Order Number 9020771

Detection of edge points using entropy

Jeanty, Henry Henrick, Ph.D.

City University of New York, 1990

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

DETECTION OF EDGE POINTS USING ENTROPY

BY

HENRY HENRICK JEANTY

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1990

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

1/25/90
Date

Joseph Barba
Chair of Examining Committee

1/25/90
Date

Jacques E. Benveniste
Executive Officer

Professor Joseph Barba

Professor Michael Conner

Professor Sanghamitra Basu
Supervisory Committee

The City University of New York

Abstract

DETECTION OF EDGE POINTS USING ENTROPY

by

HENRY HENRICK JEANTY

Adviser: Professor Joseph Barba

The problem of detecting edges in digital images is an important one in all areas of image processing. In our particular area of interest, namely digital images of microscopic biological specimens, quantitative information such as area and shape is crucial to the analysis and classification of objects. The location of edges can provide this information while minimizing the amount of data to process. An extensive body of literature is concerned with this problem, and part of this dissertation is a survey of this literature.

Edge detection does not seem to have a general solution. Different types of images demand different approaches to solving the problem. Therefore, we describe the specific types of images we deal with along with the problems and constraints associated with them. We perform an analysis of the entropy operator in the cases of noise-free and noise corrupted step and ramp edges. Finally, the methods used in implementing the operator are described as well as final results.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank the people who have supported me during the time of preparation of my thesis and during my studies over the years. To Professor Joseph Barba for many helpful discussion and his patience and support. To Dr. Joan Gil for his helpful input, to my parents for their love and encouragement and without whose support none of this would have been possible. To Renee L. Lucchese, alias Spankey, for her expert preparation and organization in preparing the final document.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure

List of Illustrations - Continued

Figure

List of Illustrations - Continued

Figure

CHAPTER I

I. INTRODUCTION

One of the major goals of artificial intelligence is the understanding of visual scenes. But the amount of information from a single image, be it a natural scene or television frame, is often so large that attempting to analyze an entire raw image can be an exercise in futility. An approach which meets with much greater success is one which attempts to simplify and extract the most important features from images. What is important in an image depends on the application, but there seems to be a group of features that can represent a major part of the information contained in images while comprising only a small part of the total number of picture elements (pixels). These features are called edges. A fact supporting the importance of edges is that in most natural scenes, distinct objects have boundaries separating them from each other and from the background. These boundaries correlate highly with edges. Edges are therefore a natural way to identify the different objects forming a scene. Another piece of evidence supporting the importance of edges is that anatomical structures which respond to abrupt changes in intensity and color both in space and time have been found [32]. Experiments show that absolute colors and intensities are not readily perceived, but rather that abrupt changes are. Another motivation for the use of edges is the relative sparsity of edge points in images. An object can be described by many less points if only its contour points are kept. Despite the loss of certain

types of information (texture, gray level, etc.) much useful information such as area or perimeter, is preserved in the contour of an object. This potentially vast reduction in data is a convincing argument, especially for the implementation of fast recognition and classification algorithms. This reduction in the amount of data is due to the way in which edges are defined. Edges convey information not about a single point, but about the neighborhood of a point. Thus a true edge point should replace a number of points located in a certain neighborhood.

Despite our daily encounters with edges, there really is no clear and unambiguous definition of what an edge is. An edge is one of those entities which we recognize effortlessly and yet are at a loss to describe accurately. The most common definition, very likely based on common sense, usually describes edges as locations of sudden change in the intensity of a scene. Many early attempts at edge location were based on this definition and were for the most part unsuccessful. Only in the simplest of images did this approach prove satisfactory. One important reason for this failure is that this simple definition of edges did not take into consideration the fact that most images are noisy and that edges are usually blurred. An algorithm looking for large differences in intensity is essentially a differentiator followed possibly by a threshold operator. But when one considers the transfer function of a differentiator, one quickly realizes that high spatial frequency components of a signal are amplified relative to the lower frequencies. Therefore, high frequency noise present in the edge is enhanced. Improvements on the

difference method can be found in algorithms which rely on estimates of the gradient. In these methods, suitably defined maxima of the gradient are considered to locate edges. Some methods use the zero-crossings of a second derivative operator and are simple variations on the gradient method. Mathematical definitions of edges try and model edges in terms of step, ramp or roof edges. Algorithms based on these definitions usually try and find the best fit to some edge model and reduce the effects of noise.

II. IMPORTANCE OF EDGE DETECTION IN PATHOLOGY

The purpose of this research is to develop methods to detect the contours or boundaries of important structures from images of biological specimens. These structures are usually cells and their nuclei. Once detected such structures can be quantified and analyzed to provide objective data which physicians can use towards diagnostic purposes.

In the United States, Pathology consists of two branches: Clinical Pathology (clinical chemistry, blood banking, microbiology) which is characterized by a high level of utilization of expensive analytical instruments, and Anatomic Pathology (surgical pathology, autopsies, cytology, immunopathology) in which all diagnoses and determinations are usually made by visual inspection of glass slides by pathologists.

In Surgical Pathology, the tissue diagnoses (generally accepted by clinicians as final and irrefutable proof of the existence or absence of a malignancy and an essential part of surgical practice) are made by a pathologist by visual inspection on the basis of criteria derived from experience. In most instances, this is done easily and rapidly and requires no instrumental assistance, but in a number of cases the diagnosis is difficult, requires additional studies and consultations and different experts may express contradictory opinions. In such cases, computerized image analysis could prove invaluable because discussions with experienced pathologists reveal that in many cases the diagnosis depends on quantitative criteria (applied subjectively) or on definable image features.

The main benefit to be expected from the development of a good edge detecting algorithm is increased reliability of cytologic and histologic techniques.

In the processing of images of histology and cytology specimens, the most fundamental and essential operation is the detection of the nuclear or cytoplasmic boundaries or contours. This crucial observation is borne out when pathologists are asked to describe verbally their criteria for diagnosis. Most pathologists mention subjectively determined quantitative descriptors such as nuclear size, cytoplasmic to nuclear size ratio, nuclear shape, texture of nuclear chromatin and other features all of which presuppose that some

structure's contour or boundary (nucleus, cytoplasmic membrane) has been detected.

III. DIFFICULTIES INVOLVED

Two prominent problems are the extreme variability of histological specimens and trying to detect and separate the contours of overlapping structures (such a two cell nuclei). We are confident that these two problems can be dealt with, but the major problem in our view will be trying to detect and quantify something for which there is no common and agreed upon definition and no general model. This leads to the application of many ad hoc and heuristic methods.

We can obtain a better understanding of some of the methods used in the area of edge detection by turning our attention to various attempts made at solving this problem.

IV. SURVEY OF EDGE DETECTION LITERATURE

In this section, I will present important papers in the area of edge detection. Although this survey does not cover the entire body of literature, it is meant to provide the reader with an overview of the different approaches used in this challenging field. The most obvious characteristic of this survey is a preponderance of heuristic approaches. This seems to be the norm rather than the exception in image analysis. A very likely reason for this is that image processing is relatively young and we don't yet have general and well established models of images. A model used in one area for a particular set of images is unlikely to fit images studied in a different environment.

Given these realities of image processing life, one should not expect a uniform, mathematically oriented approach to edge detection. Instead, one will find that each author uses a methodology and techniques most appropriate to his or her type of images and environmental (processor, amount of data, etc.) constraints. But despite this lack of uniformity one should still expect to recognize certain common mathematical methods such as convolution which plays a major role in the filtering of images.

All these facts are also reasons why the survey doesn't cover as many papers as it could have. We found that many papers simply presented variations on a main approach and do not truly lead to a better understanding of the problem. Instead they show how researchers attempt to improve a

previous approach when presented with the new technical capabilities of their equipment. This is quite evident in Nevatia [1] where color images are essentially treated as a combination of monochromatic images or in Duda and Hart [2] whose approach is a variation on Rosenfeld's [3] method which itself was a variation on Hough's [4] method. Each variation in this last example was meant to better deal with the constraints and capabilities of the day's hardware.

Given the number of different approaches there was no truly consistent way to classify the different methods. Therefore, an attempt has been made at describing general, common approaches such as convolution based methods or heuristics based methods. Papers relevant to a particular method are referenced in the text in order to allow the reader to better follow the different techniques. Finally, important papers are reviewed individually in order to provide a more detailed look at some specific approach.

V. METHODS BASED ON CONVOLUTION

One of the most common techniques used in image processing systems is convolution. Operations such as blurring and sharpening can be performed with this method. Convolution in the spatial domain is equivalent to multiplication in the frequency domain and the essential purpose of convolution is to filter an image. The basic problem in using convolution is to find the masks necessary to perform some particular operation. Some masks such as the Prewitt, Kirsch and Sobel operations [21] are used to enhance edges by using masks of size 3x3. Shanmugan et al. [5] describe an optimal frequency domain filter for edge detection in digital pictures derived by assuming a step edge model. This filter is optimal in the sense that it maximized the filter output energy in the vicinity of edges for a given bandwidth and resolution requirements. The approach used is amenable to digital implementations by Fast Fourier Transform. The transfer function of this filter is very similar to that proposed by Modestino and Fries [11] which yields the minimum mean-squared error estimate of noisy images. Robinson tried using 5 level masks [20] to obtain better results. Unfortunately his arguments concerning the advantages of these masks are not convincing. It seems that the difference between 5 level masks and 3 level masks is simply not substantial. Shaw [54] presents an excellent comparison of some of these masks. A disadvantage of these comparatively small masks is that they are orientation-dependent. This means that convolutions with separate masks must be performed to quantify the direction of an edge [5]. Marr and Hilbreth

[26] detected intensity changes over a wide range of scales and orientations by using the second derivative of a Gaussian as filter.

Despite the advantages of mathematical tractability and a thorough understanding of the frequency domain effects of convolution, a major disadvantage of using convolution is the computational expense involved in performing it with anything but the smallest masks. This expense is now being diminished with the advent of more powerful image processors capable of performing convolutions with masks of size 3x3 in real time. Despite these technological improvements, convolution is not the answer to all the problems associated with edge detection. After all, given a certain mask, the frequency effects are fixed and convolution simply performs a filtering operation. This operation may enhance certain features of the image, but does not perform certain other necessary functions such as actually locating edge points. This is one of the reasons why so many methods used in edge detection are based on heuristics. These methods will be the topic of our next section.

VI. METHODS BASED ON HEURISTICS

Most papers in the area of edge detection describe approaches based on heuristic methods. One example is Nevatia's treatment of color edge detection [1]. Given a color image, the problem is to decide on an appropriate way to detect edges. The simplest of three methods described by Nevatia is to process each individual color component image independently of the others. The other two methods detect edges in each individual color component and then try to merge these edges by satisfying some constraint. Even in Robinson [20] mentioned earlier in the context of convolution based methods, a number of problems pertaining to the linking of edges and selection of thresholds had to be solved on an ad hoc basis. Robinson's edge activity index [20] is another example of such solutions. Another example of heuristics is to be found in Duda and Hart [2]. The authors try to locate lines and curves in pictures by using the Hough transform [4]. The general approach embraced is to parameterize the objects (lines in this instance) to be located. A white background with black pixels is assumed in the paper. The method simply tests whether a pixel belongs or not to a particular straight line defined by two parameters. When a particular line passes through enough pixels, it is assumed to exist in the image. The method is appealing in that any general curve is theoretically detectable by this approach. Unfortunately there are certain practical problems. This approach is useful if the number of parameters used is small and if the parameters are bounded. Another problem pointed out the the paper is that of detecting separate objects having

the same parameters (such as two colinear but distinct line segments). In another approach, Montanari [40] used a dynamic programming method to determine the optimal curve with respect to a figure of merit. Martelli [38,39] had used a similar approach which casts the problem of detecting known shapes in noisy image in the mold of finding the shortest path on a connected graph. A study shows this approach to be a form of edge follower. Essentially, properties of edges are embedded in an objective function and the algorithm locates edges which minimize this function.

An often used feature in image analysis is the histogram of gray levels of a picture. Many heuristic methods are based on the analysis of histograms. One of the principal uses of histograms is in the computation of thresholds. By thresholding an image it becomes much easier to apply many algorithms (such as edge detectors). Weszka [42] presents an excellent survey of thresholding techniques based on histogram analysis. These methods can easily be classified as global, local or dynamic. Global methods use the histogram of the entire image to select a threshold applicable to all pixels, while local methods use the histogram of a neighborhood of a particular pixel to compute the threshold. Dynamic methods use information about neighborhood and locations of a pixel to assign a threshold.

In the following section, a fuller review of many of the papers cited earlier is conducted. Papers are introduced on a new page by the author or author's name, their year of publication, a reference number and their title.

VII. REVIEW OF REPRESENTATIVE PAPERS

Van Vliet and Young 1989 [82]

"A Nonlinear Laplace Operator as Edge Detector in Noisy Images"

This paper discusses an edge detection scheme robust enough to perform well over a wide range of signal-to-noise ratios. The authors note the poor performance of most edge detectors in images with low signal-to-noise ratios. Their edge detection approach is based on the detection of zero crossings in the output of a "Laplace" filtered image. The authors developed an edge detection scheme based upon the Marr-Hildreth model combined with the non-linear Laplace operator proposed by Beckers [83] in 1986. This Laplace-like operator uses a local 3x3 nieghborhood and may perform better than a linear Laplace filter. Furthermore, the authors extended the nonlinear Laplacian to larger filter sizes.

The complete Marr-Hildreth model for edge detection consists of a smoothing filter, a Laplace filter, and a zero-crossing detector in sequence. The authors extended this model with an edge strength detector and a threshold operator. They selected the edge strength detector proposed by Lee [84]. This detector is based on grey scale morphological operators and its inherent sensitivity to noise is much less than the more conventional gradient operators. A detailed analysis of this operator will show a strong parallel with

the nonlinear Laplace operator and includes the possibility of using circularly shaped supports. The threshold operator was not automatic but manually chosen. The authors explain this by the lack of algorithm robust enough for images with low signal-to-noise ratio.

In their evaluation they find that although the zero crossing detector and the threshold operator are essentially fixed, the smoothing filter and the Laplace filter must be optimized as to type and size. Similarly, the size of the edge strength filter is also related to the smoothing size. By optimizing these variable components, the authors generated an edge detector which performed well (using Pratt's [84] figure of merit) even in low signal-to-noise images.

McLean and Jernigan 1988 [***86***]

"Hierarchical Edge Detection"

This paper discusses the design of efficient edge detection operators. In order to quantify efficiency, the authors specify a set of requirements that the operators should satisfy. They are:

- The operator must be capable of working in a purely local context.
- The operators must be efficient when applied in any order: They cannot derive efficiency by exploiting redundancies when applied in a particular fashion.
- The operators must be sesitive to the orientation as well as to the magnitude of the edge.
- The operators, must work well in the presence of noise.
- The operators must be relatively insensitive to threshold specifications.

These requirements led the authors to the concept of local effectiveness. This in turn led them to the development of oriented 1-dimensional operators which can be cascaded together to achieve orientation selectivity. They adapted the Sobel operator, Gaussian smoothed Laplacian operator and two non-linear operators to this 1-dimensional edge detectors were then integrated

into a hierarchical scheme which preprocesses the image with a simple gradient operation before applying one of the more expensive nonlinear operators.

In order to satisfy the third requirement (sensitivity to orientation) the proposed edge operators performed detection by processing a single pixel width cross section of the image, the width being oriented along one of the four directions allowed in the eight connected grid. This scheme also satisfies the first criterion (operator works in a purely local context).

The hierarchical edge detection (HED) uses two steps. The first step is a coarse oriented gradient measurement, followed by the application in a particular orientation of one of the efficient 1-dimensional edge detectors. The first step is then an initial filter which selects only those pixels which exceed the gradient threshold. In this way, the HED scheme becomes very efficient. Because only highly probable edges are processed in the second step, the time required is related to the amount of edge and noise activity in the image. The performance of the gradient preprocessor becomes crucial and the authors selected a simple cross operator, applied over a 5x5 region around the evaluated pixel.

The results are encouraging for they suggest that very effective edge detection can be performed while maintaining a highly structured and localized approach to this aspect of image processing. Because of its

hierarchical nature, the HED scheme is very well suited for inclusion in systems encompassing multiple levels of processing.

***Yanowitz and Bruckstein 1989* [87]**

"A New Method for Image Segmentation"

In this paper the authors present a new method to segment images via an adaptive thresholding scheme. A threshold surface is determined by interpolating the image gray levels at points where the gradient is high. These points of high gradients are potential edge points. This is in a sense similar to Chow and Kaneko's [51] approach insofar as the latter generated an adaptive threshold image based solely on local histograms of the picture (see section on Weszka [42], "A survey of thresholding selection techniques"). But in this new approach, supplementary information from the gradient image is used to obtain a good threshold surface. Furthermore, a validation process is used to remove stains that are due to random variations in illumination. The algorithm can concisely be described as follows:

1. Smooth the image by replacing each pixel with the average of some neighborhood of it. This tends to reduce noise.

2. Obtain the gray level of the smoothed image.

3. Obtain good candidates for local threshold by locating those points having local gradient maxima.

4. Sample the smoothed image at those candidate points obtained by step 3.

5. Through some interpolation method, interpolate the samples gray levels over the image. This generates the threshold surface.

6. Use the threshold surface obtained in step 5 to segment the image.

7. This is the validation process which eliminates spurrious edges generated by uneven illumination and/or noise. This step is used after all objects have been labeled. The average value of the edge values (obtained from the gradient map derived earlier) doesn't exceed a certain threshold, the validation process eliminates it.

A major choice to be made here is in the kind of interpolation to be performed. There are two general approaches. In the first approach, one may be satisfied with an interpolation which simply approximates the given data points with minimal approximation error [88, 89]. The advantage of this approach is the reduced storage space required for the interpolation. The second approach is to require that the interpolation surface pass exactly through the given data points. This requires as much data storage space as a whole image does. The authors selected the second approach. They used a procedure which nulls the Laplacian of the surface P(x,y).

$$\frac{\delta^2 P(x,y)}{\delta x^2} + \frac{\delta^2 P(x,y)}{\delta y^2} = 0 \qquad (1)$$

By nullifying the Laplacian (1), they effectively generate a potential surface for which the divergence of the gradient is everywhere insured that the surface to be smooth enough for their purpose.

Shanmugam, Dickey and Green, 1979 [5].

"An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures".

This work's goal is to derive an optimal edge detection filter. One advantage of the derived transfer function is the full control of the product of resolution and bandwidth of the filter which can be set to any desired value. Another advantage is that a single parameter of the filter can be adjusted in order to trade resolution for output signal-to-noise ratio.

The optimality criterion of the derivation of this filter is the maximization of the filter output energy in the vicinity of edges for a given bandwidth and resolution requirements. The derivation was performed by assuming a step edge but is shown to be near optimum for a large class of blurred edges if the width of the resolution interval is chosen to be larger than the width of the blurred edge.

The transfer function of the optimal filter was found to be specified in terms of the prolate spheroidal wave function [6,7 and 8]. Sampled values of the first-order prolate spheroidal wave function provided the best results. Slepian [9] showed that using a closed form asymptotic approximation of these functions would be suitable. This approximation involves nth Hermite polynomials and Striefer [10] showed that the error involved in the use of such approximations to be quite acceptable. Both approaches are amenable to

digital implementation through the Fast Fourier Transform although the use of asymptotic approximations yielded the easiest digital implementations. An interesting fact found in the search of this optimum edge detection filter is that the transfer function of this filter is similar to that of a filter proposed by Modestino and Fries [11] which yields the minimum mean-squared error estimate of noisy edges. This suggests that there might possibly exist a single filter which yields the best resolution and signal-to-noise ratio for a given bandwidth.

***Shaw 1979* [*54*]**

"Local and Regional Edge Detectors: Some Comparisons"

This is an excellent paper on the art of understanding and comparing the claims made about the plethora of edge detectors. The author limits his study to the following edge detectors:

a) Heuckel edge detection algorithm [12,13]
b) Sobel operator with least-square goodness of fit [14]
c) Mero and Vassy's approximate Heuckel operator [15]
d) Hummel's optimal local template [16]
e) A new discrete Heuckel-like operator which he describes in the paper.

The author does qualify his approach by noting that certain types of edge detectors such as b), c) and d) are purely local while others like a) and e) are intended to work only in overlapping regions. These different types of detectors could not directly be compared. A way out of this impass was to test each operator against itself as various types of distortion were applied to the test image. The resulting degradation in edge response was then used as a measure of stability and discriminatory power of the edge detectors.

Edge detectors are classified as being of two general types: Local and Regional. The local edge detectors are characterized as being differentiators

of some type or another which measure the "edginess" of a pixel while regional edge detectors are expected to give good responses in regions which can be described as being divided into two regions of fairly constant gray levels. A mention is made of directional local edge detectors which tend to measure gradient values at a pixel.

The author used sets of synthetic images of perfect polygons and a "real" image of a girl's face. The synthetic images gave a measure of the performance of the different edge detectors to long straight lines while the "real" image gave a measure of the responses to textured and curved edges. Another part of the comparison was to use the detectors on perfect versions of the test images and on distorted versions of these images. The distortions introduced where wide-spectrum (approximately Poisson), uncorrelated noise, edge blurring and imposition of an overall linear ramp on the entire image. These types of distortions are all fairly common and interfere with the detection of edges.

The conclusion was that all operators were capable of locating edges in their domain. The Sobel operator performed uniformly better than the other local operators. The two regional operators had similar performances, but the regional operators might be given an edge (no pun intended) over local operators because they tend to provide their output in a more compact form than the local operators. This is a consequence of the fact that regional operators can be thought of processing their output and provide some

goodness-of-fit value. The computations needed to obtain this same information would probably be more expensive than the application of a local edge detector.

Nevatia 1977. [*1*]

"A Color Edge Detector and its Use in Scene Segmentation"

Nevatia investigates the problem of edge detection in color images. Three approaches to color edge detections are described.

The first approach is to define a metric in color space and then define edges as discontinuities in this space. The author suggests that the definition of such a metric could be based on experiments with humans [18]. This approach reduces the color edge detection problem to an edge detection in one dimension and the results should not be too different from that of edges found in equivalent achromatic images.

The second approach is the independent detection of edges in each of the three color components followed by the merging of these edges into a single edge by some specified procedure.

The third approach which is the one developed by the author consists as in the second one in independently locating edges in the different color components of the image, but to impose some uniformity constraints in order to use the three edges concurrently. It is this constraint which directly affects the computation of the three components and makes this approach different from the second one.

The achromatic edge detector used by the author was the Hueckel operator followed by an edge linker described in detail in [19] Many of the features of the edge detected derive from the properties of the Hueckel operator and the edge linking algorithm used. No conclusive evidence to the superiority of color edge detection over monochromatic edge detection is presented. It was found that edges detected in the chromatic components were often embedded in the luminance component although they might be more difficult to extract from the latter. One suggestion was that color edges might make a system more robust and that using the co-occurrence of edges in the luminance and chromatic components would decrease the number of spurious edges. An obvious generalization of the scheme to multispectral images such as LANDSAT images which use four spectral bands or aerial photographs which can use up to 12 spectral bands is mentioned.

Robinson 1977 [***20***].

"Edge Detection by Compass Gradient Masks"

It is assumed that the reader is familiar with operators such as the Prewitt and Sobel operations [21].

An edge detection system to detect and code visually significant edges in natural images is developed in this paper. The author quickly mentions three-level simple masks such as the Prewitt, Kirsch and Sobel operators. These operators are called three-level masks because they use only 3 distinct integers. The following two orthogonal masks are introduced along with 6 other five-level simple masks:

$$Mx = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} \qquad My = \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix}$$

These masks approximate the partial derivatives in the x and y directions. Some of their advantages are:

a. The direction of an edge is easily found from the output of the first four masks described by the author in his paper.

b. Five-level simple directional masks compensate for the lower visual acuity in the diagonal direction by yielding a higher output for diagonal edges.

