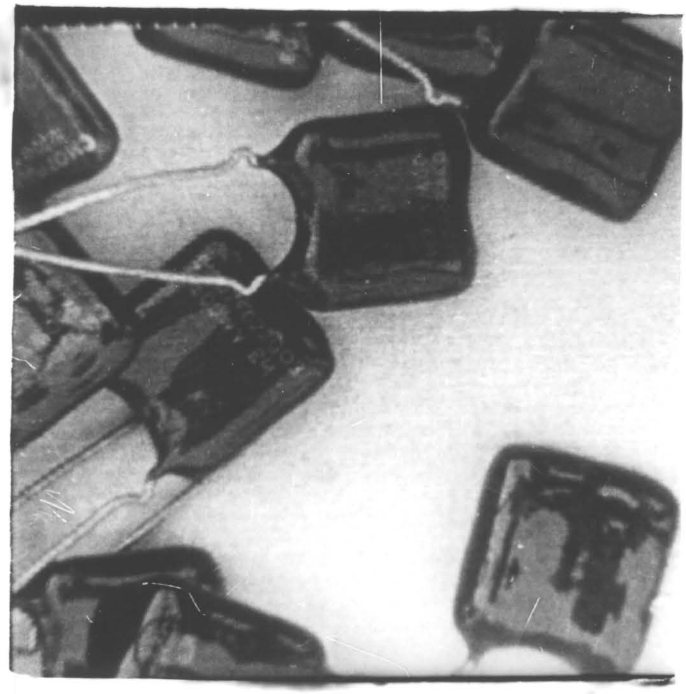
Report on Picture Function Transitions

The following tables report the frequency of occurrence of different transition points in three different images. They give some concept of the types and relative frequencies of the transitions. The information is listed for a scan (row-by-row) study of the images shown at the start of each table section. The relationships exibited by column studies are not signifigantly different.

The notation used in the following tables can be read in the following way:

Pbbbbb N=cccccc

where aaaaa is the mean value obtained for the measurement, bbbbb is the standard error, and ccccc is the number of occuennces counted for the transition pair being considered.



Picture:CAPS

Ramp Transition Table for CAPS

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading	Valley	-	1.36±0.005 N=11759	1.28±0.014 N=1563
Inflection	Peak	1.37±0.006 N=11600	-	1.30±0.014 N=1589
Point	Plateau	1.31±0.016 N=1606	1.28±0.014 N=1592	1.22±0.022 N=396

Ramp Size Transition Table for CAPS

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading Inflection Point	Valley	-	85.72±0.89 N=11759	92.91±3.45 N=1563
	Peak	87.25±0.836 N=11600	-	86.07±2.89 N=1589
	Plateau	88.32±2.61 N=1606	91.91±3.35 N=1592	87.89±8.10 N=396

Shading Length Transition Table for CAPS

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading Inflection Point	Valley	-	3.23±0.032 N=1758	3.73±0.155 N=130
	Peak	3.20±0.029 N=1858	-	3.12±0.097 N=165
	Plateau	3.28±0.096 N=161	3.47±0.129 N=149	3.14±0.218 N=28

Shading Size Transition Table for CAPS

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading	Valley	-	16.43±0.151 N=1758	12.52±0.375 N=130
Inflection	Peak	16.96±0.157 N=1858	-	12.43±0.346 N=165
Point	Plateau	11.93±0.355 N=161	11.87±0.335 N=149	10.03±0.643 N=28

Picture:HOUSE

Ramp Length Transition Table for HOUSE

Trailing Inflection Point

		Valley	Peak	Plateau
Leading	Valley	-	1.34±0.007 N=9186	1.31±0.015 N=1924
Inflection	Peak	1.38±0.007 N=9093	-	1.37±0.017 N=2010
Point	Plateau	1.35±0.016 N=2016	1.34±0.017 N=1962	1.32±0.017 N=1490

Ramp Size Transition Table for HOUSE

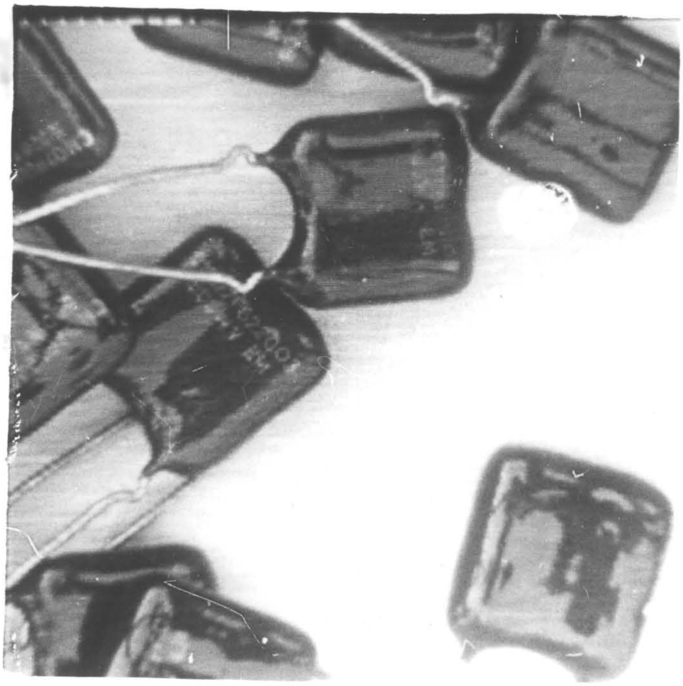
		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading Inflection Point	Valley	-	69.54 ± 0.744 N=9186	70.36 ± 2.98 N=1924
	Peak	69.37 ± 0.757 N=9093	-	77.37 ± 2.67 N=2010
	Plateau	69.20 ± 2.68 N=2016	70.47 ± 2.51 N=1962	57.30 ± 4.27 N=1409

Shading Length Transition Table for HOUSE

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading Inflection Point	Valley	-	2.87 ± 0.022 N=2731	3.29 ± 0.105 N=215
	Peak	2.85 ± 0.022 N=2731	-	3.36 ± 0.083 N=313
	Plateau	3.163 ± 0.094 N=234	3.35 ± 0.094 N=296	3.90 ± 0.215 N=77

Shading Size Transition Table for HOUSE

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading	Valley	-	16.24±0.187 N=2731	8.94±0.275 N=215
Inflection	Peak	16.34±0.183 N=2731	-	8.93±0.252 N=313
Point	Plateau	8.98±0.261 N=234	8.65±0.243 N=296	5.32±0.163 N=77



Picture: Smoothed Caps - Hysteresis Smoothing of 40

Ramp Length Transition Table for Smoothed CAPS

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading	Valley	-	1.37±0.042 N=211	1.51±0.026 N=879
Inflection	Peak	1.37±0.041 N=196	-	1.59±0.033 N=170
Point	Plateau	1.49±0.024 N=925	1.50±0.026 N=835	1.34±0.008 N=6397

Ramp Size Transition Table for Smoothed CAPS

Trailing Inflection Point

		Valley	Peak	Plateau
Leading	Valley	-	93.68±1.53 N=211	66.67±1.49 N=879
Inflection	Peak	90.87±1.30 N=196	-	80.99±1.52 N=710
Point	Plateau	72.10±1.56 N=925	73.95±1.51 N=835	65.86±1.26 N=6397