A gradient picture is obtained by finding the maximum gradient magnitude at each pixel. The direction of the edge is given by the mask which generated the largest output. In order to be consistent with the Freeman [22] chain coding scheme and allow for subsequent coding of extracted boundaries, the numbers from 0 to 7 are used to describe the eight directions in a 3x3 grid. Once the edge map has been computed, a rule is used to connect edges. two edges are connected if the direction at the center of a 3x3 grid is k ($k=0,\ldots,7$) and if the direction of the preceding and succeeding edge vectors are k-1, k or $k+1$ (mod 8). The gradient value, edge direction map and threshold map are used simultaneously to determine whether an edge is present or not. The author found that a fixed threshold gave the best results but that in the case of faint edges, significant edges were lost. It was also found that when the threshold was set too low, too many edges were found. The author's solution was the edge activity index (EAI) defined as the ratio of the maximum gradient magnitude at a pixel to the average magnitude of the gradients in eight compass directions. The EAI is set to 0 when the eight directional gradients are equal. One method used to find this threshold used the histogram of the gradient image. The mean value of this histogram was one choice for the threshold. Another method made use of a locally adaptive threshold (LAT). This threshold was defined as the ratio of the maximum gradient magnitude to the output of a low-pass filtered (blurred) version of the original image. The mask used for the low-pass filter was

$$MO = \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$$

The rationale behind the EAI is based on the premise that in regions where there is no preferred orientation, the EAI will be 0, whereas in regions where the edge activity is considerably superior in some direction, the maximum gradient will be taken. Unfortunately, no convincing argument is made for the use of five-level rather than three-level simple masks. The author even notes that if a fixed threshold is used, the Kirsch and five-level simple mask give similar binary edge maps. The use of local connectivity of these edge maps improves the performance of all the directional masks operators.

Duda and Hart 1972 [2]

"Use of the Hough Transformation to Detect Lines and Curves in Pictures".

The author addresses the recurring problem of finding straight lines in digitized images. The algorithm described here finds its roots in a method due to Hough [4] to locate colinear points. This original method was described by Rosenfeld [3] who replaced the initial problem of finding colinear points by the mathematically equivalent one of finding concurrent lines in a parameter space. Hough parameter space was the familiar slope-intercept parameters. One of the disadvantages of the resulting two-dimensional slope-intercept space is that it is unbounded and this complicates the application of the procedure. The authors thus present an alternative parameterization which eliminates the problem of unbounded parameter space. The new parameterization is the so-called Normal parameterization which specifies a line in terms of the angle (Theta) of its normal and its algebraic distance (Rho) from the origin. By restricting the angle to the range [0-Pi], lines are represented by a unique set of parameters and a line is represented by a unique point in Theta-Rho space. The problem of detecting colinear points in picture space becomes that of finding concurrent curves. The implementation of the algorithm uses a two dimensional array of accumulators to count the number of curves passing through a particular point in the picture. The accumulators are actually counting the number of points (x,y) which belong to a line with normal parameters Rho and Theta. A given column in the Theta-Rho accumulator

array is just a histogram of these counts. A consequence of this fact is that accumulators with high counts correspond to sets of nearly colinear points in the picture. One of the limitations of this approach is that it is sensitive to the quantization of both Theta and Rho. Increasing the resolution provides better results but increases computation time. Another problem is that this method locates sets of colinear points without regard for continuity. Two colinear but separate lines would be grouped as a single line by this method. One of the interesting by-product of this approach is that it points to the fact that any parameterization could be used. These alternative parameterizations could then be used to locate circular configurations of points in an image for instance. The conclusion is that any type of curve can be located with the proper parameterization. Parameterizations with bounded parameters would be preferable because the implementation of the accumulator array requires quantization of the entire parameter space and the computation grows exponentially with the number of parameters.

Marr and Hildreth 1979 [*26*].

"Theory of Edge Detection"

An approach using analytic and symbolic methods is described in this paper. Unlike most approaches which use local neighborhood operators, this method makes use of significantly larger convolution masks and provides as output a set of connected edge contours. This obviates the needs for further thinning and linking procedures. The intensity changes of an image are described using a primitive language of edge segments, blobs and terminations. This description was called the raw primal sketch. More abstract tokens were then constructed by making explicit certain geometrical relationships present in the raw primal sketch. This secondary representation was called the full primal sketch of an image.

The main analytical problem for the authors was the representation of intensity changes and localization of edges. In order to reduce the effects of noise and minimize the complexity of computations, a nondirectional Laplacian operator was convolved with an image and the zero-crossings of the output were used as localizers of edges. Since the Laplacian is roughly bandpass, only a portion of the spectrum of the image is analyzed. The operation was performed at different scales in order to locate edges appearing at different scales. The authors suspected that this representation might possibly be complete based on a theorem by Logan [27]. This would imply that the zero-crossing representation would provide

enough information to recover arbitrary intensity profiles. A point overlooked by the authors is that the theorem is only valid for one dimension while their signals are two dimensional. Buxton [28] has shown mathematically that the Marr-Hilldreth operator is the optimal operator for detecting ideal step edges in terms of zero-crossings. But this is in an ideal case. As soon as the transfer function is bandlimited by truncation it is no longer optimal. Slepian and Pollack [29], Landau and Pollack [30,31] and Shanmugam et al. [5] show that the optimal transfer function which is zero outside of a given interval is expressed in terms of a prolate spheroidal wave function. A drawback of this operator is the high density of zero-crossings obtained when it is applied to very noisy images. Canny [33] has shown that by maximizing the output of an appropriate operator a better performance can be obtained. Marr and Hildreth also cover some of the discoveries about human early visual processing mechanisms which helped to motivate their approach. An interesting note was about one practical reason why edge detecting methods have been for the most part unsuccessful. They surmise that it is a consequence of the use of only 10 to 30 pixel neighborhoods in current image processing algorithms in contrast to the smallest of Wilson's [34] four psychophysical channels which have receptive fields that cover about 500 foveal cones. Rosenfeld and Kak [35] had discussed the rather ineffective use of the Laplacian. The reason for this being that the Laplacian is not very effective unless one uses it in a band limited situation and then uses its zero-crossings. These ideas did not appear in the previous literature. The field of computer vision

had to wait for the human stereo vision theory of Marr & Poggio [36] for the idea of using narrow bandpass differential operators. this theory was also the first to primarily rely on zero-crossings.

The second part of the authors' method deals with the parsing of the zero-crossings information obtained. Three types of constructs are made:

a. Isolated edges

b. Bars

c. Blobs and terminations

Isolated edges are obtained by a selection criterion according to which the edge is obtained from the smallest channel to which the intensity change is indistinguishable from a step function. Information about edge amplitude and width are then obtained from the zero-crossings provided by that channel.

In certain cases edges run parallel and form bars. In these cases multiple channels are involved, but channels which involve wavelengths larger than about 2w, where w is the width of the bar, cannot be relied upon to provide accurate information about edge length, width, position or contrast. In these cases the information from the smaller channels is used to generate a symbolic line segment with parameters more reliable than those associated with each individual segment.

Finally, parallel segments sometimes merge or are joined by a third segment or form closed curves. These structures are reflected by anomalous effects in the larger channels and are best dealt with early. Marr [37] called the closed contours BLOBS and characterized them by length, width, orientation and average contrast while terminations were associated with positions and orientations.

Martelli 1972 [***38***], ***1973*** [***39***]

"Edge Detection Using Heuristic Search Methods"

Unlike most edge detection methods based on Laplacian or gradients, the edge detector presented by Martelli is based on a "shortest path" perspective. By embedding the properties of a curve in a figure of merit, Montanari [40] had proposed a method for detecting curves in noisy images. A dynamic programming approach was then used to determine the optimal curve with respect to the given figure of merit. Martelli had himself used this approach in [38,39] to detect edges. The author recasts the problem of finding the shortest path on a connected graph. One of the advantages claimed by the author is that known shapes can be so detected in very noisy images. A characteristic of this method is the increase of computation time with noise level. By using the A* algorithm described in the paper, the author claims that substantial reduction in computation time can be obtained. The A* algorithm can be seen to be an edge follower as it starts with the first element of the edge and finds the next most promising edge at every step. The problem is cast in the proper mold by embedding the properties of an edge or contour in an objective function. The problem of extracting these edges and contours from noisy images is then the minimization of this function. A cost is assigned to every possible edge. This cost takes into account local properties such as gray levels and curvature. The total cost would presumably contain global information. The minimization of this total cost is approached as a graph search. One

of the possible drawbacks of this method is that characteristics of the goal nodes along with the rules for expanding a node and for computing costs must be embedded in the algorithm and have to be changed for different types of edges or contours sought. When this type of heuristics is introduced in the algorithm search speed is increased, complex shapes can be searched for without significantly affecting the complexity of the search, but the solution is no longer optimal.

***Weszka 1978* [*42*]**

"A Survey of Thresholding Selection Techniques"

This is an excellent survey of thresholding techniques based on histogram analysis. Different methods used to compute appropriate thresholds are described. The stated goal of all these methods is to segment an image into regions which can later be analyzed on the basis of properties such as area, perimeter, shape, etc. Another reason for the thresholding process is the reduced storage requirements for the thresholded image as compared to the original one. The author classifies the different schemes as global, local or dynamic thresholding methods.

Global methods which select a threshold based on an analysis of the histogram of the entire image are represented by such methods as the "p-tile" method of Doyle [43] in which a threshold is computed in such a way as to map at least q% of the gray levels into the object. This can work only if the area of the object to be found is known and does not vary from image to image. Another global method is the "mode" method of Prewitt and Mendelsohn [41] in which the thresholds are placed at the minima between the peaks of a smoothed histogram. The rationale behind this method being that edges would have far less gray levels than the interior of either an object or the background and should therefore be represented by the valleys between the modes of the histrogram. Some global historgram selection techniques are based on statistics of local properties.

Some often used properties are obtained by applying a "LaPlacian" or gradient operator to the image in order to improve the shape of the histogram and make it easier to identify the gray levels associated with edges. The general goal of these operators is to generate a bimodal histogram in which the mean is selected as a threshold. If the histogram of the filtered image is unimodal, the mode can be selected as a threshold should give similar results to those obtained by using the mean in the bimodal case (Weszka and Rosenfeld [45]).

Local methods to compute thresholds depend on both the gray level of a point and some properties of the neighborhood of the point. Some of these methods rely on linear threshold operators (Bartz [46]) where a threshold is computed based on the average contrast over previously analyzed sections of the image. Similarly generated thresholds can be applied locally or to large regions. Ullman [47] described another technique to compute thresholds based on the gray levels of the neighborhood of a point. In this scheme certain points of a 5x5 neighborhood of a point along with two rules are used to select a threshold. The whitest gray level decides which rule will be selected. Morrin [48] used two-dimensional plots of gray level versus gradient value to select thresholds. Panda [49] used a scheme where the threshold applied to a point depended on the gray level as well as the edge value of the point. Trimodal distributions obtained from the plots of frequency as a function of gray level and edge value gave rise to several segmentation

procedures. One such method applied by Katz [50] uses the mean gray level of those points having high edge values as a threshold. Another scheme searched for valleys in the gray level histogram of points having low edge values. The best schemes were those which used a combination of the valley search and mean gray level methods.

Dynamic threshold selection techniques are those in which threshold values depend on the gray level, some properties of the neighborhood of a point and on the location of the point in the image. Such a method was used by Chow and Kaneko [51] to detect boundaries in radiographic images. Images were subdivided into overlapping windows of size 7x7 for which the histogram was computed along with the variance of the gray level. The histograms were modelled as one or two normal distributions depending on whether the histogram was unimodal or bimodal. Minimum error thresholds were selected for those windows whose mixture distributions satisfied a bimodality test. This is because these windows contained both background and objects and should therefore be close to the boundaries of objects. Once the thresholds had been computed for each window a linear interpolation scheme was used to select the threshold to be applied to each pixel of the windows. Interpolation was also used to find thresholds in windows which had bimodal distributions. This made the scheme truly dynamic since a pixel's threshold depended on its gray level, local statistics and its proximity to boundary points.

VIII. TYPICAL IMAGING TOOLS USED IN EDGE DETECTION

1) Image Preprocessing

Noise is probably the most common problem encountered in image processing. One of the first tasks to be performed before processing can even start is to "clean up" the image. This usually involves smoothing and noise reduction.

Although there are different types of "noise", there are some characteristic types of noise associated with pictures. Electronic sensors generate uncorrelated, additive noise also known as "snow". Unlike the small changes in pixel values observed in natural scenes, this type of noise can generate large changes in neighboring pixels. A method by Nathan [78] detects these large variations and replaces the affected points by a local average. Numerous nonlinear methods can be used, but an improvement of the order square root of (N) in signal to noise (S/N) ratio can be expected with simple averaging using N neighbors if the noise process is independent and identically distributed. Another nonlinear method is median filtering [79] which reduces noise without much image degradation. This method replaces the center pixel of a given region with the median of the values in the region.

Sometimes, the noise pattern is not random, but structured. One common example is the line pattern of TV scan lines superimposed on an image [80]

In such cases, the spectrum of structured patterns can be analyzed and the offending frequencies can be removed by frequency domain filtering methods.

2) Region Growing

Region growing is another useful technique. It consists in starting with a point within a region and then "growing" the region by grouping all neighboring points with similar properties. A good example of this method can be found in [57]. The basis of region growing is to somehow quantify the similarity of two pixels with respect to some criterion. We once again feel that the entropic image may provide a simple way to quantify this similarity value.

3) Template Matching

In [52] we describe an edge detector for binary images based on the recognition of edges in patterns of size 3x3. This is simple template matching. We intend to use this method after thresholding. Thresholding results in a binary (2 level) image. Therefore there are 2∧9 (512) patterns possible for any region of size 3x3. This number is manageable enough for a computer to store in memory all of the patterns representing edges. A disadvantage of this method is that 3x3 is the largest practical size for the templates. A template of size 4x4 would create 2,16 (65536) possible patterns. Yet this method is very flexible in that by storing different tables with different entries, different types of edge patterns can be detected. One table can be generated to detect fine lines while another might detect coarse edges and ignore isolated points. This

type of operation is called a Hit or Miss Transform (HMT) in Morphometry, a subject we will discuss later.

4) Correlation Methods

Certain types of patterns occur fairly often in medical images. This is due to the fact that although variability is common there are nonetheless recurring cell types with distinctive shapes or coloration. Cross-correlation between a region to classify and a number of models of master templates can help in determining the class of which a region should be assigned [81,82]. This scheme might allow a region growing algorithm to detect when a particular region's boundary has been reached. This would occur when the cross-correlation between the region under test and some master template falls off below a certain threshold. This information would in turn localize the general position of a gray-level, color or textural edge.

5) Convolution Methods

Convolution is one of the basic operations used in image processing to filter, smooth or sharpen images. One of the advantages of convolution is that, together with Fourier analysis, we can view an image either in the spatial or frequency domain. Also, despite the computational complexity involved in using it with large masks, we now have hardware capable of performing convolution with reasonably sized masks in real or near real-time. Unfortunately, convolution by itself cannot detect a cell or a nucleus. It can

however pave the way for other algorithms such as an edge follower by enhancing edges. We can also use convolution to remove some of the structured type of noise mentioned earlier.

6) Morphological Methods

Some of the techniques we have used [52] are based on a field called mathematical morphology. Since most of the work in that area is not well known to engineers, let us first give a brief introduction of mathematical morphology.

Morphology theory was developed out of work in the field of cellular automata and is a synthesis of many mathematical systems, including integral geometry, statistics and topology. G. Matheron and J. Serra developed the first theoretical notions of morphology in the 1960s [23,24]. Very notably, Matheron used morphology to investigate the relationship between the geometry of porous media and their relative permeabilities. A texture analyzer [25] based on the principles of mathematical morphology was developed by Serra and J. Klein. Although morphology theory was first applied only to binary images, much effort is now being applied to extending the theory to gray level images.

In mathematical morphology, the values of pixels are changed as a function of their neighboring pixel values. This is somewhat analogous to convolution, but while convolution uses additions and multiplications, morphology uses

point transformations such as union, intersection and geometric transformations such as translation and reflection. Morphology also uses measurements related to transformations such as the number of changed and unchanged pixels between applications of transformations. The Hit or Miss Transform mentioned earlier is one of the fundamental operations performed in morphology. In this operation, a binary template is scanned over an image in a manner similar to that performed in convolution and matches are registered by setting the center pixel of the pattern. Other operations such as dilations and erosions are generated from HMT's. Dilation expands objects while erosion shrinks objects. Many similarities between these operations make their hardware implementation straightforward and image processing companies are now starting to include these operators in their machines.

However, despite being very useful, morphology requires quite a bit of preprocessing (filtering, smoothing, thresholding) before it can be used and must therefore be treated as an additional and complementary tool in image processing.

IX. BASIC APPROACH

Our main focus is the detection of cells in histological and cytological specimens. In our particular case the nucleus of cells is the most important regions to study since the size, shape and texture of chromatin contained in the nucleus provide much information about the status of a cell. The problems of edge detection and segmentation are actually the same since we are simply trying to locate two regions which are somehow different. Whether we locate the region itself or the boundary of the region is equivalent.

The basic concept we use to determine the location of edges and of regions of interest is that of entropy. Entropy is defined as:

$$H = -\Sigma [P(G) \text{ Log } P(G)] \quad G = 0,1,2,\ldots,255 \qquad (2)$$

and can be interpreted as a measure of the lack of knowledge we have about an event. The higher the entropy, the less we know about an event. The premise is that regions about which we know very little are considered interesting. On the other hand, regions about which we know much add little information about the image and present little interest. If we take the histogram of gray levels of a region of an image, we can compute the entropy associated with the distribution of gray levels as:

$$H = -\Sigma[P(i) \log P(i)] \quad i = 0,1,\ldots,255 \qquad (3)$$

where P(i) is the probability of occurrence of gray level i.

Regions of fairly constant gray level have low entropies while regions with more uniform distributions have high entropies. This simple observation points to the application of entropy to threshold histograms [44]. But the use of entropy is not limited to analyzing only the gray levels of an image. It is quite possible to compute the entropy associated with other characteristics of an image, such as co-occurrence matrices or textural descriptors. An image could be segmented in terms of interesting and uninteresting regions, where the level of interest would be defined not simply in terms of the distribution of gray levels, but also in terms of the distribution of other parameters. This leads to the idea of a generalized edge described as a transition between two regions with different values of interest. A new image which we shall call an entropic image can be obtained from a gray level image. This entropic image can be generated by computing the entropy (with respect to some parameter) of regions of the image. The entropic image provides a simple way to represent the values of interest of different regions of a gray level image WITH RESPECT TO DIFFERENT PARAMETERS. In some instances, information of interest might be localized in the absorption image (an image representing the amount of light absorbed by the specimen). Computing the entropic image with respect to the absorption level in a particular color or combination of colors (i.e. hue, saturation, luminance, etc.) will allow for the detection of interesting regions in those particular features. From this point on, the term entropic image will refer to an image computed by replacing entire regions with the value of the entropy computed by replacing entire regions with the value of the entropy, computed with respect to some feature in a neighborhood

of those regions. Those regions could be as small as an individual pixel, in which case the regions used for the computation of entropy would be neighborhoods of these pixels.

By using this approach [52], we have obtained some encouraging results. One of the advantages of the entropic image was that fairly large regions (12x12 to 16x16 pixels) can be described in terms of their entropy (a single number) and the original image of 512x512 pixels was transformed into an entropic image of size about 32x32 entropy values. These entropy values were enough to quickly locate areas of interest in the image. Another advantage of entropy is that its value is obtained from a histogram. The histogram of gray level values is one of the most common features available on current image processors. Some processors can compute histograms of specified regions in real time.

The entropic image is not the final solution to the problem and must itself be processed. But unlike typical gray level images whose values can change erratically, entropic images should have slowly varying values and should be easier to process. This is a consequence of the fact that the histogram of an image doesn't change much as one moves from one pixel to a neighboring one.

Chapter II: Analysis of Entropy Operator

I. Image Characteristics and Acquisition

Before we proceed it will be informative to quickly describe some of the more important characteristics, problems, constraints and a priori information associated with the processing of these particular images.

a) IMAGE ACQUISITION

The images we processed are monochrome images obtained by digitizing an analog signal. This signal is from a black and white camera with 57 db signal to noise (S/N) ratio. The digitization is done on a grid of 512x512 picture elements (Pixels). The value of each pixel ranges from 0 (darkest) to 255 (brightest) and requires 8 bits (8 bits = 1 byte). The images we will analyze are images of biomedical slides (Histology and Cytology specimens) mounted on a microscope. The magnification is anywhere from 10X to 100X. These images generally offer good contrast but also tend to be defocussed in certain areas. Also very common are artifacts (dust, smearing...) due to specimen preparation.

b) CONSTRAINTS

A major constraint is processing time. If we are to detect the edges of nuclei or cells on a slide, we must do it in amounts of time similar to that used by physicians to warrant the use of such a system. But given that more powerful image processing equipment appears yearly, we did not concern ourselves with meeting stringent time requirements.

c) A PRIORI INFORMATION

The prominent feature of the type of images we deal with is that we most often look for cells and/or nuclei. Given the structure of cells and nuclei we can surmise that we will mostly be trying to locate objects with fairly regular (although not always) and closed contours. The approximate size of these structures is also be known as a function of magnification, tissue type, etc.

d) Typical cytological images and the problems they present.

Very typical examples of the type of images we have to process can be seen in Figures 47a, 48a and 49a. These images are contrast enhanced versions of the originals. The enhancement operation simply consisted of linearly stretching the range of gray levels of pixels so that it went from 0 (totally dark) to 250 (very bright).

Fig. 47a shows one of the better images we have to deal with. The contrast is q uite good as the three sections we are interested in (background, cytoplasm and nucleus) form three fairly distinct regions. Yet one can notice a number of artifacts in the picture. These artifacts can be broadly classified in two categories:

a. Artifacts due to specimen preparation. Typically they are represented by smears, streaks and blobs seen in the image.

b. Artifacts due to the inherent texture of the specimen being studied. Most often, with cytological specimens, we have to deal with the texture of the chromatin in the nucleus (Figs. 49a, 47a and 48a represent increasing levels of such artifacts). But very often a substantial amount of texture is present in the cytoplasmic area (Fig. 48a is a good illustration of that problem).

Another problem with such images is the sometimes poor contrast between background and object (Fig. 49a).

This list of characteristics is short, yet it is very representative of the problems we have to solve. As such these characteristics more or less dictate the path we must follow in order to solve the problems associated with them. This is an important fact for it implies that the solution we find for our particular type of image may not be appropriate for others. The converse is also true. Solutions that have been found satisfactory for other

images simply are not well suited to our images. We will show this in the next chapter as we provide the results of the two most common and available edge detection methods when applied to our images. In fact it has been our experience that each image type represents a specific problem and that there simply may not be a simple, general edge detection algorithm.

Images of faces have different characteristics from LANDSAT images or from X-ray images or from CATSCAN images. As such the type of preprocessing used in each area and the choice of edge detector subsequently used would be different. In fact, the types of edges found in these images have different characteristics and warrant different approaches.

We therefore warn readers who expect to read about a general approach that they may be disappointed. Our intent is not to propose a general solution to the problem of edge detection, but rather to add another tool to be used by members of the image processing community. Perhaps others can use this tool as is or in conjunction with other methods.

II. Performance of the Most Common Edge Detectors on Cytological Images

The problems mentioned above are few in numbers, but are difficult to deal with in general and make the application of usual edge detecting algorithms very dissapointing.

Figs. 56a,b,c represent the results of applying a Laplacian operator to the images shown in Figs. 47a, 48a and 49a which are contrast enhanced. The results are totally disappointing. Yet the Laplacian is one of the better and most commonly used edge detectors because of its rotational invariance. The most obvious problems are the spurious edges associated with practically every artifact and the total lack of visible continuous edge. One can barely make out some of the parts of the cytoplasmic membrane in the case of Fig. 56a. In the case of Fig. 56c the entire nucleus stands out and could be made out as a consequence of the high contrast between the nucleus and the cytoplasm, but only the lower right (South-East) boundary of the cell can be guessed at due to the relatively high contrast of that region in the original image (Fig. 49a).

The Laplacian is only one of the many differential operators that can be used to detect edges. We also investigated the results obtained when using a gradient edge detector. The masks we used were the directional Sobel edge operators. Figs. 58a,b,c,d,e and f are the results obtained when

applying Sobel edge operators on the image of Fig. 47a. The Sobel operators used where detectors for edges in the following directions: North (Fig. 58a) North-West (Fig.58b), West (Fig.58c), South-West (Fig.58d), South (Fig. 58e) and South-East (Fig. 58f). There is an improvement on fig 56a but still quite unsatisfactory. There are still numerous spurious edges associated with our artifacts. Another problem with the use of directional masks such as the Sobel edge detectors is that masks attuned to different directions (up to 8) must be applied separately and their results combined to obtain an intermediary edge image. This edge image must then be further processed to generate the edge map which represents the final selection of pixels considered to be on an edge. The typical approach is to threshold the intermediary edge image. The problem is that threshold selection techniques are far more an art than a science and are very much ad hoc procedures.

In all fairness though, one must say that no operator will perform well on such raw images as Fig. 47a, 48a and 49a. They need to be preprocessed before edge detectors can hope to perform usefully on them. For instance, Marr and Hildreth [26] determined edges by first smoothing the image with a Gaussian filter and then taking the Laplacian of the resulting image. Modestino and Fries suggested the use of Wiener filtering for detecting edges in noisy pictures [17]. Shanmugan et al. [5] used an optimal two-dimensional operator approximated by the Laplacian of a Gaussian.