Shading Length Transition Table for Smoothed CAPS

Trailing Inflection Point

		Valley	Peak	Plateau
Leading	Valley	-	3.31±0.057 N=553	3.14±0.057 N=534
Inflection	Peak	3.11±0.053 N=592	-	3.64±0.064 N=718
Point	Plateau	3.10±0.053 N=543	3.33±0.055 N=724	3.69±0.059 N=907

Shading Size Transition Table for Smoothed CAPS

		Trailing Inflection Point		
		Valley	Peak	Plateau
Leading	Valley	-	59.96±1.95 N=553	20.14±0.646 N=534
Inflection	Peak	66.67±1.73 N=592	-	21.43±0.651 N=718
Point	Plateau	19.11±0.650 N=543	19.08±0.632 N=724	5.31±0.111 N=907

APPENDIX B

A Brief Glossary of Common Image Processing Terminology

Anti-Skating -

A technique devised to limit the scope of merging in colinearity smoothing.

Blocks world picture -

A picture composed solely of those things found in a child's building block set. Usually illuminated by a small, single source of light. These images were originally used in image studies because of their great simplicity.

Blurring -

The process of replacing every pixel in a digital picture function by the local average over several pixels. The size of the local neighborhood limits the degree of blurring.

Critical Point -

In an intensity profile, any of the following types of point : local maximum, local minimum, or plateau point.

<u>Digital Picture Function</u> -

A discrete, quantized function of two variables where each function value is formed by sampling the original image according to some regular grid. All research to date has used a rectilinear system of sampling; however, any other could work just as well.

<u>Edge</u> -

The dividing line between two homogeneous regions of different intensities

<u>Gradient</u> -

The n-dimensional derivative of some n-dimensional function. In scene analysis, this approximated by estimating a one-dimensional derivitive various axes, using a differencing algorithm, and combining the results of this differencing to get some higher-dimensional derivitive.

<u>Gradient Edge Detection</u> -

Applying a Gradient estimation algorithm to a digital picture function to yield an approximate two-dimensional derivative of the surface. In the past, these have been used to determine line drawings of objects.

<u>Hystersis Smoothing</u> -

A non-linear algorithm for removing small amplitude fluctuations in an intensity profile without disturbing large amplitude variations.

<u>Image</u> -

The pattern of light intensities which fall on a plane when focused by a lens system.

<u>Intensity</u> -

The measure of the amount of light reflected from a given region. Two standard scales of intensities are used: linear and logrithmic.

<u>Intensity Profile</u> -

A plot of a digital picture function as a function of one of its variables(i or j), while the other is held constant.

<u>Line Drawing</u> -

A binary picture function which is the representation of some other picture function in which only the boundaries delineating regions are marked.

<u>Mark</u> -

To label a pixel, or pixels, because of certain properties.

<u>Mole</u> -

See Transition Mode.

Noise -

Any variation in the picture function which does not reflect the actual intensity variations in the object.

Noise Reduction -

The process of reducing the small intensity variations from pixel to pixel in an image.

Operator -

A mathematical tool which can be implemented on the computer to some fair approximation. Most operators in image processing are differential operators, which compute differences among local regions.

Shading Edge -

Any intensity variations which are largely nonlinear.

Peak -

A region surrounding alocal intensity maximum that is bounded by the highest of its adjacent valleys.

Picture Element -

One unit of a sampled image.

Picture Function -

A function of two variables which describes the intensity variations in an image. No restrictions are set on the variables the function is over, or on the function value.

Plateau -

A region in the intensity profile over which there is no change in intensity.

Pixel -

Shortened form of the words "picture element".

Skating -

The process by which edges might be extended beyond their original bounds during co-linearity smoothing by aggreation of several small intensity changes.

Transition Mode -

The manner by which intensity profiles vary between critical points. Transition mode is usually classified as linear or non-linear.

Real-World Image -

An image which is composed of representations of scenes commonly experienced by most people. A landscape is a real-world image, while a blocks world picture is not.

<u>Valley</u> -

A region of local intensity minimum that is bounded by the lowest of its adjacent valleys.

APPENDIX C

A Comparison of Various Edge Detectors

In this section, the results of applying several gradient edge detectors to two pictures are examined. As can easily be seen, the resultant images leave much to be desired. The problems with such an of edge finder where characterized in the introduction as:

1) The broader the filter width (larger region sampled for the gradient aproximation), the broader will be the response of the detector in the filter output image, f(i,j). This smoothing is proportional to the filter width.

2) The broader the image gradient (the longer the period it takes to reach some local maximum) the larger the filter mask must be to optimally detect the gradient.

3) Broader filters are more noise immune (less sensitive to the effects of points that might be noise) while small filters show good transient response with poor noise immunity.

4) The maximum value of the filter output is a function of the difference in image intensity between two regions. The relationship varies from filter to filter, and is a function of the size of the filter as well as the picture function being examined.

5) The determination of a "representative value" for the one-directional differencing requires some degree of interpretation of the pixels which may not be warranted. Some of the pixels contain noise, and to weigh these equally with noise-free pixel values would be a mistake.

The pictures were chosen on the basis of the extreme variations in texture and detail available in each of these real world images. No hysteresis smoothing or other operators have been applied.

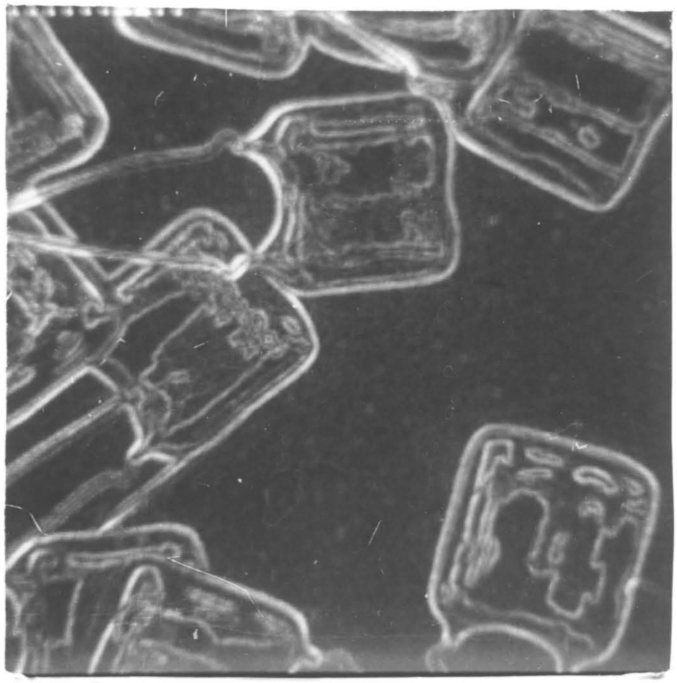


FIGURE 24

Roberts Cross Operator Applied to CAPS

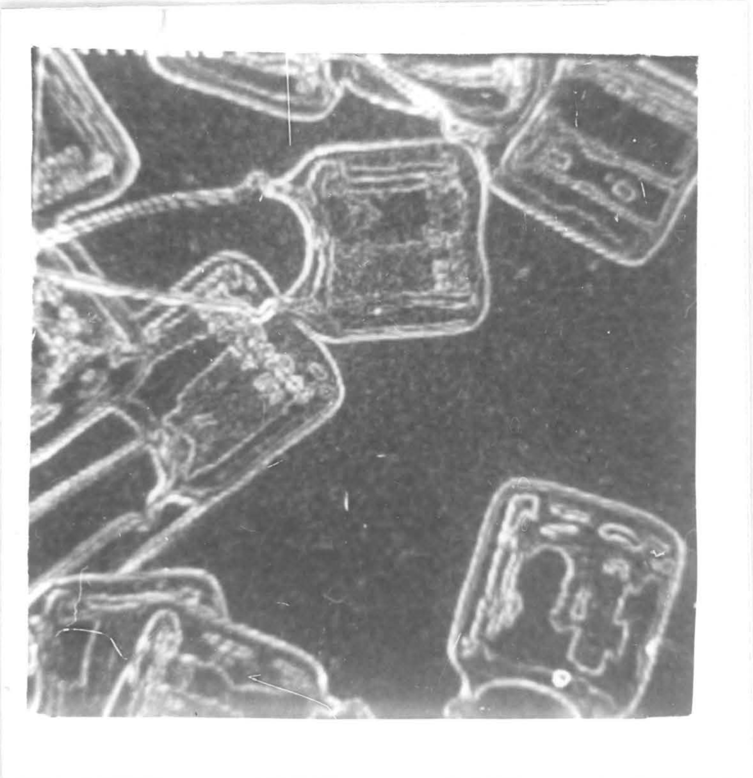


FIGURE 25

Sobel Operator Applied to CAPS

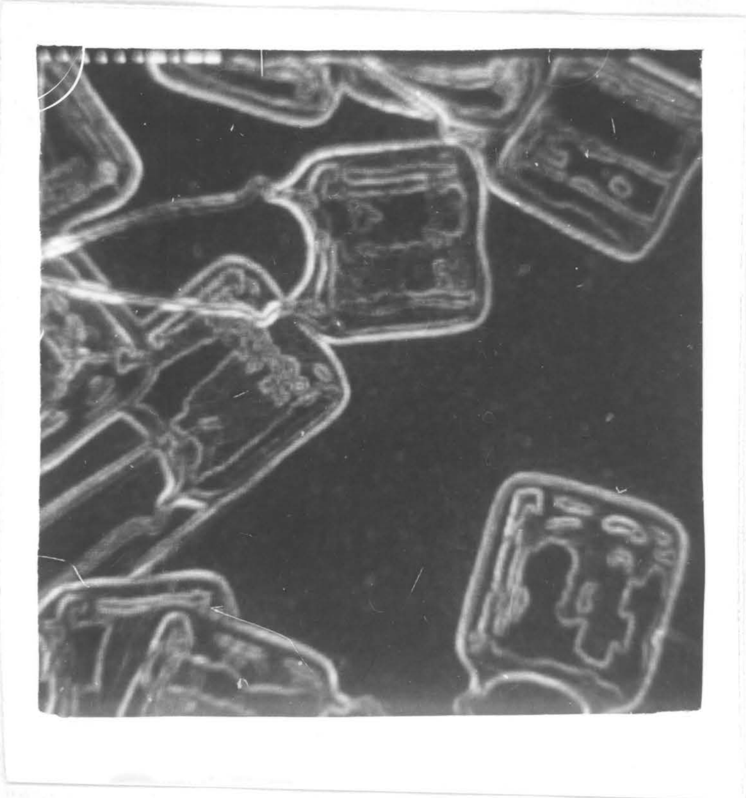


FIGURE 26

Kirsch Operator Applied to CAPS

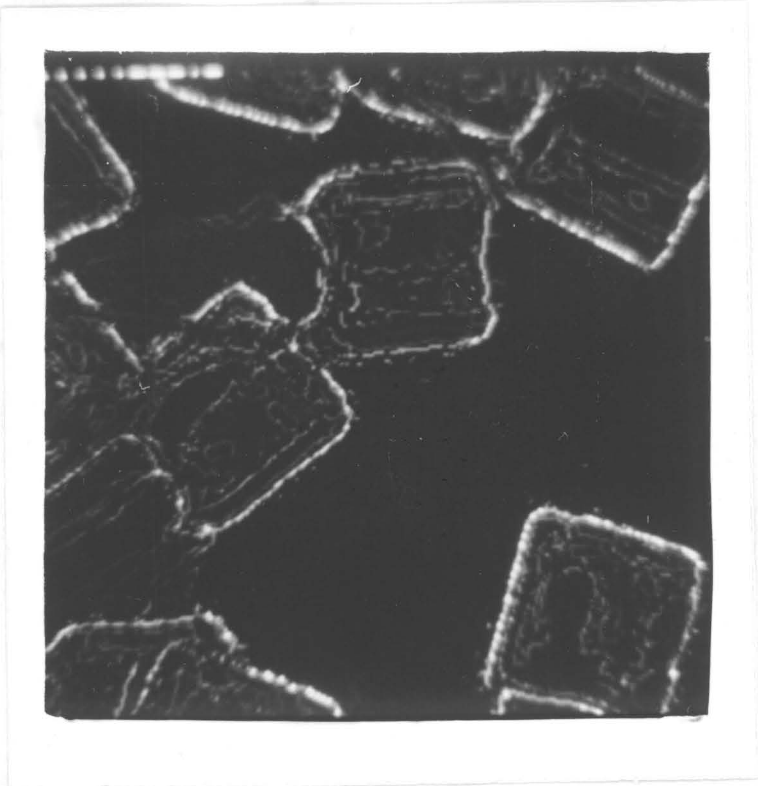


FIGURE 27

Semantic Operator Applied to CAPS

FIGURE 28

Roberts Cross Operator Applied to HOUSE

FIGURE 29

Sobel Operator Applied to HOUSE

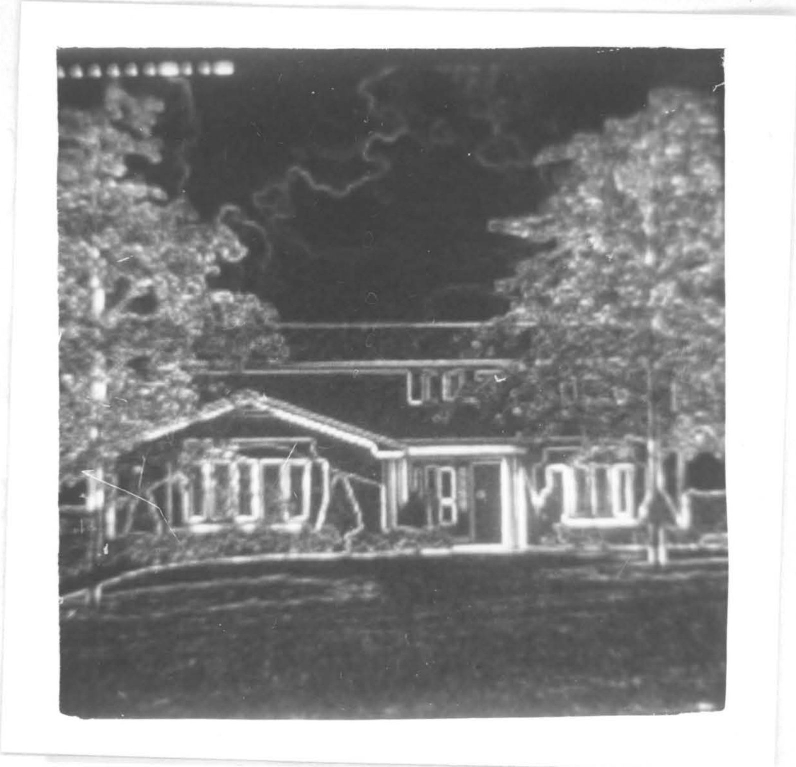


FIGURE 30

Kirsch Operator Applied to HOUSE

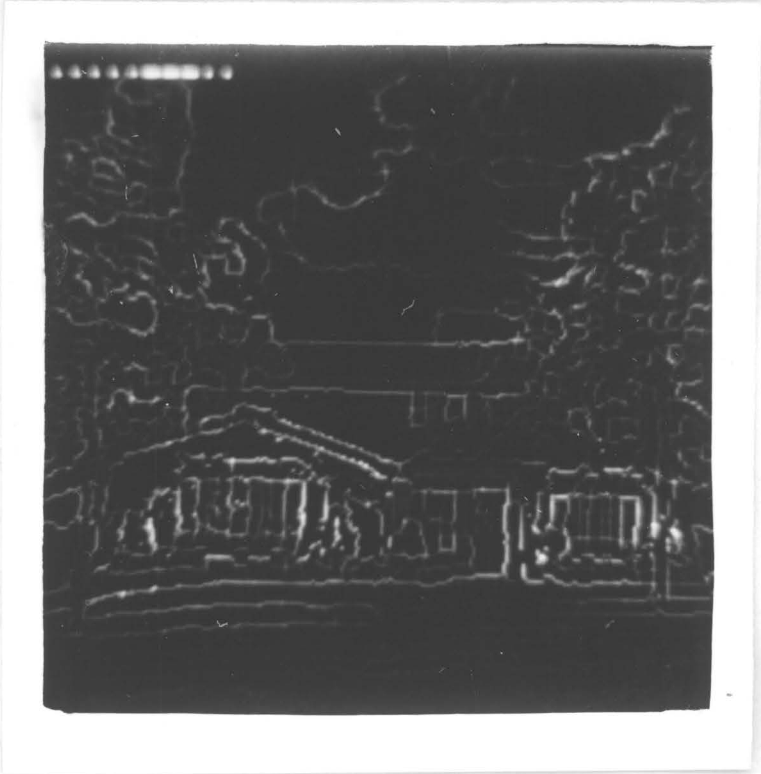


FIGURE 31

Semantic Operator Applied to HOUSE

APPENDIX D

A Simple Program to do Semantic Edge Detection

The following program was written to run on the HP2100 computer under DOS IIIC using the 24177B-JULY 1971 FORTRAN compiler. The operating system has undergone great modification at this instilation.