All these approaches are testimony to the fact that the theory is still being developed and that the approaches are rather ad hoc.

Despite this, we agree that the images must be preprocessed and by this we mean smoothed, in order to remove the kinds of artifacts that plague our images. We have seen that in the type of images we deal with (cytological specimen), the major problem are artifacts such as the ones seen in nuclear and cytoplasmic areas of the cell depicted in Fig. 48a.

The problem is now to find the smoothing method best suited for our kind of images. One method is simple neighborhood averaging in which the gray level of each pixel of an image is replaced by the average gray level in the neighborhood of our target pixel. Due to its ease of implementation, a 3x3 neighborhood is often used. Unfortunately, this is not enough in our case as many of the artifacts are larger in size than 3x3. The solution might be to use a larger averaging neighborhood, but the blurring effect is strongly proportional to the size of the neighborhood used. The kind of neighborhood size that would smooth out our artifacts would also blur the edges we want to locate. The blurring effect can be reduced by using a thresholding method by which only regions with small variations are blurred. This is again unsatisfactory in our case since we often have edges located in regions of small contrast (Fig. 49a). Also, the introduction of an ad hoc step (what threshold should be chosen ?) should be avoided if possible.

Another possibility is to use lowpass filtering. Unfortunately, lowpass filters can have sharp transitions at their cut-off frequency and may not pass enough of the high frequencies forming many edges. In fact lowpass filtering will remove spurious effects but very often at the expense of reducing the sharpness of edges we would like to keep. As such lowpass filtering is more of a cosmetic process.

Using a Gaussian filter as Marr and Hildreth [26] propose means approximating a Gaussian filter by some mask and convolving the image with it. This is a time consumming operation for all but the smallest of masks, generally 3x3, which then are poor approximations.

What we need is a smoothing operator that is simple to implement, fast and will not blur the edges we are interested in. A very good candidate is median filtering in which we replace the gray level of each pixel by the median of the gray levels of some neighborhood of that pixel. This filter is extremely good when edge sharpness is to be preserved. Being very non-linear, the analysis of median filters is difficult at best, but one can see that the principal property of median filtering is to force pixels which that have values quite different from their neighbors to have values closer to that of their neighbors. By changing the size of the neighborhoods used, one can control (although by trial and error) the size of artifacts that are smoothed out. Figures 47a,b, 48a,b and 49a,b represent original, contrast enhanced images (Figs. 47a, 48a and 49a) along with their median filtered

versions (Figs. 47b, 48b and 49b). The size of the neighborhood used was 15x15. One can notice how the artifacts were smoothed out, particularly in the case of Fig. 48b, as well as how the contrast between the different regions (background, cytoplasm and nucleus) was improved.

Another advantage of median filtering over other smoothing methods that might give similar results is the ease with which it can be implemented. With the advent of more and more sophisticated image processing hardware, we now have image processors which can compute the histogram of an entire image in 1/30 second, which is one frame time. This means that the histogram can be computed in real time. Along with this capability, some can also compute the histogram of any part of the image in less than 1/30 second. It is a simple matter to obtain the median of a neighborhood from its histogram. There are even now digital chips which compute the median of a set of values given them. But the property that the median is obtained from the histogram will prove very useful later since our entropic operator will also use the histogram of a region as a basis for locating edges. By sharing a common operation we kill two birds with one stone.

Now that we have found a good smoothing operator which blurs out the unwanted artifacts while preserving the sharpness of our edges one could legitimately ask why not now use a typical edge detector, such as a Laplacian or a gradient edge detector. The answer is threefold:

a. The results simply are not satisfactory in our case.

b. The operator we propose can be computed more efficiently (faster)

c. We believe the results to be superior, especially in the presence of noise

As examples of the performance of typical other edge detectors, we applied a Laplacian operator to the median filtered images shown in Figs. 47b, 48b and 49b. The results are shown in Figs. 55a,b, and c respectively. Possibly the best results are shown shown in Fig. 55a. This should not be surprising since Fig. 47b was about the best of the three original images. Yet we notice that the edges are not uniformly detected. Diagonal edges seem to be brighter. This is one of the features of the Laplacian. Although the Laplacian has the advantage of detecting edges in all directions, it has a bias in favor of diagonal edges. One will also note the many contours in the resulting images. This is most evident in Fig. 55c. These contours are the boundaries between regions which have constant gray levels. These regions are a consequence of our previous median filtering. In regions of the original image where the gray levels were fairly constant, the median was also fairly constant, resulting in output areas with fairly constant gray level values. One could possibly eliminate those contours in Fig. 55a but it would be difficult to do so in Fig. 55c.

There is a ring of brighter contours around the location of the cytoplasmic and nuclear membranes, but there are also contours just as bright in Fig. 55c.

Fig. 27b is the worse of all 3 and would be very difficult to process despite the fact that the median picture from which it was obtained (Fig. 48b) had very good contrast and practically no artifact. At this juncture, it seems that using a Laplacian after median filtering an image is just not the solution. In fact, the commonly used sequence of operations when using a Laplacian is to first filter noisy images with a Gaussian filter and then apply a Laplacian operator. Edges are then located at zero crossings of the Laplacian (Marr and Hildreth [26]). The problems with this optimal use of the Laplacian is that the Gaussian filter can shift the position of most of the edges in real images. Another problem is that of the use of the Gaussian filter. Its purpose is to remove noise and it will not remove artifacts. This is because like all difference based edge operators the Laplacian is very sensitive to noise and needs to operate on a noise free image. In fact, since the Laplacian is a second-derivative operator, it is unacceptably sensitive to noise.

At this point we seem to have a contradiction. The Laplacian might perform acceptably on a Gaussian smoothed image, but such an image is not what we need with our type of images since we really need images from which artifacts (which are not noise) are to be removed.

Our earlier conclusion was that median filtering was the best in term of performance and efficiency (speed) that we could find. Unfortunately, a Laplacian performs poorly on median filtered images.

Another conclusion is that we should look for something better than the Laplacian when dealing with our type of images.

Another choice is that of using a gradient operator on our median filtered images. The results of such an attempt are shown in Figs. 57a,b,c, d,e and f. As before these figures represent the detection of edges in different directions. Typically 4 to 8 directions are chosen. The advantage of using more directions is more accuracy in locating areas of high gradient but the disadvantage is that of having to process the image more times since one pass of a directional mask is required for each direction. That is a major drawback, especially when speed is of the essence.

Another disadvantage is that all these directional edge images must somehow be combined into a single edge image. The usual method is to compute the sum of the squares or absolute values of the directional images and then apply a threshold. All pixels with gray levels larger than the threshold are considered edge points. The results of Figs. 57a,b,c,d,e and f are much better than those obtained with the Laplacian, but at the expensive cost of more processing time. Also, since the contours we are interested in are those of circular and closed objects, we have edges in all

directions. This forces us to use at least 8 directional masks. Using fewer masks means missing certain edges. We must also deal with the thorny issue of thresholding the edge image obtained after summing our directional edge images. All of these thresholding techniques are ad hoc and require trial and error approaches.

In conclusion, we find that we simply are not satisfied with the two most common edge detectors, the Laplacian and gradient operators. The Laplacian is simply not well suited to our kind of images and the gradient operator though better suited requires a large amount of time. Also, we found that the preferred smoothing technique given our kind of images is median filtering and this adds the constraint that whatever operator we wish to apply after the smoothing phase be able to perform well on the output of the median filter.

III. Rationale for Using the Entropy in Edge Detection

After our review of the results produced by typical edge detectors we were faced with the conclusion that we need to median filter our images and then use an edge detector that performs well on such median filtered images. Median filtering requires the generation of an histogram of the gray levels in a neighborhood of some pixel. The median value is then readily obtained from such an histogram. Since we are always trying to process our images as quickly as possible, a natural question we posed

ourselves was what other relevant information can we gather from the histogram. By garnering as much information from the histogram as we could we would save precious time. Another point we mentioned earlier was that histogram computation is now a widely available function on many image processors and we wanted to obtain as much information as possible from this resource. A first thought was to simply use the local histograms to compute a threshold and use this threshold for segmentation. The edge points would be points at the borders of two different areas. The results were disappointing even though we supervised the selection of thresholds by manually choosing them. We then thought on a purely intuitive basis about computing the information content of a region. This is closely related to the entropy associated with the gray level distribution in the neighborhood of interest. Areas with low entropy (low information) should be unlikely to contain edges while areas of high entropy (high information content) would be likely to have edges. We then realized that the entropy would change as we moved from areas of low interest (such as background areas) to areas of more interest (such as the texture rich nuclear area). Finally we surmised that edge areas which by definition separate two dissimilar regions would have more information than either region surrounding them.

Putting the last two statements together we concluded that if our intuition was correct, peaks in an entropic image (the image obtained after

computing the entropy around every pixel of an image) should be good indicators of edge points.

Edge detection and segmentation are essentially two approaches to solving a common problem. In edge detection one tries to locate the edge or boundary which separates regions having different characteristics.

In segmentation one locates the different regions (based on some property) which form an image. Once these regions have been located, edges are simply defined as the set of pixels where different regions meet.

By using the entropy we can quantify different properties of images. Since the computation of entropy is based on a histogram, the property which is being quantified depends on what histogram is being used. If we use the gray levels of the pixels of a sub image as the basis for the histogram we obtain a measure of the amount of information we have about the gray levels of the subimage.

In general if we compute the histogram $P_p(x)$ = probability that property p has value x, of a property p of a subimage, the entropy computed from that histogram provides us with a measure of our knowledge (or lack thereof) concerning that property within the subimage

$$P_p(x) = [\text{Probability } (p = x)] = \Pr(p = x)$$

The basis of edge detection (or equivalently segmentation) using entropy is to consider that regions which have similar properties will also have similar entropy values.

One simple application of this use of entropy is the location of regions which are similar. For instance when looking at the background areas of an image the gray levels vary little. The entropy computed from the histogram of gray levels $P_p(x)$ would be small and vary little over the background area.

The major question which must be resolved when using the entropy is deciding which property p to use when computing the histogram $P_p(x)$. The most readily available properties of the pixels of a gray level image are their spatial coordinates and their gray levels. It is not clear which property is best to use to quantitate certain properties. For instance the distribution of gray levels of pixels would probably not be as useful as the joint distribution of gray levels in quantitating a property such as texture.

The major emphasis of this dissertation is placed on the use of entropy obtained from gray level distributions.

IV. BASIC APPROACH

Our main focus is the detection of cells in histological and cytological specimens. In our particular case the nucleus of cells is the most important regions to study since the size, shape and texture of chromatin contained in the nucleus provide much information about the status of a cell. The problems of edge detection and segmentation are actually the same since we are simply trying to locate two regions which are somehow different. Whether we locate the region itself or the boundary of the region is equivalent.

The basic concept we use to determine the location of edges and of regions of interest is that of entropy. Entropy is defined as:

$$H = - \Sigma [P(G) \text{ Log } P(G)] \quad G = 0,1,2,....,255 \quad (2)$$

and can be interpreted as a measure of the lack of knowledge we have about an event. The higher the entropy, the less we know about an event. The premise is that regions about which we know very little are considered interesting. On the other hand, regions about which we know much add little information about the image and present little interest. If we take the histogram of gray levels of a region of an image, we can compute the entropy associated with the distribution of gray levels as:

$$H = - \Sigma[P(i) \log P(i)] \quad i = 0,1,....,255 \quad (3)$$

where P(i) is the probability of occurrence of gray level i.

Regions of fairly constant gray level have low entropies while regions with more uniform distributions have high entropies. This simple observation points to the application of entropy to threshold histograms [44]. But the use of entropy is not limited to analyzing only the gray levels of an image. It is quite possible to compute the entropy associated with other characteristics of an image, such as co-occurrence matrices or textural descriptors. An image could be segmented in terms of interesting and uninteresting regions, where the level of interest would be defined not simply in terms of the distribution of gray levels, but also in terms of the distribution of other parameters. This leads to the idea of a generalized edge described as a transition between two regions with different values of interest. A new image which we shall call an entropic image can be obtained from a gray level image. This entropic image can be generated by computing the entropy (with respect to some parameter) of regions of the image. The entropic image provides a simple way to represent the values of interest of different regions of a gray level image WITH RESPECT TO DIFFERENT PARAMETERS. In some instances, information of interest might be localized in the absorption image (an image representing the amount of light absorbed by the specimen). Computing the entropic image with respect to the absorption level in a particular color or combination of colors (i.e. hue, saturation, luminance, etc.) will allow for the detection of interesting regions in those particular features. From this point on, the term entropic image will refer to an image computed by replacing entire regions with the value of the entropy computed by replacing entire regions with the value of the entropy, computed with respect to some feature in a neighborhood

of those regions. Those regions could be as small as an individual pixel, in which case the regions used for the computation of entropy would be neighborhoods of these pixels.

By using this approach [52], we have obtained some encouraging results. One of the advantages of the entropic image was that fairly large regions (12x12 to 16x16 pixels) can be described in terms of their entropy (a single number) and the original image of 512x512 pixels was transformed into an entropic image of size about 32x32 entropy values. These entropy values were enough to quickly locate areas of interest in the image. Another advantage of entropy is that its value is obtained from a histogram. The histogram of gray level values is one of the most common features available on current image processors. Some processors can compute histograms of specified regions in real time.

The entropic image is not the final solution to the problem and must itself be processed. But unlike typical gray level images whose values can change erratically, entropic images should have slowly varying values and should be easier to process. This is a consequence of the fact that the histogram of an image doesn't change much as one moves from one pixel to a neighboring one.

V. Detecting An Ideal, Noise-Free Step Edge

Our first analysis will be to measure the performance of our entropy based operator on an ideal noise-free, one-dimensional step edge $G(x) = L_o + (L_1 - L_o)\, U(x\text{-}E)$ where G(x) is the gray level of the pixel at coordinate x, U(x-E) is a unit step function occurring at $x = E$ and L_o, L_1 are the two gray levels surrounding the edge (Fig. 1).

If we compute the histogram $P_G(g)$ of the gray levels within a window of length $2L + 1$ centered at $x = W$ we obtain the following

Case a) $W + L < E$

See Fig. 2

The gray level distribution is:

$$P_G(g) = \begin{cases} 1 & g = L_o \\ 0 & g \neq L_o \end{cases}$$

See Fig. 3

the entropy associated with such a distribution is: $H_G(W) = -1 \log 1 = 0$.

Case b) $W\text{-}L < E \leq W + L$

See Fig. 4

The gray level distribution is

$$P_G(g) = \begin{cases} \dfrac{E - W + L}{2L + 1} & g = L_o \\ \dfrac{W + L - E + 1}{2L + 1} & g = L_1 \end{cases}$$

See Fig. 5

The entropy in this case is:

$$H_G(W) = -[P_G(L_o)\text{Log}(P_G(L_o) + P_G(L_1)\text{ Log }(P_G(L_1))]$$

Case c) $E \leq W\text{-}L$

See Fig. 6

The gray level distribution is:

$$P_G(g) = \begin{cases} 1 & g = L_1 \\ 0 & g \neq L_1 \end{cases}$$

See Fig. 7

The entropy in this case is

$$H_G(W) = -1 \log 1 = 0$$

As can be seen, the entropy is 0 in cases a) and c) when the window encompasses a region with constant gray levels.

The only case where $H_G(W) \neq 0$ is when different gray levels occur within the window of interest, as in case b). In case b) with a noise-free step edge the distribution of gray levels within the window is binomial with

$$P_G(L_o) = p_o$$

and

$$p_G(L_1) = p_1 \text{ where } p_o + p_1 = 1$$

This distribution is dependent on the position of the window. We can therefore define the function $H_G(W)$ as the value of the entropy associated with the gray levels of the pixels contained in a window centered at x. Fig. 8 is a plot of $H_G(W)$.

As can be seen, with the window starting on the left, the entropy is 0. As soon as the window's right end crosses the gray level edge at E the entropy starts to increase, reaching a maximum of $H_G(E) = 1$ when the window's center coincides with the edge at W = E. The entropy then decreases as the window's center passes the edge to the right. Finally $H_G(E) = 0$ when the edge transition is no longer within the window.

The reason why the entropy is maximum for W=E is that this condition corresponds to a distribution of gray levels which is

$P_G(L_0) = P_G(L_1) = 0.5$

this is a uniform distribution which also generates maximum entropy.

Therefore the location of maximum entropy can be used to detect an ideal one-dimensional step edge. By extending this to two dimensions we can also locate ideal noise-free two-dimensional step edges.

Although the conclusion that the location of maximum entropy occurs when the window is centered on the edge was obtained from the analysis of an ideal step edge, we will find this to be a general property of our operator. As we subsequently study the cases of noisy step edges, ideal ramp edges and noisy ramp edges, we will find that the entropy reaches its peak value when the window is centered on the edge.

This property will allow us to devise simple edge detectors, namely peak detectors.

Another advantage inherent in the use of the entropy operator is the redundancy to be found in the location of the edge. Indeed, by looking at Fig. 8 which represents the entropy as a function of the position of the center of the window, we see that not only does the peak occur when W is equal to E, but

that the entropy starts to rise on the left when W is equal to E-L and returns to 0 when W is equal to E + L.

By locating these points one can help better locate the position of an edge.

Another advantage of the entropy operator is that the image needs not be contrast enhanced in order to obtain discernible peaks. This is a consequence of the fact that the entropy is invariant under any one to one transformation of gray levels. Contrast enhancement is one such operation where the gray levels of all pixels are changed to some other value in order to stretch the range of gray levels. This means that by using the entropy operator we can eliminate the usual step of contrast enhancement or any previous preprocessing operation which performed a one to one transformation on the gray levels of the original image.

VI. Detecting An Ideal Step Edge Corrupted by Noise

Although we have seen that a perfect step edge can easily be located through the use of entropy, we rarely encounter such perfect edges in real images. Most often some type of noise or artifact blurs edges and region boundaries in general.

Our next step is to analyze how a corrupted step edge will be detected through the use of entropy.

This time we model our noisy edge G(x) as an ideal step edge I(x) plus noise N(x).

$$G(x) = I(x) + N(x)$$

See Fig. 9

where $I(x) = L_o + (L_1 - L_0)\, U(x-E)$

where L_o and L_1 are the two levels on either side of the edge.

At this point we need to know the type of noise process affecting our edge. There are numerous models we could use for N(x), but our approach here will be to select the N(x) which produces the worst results. As we shall show later, the N(x) producing the worst behavior is N(x) such that $P_N(x)$ = constant. In other words, a noise process with uniform distribution. We can now perform

an analysis very similar to the one used in the case of a noise-free step edge. The one difference is that we will have to subdivide our analysis into two cases. In case I, the range R of values taken by N(x) is less than $|L_1 - L_o|$, while in case II the range of N(x) is greater than $|L_1 - L_o|$. We also choose N(x) with 0 mean, E(N(x)) = N(x) = 0, where E( ) represents the expected or average value.

Case I: $R < |L_1 - L_o|$

Subcase a) $W + L < E$

See Fig. 10

the gray level distribution is:

$$P_G(g) = \begin{cases} \frac{1}{R} & L_o - \frac{R}{2} < g < L_o + \frac{R}{2} \\ 0 & \text{otherwise} \end{cases}$$

See Fig. 11

the entropy associated with this distribution is $H_G(W) = \log R$.

Subcase b) $W\text{-}L < E \leq W + L$

See Fig. 12

The gray level distribution is

$$P_G(g) = \begin{cases} \left(\frac{E - W + L}{2L + 1}\right)\frac{1}{R} & L_o - \frac{R}{2} < g < L_o + \frac{R}{2} \\ \left(\frac{W + L - E + 1}{2L + 1}\right)\frac{1}{R} & L_1 - \frac{R}{2} < g < L_1 + \frac{R}{2} \end{cases}$$

See Fig. 13

the entropy in this case is

$$H_G(W) = (\alpha \log R - \alpha\log\alpha) + (\beta \log R - \beta\log\beta)$$

where $\alpha = \frac{E - W + L}{2L + 1}$, $\beta = \frac{W + L - E + 1}{2L + 1}$

note $\alpha + \beta = 1$.

$$=> H_G(W) = \log R - (\alpha\log\alpha + \beta\log\beta)$$

since $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$ we have $(\alpha\log\alpha + \beta\log\beta) \leq 0$, therefore $H_G(W) \geq \log R$

Subcase c) $E \leq W\text{-}L$

See Fig. 14

The gray level distribution is

$$P_G(W) = \begin{cases} \frac{1}{R} & L_1 - \frac{R}{2} < g < L_1 + \frac{R}{2} \\ 0 & \text{Otherwise} \end{cases}$$

See Fig. 15

The entropy is

$$H_G(g) = -R\left(\frac{1}{R}\log\frac{1}{R}\right) = \text{Log } R$$

As in the case of the noise free edge we can define $H_G(W)$. Once again, on the left side of the edge $H_G(W) = \log R$ As the right side of the window crosses the edge the entropy starts to increase to a maximum and then decreases back to logR as the left side of the window crosses the edge. If we note that $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$ and $\alpha + \beta = 1$ we have $(\alpha\log\alpha + \beta\log\beta) < 0$ and therefore $H_G(W)$ is maximum when $\alpha\log\alpha + \beta\log\beta$ is minimum. This occurs when $\alpha = \beta = 0.5$. Therefore, $H_{G\,max}(W) = 1 + \log R$. This is the case when the window is centered exactly on the edge. Therefore, irrelevant of the noise process, so long as the range of N(x) is less than the height of the step, the use of entropy can detect the location of the edge.

This is a very interesting result as we find the shape of our output unchanged except for an upward shift. The location of our edge is still given by the peak value of $H_G(W)$ and the reference points at E+L and E-L can still be used for better accuracy in locating the edge.

Case II: $R \geq |L_1 - L_0|$

Subcase a) $W + L < E$

See Fig. 18

The gray level distribution

$$P_G(g) = \begin{cases} \frac{1}{R} & L_o - \frac{R}{2} < g < L_o + \frac{R}{2} \\ 0 & \text{Otherwise} \end{cases}$$

The entropy associated with this distribution is $H_G(W) = \log R$

Subcase b) $W\text{-}L < E \leq W + L$

See Fig. 20

The gray level distribution is

$$P_G(G) = \begin{cases} (\frac{E-W+L}{2L+1})\frac{1}{R} & L_o - \frac{R}{2} < g < L_1 - \frac{R}{2} \\ (\frac{E-W+L}{2L+1})\frac{1}{R}(\frac{W+L-E+1}{2L+1})\frac{1}{R} = \frac{1}{R} & L_1 - \frac{R}{2} < g < L_o + \frac{R}{2} \\ (\frac{W+L-E+1}{2L+1})\frac{1}{R} & L_o + \frac{R}{2} < g < L_1 + R_2 \end{cases}$$

See Fig. 21

The entropy in this case is:

$$H_G(W) = (L_1 - L_o)\frac{1}{R}[\log R - (\alpha\log\alpha + \beta\log\beta)] + (L_o - L_1 + R)\frac{1}{R}\log R.$$

where $\alpha = \frac{E-W+L}{2L+1}$ and $\beta = \frac{W+L-EH}{2L+1}$

Subcase c) $E \leq W\text{-}L$

the gray level distribution is

$$P_G(W) = \begin{cases} \frac{1}{R} & L_1 - \frac{R}{2} < g < L_1 + \frac{R}{2} \\ 0 & \text{Otherwise} \end{cases}$$

See Fig. 23

the entropy is

$$H_G(W) = \log R$$

Once again the entropy starts at a low level of logR and increases as the window crosses the edge. Finally, H_G returns to a value of logR as the window no longer contains the edge.

If we consider the ratio $\frac{L_1 - L_o}{R} = K$

we have for subcase b:

$$H_G(W) = \log R - k(\alpha \log \alpha + \beta \log \beta)$$

The entropy rises to a maximum when $\alpha = \beta = 0.5$

$$H_{Gmax} = \text{LogR - Klog}\frac{1}{2} = \text{LogR} + \text{K}$$

But since $K = \frac{L_1 - L_o}{R}$ and $R > L_1 - L_o$

we have $K < 1$.

and therefore the maximum peak value of $H_G(W)$ is lower than in the case of

$R < L_1 - L_o$

in that previous case

$H_{Gmax} = 1 + \log R$.

The effect of increasing the range of the noise is therefore to lower the peak value of the entropy. This in turn makes it more difficult to locate the edge. See Fig. 24 for a plot of $H_G(W)$.

Nevertheless, the property that the edge is located at the peak of the output, when the window is centered on the edge is unchanged. The reference points of E-L and E+L are also still present and could be used to better locate the peak.