More information on this program may be found in Chapter 7.

```
      FTN4
      PROGRAM SEGE
C
C     PURPOSE: TO SEMANTICALLY DETECT THE EDGES IN A IMAGE
C               USING THE GUIDLINES FOUND IN THE THESIS OF
C                THE AUTHOR, WILLIAM BRENT LANDER
C     WRITTEN: 13 AUGUST 77
C
C     NOTES:
C     THE PROGRAM CAN BE UNDERSTOOD MUCH MORE EASILY WHEN THE
C     MEANINGS OF THE FOLLOWING VARIABLES IS UNDERSTOOD.
C     ALLOWED VALUES ARE ALSO DESCRIBED.
C
C     IDA - THE TRANSITION MODE. IDA HAS VALUES FROM 1 TO 9,
C           INCLUSIVE.
C           THE FOLLOWING TABLE GIVES THE MEANINGS OF
C           THE VALUES OF IDA.
C
C                      TRAILING TRANSITION POINT
C                      VALLEY  PEAK  PLATEAU
C                      ---------------------
C     LEADING|VALLEY |    1      2      3
C     TRANS. | PEAK  |    4      5      6
C     POINT  |PLAT.  |    7      8      9
C
C
C     TABLE SUMMATIONS ARE MADE IN THE 9-VECTORS
C     LISTED BELOW.
C
C          TRANS1 - COUNT OF NUMBERS OF LINEAR TRANSITIONS.
C          TRANS2 - SUM OF LENGTH OF LINEAR TRANSITIONS.
C          TRANS3 - SUM OF SQUARES OF LENGTH OF LINEAR
C                   TRANSITIONS.
C          TRANS4 - COUNT OF NUMBERS OF SHADING TRANSITIONS.
C          TRANS5 - SUM OF LENGTH OF SHADING TRANSITIONS.
C        . TRANS6 - SUM OF SQUARES OF LENGTHS OF SHADING
C                   TRANSITIONS.
C          TRANS7 - SUM OF SIZE (INTENSITY VARIATION) OF
C                                 LINEAR TRANSITIONS.
C          TRANS8 - SUM OF SQUARE OF SIZE OF
C                                 LINEAR TRANSITIONS.
C          TRANS9 - SUM OF SIZE OF SHADING TRANSITIONS.
C          TRANSA - SUM OF SQUARES OF SIZE OF SHADING
C                   TRANSITIONS.
C          TRANSB - MAXIMUM SHADING TRANSITION SIZE.
C
C
C     AFTER ANALYSIS, THE STATISTICS ARE GENERATED.
C     THE TABLES HAVE THE FOLLOWING MEASURES:
C
```

```
C       COUNT - TRANS1, TRANS4
C       AVERAGE - TRANS2, TRANS5, TRANS7, TRANS9
C       ST. DEV. - TRANS3, TRANS6, TRANS8, TRANSA
C       MAXIMUM  - TRANSB
C
C
C
C       <><><><><><><><><><><><><><><><><><><><><><><><><>
C
        COMMON IMAGEA(256), IMAGEB(256), IMAGFC(1920),
      1        IFILE1(1940)
        INTEGER TRANS1(9),TRANS4(9),TRANSB(9)
        REAL TRANS6(9),TRANS7(9),TRANS8(9),TRANSA(9)
        REAL TRANS2(9),TRANS3(9),TRANS5(9),TRANS9(9)
        INTEGER NAM(3),NAM2(3),TABLE1(50),TRACK(256),
      1         IFILE(128, 16)
        EQUIVALENCE (IMAGEB(129), IFILE(1,1))
        WRITE(1,100)
100     FORMAT("SEGE")
        CALL FILNM(NAM,1)
        CALL FILNM(NAM2,2)
C
C *** SET TABLES TO ZERO INITIALLY
C
        DO 1 I=1, 9
        TRANS1(I)=0
        TRANS2(I)=0
        TRANS3(I)=0
        TRANS4(I)=0
        TRANS5(I)=0
        TRANS6(I)=0
        TRANS7(I)=0
        TRANS8(I)=0
        TRANS9(I)=0
        TRANSA(I)=0
        TRANSB(I)=0
1       CONTINUE
        EPS=1
        EPS2=1
        WRITE(1,110)
110     FORMAT("ENTER EPS1, EPS2")
        READ(1,*) EPS , EPS2
        WRITE(1,600)
600     FORMAT("CO-LINEARITY SMOOTHING??")
        IF (JGETC(ICHAR2) .NE. 131B) GOTO 2
        WRITE(1,610)
610     FORMAT("CO-LINEARITY SMOOTHING WILL BE DONE."/
      1        "ENTER DEG1, DEG2, BIAS")
        DEG1=10
        DEG2=10
```

```
      BIAS=10
      MERGE=0
      READ(1,*) DEG1, DEG2, BIAS
      GOTO 630
2     WRITE(1,620)
620   FORMAT("CO-LINEARITY SMOOTHING WILL NOT BE DONE.")
630   CONTINUE
C
C *** MAIN PROGRAM LOOP
C *** HERE 1) TRANSITIONS ARE CLASSIFIED AS TO
C ***               TYPE AND MODE
C ***        2) STATISTICAL MEASUREMENTS ARE MADE
C
      DO 90 I=1, 16, 1
      CALL DRD (NAM,IFILE ,(I-1)*16,16)
      DO 85 J=1,16
           DO 3 K=1,128
           IMAGEA(K)=IFILE(K,J)
           TRACK(K)=0
           TRACK(K+128)=0
3          CONTINUE
      CALL UPACK(IMAGEA,256)
      CALL MINMX
      IF (ICHAR2 .EQ. 1313) CALL SIMPL(DEG1,DEG2,BIAS,0,
     1                                       MERGE)
      POSA=0
5     CALL NEXT(A,POSA,ICODE1)
      IF (POSA.LE.3.) GOTO 5
      ICODE1=ICODE1+1
      POSB=POSA
10    CALL NEXT (B, POSB, ICODE2)
      IF(ICODE2 .LT. 0) GO TO 80
      ICODE2=ICODE2 + 1
C
C *** CHECK TO SEE IF NEW POINT PAIR IS AN EDGE
C
      IDA=(ICODE1-1)*3+ICODE2
      IF (IDA .GT. 9 .OR. IDA .LT. 1) WRITE(1,3000)IDA
3000  FORMAT("IDA=",I10)
      GOTO (75,20,20,20,75,20,20,20,15),IDA
C
C *** CHECK TO SEE IF BOTH ENDS OF PLATEAU TO PLATEAU
C *** TRANSITION ARE ON THE SAME LEVEL.
C
15    IF (A .EQ. B) GO TO 75
C
C *** VALID, INTERESTING TRANSITION, .
C *** CHECK TO SEE IF LINEAR.
C
20    RATIO=(A-B)/(POSA-POSB)
```

```
      K=0
      DO 25 L=POSA, POSB
      IF (ABS(RATIO*K+A - IMAGEA(L)) .GT. EPS) GOTO 30
      K=K+1
25    CONTINUE
C
C *** EDGE TRANSITION IS LINEAR, MARK CENTER WITH TOTAL
C
C *** DIFFERENCE
      TRACK((POSA+POSB)/2.0+0.5) = ABS(A-B)
      TRANS1(IDA)=TRANS1(IDA)+1
      TRANS2(IDA)=TRANS2(IDA)+POSB-POSA
      TRANS3(IDA)=TRANS3(IDA)+(POSB-POSA)**2
      TRANS9(IDA)=TRANS9(IDA)+ABS(A-B)