The previous analysis has shown us that by using the entropy H_G(W) associated with the gray level distribution of pixels wihtin a window, we can locate the position of a step edge, ideal or corrupted by noise. We have seen that the effect of noise is to lower the relative difference between $H_G(W_0)$ and H_G(f.W sub e) where W_0 represents a window centered on the edge. The lower the signal to noise ratio, the more difficult it will be to locate the step edge.

Despite this promising start we must realize that step edges are not that common in images, but that gradual edges, edges with an extent greater than one pixel, are by far the most common. These edges are called ramp edges and occur in natural scenes. They are also the results of any operation which smoothes the image.

Our next analysis will deal with the performance of our entropic edge detector when applied to ramp edges.

First we will tackle the case of an ideal ramp edge and then we will analyze the more general case of a ramp edge corrupted by noise. That last case will prove to be very general since any edge can be thought of as a ramp edge corrupted by noise.

VII. Detecting An Ideal, Noise-Free Ramp Edge

To simplify the analysis, we will approximate the width of our window by 2L and the extent of the edge by 2S. The slight error introduced will not detract from our goal of showing how the function H_G (x) reaches its maximum value when the window is centered on the edge. See Fig. 25 for a plot of the ideal, noise-free ramp edge.

In the case of a noise-free ramp edge, we have two cases to study:

Case 1: The width of the window is greater than the extent of the edge:

$2L \geq S \Rightarrow L \geq S$

Case 2: The width of the window is less than the extent of the edge: $2L < 2S \Rightarrow L < S$

Case I) $L \geq S$

Subcase a) $W + L \leq E\text{-}S$

See Fig. 26

$$P_G(g) = \begin{cases} 1 & g = L_0 \\ 0 & g \neq L_0 \end{cases}$$

See Fig. 27

and $H_G(W) = 0$

Subcase b) $E\text{-}S < W+L \leq E+S$

See Fig. 28

$$P_G(g) = \begin{cases} \dfrac{(E-S)-(W-L)}{2L} & g = L_0 \\ \dfrac{1}{2L} & L_0 < g \leq G(W+L) \\ 0 & \text{elsewhere} \end{cases}$$

See Fig. 29

where $G(W+L)$ is the gray level at $(W+L)$

$$=> G(W+L) = \frac{(L_1 - L_o)}{2S}(W+L) + L_o - \frac{(L_1 - L_o)}{2S})\,(E\text{-}S)$$

This represents a distribution with mass

$\dfrac{(E-S)-(W-L)}{2L}$ at $g = L_0$ and a uniform distribution for $L_0 < g \leq G(W+L)$

The entropy associated with this distribution will depend on W. The important point here is not the exact value of $H_G(W)$, but that $H_G(W)$ is a monotone, increasing function of W for $E\text{-}S < W+L \leq E+S$. This can be seen by noting that as W increases, the probability mass at L_o decreases while the extent of the uniform part of the distribution increases. We can therefore state at this point that as the right side of the window crosses the left side ($W = E\text{-}S$) of the ramp edge, the entropy begins to increase. This increase continues as long as W increases and as long as $W+L \leq E+S$

Subcase c) $(W+L) > (E\text{-}S)$ and $(W\text{-}L) < (E\text{-}S)$

See Fig. 30

In this case, since the extent of the window is larger than the extent of the edge, the entire edge is inside the window and:

$$P_G(g) = \begin{cases} \dfrac{(E - S) - (W - L)}{2L} & g = L_0 \\ \dfrac{1}{2L} & L_0 < g < L_1 \\ \dfrac{(W + L) - (E + S)}{2L} & g = L_1 \\ 0 & \text{elsewhere} \end{cases}$$

See Fig. 31

this distribution corresponds to two probability masses at $g = L_0$ and $g = L_1$ and a uniform distribution for $L_0 < g < L_1$.

The entropy associated with this distribution is larger than or equal to the entropy in subcase b). This is because the function $\psi(p) = -p\log(p)$ is convex. That is: $\psi(p_1 + p_2) < \psi(p_1) + \psi(p_2) < \psi(p_1 + \varepsilon) + \psi(p_2 - \varepsilon)$ where $p_1 < p_1 + \varepsilon \leq p_2 - \varepsilon < p_2$ The entropy in subcase c) is larger than that in subcase b) because the partition containing the possible gray levels within the window in subcase c) is a refinement of the partition containing the possible gray levels within the window in subcase b).

One can also note that as the window moves from left to right, the distribution of gray levels varies only for levels $g = L_0$ and $g = L_1$. Therefore, the changes in $H_G(w)$ come only from the probabilities of these two terms.

$H_G(w)$ will be maximum when $P_G(L_0) = P_G(L_1)$.

For maximum entropy we need $P_G(L_0) = P_G(L_1)$

$$\frac{(E - S) - (W - L)}{2L} = \frac{(W + L) - E + S)}{2L}$$

$$=> E-W = W-E => 2E = 2W => W=E$$

and therefore the entropy is maximum when the center of the window coincides with the center of the ramp edge. In the case $W = E$, the entropy is maximum when all levels are equally likely. This occurs when $L = S$, the window has the same extent as the edge.

Thus, when $L = S$ and the window is centered on the edge we have: exact formula

$$P_G(G) = \begin{cases} \dfrac{1}{|L_1 - L_0 + 1|} & L_0 \leq g \leq L_1 \\ 0 & \text{Elsewhere} \end{cases}$$

And $H_G(W) = \log |L_1\text{-}L_0 + 1|$ for $(E\text{-}S < W+L \leq (E+S)$. This formula assumes that all $(L_1\text{-}L_0 + 1)$ gray levels actually occur within the window. In reality, because of quantization effects the actual number of distinct gray levels that occur in the window will be $\leq |L_1\text{-}L_0 + 1|$.

Since we are studying the case where $L \geq S$, we now investigate what happens if $L > S$.

In this case, the previous conclusion that the entropy is maximum when $W=E$ still holds. The difference is that now the distribution of P_G (g) is no longer uniform when $W=E$.

If we denote the ratio $\frac{L}{S} = R$ we have

$$P_G(g) = \begin{cases} \frac{R-1}{2R} & g = L_0 \\ \frac{1}{R} & L_0 < g < L_1 \\ \frac{R-1}{2R} & g = L_1 \end{cases}$$

as $R \rightarrow \infty$ we asymptotically approach the following

$$\text{Lim}_{R\to\infty} \quad P_G(g) = \begin{cases} \frac{1}{2} & g = L_0 \\ 0 & L_0 < g < L_1 \\ \frac{1}{2} & g = L_1 \end{cases}$$

and $\text{Lim}_{R\to\infty} H_G(W=E) = 1$

This in effect is nothing else than the case of a noise free step edge revisited! Indeed, to a window with extent much larger than the extent of the ramp edge, the ramp looks like a step. We can conclude two facts from the analysis of subcase c)

a. Whatever the size of the window (so long as $L \geq S$) the entropy associated with the gray level distribution inside the window is maximum when $W=E$, that is when the window is centered on the edge.

b. The optimum window is one whose extent is equal to that of the ramp edge.

For $R = 1$, $H_{G\,max} = H_G(W=E) = \text{Log}_2 |L_1 - L_0 + 1|$

as R increases

$H_{G\,max} = H_G(W=E)$ decreases asymptotically towards 1.

See Fig. 32 for a plot of $H_G(E)$ as a function R.

Subcase d) $(W-L) < (E+S) < W+L$

See Fig. 33

In this case:

$$P_G(g) = \begin{cases} \dfrac{E + S - (W - L)}{2L} & G(W\text{-}L) \leq g < L_1 \\ \dfrac{W + L - (E + S)}{2L} & G = L_1 \\ 0 & \text{elsewhere} \end{cases}$$

See Fig. 34

where G(W-L) is the gray level at (W-L)

$$G(W\text{-}L) = \frac{(L_1 - L_0)}{2L}(W\text{-}L) + L_0 - \frac{(L_1 - L_0)}{2S}(E\text{-}S)$$

this represents a distribution with mass

$$\frac{W + L - (E + S)}{2L} \text{ at } g = L_1$$

an a uniform distribution for $G(W\text{-}L) \leq g < L_1$

As in subcase b) the important point is not the exact value of H_G (W) but that now H_G(W) is a monotonically decreasing function of W. This can be seen by realizing that subcase d) is a mirror image of subcase b). Increasing W in subcase d) is the the equivalent of decreasing W in subcase b). Therefore, as

the window's center moves to the right of the center of the edge, $H_G(W)$ starts to decrease.

Subcase e) $E+S < W-L$

See Fig. 35

$$P_G(g) = \begin{cases} 1 & g = L_1 \\ 0 & g \neq L_1 \end{cases}$$

$H_G(W) = 0$

See Fig. 36

From this analysis of case A) it can be seen that as the window starts on the left, without encroaching on the edge, we have $H_G(W) = 0$.

As the window crosses the left side of the edge $H_G(W)$ starts to increase and reaches a maximum when $W = E$, in other words when the window is centered on the edge. The maximum value of $H_G(W = E)$ depends on the ratio $R = \frac{L}{S}$ and is maximum for $L = S$ (the window has the same extent as the edge), but $H_G(W = E)$ is never less than 1, with

$$\text{Lim}_{R \to \infty} H_G(W = E) = 1.$$

As the window moves further to the right, the entropy starts to decrease until finally it reaches the value $H_G(W) = 0$ when no part of the edge is contained in the window. See Fig. 37 for a plot of $H_G(W)$.

Case II) L < S

This case is identical to case I in all respects except for the extra case where the entire window is within the extent of the edge. We will call this subcase c.

The analysis is as follows:

Subcase c) E-S < W-L and W+L < E+S

See Fig. 38

$$P_G(G) = \begin{cases} \dfrac{1}{|G(W+L) - G(W-L)|} & G(W\text{-}L) < g \leq G(W+L) \\ 0 & \text{Elsewhere} \end{cases}$$

See Fig. 39

The entropy associated with this distribution is

$$H_G(W) = \text{LogD}$$

where D = |G(W+L) - G(W-L)|

But since there are at most 2L pixels in the window $H_G(W) \leq \text{Log}(2L) =>$ $H_G(W) \leq 1 + \text{LogL}$.

This value of $H_G(W)$ = LogD is the largest value which can be attained. Once again this is a consequence of the fact that the function $\psi(g) = P_G(g)$ Log $(P_G(g))$ is convex. H_G (W) is constant as long as the window is fully contained in the ramp. See Fig. 40 for a plot of $H_G(W)$.

Here again we find that as the window is centered on the edge the value of $H_G(W)$ takes on its peak value. For a window of extent less than 2L this peak value extends from W = E-S+L to W = E+S+L. The remedy to this situation is to use a window with length at least equal to the extent of the window.

VIII. Detecting a Ramp Edge Corrupted by Noise

We now analyze the performance of our entropic edge detector when applied to a noisy ramp edge. We will model our noisy ramp edge as a perfect ramp edge I(x) plus some uniform random noise N(x) with zero mean.

$$G(x) = I(x) + N(x)$$

See Fig. 41

Much of the ground work for this part of the analysis has already been laid down in our analysis of the detection of an ideal, noise free ramp edge. In fact, this part of the analysis will parallel that of the ideal, noise free ramp edge.

This is explained by realizing that all the cases and subcases analyzed in the ideal, noise free ramp edge apply equally to this case. The only difference is that we have noise added to our ramp edge. Since we are dealing with additive noise the distribution of gray levels in the noise corrupted ramp edge will be the convolution

$$P_G(g) = P_I(g) \bigstar P_N(g) \qquad (4)$$

where $P_I(g)$ is the distribution obtained in the noise free case, and $P_N(g)$ is the gray level distribution of the noise, and $\bigstar$ represents the convolution operation.

Case I) $L \geq S$

Subcase a) $W + L \leq E\text{-}S$

See Fig. 42

$$P_G(g) = \begin{cases} \frac{1}{R} & L_0 - \frac{R}{2} < g \leq L_0 + \frac{R}{2} \\ 0 & \text{Elsewhere} \end{cases}$$

See Fig. 43

and $H_G(W) = \text{LogR}$

Subcase b) $E\text{-}S < W + L \leq E + S$

See Fig. 44

P_G (G) is given by (4) where $P_I(g)$ is the distribution $P_G(g)$ found in Case 1, subcase b) of the analysis of the ideal ramp edge.

What interests us is the value of $H_G(W)$. Once again it is difficult to compute the exact value of $H_G(W)$. But as in the parallel case in the analysis of the ideal ramp edge we find that H_G (W) is monotone and increasing. This is a positive result, but we still don't know what the effect of the added noise is in our localization of the edge.

We can logically suspect that the major effect of the noise should be to make it more difficult to locate the edge.

We now show that this is exactly what happens. First we argue that $H_G(W)$ will be monotone, decreasing when we reach the conditions of subcase d). This argument is the same as the one used in the ideal ramp case.

This means that $H_G(W)$ reaches its maximum value H_{Gmax} as before when the conditions are those of subcase c, when the window is centered on the edge.

Let us now show that the peak value H_{Gmax} is such that the spread in entropy (let us call is Z) will also be less in this case than in the case of the ideal ramp edge.

$$Z = H_{Gmax} - 0 = H_{Gmax}$$

because the lowest value of $H_G(W)$ was 0. In the case of the noise corrupted ramp edge the smallets value of $H_G(W)$ is:

$$H_{Gmin} = \text{LogR}$$

and occurs in subcases a and e. We thus see that the effect of the noise in the background ($g = L_0$, and $g = L_1$) regions is to raise the value of $H_G(W)$ by the full amount LogR of the entropy of the noise. This is a consequence of the

shape of the distribution (a single pulse) in subcases a and e. Indeed, the convolution of a single pulse with any other function F is a shifted version of F and therefore the entropy of the distribution resulting from the convolution is simply the entropy associated with F.

As the window moves to the right and encroaches on the edge, the distribution $P_I(g)$ of gray levels in the window is no longer a single pulse and will have some spread, (let us call it S_I). The distribution $P_G(g) = P_I(g) \bigstar P_N(g)$ will have a spread

$S_G = S_I + S_N$ where S_N is the spread of $P_N(g)$

Referring once more to the convexity of

$\psi(g) = P_G(g)$ Log $(P_G(g))$, we see that the entropy associated with $P_G(g)$ is greater than that associated with $P_I(g)$

$H_G > H_I$

Defining $D = H_G - H_I$

We can see that

$D <$ Log S_N

This means that in the case of a noise corrupted ramp edge

$Z < \log R$

and that therefore, although we still reach a peak when the window is centered on the edge the difference $H_{Gmax} - H_{Gmin}$ is smaller than in the case of the ideal step edge.

We therefore reach the same conclusion here as we did in the case of the noise corrupted step edge, namely that the effect of noise is to make it more difficult to localize the edge, this is a consequence of lowering the difference between H_{Gmax} and H_{Gmin}

Given this result, we no longer need to analyze the remaining subcases associated with Case 1.

Case II) $L < S$

Subcase c) $E-S < W-L$ and $W+L < E+S$

$P_G(g) = P_I(g) \bigstar P_N(g)$ for $G(W-L) < g \leq G(W+L)$

The entropy $H_G(W)$ associated with this distribution is constant as long as the window is fully enclosed within the ramp.

Arguing as we did in the previous analysis, H_G(W) in this case attains its maximum value.

Although it is difficult to compute, H_G (W) is maximum when the window is centered on the edge. But once again we see that when the window is smaller than the extent of the ramp edge it becomes more difficult to localize the edge.

The conclusion one reaches from the analysis of Case 2 is that even if the window is of smaller extent than the ramp edge, the entropy still reaches its maximum value when centered on the edge. The only difference is that there is a region centered around the center of the edge for which H_G (W) is maximum and constant. This simply means that the resolution of a small window is too fine. To a small window, almost every part of a wide edge is an edge.

The analysis of Cases 1 and 2 has shown that using the entropy of gray levels in a window one can detect the location of an ideal noise free ramp edge.

IX. Advantages of Entropy Operator

At this point we can look back at the results of the analysis of the entropy operator and see a number of advantages over other edge detectors.

The major advantage is the ease with which one can locate edges in an entropic image. All that is required is a peak detector. Unlike the Laplacian its sensitivity to noise is small. Also, only one pass of the operator is needed. This is quite a time saver, especially when we consider the 4 to 8 passes necessary with the Gradient operator for proper coverage of edges at all possible orientations.

One could argue that the Laplacian, Gradient and other convolution based operators can be improved by using larger masks, such as a 5x5 mask. All that this would achieve is better noise immunity at the cost of far more processing time. One must remember that the number of multiplications (a time consuming operation) is of the order of NxM, where NxM is the size of the mask used. By going from a 3x3 mask to a 5x5 mask, one practically triples the amount of time needed to process the image. This still would not solve one of the major problems of the Laplacian which is its practically null response on the ramp part of ramp edges as a consequence of being a second derivative operator.

Since ramp edges are far more common than step edges in real images this would seem to leave only the Gradient operator. But we have seen how expensive that is in terms of the number of applications necessary. This expense is now looming even larger if we decide to go for larger masks. This is a drawback of all convolution based methods which employ masks that are not rotationally invariant.

The entropy operator is very efficient since with current hardware the time it takes to compute the histogram of a subimage does not depend on the size of subimage. This means that if need be, one could compute the histogram of a region of size 5x5 in the same time as it would take to compute the histogram of a region of size 15x15.

Another advantage in the computation of the entropy is that it can be performed in a recursive fashion. Indeed, given the entropy of a neighborhood of size NxM (N is the vertical dimension, M the Horizontal dimension) located at (x,y) it is easy to compute the value of the entropy of the NxM neighborhood located at (x + 1,y) or (x-1,y) or (x,y-1) or (x,y + 1). In the case of (x + 1,y) which corresponds to a horizontal shift of the window one pixel to the right we see that only the pixels on the left and vertical edges of window are affected. This means that there is no need to recompute the entire histogram, but that one simply needs to update the number of gray levels lost from the left edge and gained on the right edge. This represents a complexity of the order of 2N since there are N pixels along each of these two edges. As this update is being

performed, the value of the entropy can also be updated. Since as we have just seen, the entropy operator is rotation invariant, we can assume for the sake of argument that N < = M. One can therefore see that the time penalty incurred in increasing the size of the neighborhood when using the entropy operator grows proportionately to N rather then to NxM as in the case of convolution operators.

This is not a property of the entropy operator, but rather a property of one implementation of the operator. Because the implementation allows for such a low time penalty in changing the size of the window, we feel that it makes it easier to use the size most appropriate to solving the problem. If the problem warrants using a window of size 15x15, it will take only about three times as much time as using a 5x5 window. The time penalty would be of the order of 9 times if using convolution. Incidentally, performing a convolution with masks of size 5x5 would already be very time consumming.

Another major advantage of the entropy operator is the freedom of choosing any shape window. This again is not really a property of the operator, but again a property of the underlying source of information, the histogram. Indeed, current hardware allows one to compute the histogram of any shaped region. With convolution based methods one is limited by practical reasons to square masks.

Finally, another advantage of the entropy operator is that it is very compatible with median filtering (a practical requirement for our images) as both operators require a histogram. This means that by using this combination, one can economize on special hardware. Also, unlike many other approaches which count on future, improved and very likely expensive hardware for fast implementations, this operator makes the best use of hardware already available

CHAPTER III: Method and Results

METHOD

As with many edge detectors, a number of steps must be taken to obtain edge points from a gray level image. Most methods generate an intermediate gray level image in which high values represent potential edge points. In Fig. c, for instance, a number of contours with relatively high levels appear. This intermediate image is usually not the desired result. The desired goal is an image in which edges or contours are deterministically recognized. This means that we are not so much interested in obtaining an image with varying gray levels as in an image in which pixels are labeled as edge points or non-edge points. Such an image is called an edge map.

We perform a three step approach to compute this edge map.

The first step is median filtering the original image. Median filtering is a nonlinear signal processing technique which removes small artifacts while still maintaining the important edges. Median filtering is also very useful in removing noise, in particular in removing discrete impulse noise. Because of the nonlinear nature of median filtering, only a limited analysis of its effects can be performed and this leads to various strategies for its application. As such it is an adhoc tool and should be used in an interactive fashion. The effect of different applications of the filter should be monitored and the process terminated when the filter no longer improves the resulting image.

In our case we found that using a window of size 15 by 15 pixels removed small artifacts but left the major edges (those we want) untouched (compare originals with median filtered images).

2) Compute Entropic Image

Once the image has been smoothed out (by median filtering in our case) one can compute the entropic image. This image is obtained by sliding an N * M window over the entire image. As the center pixel of the window sweeps over the entire image, a histogram is computed at each point. This histogram depends on the coordinates of the center pixel of the window and can be denoted $P_G(x,y,g)$. The represents the probability of gray level g occurring in the window centered at (x,y). Associated with this probability is the entropy $H_G(x,y)$.

From our previous analysis we know that this value will be larger in areas where the gray level distribution is large and will be small where the gray level distribution is small. We also know that around a step edge the values of $H_G(x,y)$ have a distinctive profile in which the entropy increases from a low value to a maximum and then back to a low value along a direction perpendicular to the edge. The maximum value is centered on the location of the edge. Looking for edges is equivalent to looking for these peaks in $H_G(x,y)$. Another property of H_G is that one can be fairly sure that the gray level distribution is small where $H_G(x,y)$ is small. THis means that as $H_G(x,y)$ is

computed those pixels for which H_G (x,y) is smaller than some threshold can be eliminated from contention. This is most easily done by computing H_G(x,y) and then performing the following:

if $H_G(x,y) <$ threshold

then $H_G(x,y) = 0$

Finding an appropriate threshold is another ad hoc procedure which needs to be monitored. Despite this we feel that automatic methods for its selection can be devised. One possibility is to relate the threshold to the size of the window being used. Indeed if we compute $N_G(x,y) = 2^{H_G(x,y)}$ where $^{\cdot}$ represents exponentiation we obtain the number of gray levels of a uniform distribution which would have entropy H_G(x,y). We could then decide that in a window of size S = N*M pixels, we should not have $N_G(x,y) <$ (S/3). In other words threshold = $\log_2(S/3)$. Basically $N_G(x,y) >=$ threshold means that we consider the contents of a window interesting only if the average number of gray levels in the window is greater than some fraction of the total number of pixels in the window.

Another factor to be chosen is the size of the window. Once again experimentation is in order although some guidelines can be formulated.

The window size should be large enough so that the histogram of gray levels within it will be some statistical significance. As such windows of size

less than (5 * 5 = 25) pixels are not recommended. Also the window size should not be too large as this would render the processing very slow. Also, we know that the size of the window should be related to the size of edges we are interested in. We feel that edges 5 pixels wide or less are the most interesting ones. We find windows of size 25 as in (5 * 5) and 49 (7 * 7) pixels quite adequate. When using windows of (5 * 5) pixels, a lower threshold of 3 removes the uninteresting regions quite well.

Finally, the shape of the window can be tailored to the type of edge being detected. Square windows (N * N) pixels pick up edges of all orientations while vertical edges are better recognized by horizontal windows (N horizontal * M vertical, N > M) pixels. Horizontal edges are better found with vertical windows (N horizontal * M vertical, N < M) pixels. This is a consequence of the fact that horizontal windows "see" more of the horizontal neighbors of a vertical edge and can better locate the peak when the window is centered on the edge. A similar argument holds for vertical windows and horizontal edges. Also, if one is only looking for vertical edges, a horizontal window need only be scanned horizontally whereas the search for horizontal edges could be performed with a vertical window scanned vertically.

One might wonder how efficient it is to compute so many histograms and entropies for each pixel. We use a method which updates the histogram as the window is moved one pixel to the right so that $P_G(x,y,g) = F(P_G(x-1,y,g))$. This implies that once the histogram at (x,y) has been computed it need not be

recomputed as the window is moved one pixel to the right, but simply updated. Similarly, $H_G(x,y) = Q(H_G(x-1,y))$ and means that the entropy at (x,y) is simply computed by updating the entropy at (x-1,y). These updates on $H_G(x,y)$ and PG(x,y,g) make the computational load to be of the order (N * M * W) for an image of size (N * M) pixels and with a window of size (L * H), with W of the order of (L + H)/2.

3) Compute Edge Map

This is the last step in the location of edge points. The result of this step is a binary image in which non-zero pixels represent points which belong to an edge. In most other methods this is a difficult step to cross. Usually a simple threshold operator is used where pixels whose "edge" value is larger than some threshold are considered edge points while all others are eliminated. This is what is usually done after convolving some mask (sobel, laplacian, etc...) with the image. The problem here is the selection of the threshold. A low threshold appropriate for an image with faint edges is not appropriate for an image where the edges have high contrast. A survey of threshold selection techniques can be found in [42]

With the entropic image approach there is no need to use thresholds at this stage. Edges can be associated with the local peaks in the entropic image. In our implementation, we consider a pixel to be an edge point whenever:

$H_G(x,y) > H_G(x-1,y)$ and $H_G(x,y) > H_G(x+1,y)$

or

$H_G(x,y) > H_G(x,y-1)$ and $H_G(x,y) > H_G(x,y+1)$

RESULTS

We now show results obtained with the method outlined above.