      TRANSA(IDA)=TRANSA(IDA)+(A-B)**2
      GOTO 75
C
C
C *** POOR FIT AT  "I" - FIND THE BEST EXISTING LINEAR
C *** POINT
C
C *** GENERATE TABLE
C
30    I1=POSA
      I2=POSB
      K=0
      DO 32 L=I1,I2-1
      K=K+1
      TABLE1(K)=IMAGEA(L)-IMAGEA(L+1)
32    CONTINUE
      IF (K .GT. 50) WRITE(1,1000) ISECT
1000  FORMAT("TABLE OVERFLOW AT SECTOR ",I5)
      LRUN=1
      ISTART=I1
C
C *** AND NOW THE SLOW SEARCH
C
      DO 34 L1=1,K
      NRUN=1
      RATIO=TABLE1(L1)
      IF (L1 .GE. K-1) GO TO 31
      DO 33 L2=L1+1,K
      IF (ABS(TABLE1(L2)-RATIO) .GT. EPS2) GO TO 31
      NRUN=NRUN+1
33    CONTINUE
C
C *** NOW LOOK BACK
C
31    IF (L1 .LE. 1) GO TO 40
      DO 35 L3=1,L1-1
```

```
      IF  (ABS(TABLE1(L1-L3)  - RATIO)  .GT.  EPS)  GO TO 40
      NRUN=NRUN+1
35    CONTINUE
40    IF(NRUN .LE. LRUN) GO TO 34
      LRUN=NRUN
      ISTART=I1+L1-L3
34    CONTINUE
C
C *** LONGEST SPAN FOUND AT 'ISTART' HAVING LENGTH 'LRUN'
C
70    ISTART=ISTART+(LRUN/2.0)-0.5
      IF  (ISTART .LE. 0)  ISTART=1
      TRACK(ISTART)=ABS(A-B)
C
C *** GATHER STATISTICS ON THE NATURE OF SHADING MODE
C *** TRANSITIONS.
C
      TRANS4(IDA)=TRANS4(IDA)+1
      TRANS5(IDA)=TRANS5(IDA)+POSB-POSA
      TRANS6(IDA)=TRANS6(IDA)+(POSB-POSA)**2
      TRANS7(IDA)=TRANS7(IDA)+ABS(A-B)
      TRANS8(IDA)=TRANS8(IDA)+(A-B)**2
      IF(TRANSB(IDA)  .LT.  ABS(A-B))  TRANSB(IDA)=ABS(A-B)
C
C *** SETUP FOR NEXT LOOP
C
75    ICODE1 = ICODE2
      POSA=POSB
      A=B
      GOTO 10
C
C *** GO BACK FOR NEXT LINE
C
80    CALL PACK(TRACK,256)
      CALL DWR(NAM2, TRACK,  (I-1)*16+J-1,  1)
      CALL OUTS((I-1)*16+J-1)
85    CONTINUE
90    CONTINUE
C
C *** ANALYSIS SECTION- - -  REDUCE THE MEASUREMENTS.
C
      DO 905 I=1, 9
      IF  (TRANS1(I)  .EQ.  0)  TRANS1(I)=1
      IF  (TRANS4(I)  .EQ.  0)  TRANS4(I)=1
C
C *** COMPUTE AVERAGES
C
      TRANS2(I)=TRANS2(I)/FLOAT(TRANS1(I))
      TRANS9(I)=TRANS9(I)/FLOAT(TRANS1(I))
      TRANS5(I)=TRANS5(I)/FLOAT(TRANS4(I))
```

```
      TRANS7(I)=TRANS7(I)/FLOAT(TRANS4(I))
C
C *** COMPUTE STANDARD DEV.
C
      TRANS3(I)=SQRT((TRANS3(I)/FLOAT(TRANS1(I)))
     1                -TRANS2(I)**2)
      TRANS6(I)=SQRT((TRANS6(I)/FLOAT(TRANS4(I)))
     1                -TRANS5(I)**2)
      TRANS8(I)=SQRT((TRANS8(I)/FLOAT(TRANS4(I)))
     1                -TRANS7(I)**2)
      TRANSA(I)=SQRT((TRANSA(I)/FLOAT(TRANS1(I)))
     1                -TRANS9(I)**2)
      IF(TRANS1(I) .EQ. 1) TRANS1(I)=0
      IF(TRANS4(I) .EQ. 1) TRANS4(I)=0
905   CONTINUE
      WRITE(1,200)
200   FORMAT("ANALYSIS COMPLETE. DO YOU WANT THE REPORT?")
      IF (JGETC(ICHAR) .NE. 131B) CALL SUPER(1)
      WRITE(1,210)
210   FORMAT("ENTER 1 FOR SYSTEMS CONSOLE, 2 FOR PRINTER")
      READ (1,*) I
      IF (I .NE. 1) I=11
      IF (I .EQ. 1) CALL NEWPG
      IF (I .EQ. 1) GOTO 907
      CALL EXEC (3,1113B,-1)
      WRITE(I,650) NAM, EPS1, EPS2
650   FORMAT(////20X,"STATISTICAL TABLES FOR ",3A2,
     1  20X,"ALLOWED VARIATION IN TOTAL LINEARITY =",I10
     2  20X,"ALLOWED VARIATION IN LINEAR SEGMENT  =",I10)
      IF (ICHAR2 .EQ. 131B) WRITE(I,660) DEG1, DEG2, MERGE
660   FORMAT(20X,
     1  "THE RESULTS CONTAINED REFLECT A COLINEARITY"
     2   "SMOOTHING ON THE ORIGINAL"///
     3   20X,"FIRST FREEDOM VARIATION=",F10.2," DEGREES"//
     4   20X,"SECOND FREEDOM VARIATION=",F10.2," DEGREES"//
     5   20X,I10," POINTS WERE REMOVED ")
      CALL EXEC(3,1113B,-1)
907   CONTINUE
      WRITE(I,220) NAM
220   FORMAT(////20X,"RAMP TRANSITION TABLE FOR ",3A2)
      WRITE(I,510) TRANS1
      IF (I .EQ. 11) GO TO 91
      CALL LINE (50,0,1,1)
      WRITE(I,230)
230   FORMAT(" HIT SPACE BAR TO CONTINUE ")
      IF (JGETC(ICHAR) .EQ. 15B) CALL SUPER(2)
91    WRITE(I,240) NAM
240   FORMAT(////20X,
     1   "AVERAGE RAMP TRANSITION LENGTH TABLE FOR ",3A2)
      WRITE(I,500) TRANS2
```

```
      IF (I .EQ. 11) GO TO 915
      CALL LINE (50,0,1,1)
      WRITE(I,230)
      IF (JGETC(ICHAR) .EQ. 15B) CALL SUPER(1)
      CALL NEWPG
915   WRITE(I,235) NAM
235   FORMAT(////20X,
     1"ST. DEV. OF RAMP LENGTH TRANSITION TABLE FOR ",3A2)
      WRITE(I,500) TRANS3
      IF (I .EQ. 11) CALL EXEC (3,1113B,-1)
      IF (I .EQ. 11) GOTO 92
      CALL LINE (50,0,1,1)
      WRITE(I,230)
      IF (JGETC(ICHAR) .EQ. 15B) CALL SUPER(1)
      CALL NEWPG
92    WRITE(I,250) NAM
250   FORMAT(////20X,
     1   "AVERAGE RAMP SIZE TRANSITION TABLE FOR ",3A2)
      WRITE(I,500) IRANS9
      IF (I .EQ. 11) GOTO 93
      CALL LINE (50,0,1,1)
      WRITE(I,220)
      IF (JGETC(ICHAR) .EQ. 15B) CALL SUPER(1)
      CALL NEWPG
93    WRITE(I,260) NAM
260   FORMAT(////20X,
     1   "ST. DEV. OF RAMP TRANSITION SIZE TABLE FOR ",3A2)
      WRITE(I,500) TRANSA
      IF (I .EQ. 11) CALL EXEC (3,1113B,-1)
      IF (I .EQ. 11) GOTO 94
      CALL LINE(50,0,1,1)
      WRITE(I,230)
      IF (JGETC(ICHAR) .EQ. 15B) CALL SUPER(1)
      CALL NEWPG
94    WRITE(I,270) NAM
270   FORMAT(////20X,
     1   "COUNT OF SHADING TRANSITIONS TABLE FOR ",3A2)
      WRITE(I,510) TRANS4
      IF (I .EQ. 11) GOTO 95
      CALL LINE(50,0,1,1)
      WRITE(I,230)