Figures 47a and 47b represent an original contrast enhanced image (fig. 47a) and its median filtered version (Fig. 47b). The size of the window used for median filtering was 15x15.

Figures 48a and 48b represent an original contrast enhanced image (fig. 48a) and its median filtered version (Fig. 48b). The size of the window used for median filtering was 15x15.

Figures 49a and 49b represent an original contrast enhanced image (fig. 49a) and its median filtered version (Fig. 49b). The size of the window used for median filtering was 15x15.

Figures 50a and 50b represent the median filtered image of fig. 47b along with the edge map obtained from the entropy operator. The window size used was 5x5 and the threshold was 3.0

Figures 51a and 51b represent the median filtered image of fig. 48b along with the edge map obtained from the entropy operator. The window size used was 5x5 and the threshold was 3.0

Figures 52a and 52b represent the median filtered image of fig. 49b along with the edge map obtained from the entropy operator. The window size used was 5x5 and the threshold was 1.7

Figures 53a and 53b show a synthetic image of a black circle on a dark background and the result of applying the entropy operator to it.

Figures 54a, 54b and 54c represent the raw entropic images of figures 47a, 48a and 49a. Finding the peaks in these images generated the images seen in figures 50b, 51b and 52b respectively.

Figures 55a, 55b and 55c show the result of applying a Laplacian operator to figures 47b, 48b and 49b respectively.

Figures 56a, 56b and 56c show the result of applying a Laplacian operator to figures 47a, 48a and 49a respectively.

Figures 57 a,b,c,d,e and f are the result of applying different directional Sobel masks to figure 47b.

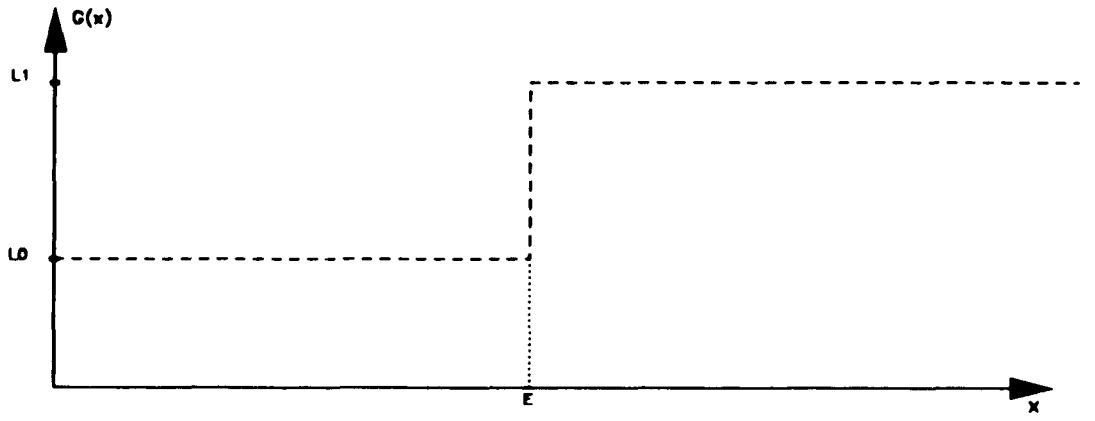
Figures 58a,b,c,d,e and f are the result of applying different directional Sobel masks to figure 47a.

CONCLUSION

The purpose of this dissertation has been to add a new method of edge detection to the many existing ones. This new method based on entropy uses the idea that in the area of edges the distribution of gray levels has a larger entropy than outside of edges. We have shown that this implies that the entropy reaches a peak when the window is centered on an edge. This last property makes it much easier to locate edge pixels and therefore makes it much easier to generate the edge map image. We have shown that the computational load of this method can be optimized in a way that reduces the actual number of operations to be performed. An advantage of the entropic edge detector is that edges of different contrasts will generate recognizable peaks in $H_G(W)$. In convolution based methods edges with different intensities generate different gray levels which are difficult to locate with a single threshold.

As was mentioned earlier, not only can the gray level distribution of pixels be used, but other statistics of interest can be used to locate edges. This means that not only gray level edges, but other kind of edges can be detected with this method. The information from the entropic image could also be used with other algorithms by allowing processing only on regions deemed interesting [52]. This can speed processing by disregarding areas of little interest and not wasting valuable time processing them.

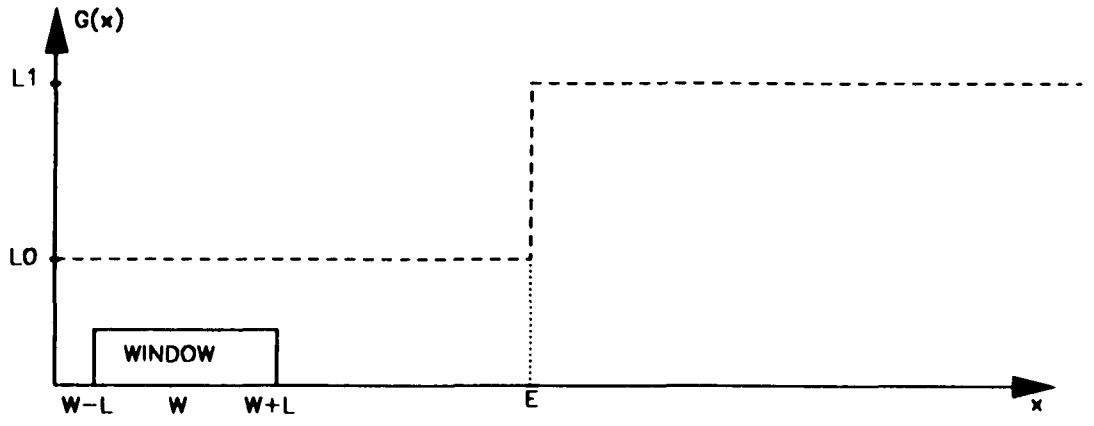
Following are the figures associated with the different cases:



An ideal step edge at x=E

$$G(x) = L0 + (L1 - L0) * U(x - E)$$

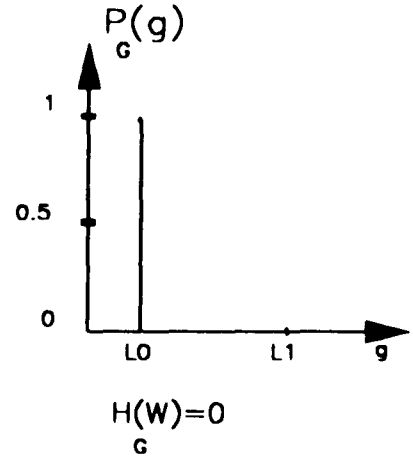
Fig. 1



Noise free step edge

Case a) (W+L) < E

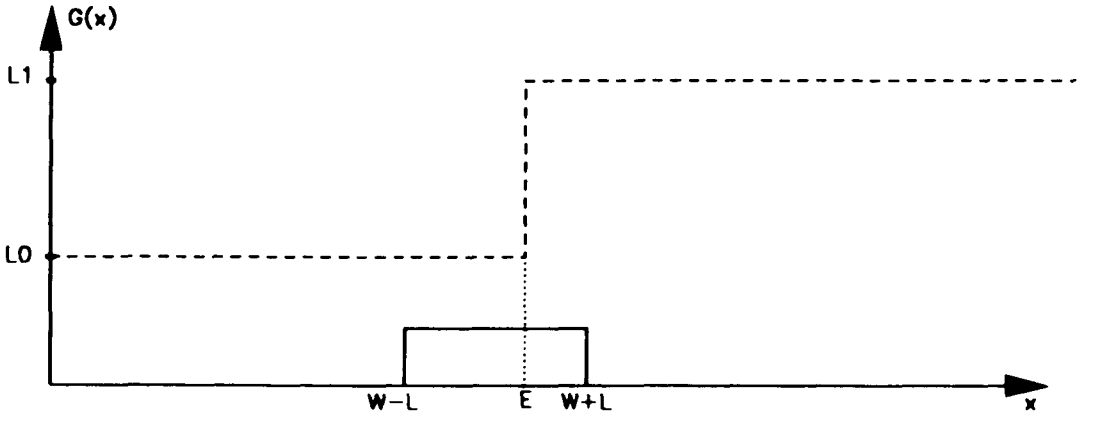
Fig. 2



Distribution of gray levels for

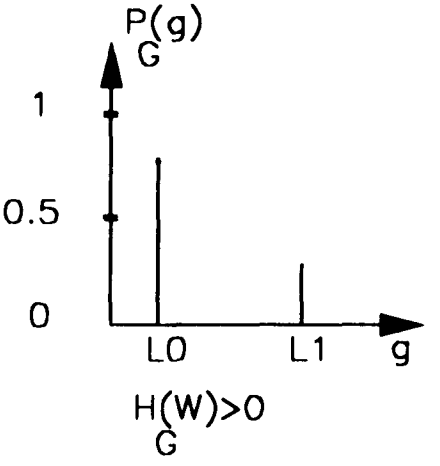
Case a) (W+L) < E

Fig. 3



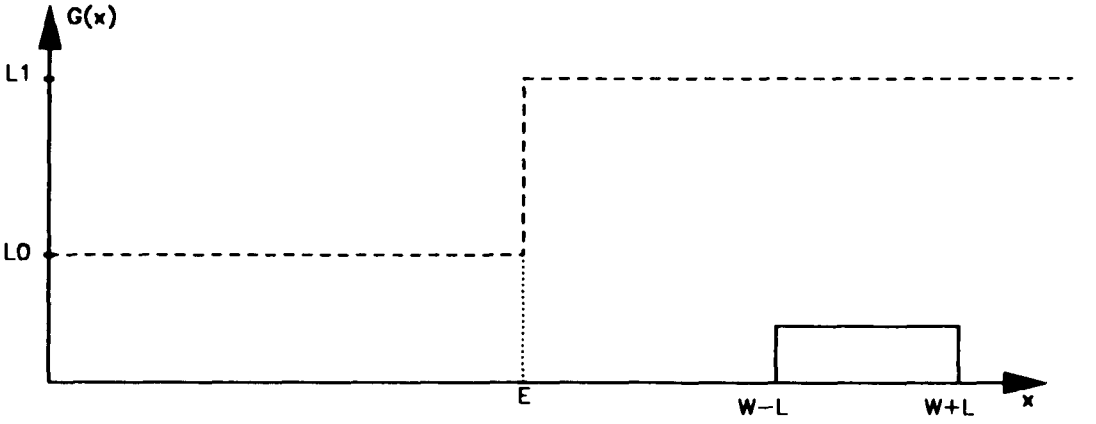
Noise free step edge
Case b) (W-L) < E <= (W+L)

Fig. 4



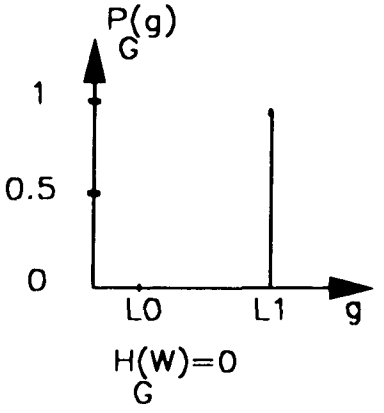
Noise free step edge
Case b) (W−L) < E <= (W+L)

Fig. 5



Noise free step edge
Case c) E <= (W-L)

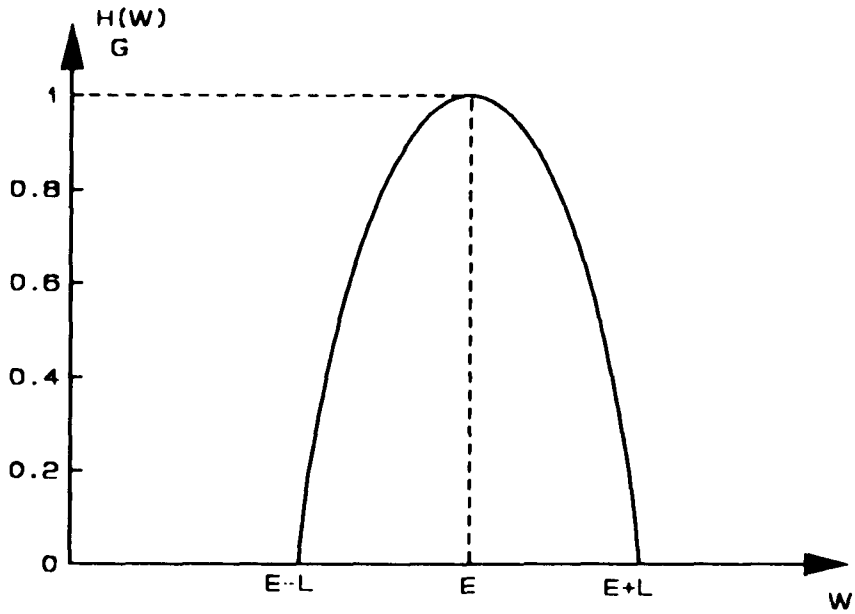
Fig. 6



Noise free step edge

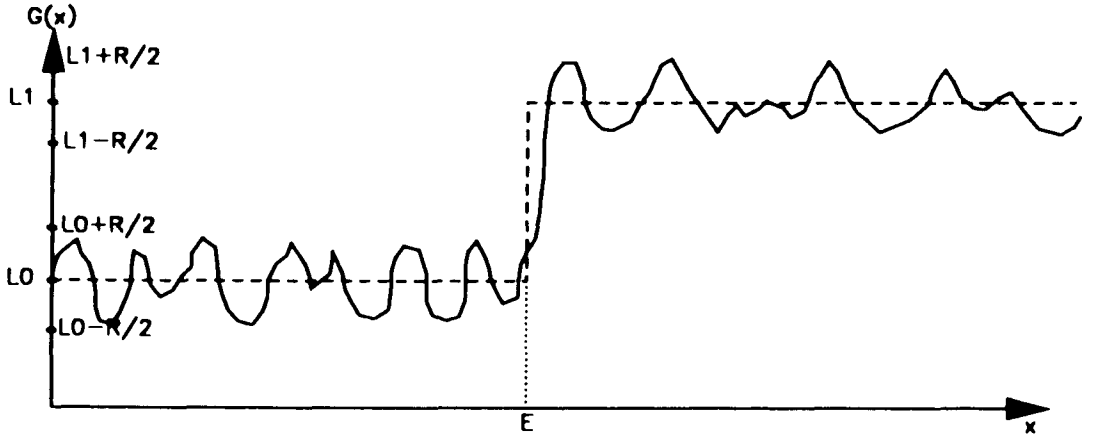
Case c) E <= (W-L)

Fig. 7



H(W) for a noise free step edge
G
Located at E
With a window of length 2L

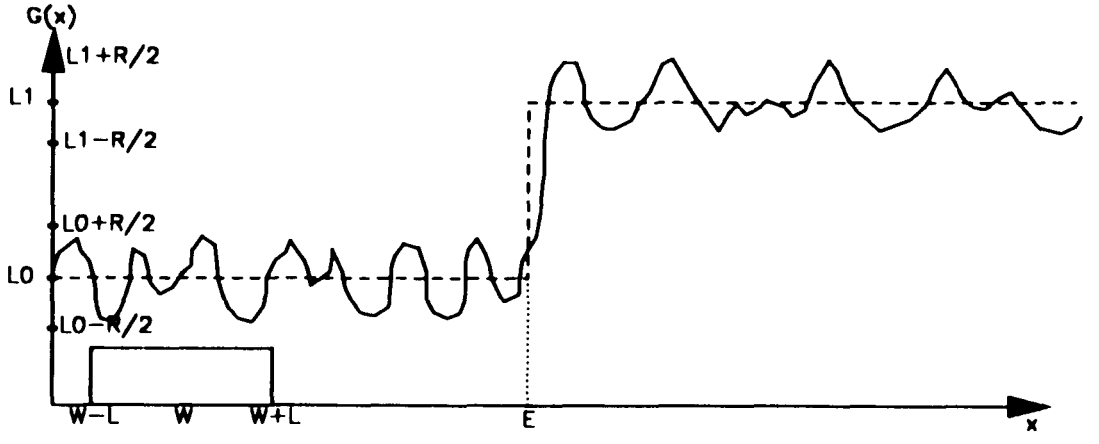
Fig. 8



Noisy step edge

Case 1: $R < |L1 - L0|$

Fig. 9

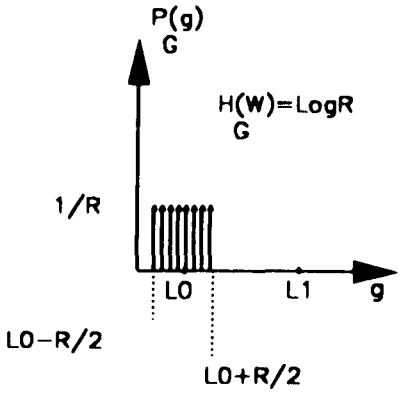


Noisy step edge

Case 1: R < |L1−L0|

Subcase a) (W+L) <= E

Fig. 10

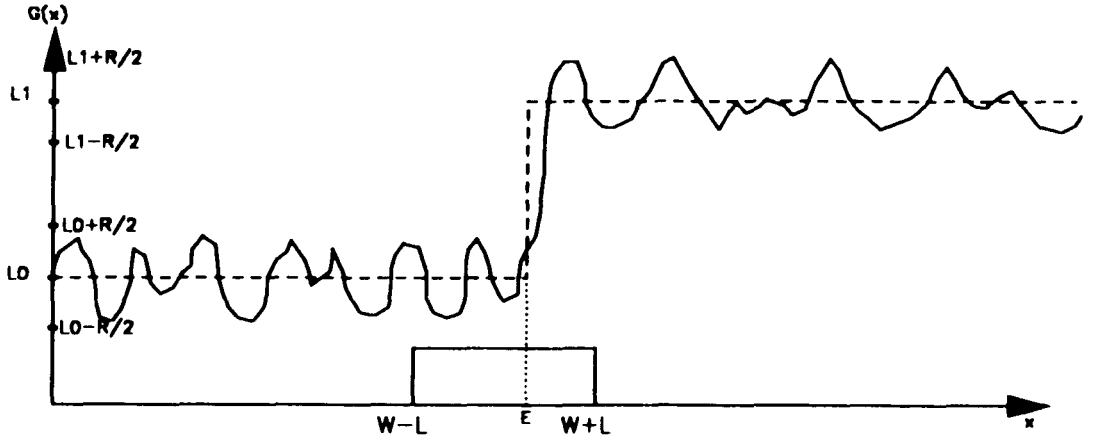


Noisy step edge

Case 1: R < |L1-L0|

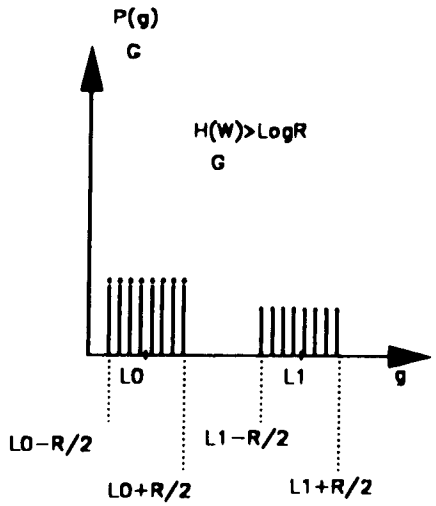
Subcase a) (W+L) <= E

Fig. 11



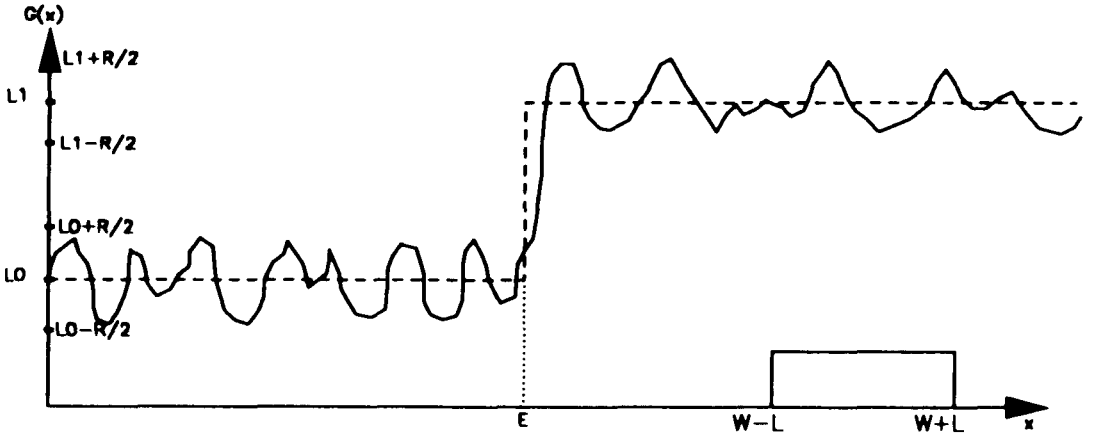
Noisy step edge
Case 1: R < |L1-L0|
Subcase b) (W-L) < E <= (W+L)

Fig. 12



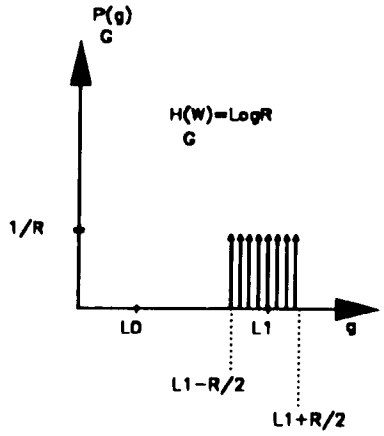
Noisy step edge
Case 1: R < |L1-L0|
Subcase b) (W-L) < E <= (W+L)

Fig. 13



Noisy step edge
Case 1: R < |L1 - L0|
Subcase c) E <= (W-L)

Fig. 14

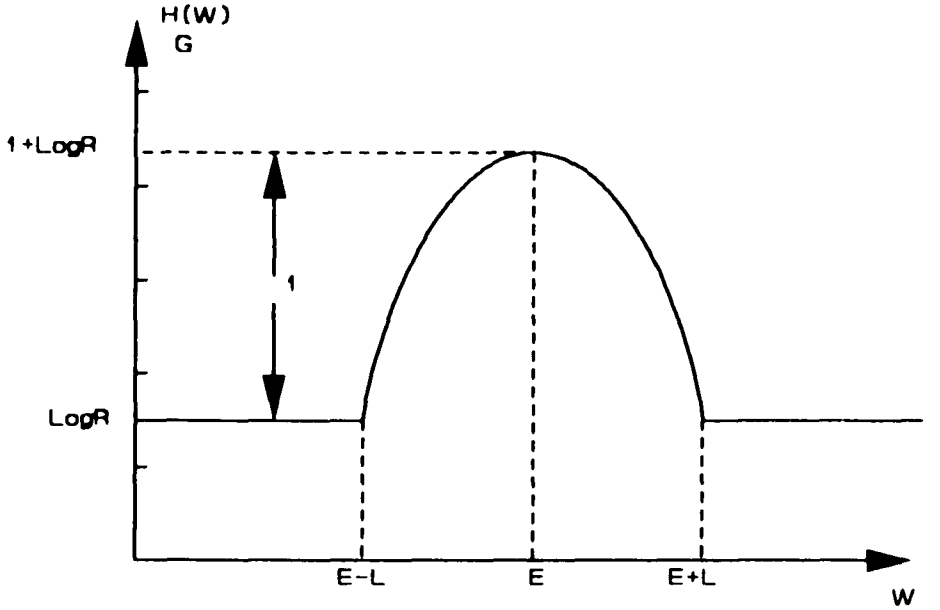


Noisy step edge

Case 1: R < |L1 - L0|

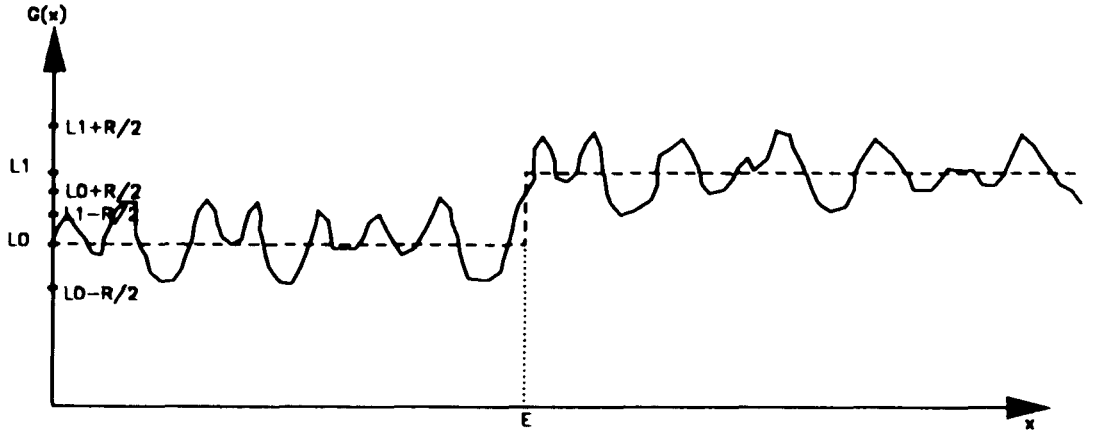
Subcase c) E <= (W-L)

Fig. 15



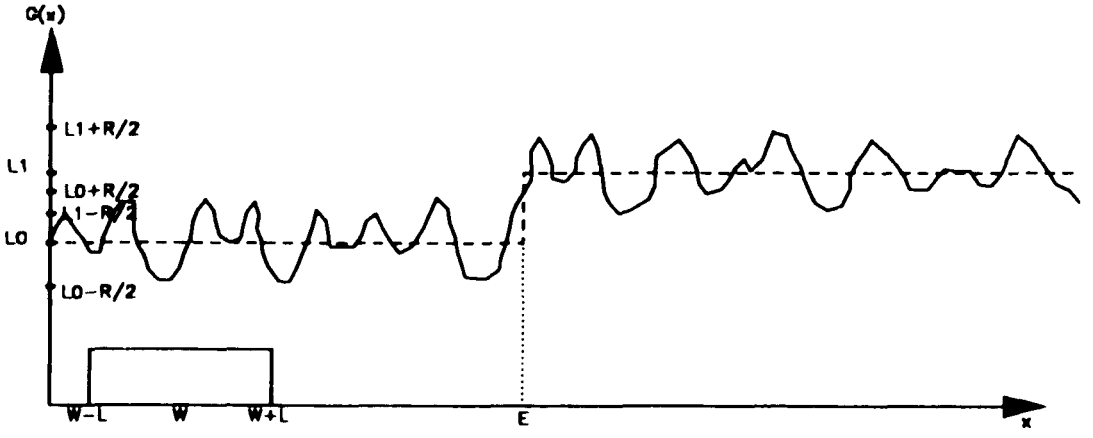
H(W) for a noisy step edge
G Located at E with a window of length 2L
(CASE 1)

Fig. 16



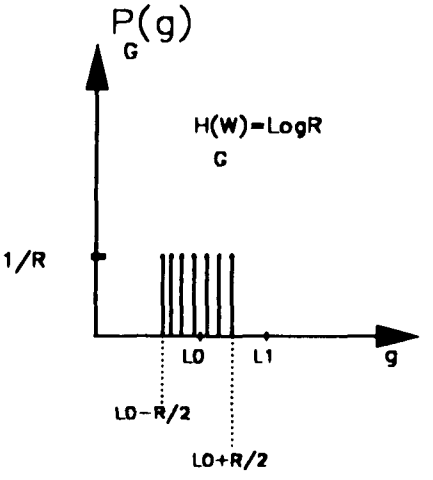
Noisy step edge
Case 2: R >= |L1 - L0|

Fig. 17



Noisy step edge
Case 2: R >= |L1 - L0|
Subcase a) (W+L) <= E

Fig. 18

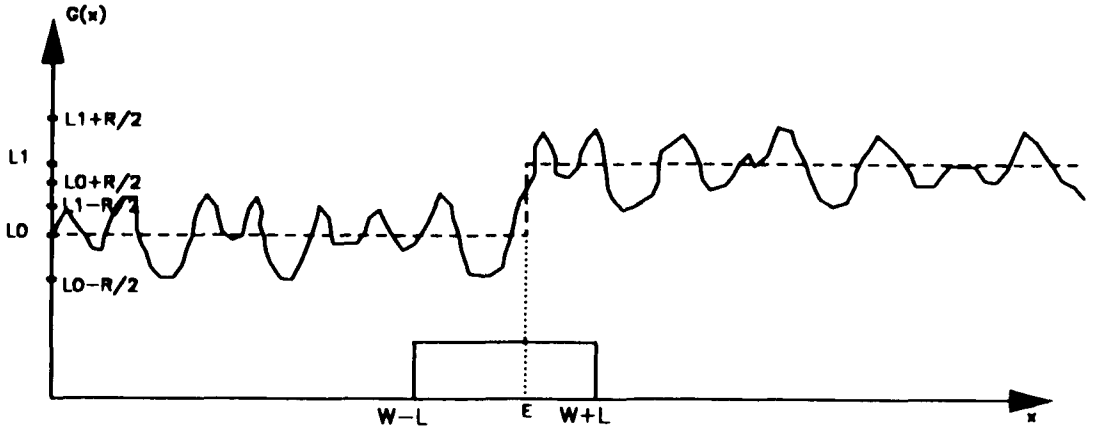


Noisy step edge

Case 2: R >= |L1 - L0|

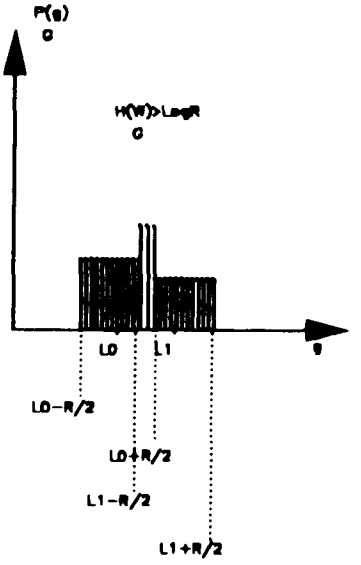
Subcase a) (W+L) <= E

Fig. 19



Noisy step edge
Case 2: R >= |L1 − L0|
Subcase b) (W−L) < E <= (W+L)

Fig. 20

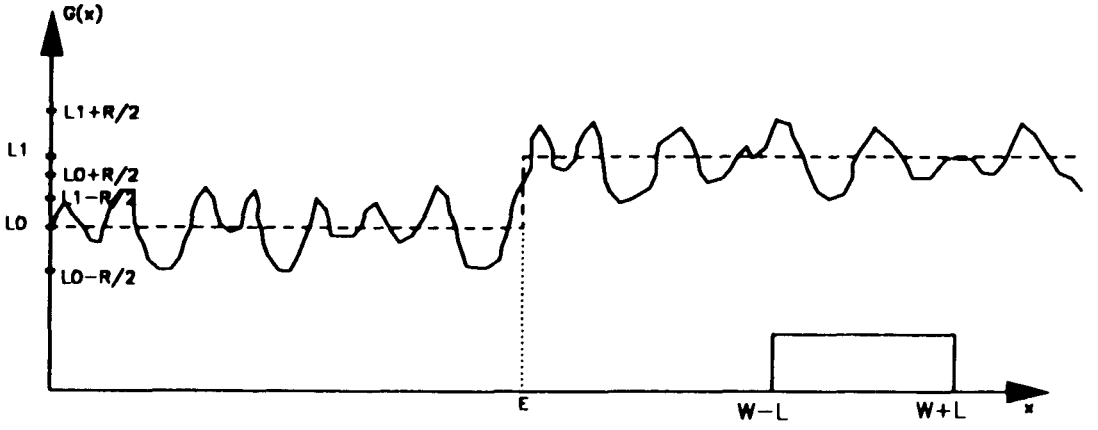


Noisy step edge

Case 2: R >= |L1 - L0|

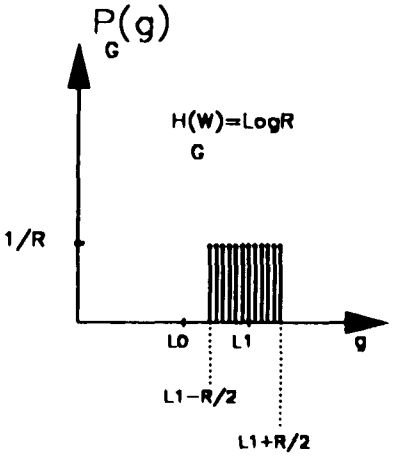
Subcase b) (W-L) < E <= (W+L)

Fig. 21



Noisy step edge
Case 2: R >= |L1 - L0|
Subcase c) E <= (W-L)

Fig. 22

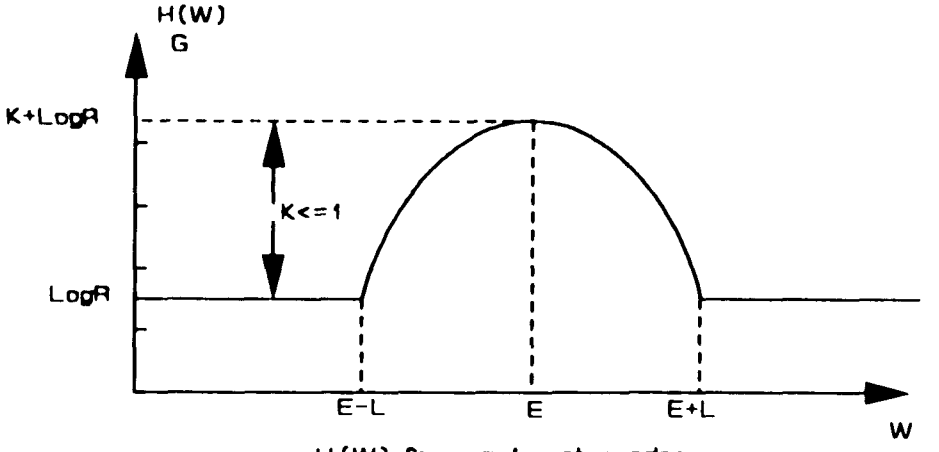


Noisy step edge

Case 2: R >= |L1 - LO|

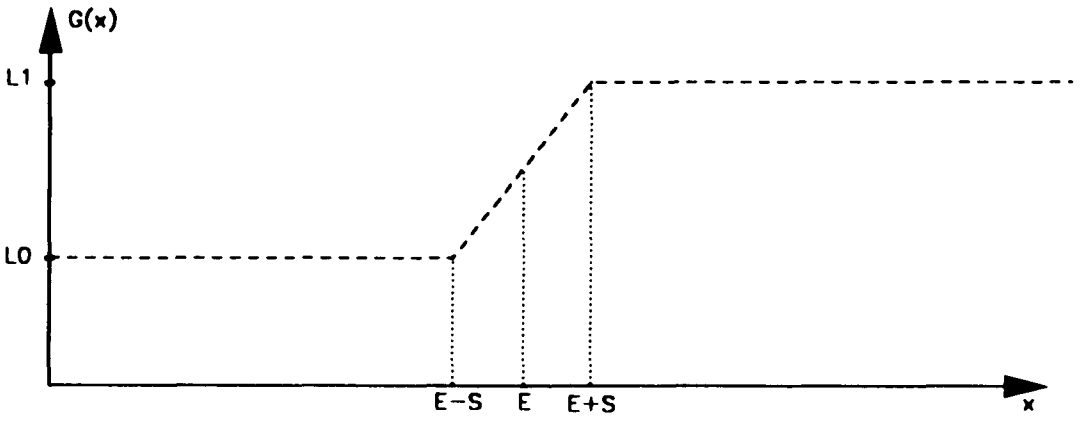
Subcase c) E <= (W-L)

Fig. 23



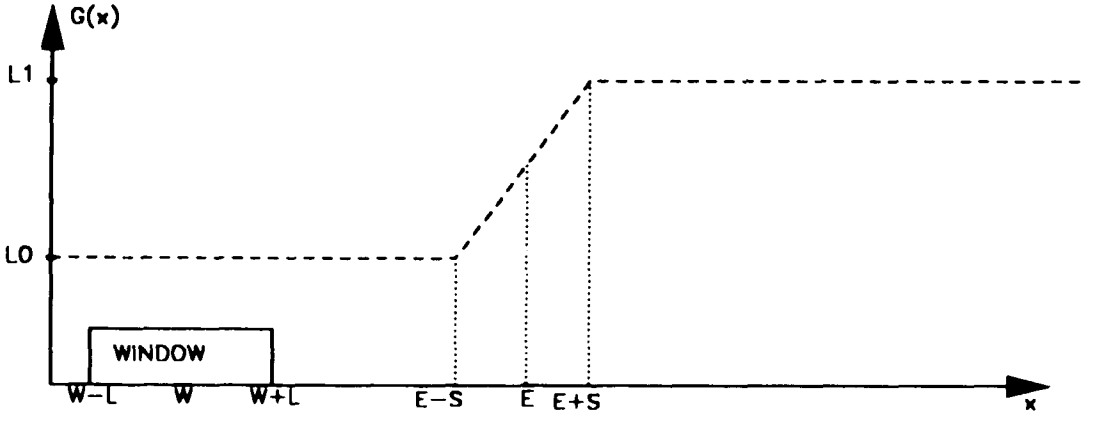
H(W) for a noisy step edge
G
Located at E with window of length 2L
(CASE II)

Fig. 24



Ideal noise free ramp edge

Fig. 25

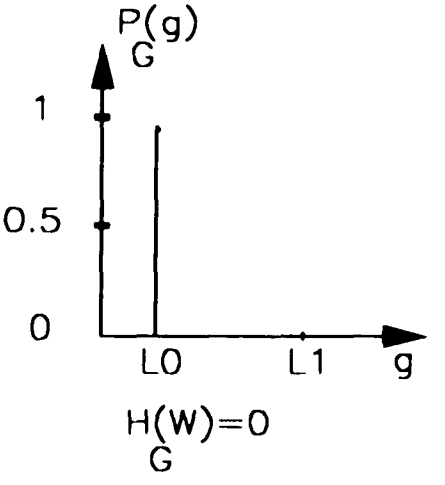


Ideal noise free ramp edge

Case 1: L >= S

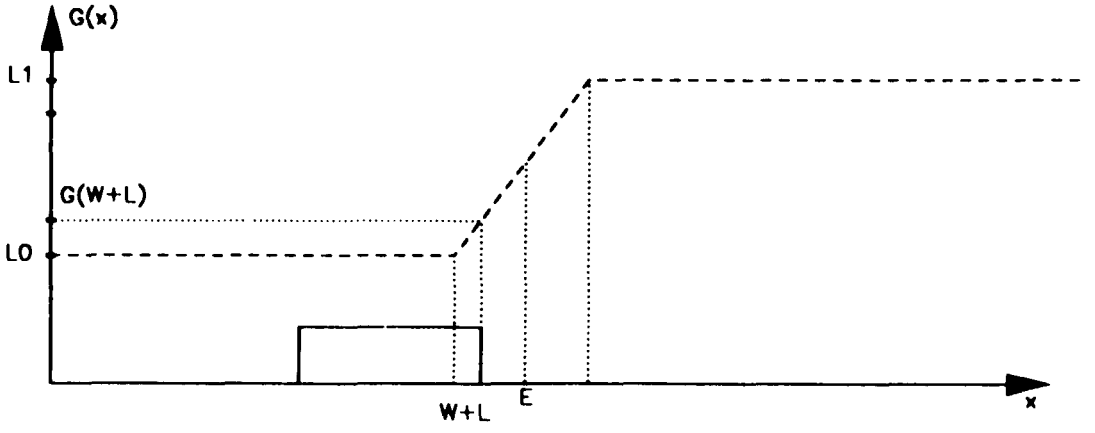
Subcase a) (W+L) <= (E−S)

Fig. 26



Ideal noise free ramp edge
Case 1: L >= S
Subcase a) (W+L) <= (E-S)

Fig. 27

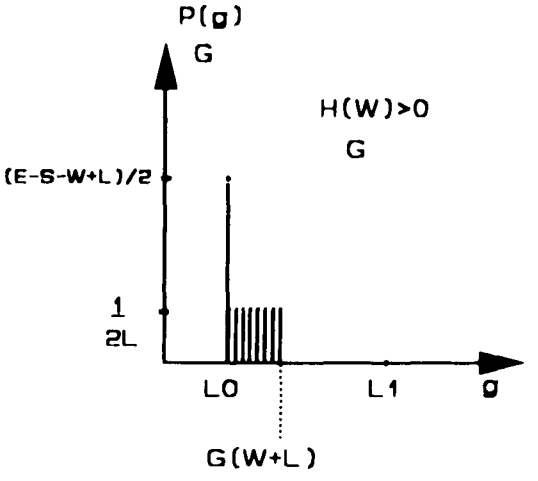


Ideal noise free ramp edge

Case 1: L >= S

Subcase b) (E−S) < (W+L) <= (E+S)

Fig. 28

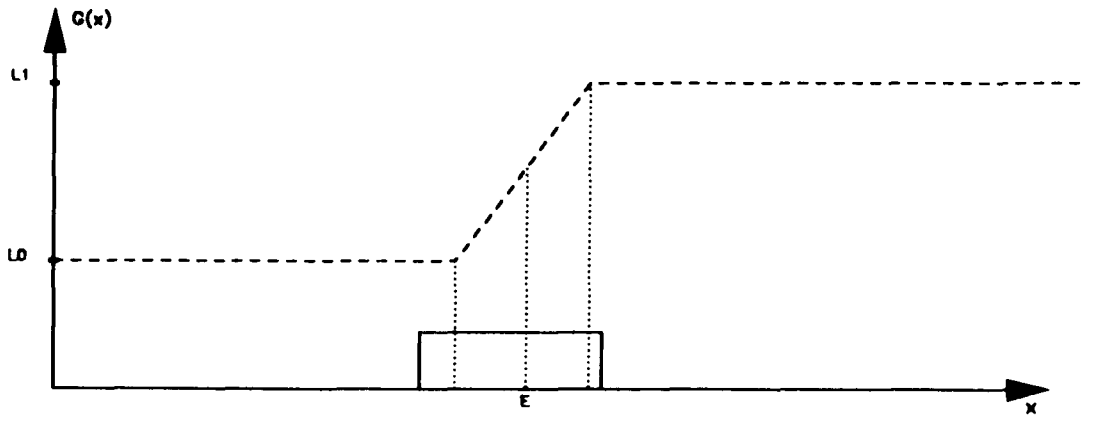


Ideal noise free ramp edge

Case 1 : L >= S

Subcase b) (E-S) < (W+L) <= (E+S)

Fig. 29

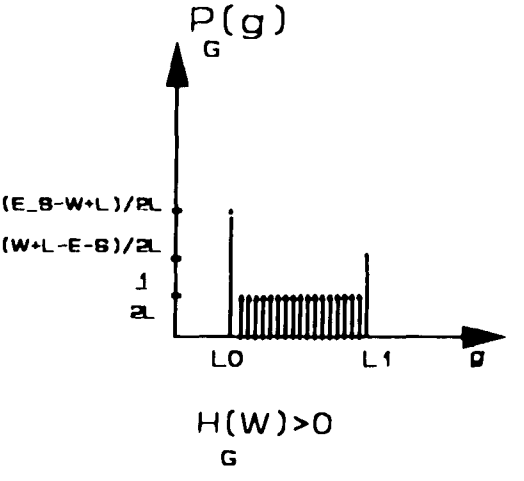


Ideal noise free ramp edge

Case 1: L >= S

Subcase c) (E−S) < (W+L) <= (E+S)

Fig. 30



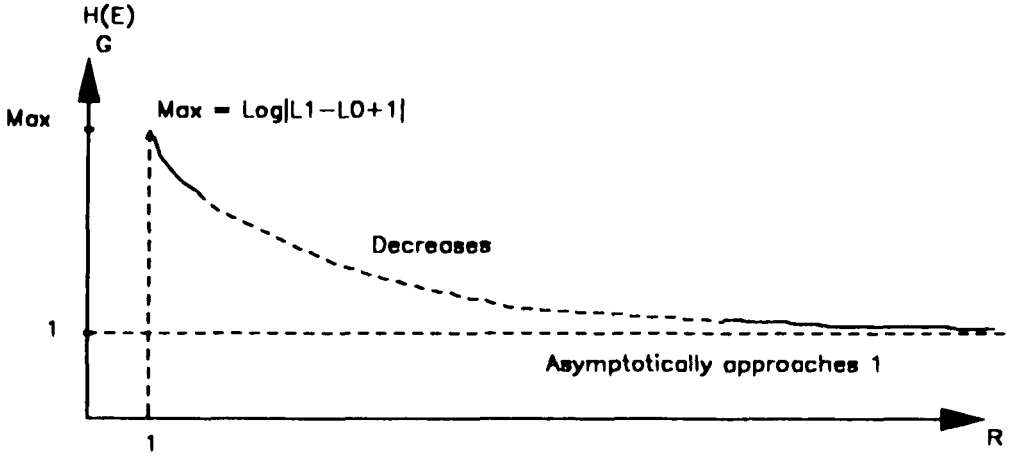
H(W)>0
G

Ideal noise free ramp edge

Case 1: L >= S

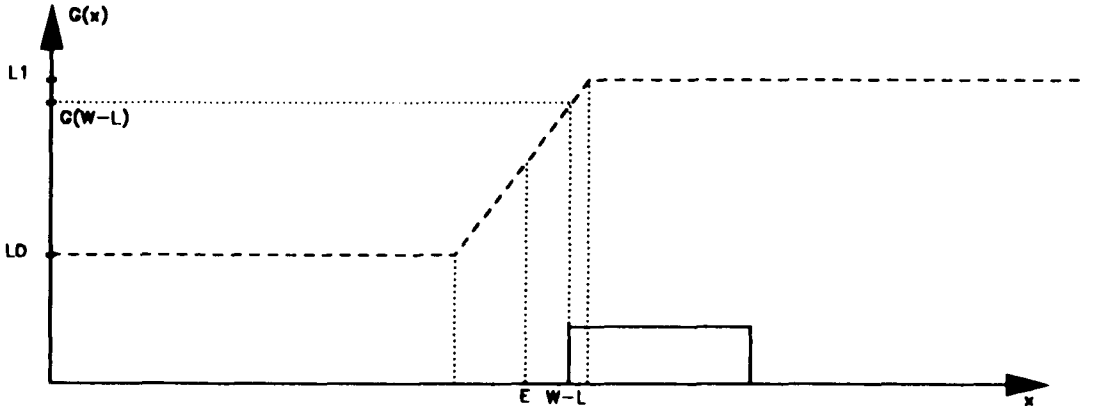
Subcase c) (E-S) < (W+L) <= (E+S)

Fig. 31



Behaviour of H(E) as a function of R
G

Fig. 32

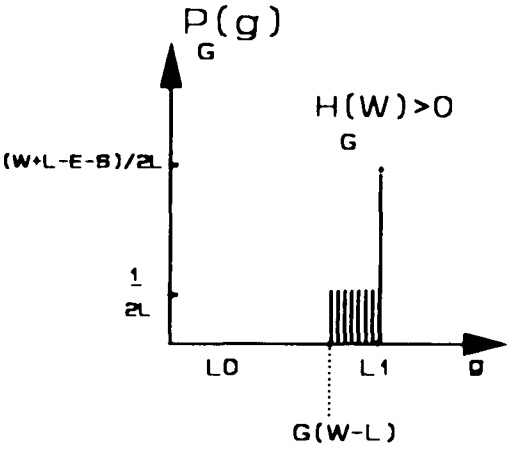


Ideal noise free ramp edge

Case 1: L >= S

Subcase d) (W-L) < (E+S) < (W+L)

Fig. 33

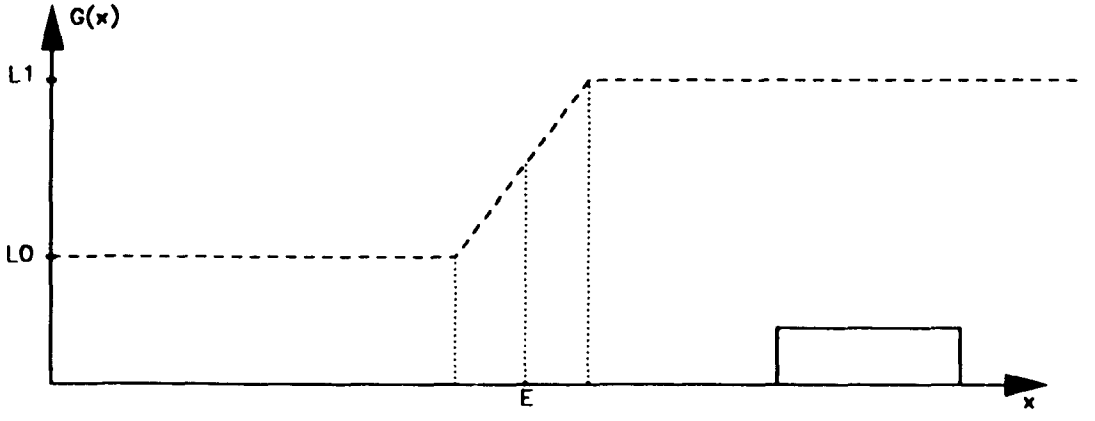


Ideal noise free ramp edge

Case 1: L >= S

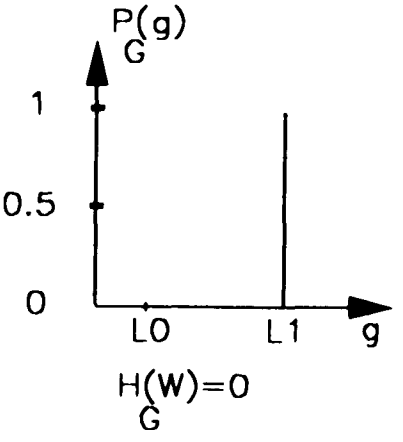
Subcase d) (W-L) < (E+S) < (W+L)