      CALL NEWPG
95    WRITE(I,280) NAM
280   FORMAAT(////20X,
     1   "AVERAGE LENGTH OF TRANSITION TABLE FOR ",3A2)
      WRITE(I,500) TRANS5
      IF (I .EQ. 11) GOTO 96
      CALL LINE( 50, 0, 1, 1)
      WRITE(I,230)
      IF (JGETC(ICHAR) .EQ. 15B) CALL SUPER(1)
```

```
      CALL NEWPG
96    WRITE(I,290) NAM
290   FORMAT(///20X,
     1"SHADING LENGTH ST. DEV. TRANSITION TABLE FOR ",3A2)
      WRITE(I,500) TRANS6
      IF (I .EQ. 11) CALL EXEC (3, 1113B,-1)
      IF (I .EQ. 11) GOTO 97
      CALL LINE(50,0,1,1)
      WRITE(I,230)
      IF (JGETC(ICHAR) .EQ. 15B) CALL SUPER(1)
      CALL NEWPG
97    WRITE(I,300) NAM
300   FORMAT(///20X,
     1"AVERAGE SIZE OF SHADING TRANSITION TABLE FOR ",3A2)
      WRITE(I,500) TRANS7
      IF (I .EQ. 11) GOTO 98
      CALL LINE (50,0,1,1)
      WRITE(I,230)
      IF (JGETC(ICHAR) .EQ. 15B) CALL SUPER(1)
      CALLNEWPG
98    WRITE(I,310) NAM
310   FORMAAT(///20X,
     1"SIZE ST. DEV. OF SHADING TRANSITION TABLE FOR",3A2)
      WRITE(I,500) TRANS8
      WRITE(I,320) NAM
320   FORMAT(///20X,
     1   "MAXIMUM SHADING EDGE SIZE TABLE FOR ",3A2)
      WRITE(I,500) TRANS8
      IF (I .EQ. 11) CALL EXEC (3,1113B,-1)
      IF (I .EQ. 11) CALL EXEC (3,1113B,-1)
      CALL LINE(50,0,1,1)
      WRITE(1,230)
      ICHAR=JGETC(ICHAR)
C
C *** END PROGRAM.
C
      WRITE(1,9000)
9000  FORMAT("END OF PROGRAM")
      CALL SUPER (1)
500   FORMAT(25X,"TRAILING INFLECTION POINT"//
     123X," VALLEY       PEAK       PLATEAU"/
     221X,40("-")/
     3"  LEADING |  VALLEY |  ",3(F8.3,2X)/10X,"|",9X,"|"/
     4"INFLECTION|   PEAK  |  ",3(F8.3,2X)/10X,"|",9X,"|"/
     5"  POINT   | PLATEAU |  ",3(F8.3,2X)/10X,"|",9X,"|")
510   FORMAT(25X,"TRAILING INFLECTION POINT"//
     123X," VALLEY       PEAK       PLATEAU"/
     221X,40("-")/
     3"  LEADING |  VALLEY |  ",3(I8,2X)/10X,"|",9X,"|"/
     4"INFLECTION|   PEAK  |  ",3(I8,2X)/10X,"|",9X,"|"/
```

```
5"  POINT    | PLATEAU |   ",3(I8,2X)/10X,"|",9X,"|")
     END
```

```
      SUBROUTINE SIMPL (DEG1, DEG2, BIAS, SKATE, MERGE)
C     PROGRAM TO DO NOISE REDUCTION BY THE COMBINATION OF
C     REGIONS OF EQUAL SLOPE AND CHARACTERISTICS.
C     WRITTEN BY BRENT LANDER, JUNE3, 1977
C     PARAMETERS - DEG1 -ANGLE IN DEGREES OF THE MAX
C                          VARIATION IN THE SLOPE
C                        OF TWO LINES TO BE CONSIDERED
C                        CANDIDATES FOR MERGING
C                  DEG2 - ANGLE IN DEGREES OF THE MAX
C                         VARIATION IN SLOPE
C                  BIAS  - THE LENGTH BELOW WHICH THE
C                          ANGULAR DEVIATION ALLOWED
C                          IS DOUBLED.
C                  SKATE - IF ZERO, THEN INITIAL SLOPE
C                                  NOT ALLOWED TO VARY
C                                  WITH THE ADDITION
C                                   OF NEW SECTIONS.
C                             NOT ZERO, THEN SLOPE CAN VARY.
C                   MERGE - ADD UP THE TOTAL NUMBER OF
C                           MERGED LINES
C                           IN THE ENTIRE LINE FROM
C                           FROM ONE OF THE SEGMENTS.
C    PROGRAM VARIABLES - A, B, C, D- THE INTENSITY OF THE
C                                    POINTS ALONG THE
C                                    CURVE BEING
C                                     APROXIMATED.
C                        POSA,POSB,POSC,POSD-THE POSITION
C                                    OF THE
C                                    POINTS MARKED A,B,C,D
C    SUBROUTINES CALLED : ATAN2
C                          NEXT
C
C    <><><><><><><><><><><><><><><><><><><><><><><><><><><><>
      COMMON IMAGEA(256), IMAGEB(256)
      LOGICAL RESET, NOTNEW
      DATA PI/3.141592654/
      ANG1=DEG1*PI/180.0
      ANG2=DEG2*PI/180.0
```

```
      ANGX=ANG1
      ANGY=ANG2
      POSD=0
      RESET = SKATE .NE. 0
C
C *** SCANING LOOP
C
10    POSA=POSD
      CALL NEXT (A,POSA, ICODE)
15    IF (ICODE .EQ. 2) CALL NEXT(A, POSA, ICODE)
      NOTNEW = .FALSE.
      IF (ICODE .EQ. -1) RETURN
20    POSB=POSA
      CALL NEXT (B, POSB, ICODE)
      IF (ICODE .EQ. -1) RETURN
30    POSC=POSB
      CALL NEXT (C, POSC, ICODE)
      IF (ICODE .EQ. 2) GO TO 50
      IF (ICODE .EQ. -1) RETURN
40    POSD=POSC
      CALL NEXT (D, POSD, ICODE)
      IF (ICODE .EQ. -1) GO TO 50
C
C *** STANDARD SETUP
C
      IF (RESET .OR. .NOT. NOTNEW)
     1               ANGAB=ATAN2(A-B, POSA-POSB)
      ANGCD=ATAN2(C-D,POSC-POSD)
       Z=POSB-POSA+ABS(B-A)
       Z2=POSD-POSC+ABS(D-C)
      IF (Z .LT. BIAS .OR. Z2 .LT. BIAS) ANG1=ANG1*2
C
C *** CHECK ON PREMISE 1
C
      IF (ABS(ANGAB-ANGCD) .GT. ANG1) GO TO 45
      ANGAD=ATAN2(A-D,POSA-POSD)
C
C *** CHECK ON PREMISE 2
C
      IF(POSD-POSA+ABS(D-A) .LT. BIAS) ANG2=ANG2*2
      IF(ABS(ANGAD-ANGAB).GT.ANG2
     1               .OR.
     2  ABS(ANGAD-ANGCD).GT.ANG2)
     3      GO TO 45
C
C *** IT WORKED!!!! CONCATINATE.
C
      NOTNEW = .TRUE.
      ANG1=ANGX
      ANG2=ANGY
```

```
      I=POSB
      IMAGEB(I) = 0
      I=POSC
      IMAGEB(I) = 0
      POSB=POSD
      B=D
      MERGE = MERGE+1
      IF (ICODE .NE. 2) GO TO 30
      POSA=POSD
      GO TO 15
C
C *** PREMISES NOT SATISFIED, MOVE ON DOWN THE LINE
C
45    A=B
      B=C
      C=D
      POSA=POSB
      POSB=POSC
      POSC=POSD
      ANG1=ANGX
      ANG2=ANGY
      NOTNEW = .FALSE.
      IF (ICODE .NE. 2) GO TO 40
C
C *** PLATEAU OR END AS THIRD MEMBER LINE SEGMENT
C
50    ANGAB=ATAN2(A-B,POSA-POSB)
      ANGAC=ATAN2(A-C,POSA-POSC)
      IF (POSB-POSA+ABS(B-A).LT.BIAS
     1           .OR.
     2    POSC-POSB+ABS(C-A).LT.BIAS)
     3                     ANG2=ANG2*2
      IF (ABS(ANGAB - ANGAC) .GT. ANG2) GO TO 55
C
C *** IT WORKED!!!!! CONCATINATE.
C
      ANG1=ANGX
      ANG2=ANGY
      I=POSB
      IMAGEB(I)=0
      POSA=POSC
      MERGE = MERGE+1
      GO TO 15
C
C *** NOT POSSIBLE TO MERGE-- BEGIN LOOKING AT THE OTHER
C ***                         SIDE OF THE PLATEAU
C
55    POSA=POSC
      ANG1=ANGX
      ANG2=ANGY
```

```
GO TO 15
END
```

```
      SUBROUTINE NEXT (VALUE, POS, ICODE)
C     THIS SUBROUTINE FINDS THE NEXT VALUE TO BE LOOKED
C     AT IN THE IMAGE FILE FOR THE LINE SIMPLIFIER
C     ROUTINE.
C     WRITTEN JUNE3, 1977 BY BRENT LANDER.
C     PARAMETERS: VALUE- VALUE OF THE NEXT INFLECTION
C                        POINT IN THE IMAGE
C                 POS   - POSITION OF THE NEXT INFLECTION
C                         IS RETURNED.
C                          STARTING VALUE OF SEARCH IS
C                          PASSED. INITIATE WITH
C                          0 BEING PASSED.
C                 ICODE -RETURN CODE- -1 END OF LINE
C                                      0 VALLEY
C                                      1 PEAK
C                                      2 PLATEAU
C
C     <><><><><><><><><><><><><><><><><><><><>
      COMMON IMAGEA(256), IMAGEB(256)
      IF (POS .LE. 0) POS=0
      IF (POS .GE. 256.0) GO TO 15
      I2=POS+1
      DO 10 I=I2,256
      IF (IMAGEB(I) .NE. 0) GO TO 20
10    CONTINUE
15    ICODE = -1
      RETURN
C
C *** VALUE NOW FOUND
C
20    POS=I
      VALUE=IMAGEB(I)
      IF (VALUE .GT. 0.0) GO TO 30
C
C *** VALLEY
C
      VALUE=-VALUE
      ICODE=0
      RETURN
C
C *** PEAK OR PLATEAU
C
30    IF (VALUE .GE. 1000) GO TO 40
C
C *** PEAK
C
      ICODE=1
      RETURN
C
C *** PLATEAU
```

```
C
40    VALUE = VALUE-1000
      ICODE=2
      RETURN
      END




      SUBROUTINE MINMX
C     SUBPROGRAM TO FIND PEAKS, VALLEYS, AND PLATEAUS
C    ON AN IMAGE WAVEFORM (IN 1-DIMENSION)
C     BY TRACING ALONG THE PATH FOLLOWED BY A
C     LINEAR PROGRESSION.
C
C     THE VALUE OF THE VARIABLE M IS A MARKING OF THE
C     PREVIOUS TRACKING.
C     DIRECTION. M=0 TRACKING DOWN TOWARDS VALLEY.
C                M=1 TRACKING UP   TOWARDS PEAK.
C                M=3 TRACKING OVER ACROSS PLATEAU.
C
C     THE IMAGE COMES ACROSS IN THE COMMON BLOCK AREA
C     IMAGEA.  THE RETURNED IMAGE GOES BACK IN THE
C     COMMON BLOCK AREA IMAGEB.
C     IMAGEA REMAINS UNCHANGED BY THE PRODUCTION.
C
C     IMAGEB RETURNS THE FOLLOWING VALUES:
C                      1<IMAGEB(I)<255 ---A PEAK IS FOUND
C                                            AT THIS I OF
C                                      MAGNITUDE IMAGEB(I)
C                  -255<IMAGEB(I)<=0  ---A VALLEY IS
C                                       FOUND AT THIS I OF
C                                       MAGNITUDE |IMAGEB(I)|
C                  1001<IMAGEB(I)<1255 --A PLATEAU WAS
C                                 DELIMITED BY THIS POINT.
C                                 PLATEAU IS OF HEIGHT
C                                    IMAGEB(I)-1000
C
      COMMON IMAGEA(256), IMAGEB(256)
      LOGICAL TRACKD, NEXT, NEXT2
C
C *** DEFAULT TO M=0 MEANS TRACKING DOWN.
C
      M=1
```

```
      IMAGEB(1)=IMAGEA(1)
      IF (IMAGEA(1) .LT. IMAGEA(2)) M=0
      IF (M .EQ. 0) IMAGEB(1) =-IMAGEB(1)
      IF (IMAGEA(1) .NE. IMAGEA(2)) GO TO 1
      M=3
      IMAGEB(1)=IMAGEB(1)+1000
1     DO 10 I=2,255
      IF (M .EQ. 3) GO TO 8
      TRACKD = M .EQ. 0
      NEXT = IMAGEA(I) .GT. IMAGEA(I+1)
      IF (IMAGEA(I) .EQ. IMAGEA(I+1)) GO TO 85
      IF (TRACKD .AND. NEXT) GO TO 3
      IF (TRACKD .AND. .NOT. NEXT) GO TO 5
      IF (.NOT. TRACKD .AND. NEXT) GO TO 7
C     .NOT. TRACKD .AND. .NOT. NEXT
C
C *** TRACKING UP AND REACHED PEAK
C
      M=0
      IMAGEB(I)=IMAGEA(I)
      GO TO 10
C
C *** VALLEY - MARK AND SET TO TRACK UP
C
3     IMAGEB(I) =-IMAGEA(I)
      M=1
      GO TO 10
C
C *** GOING INTO VALLEY-- JUST CONTINUE
C
5     IMAGEB(I) = 0
      GO TO 10
C
C *** TRACKING TOWARDS PEAK-- CONTINUE
C
7     IMAGEB(I) = 0
      GO TO 10
C
C *** TRACKING PLATEAU
C
8     IF (IMAGEA(I) .NE. IMAGEA(I+1)) GO TO 9
      M=3
      IMAGEB(I) = 0
      GO TO 10
C
C *** SET UP PLATEAU TRACKING
C
85    M=3
      IMAGEB(I)=IMAGEA(I) + 1000
      GO TO 10
```

```
C
C *** RESET FROM TRACKING PLATEAU
C
9       M = 1
        IF (IMAGEA(I) .LT. IMAGEA(I+1)) M=0
        IMAGEB(I) = IMAGEA(I) + 1000
10      CONTINUE
        IMAGEB(256)=IMAGEA(256)*(-1)**(M+1)
        RETURN
        END
```