Fig. 34



Ideal noise free ramp edge
Case 1: L >= S
Subcase e) (E+S) < (W-L)

Fig. 35

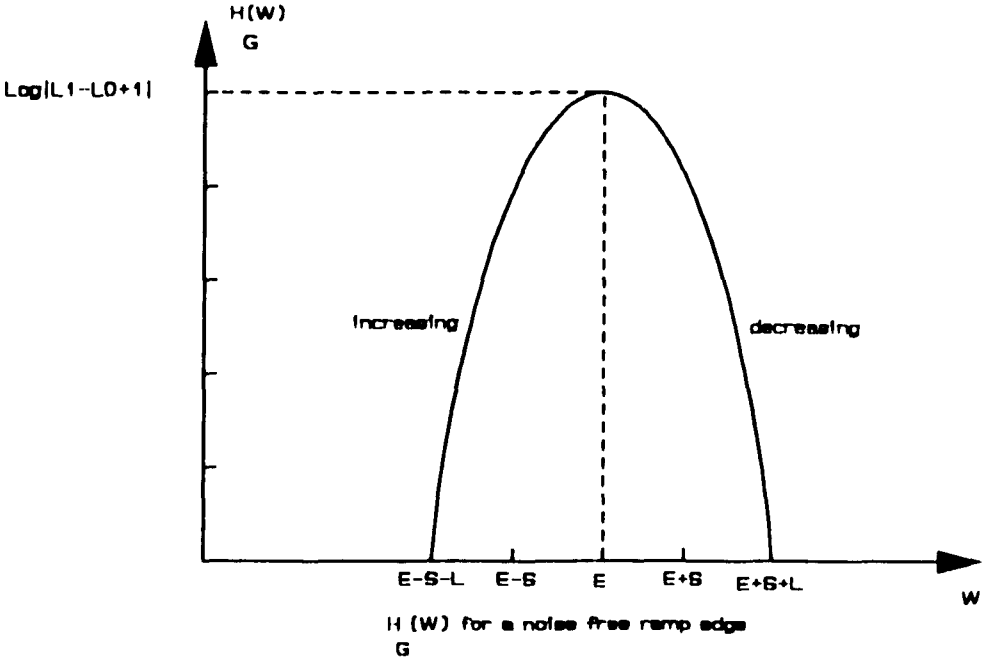


Ideal noise free ramp edge

Case 1: L >= S

Subcase e) (E+S) < (W-L)

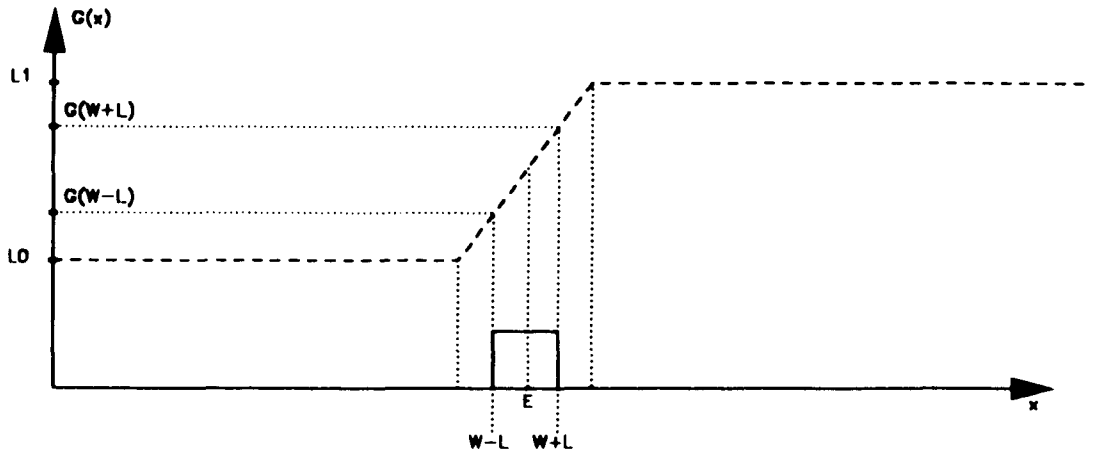
Fig. 36



H (W) for a noise free ramp edge
G

Located at E with extent 2S and with L > S

Fig. 37

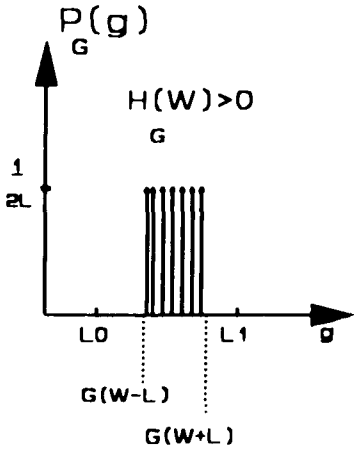


Ideal noise free ramp edge

Case 2: L < S

Subcase c) (E--S) < (W-L) and (W+L) < (E+S)

Fig. 38

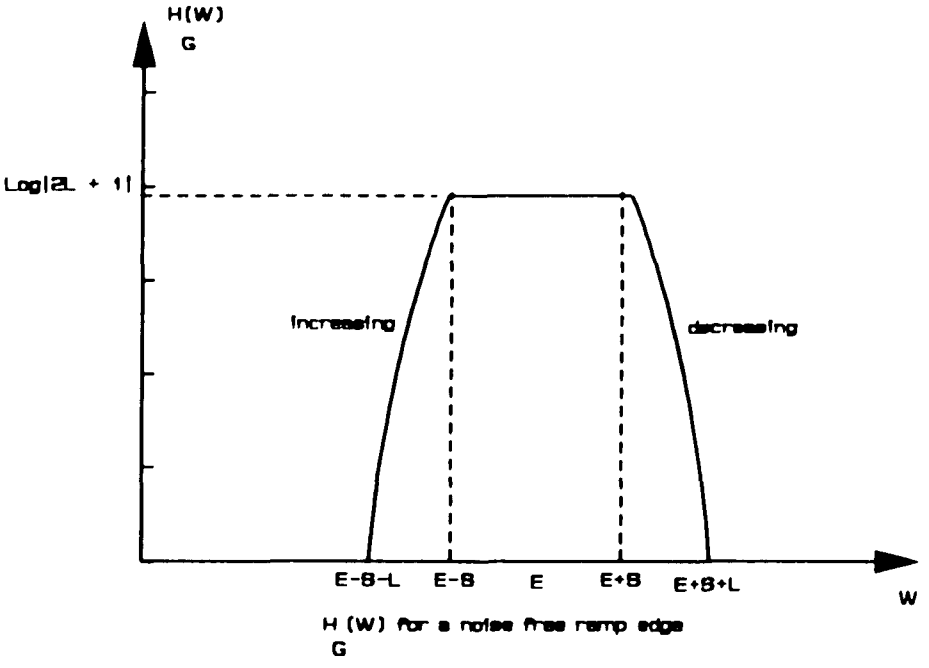


Ideal noise free ramp edge

Case 2: L < S

Subcase c) (E-S) < (W-L) and (W+L) < (E+S)

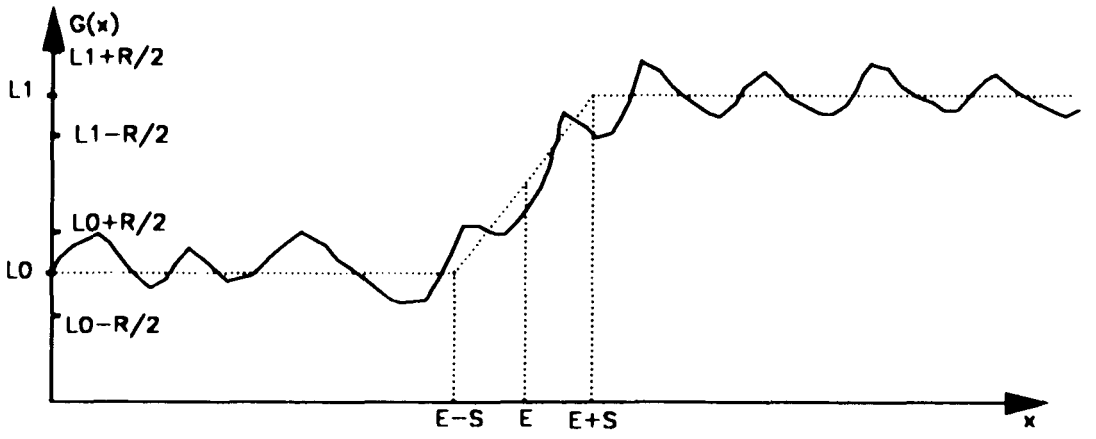
Fig. 39



H (W) for a noise free ramp edge
G

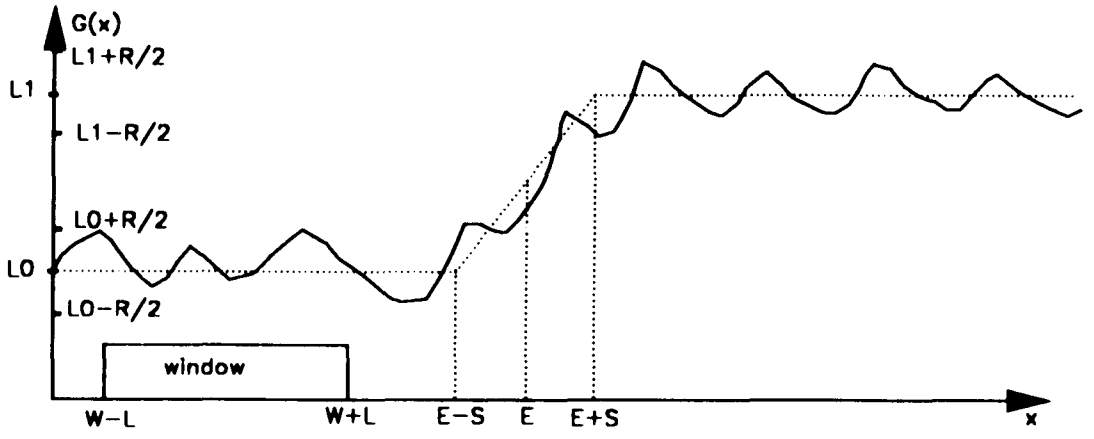
Located at E with extent 2B and with L < B

Fig. 40



Ramp edge corrupted by noise

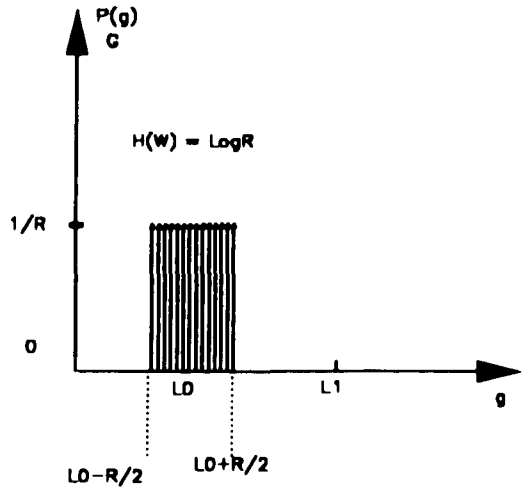
Fig. 41



Ramp edge corrupted by noise

Case a) (W+L) < (E−S)

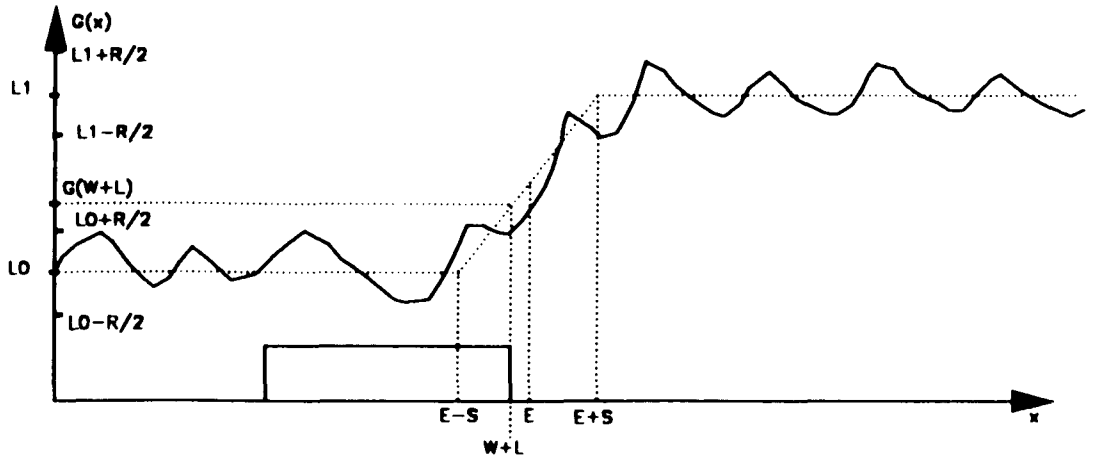
Fig. 42



Ramp edge corrupted by noise

Case a) (W+L) < (E−S)

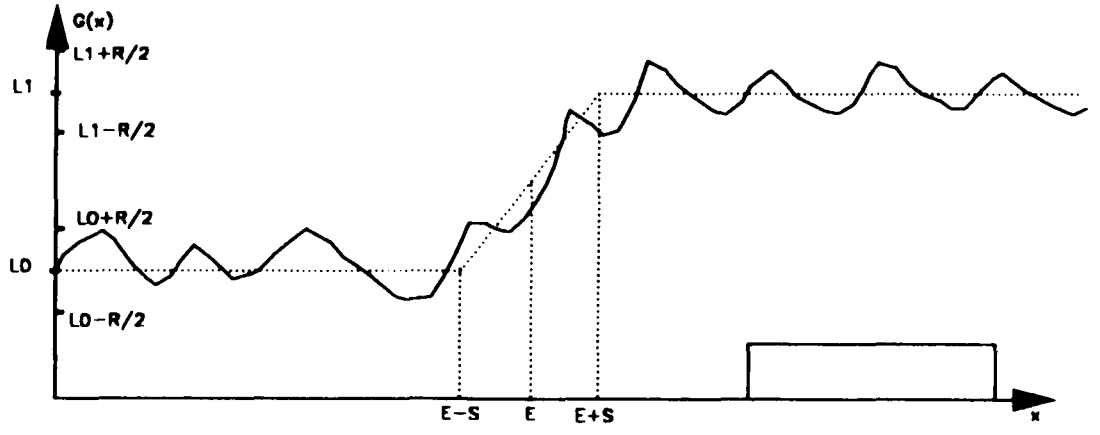
Fig. 43



Ramp edge corrupted by noise

Case b) (E−S) < (W+L) <= (E+S)

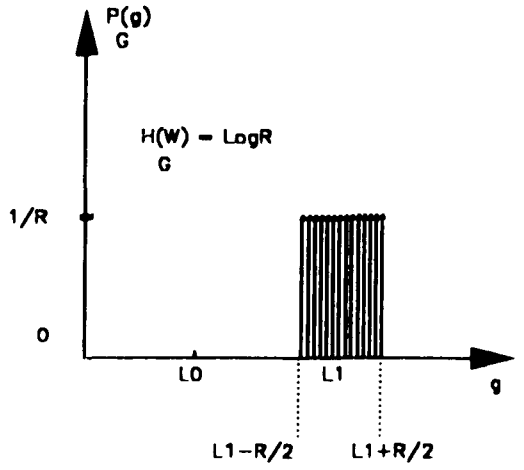
Fig. 44



Ramp edge corrupted by noise

Case e) (E+S) < (W-L)

Fig. 45



Ramp edge corrupted by noise

Case e) (E+S) < (W−L)