APPENDIX E

Index

APPENDIX F

Bibliography

Binford, and Horn, B, Edges in Blocks-World Figures., MIT Research Report, 1975

Ehrich, R.W., A Symetric Hystersis Smoothing Filter for Non-Linear Noise Reduction, to be published in the *IEEE Transactions on Electronic Computers*.

Kirsch, R. A.,"Computer Interpretation of English Text and Picture Patterns," *IEEE Transactions on Electronic Computers*, EC-13, 363-376. August 1964.

Marr, D., "Early Processing of Visual Information", Massachusetts Institute of Technology Artificial Intelligence Labortory A.I. Memo No. 340. December 1975.

Roberts, L.G., "Machine Perception of Three-Dimensional Solids," in *Optical and Electro-Optical Information Processing*, pp.159-197, J.T. Tippett, et al., editors. MIT Press, Cambridge, Mass., 1965.

Sobel, Irwin, Camera Models and Machine Perception, Stanford Artificial Intelligence Project, Memo AIM-121, Computer Science Department, Stanford University. May, 1968.

Waltz, David, Understanding Line Drawings From Scenes With Shadows, *The Psychology of Computer Vision*, Mc Graw Hill. 1975.

The two page vita has been removed from the scanned document. Page 1 of 2

The two page vita has been removed from the scanned document. Page 2 of 2

The Semantics of Edges

by

William Brent Lander

(ABSTRACT)

One of the major problems in image processing is information reduction. The research reported in this thesis examines the semantics of context-independent edges in a real-world environment. Topics discussed include a classification scheme for edges, edge semantics, and a method for minimizing noise effects in edge computations.