Fig. 46

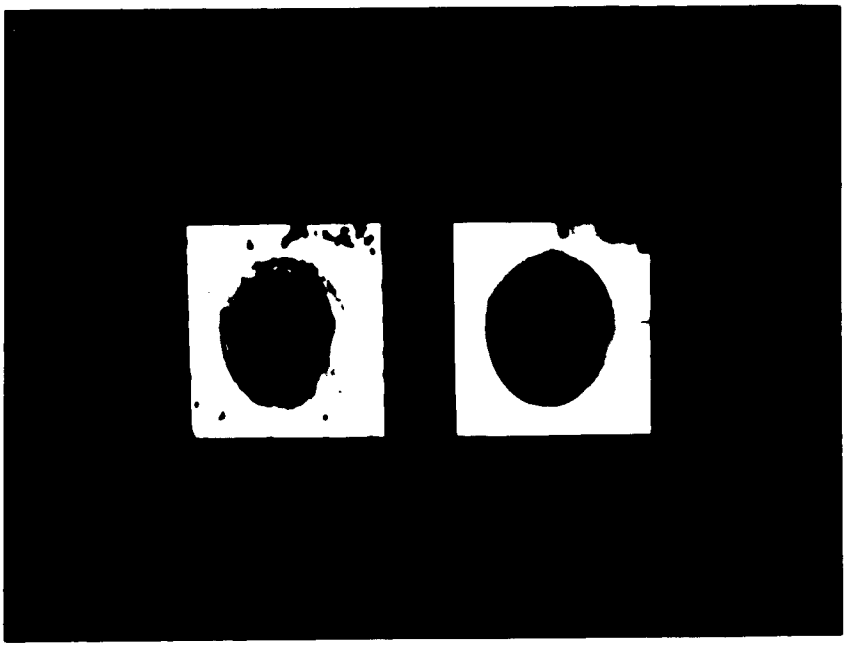


Fig. 47a,b

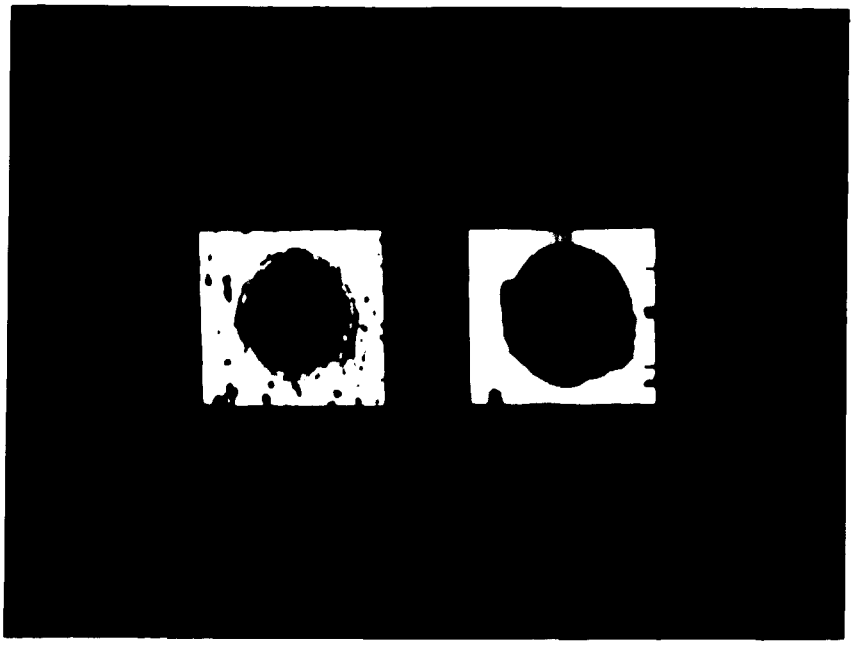


Fig. 48a,b

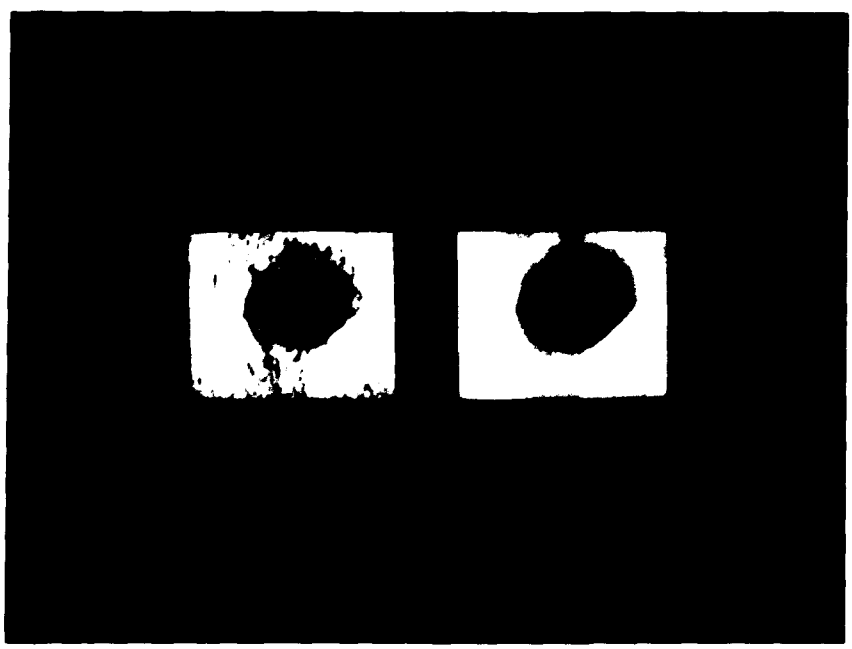


Fig. 49a,b

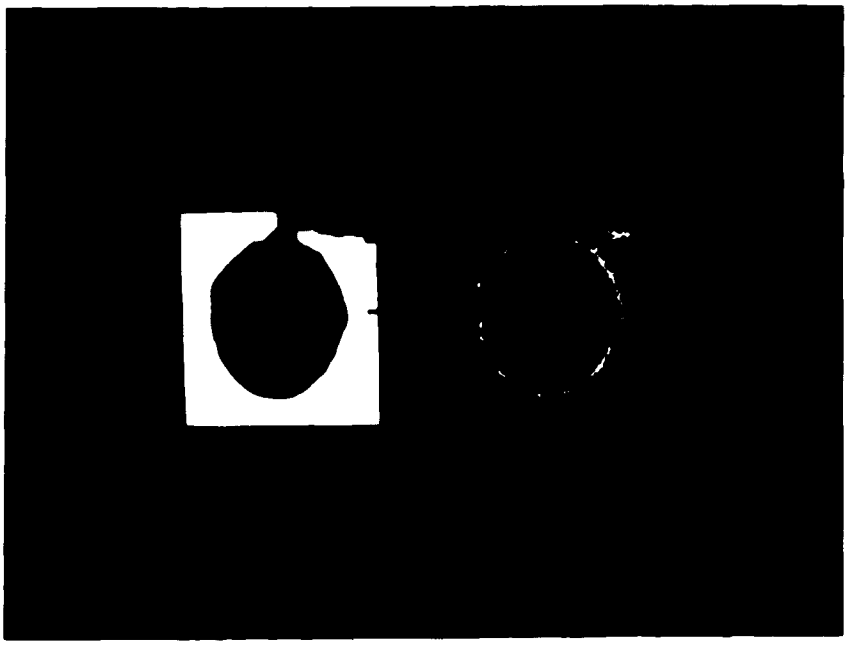


Fig. 50a,b

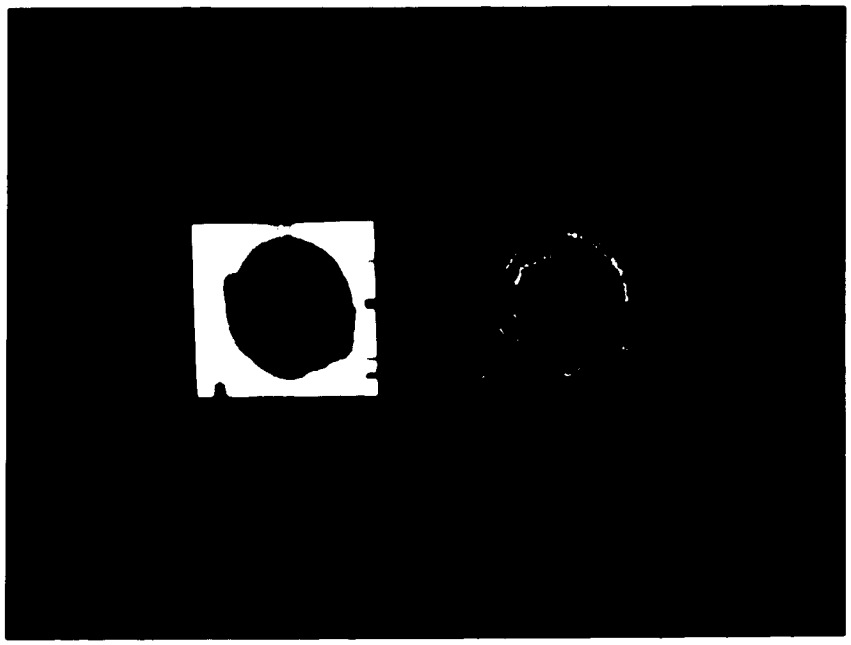


Fig. 51a,b

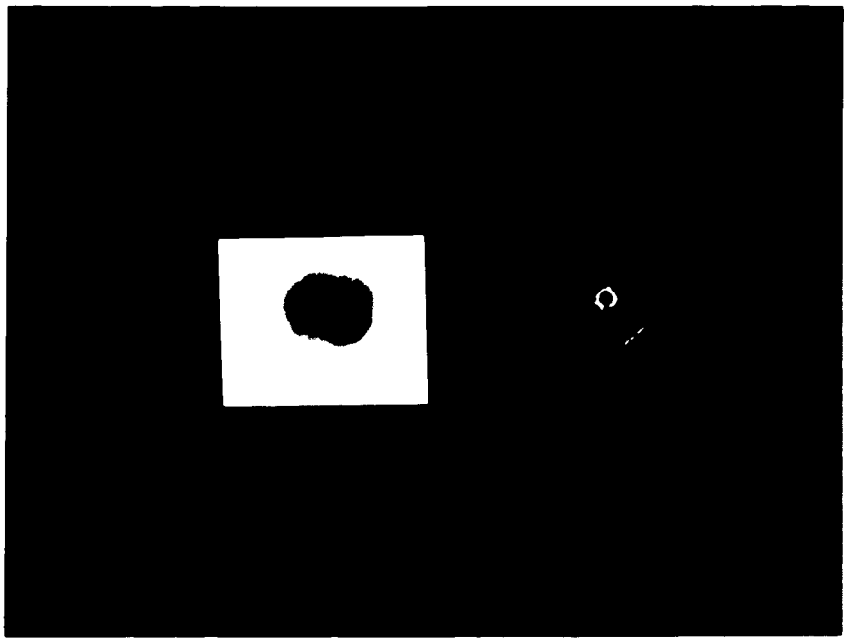


Fig. 52a,b

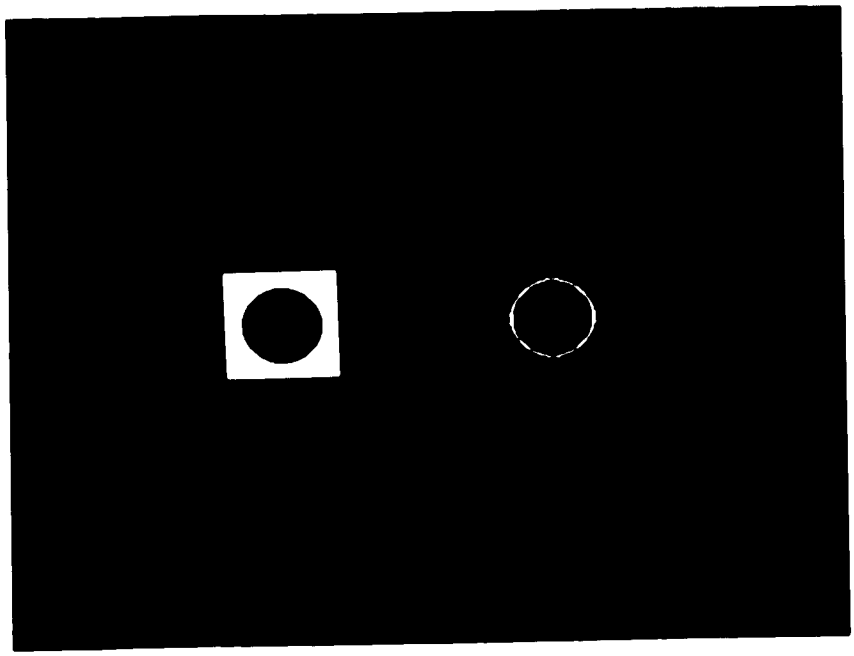


Fig. 53a,b

Fig. 54a,b,c

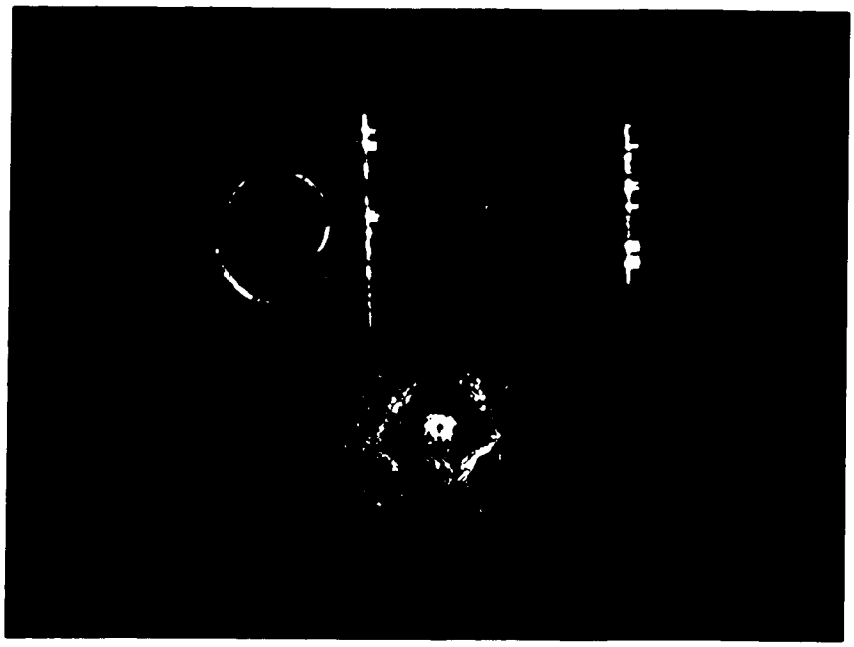


Fig. 55a,b,c

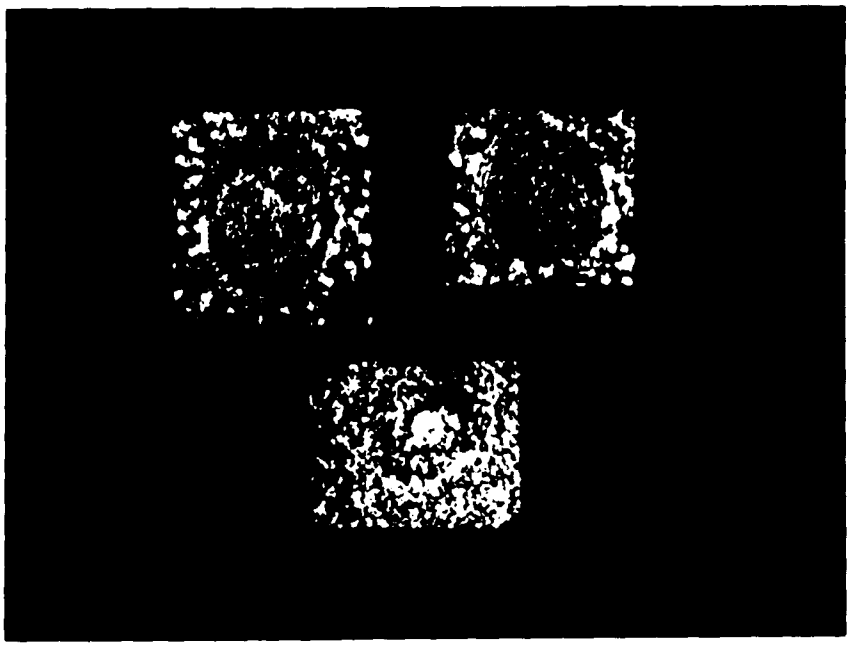


Fig. 56a,b,c

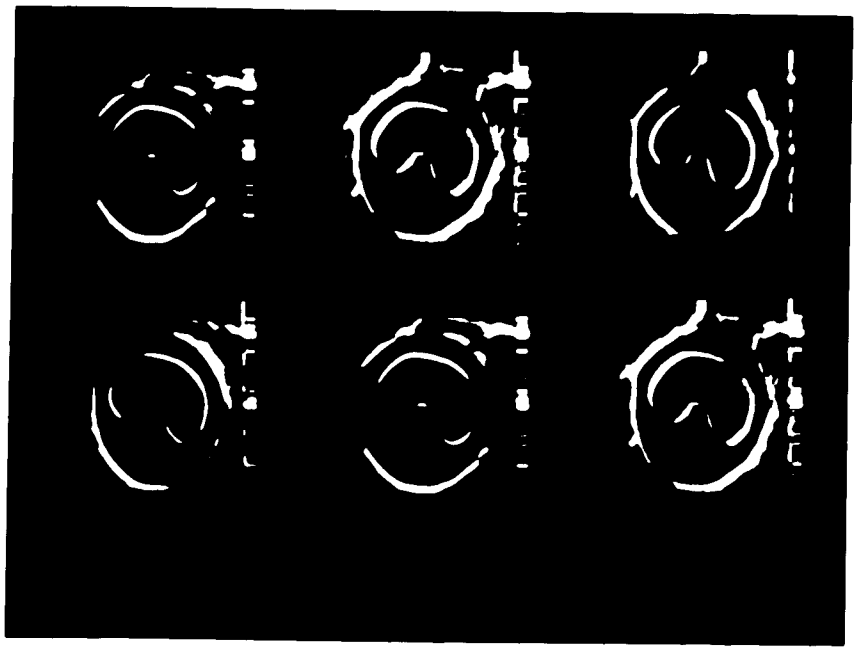


Fig. 57a,b,c,d,e,f

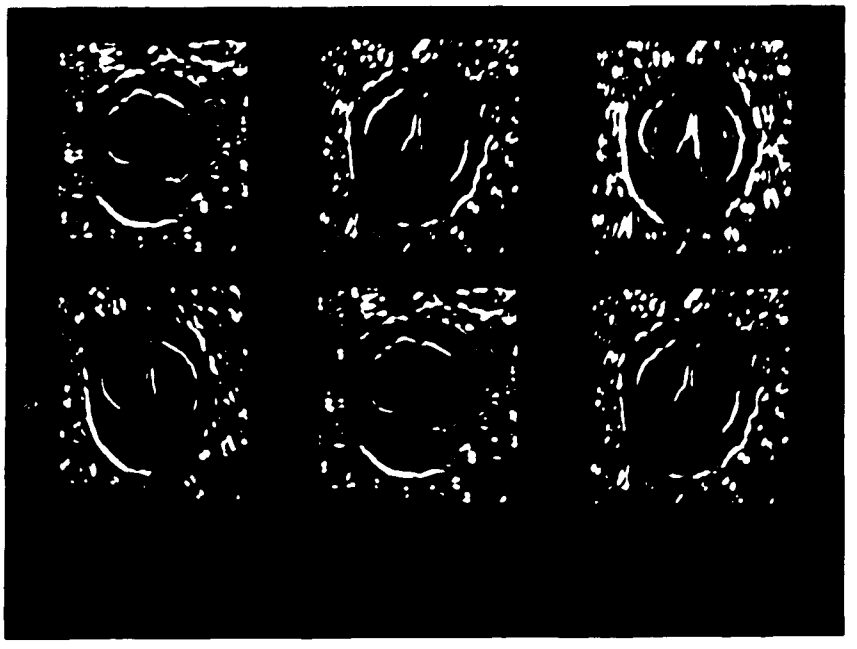


Fig. 58a,b,c,d,e,f

APPENDIX A

A description of the normal parameterization procedure is given in this appendix.

The most common representation of a straight line is of the form:

$$y = Ax + B \qquad (1)$$

This equation must be satisfied by any point (x,y) which belongs to the line. A is the slope of the line and B the y intercept of the line (the value of y where the line crosses the y axis). The problem with this representation is that as a line becomes practically parallel to the y axis, the values of A and B grow without bound and quantizing these values becomes impossible. A parameterization which avoids these problems is the normal parameterization in which a straight line is represented by a duple (R,T) where T is the angle of its normal and x axis and R is the distance from the line to the origin (see Fig. I). In this representation, points belonging to the line will satisfy the equation:

$$X\,COS(T) + Y\,SIN(T) = R \qquad (2)$$

With T restricted to [0,Pi] the normal parameters for a lines are unique.

This parameterization has the following properties:

a. It can easily be seen that sets of colinear points in the (x,y) plane correspond to sinusoidal curves with a common point of intersection in the (R,T) plane. The common point, say (R0,T0) defines the straight line passing through the colinear points.

b. A point in the (x,y) plane is represented by a sinusoidal curve in the (R,T) plane.

c. A point in the (R,T) plane is represented by a straight line in the (x,y) plane.

d. Points lying on the same straight line in the (x,y) plane correspond to curves through a common point in the (R,T) plane.

e. Points lying on the same curve in the (R,T) plane correspond to lines through the same points in the (x,y) plane.

REFERENCES

1. Nevatia, R., "A color edge detector and its use in scene segmentation", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-7, no. 11, November 1977, 820-826.
2. Duda, R.O. and Hart, P.E., "Use of the Hough transformation to detect lines and curves in pictures", Comm. ACM 15, no. 1, 1972, 11-15.
3. Rosenfeld, A., "Picture Processing by Computer", Academic Press, New York, 1969.
4. Hough, P.V.C., "Method and means for recognizing complex patterns", U.S. Patent 3,069,654, Dec. 18, 1962.
5. Shanmugam, K.A., Dickey, F.M. and Green, J.A., "An optimal frequency domain filer for edge detection in digital images", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, January 1979, 39-47.
6. Slepian, D. and Pollak, H.O., "Prolate spheroidal wave functions, Fourier analysis and uncertainty-I", Bell Syst. Tech. J., 1961, Vol. 40, pp. 43-46.
7. Landau, H.J. and Pollak, H.O., "prolate spheroidal wave functions, Fourier analysis and uncertainty-II", Bell Syst. Tech. J., 1961, Vol. 40, pp. 65-80.
8. Landau, H.J. and Pollak, H.O., "Prolate spheroidal wave functions, Fourier analysis and uncertainty-III: The dimension of the Space of Essentially Time- and Ban-Limited Signals", Bell Syst. Tech. J., July 1962, Vol. 41, pp. 1295-1336.
9. Slepian, D., J. Math. Phys., MIT, 1965, Vol. 44, pp. 93-99.
10. Striefer, W., "Optical resonator modes-Rectangular resonators of spherical curvature", J.O.S.A., Jul. 1965, Vol. 55, pp. 868-879.
11. Modestino, J.W. and Fries, R.W., "Edge detection in noisy images using recursive filering", Comput. Graph. Image Processing, 1977, Vol. 6, pp. 409=433.
12. Hueckel, M.H., "An operator which locates edges in digital pictures", JACM, Vol. 18, no. I, January 1971, 113-125. Erratum in 21, 1974, 350.
13. Hueckel, M.H., "A local visual operator which recognizes edges and lines", JACM, Vol. 20, no. 4, October 1973, 634-647.
14. Duda, R.O., and Hart, P.E., "Pattern Classification and Scene Analysis", Wiley, New York, 1971.
15. Mero, L. and Vassy, Z., "A simplified and fast version of the Hueckel operator for finding optimal edges in pictures", in Proc. 4th Intl. COnf. on Artificial Intelligence, Tbilisi, USSR, Sept. 1975, pp. 650-655.
16. Hummel, R.A., "edge detection using basis functions", Univ. of Maryland technical report TR-569, 1977.

17. Modestino, J.W. and Fries, R. W., "Edge Detection in noisy images using recursive digital filtering," Comput. Graphics Image Processing, vol. 6, 1977, pp. 409-433.

18. Frei, W., "A quantitative model of color vision", in Univ. Southern California USCIPI Report 540, Sept. 1974, pp. 69-83.

19. Nevatia, R., "Locating object boundaries in textured environments", IEEE Trans. Comput., Nov. 1976, Vol. 25, no. 11, pp. 1170-1175.

20. Robinson, G.S., "Edge detection by compass gradient masks", Computer Graphics and Image Processing", Vol. 6, no. 5, October 1977, 492-501.

21. Gonzalez, R.C. and Wintz, P., "Digital Image Processing", Addison-Wesley, Massachusetts, 1977, pp. 320-344.

22. Freeman, H., "Boundary encoding and processing", in Picture Processing and Psychopictorics, Academic Press, New York, 1970, pp. 241-266.

23. Matheron, G., Random sets and integral geometry", Wiley, New York, NY, 1975.

24. Serra, J., "Image analysis and mathematical morphology, AcademicPress, New York, NY, 1982.

25. Serra, J., and Klein, J., "The texture analyzer", J. Microscopy, Vol. 95, pp. 349-356, 1972.

26. Marr, D. and Hildreth, E., "Theory of edge detection", AI memo 518, MIT AI lab, April 1979. Also Proc. R. SOc. Lond. B., 1980, 207, 187-217.

27. Logan, B.F., Jr., "Information in the zero-crossings of bandpass signals", Bell. Syst. Tech. J. 1977, Vol. 56, pp. 487-510.

28. Buxton, B., "Early Image processing-structural techniques motivated by human visual response", Noate on lectures given at the University of Surrey, Mar.

29. Slepian, D. and Pollak, H.O., "Prolate spheroidal wave functions", Fourier analysis and uncertainty-I", Bell Syst. Tech. J., 1961, Vol. 40, pp. 43-46.

30. Landau, H.J. and Pollak, H.O., "Prolate spheroidal wave functions, Fourier analysis and uncertainty-II", Bell Syst. Tech. J., 1961, Vol. 40, pp. 65-80.

31. Landau, H.J. and Pollak, H.O., "Prolate spheroidal wave functions, Fourier analysis and uncertainty-III: The dimension of the Space of Essentially Time- and Band-Limited Signals", Bell Syst. Tech. J., July 1962, Vol. 41, pp. 1295-1336.

32. Lettvin, J.Y., Matturana, H.R., McCulloch, W.S., and Pitts, W.J., "What the frog's eye tells the frog's brain", Proceedings of Institute of Radio Engineers 47 (1959): 1940-1951.

33. Canny, J.F., "Finding edges and lines in images", MIT AI Lab. Tech. Rep. TR-720, 1983.

34. Wilson, H.R. and Bergen, J.R., "A found mechanism model for spatial vision", Vision Res., 19, pp. 19-31.

35. Rosenfeld, A. and Kak, A.C., "Digital picture processing", New York Academic Press, 1976.

36. Marr, D. and Poggio, T., "A computational theory of human stereo vision", Proc. R. Soc. Lond. B 204, 1979, pp. 301-328.

37. Marr, D., "Early processing of visual information", Phil. Trans. R. Soc. Lond. B 275, 1976, pp. 483-524.

38. Martelli, A., "Edge detection using heuristic search methods", Computer Graphics and Image Processing I, 2 (Aug. 1972), pp. 169-182.

39. Martelli, A., "Contour detection in noisy pictures using heuristic search methods", Proc. Ist. Int. Joint Conf. on Pattern Recognition, Washington, 1973, pp. 375-388.

40. Montanari, U., "On the optimal detection of edges in noisy pictures", COmm. ACM 14, May 1975, 5, pp. 335-345.

41. Prewitt, J.M.S. and Mendelsohn, M.L., "The analysis of cell images", Ann. N.Y. Acad. Sci.: 128, 1966, 1035-1053.

42. Weszka, J.S., "A survey of threshold selection techniques", Computer Graphics and Image Processing, 7, pp. 259-265, 1978.

43. Doyle, W., "Operations useful for similarity-invariant pattern recognition", J. Assoc. Comput. Mach. 9, 1962, 259-267.

44. Johannsen, G. and Bille, J., "A threshold selection method using information measures", Proc. of the 6th International Conference on Pattern Recognition", Oct. 1982 (IEEE 82CH1801-0).

45. Weszka, J.S. and Rosenfeld, A., "Threshold selection techniques, 5", University of Maryland Computer Science Center TR-349, Feb. 1975.

46. Bartz, M.R., "Optimizing a video processor for OCR", Proc. of the Internat. Joint COnference on Artificial Intelligence, Mitre Corp., Bedford, Mass., 1969: 79-90.

47. Ullman, J.R., "Binarization using associative addressing", Pattern Recognition: 6, 1974, 127-135.

48. Morrin, T.H., "A black-white representation of a gray-scale picture", IEEE Trans. Computers: 23, 1974, 184-186.

49. Panda, D.P., "Segmentation of FLIR Images by Pixel Classification", Univ. of Maryland Computer Science Center TR-508, Feb. 1977.

50. Katz, Y.H., "Pattern recognition of meteorological satellite cloud photography", in Proc. Third Symposium on Remote Sensing of Environment, Institute of Science and Technology, Univ. of Michigan, Feb. 1965, 173-214.

51. Chow, C.K. and Kaneko, T., "Boundary detection of radiographic images by a threshold method", in Proc. IFIP Congress 71, North-Holland, Amsterdam, Booklet TA-7, 1972, 130-134.

52. Jeanty, H.H., Barba, J., Gil, J., "Edge detection using local statistics and binary templates", presented a the 7th International COngress for Stereology, Sept. 1987 (caen, France). To appear in the proceedings of that conference, in ACTA STEREOLOGICA, December 1987.

53. Flammer, C., "Spheroidal wave functions", Stanford, CA., Stanford Univ. Press, 1956.

54. Shaw, G.B., "Local and regional edge detectors: Some comparisons", Computer Graphics and Image Processing, Vol. 9, No. 2, February 1979, 135-149.

55. Abdou, I.E., "Quantitative methods of edge detection", Ph.D. Thesis, University of Southern California, July 1978. Also USCIPI Report 830.

56. Beaudet, P.R., "Rotationally invariant image operators", in proceedings of the Fourth International Joint Conference on Pattern Recognition (IJCPR-78) held in Kyoto, Japan, November 7-10, 1978, 579-583.

57. Brice, C.R. and Fenema, C.L., "Scene analysis using regions", Artificial Intelligence Group Technical Note 17, Stanford Research Institute, April 1970.

58. Canny, J.F., "Finding edges and lines in images", MIT Master's Thesis, MIT AI-TR-720, June 1983.

59. Davis, L., "A survey of edge detection techniques", TR-273, University of Maryland, Computer Science Center, 1973.

60. Fenema, C.L. and Brice, C.R., "Scene analysis of pictures using regions", Artificial Intelligence Journal I, 1970, 205-226.

61. Gonzalez, R.C. and Wintz, P., "Digital Image Processing", Addision-Wesley, Massachusetts, 1977, pp. 320-344.

62. Griffith, A.K., "Mathematical models for Automatic line detection", Journal of the ACM, vol. 20, no. I, January 1973, p. 62.

63. Griffith, A.K., "Edge detection in simple scenes using a priori information", IEEE Transactions of Computers, vol. C-22, April 1973, 371-381.

64. Haralick, R.M., "Edge and region analysis for digital image data", Computer Graphics and Image Processing, August 1981, 285-291.

65. Haralick, R.M., "Digital step edges from zero crossings of second directional derivatives", IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. PAMI-6, no. I, January 1984, 58-68.

66. Hough, P.V.C., "Method and means for recognizing complex patterns", U.S. Patent 3,069,654, December 18, 1962.

67. Martelli, A., "Edge detection using hsuristic search methods", Computer Graphics and Image Processing, I, 1972, 169-182.

68. Martelli, A., "An application of heuristic search methods to edge and contour detection", Comm. ACM 19, 1976, 73-83.

69. Montanari, U., "On the optimal detection of curves in noisy pictures", Comm. ACM 14, May 1971, 335-345.

70. O'Gorman, F., "Edge detection using walsh functions", Proc. AISB, July 1976, p. 195. Also: Artificial Intelligence 10, 1978, 215-233.

71. O'Gorman F. and Clowes, M., "Finding picture edges through collinearity of feature points 4", IEEE Transactions on Computers, Vol. C-25, no. 4, April 1976.

72. Ohlander, R.B., "Analysis of natural scenes", Dept. of Computer Science, Carnegie-Mellon University, April 1975. (Ph.D. Thesis).

73. Prewitt, J.M.S., "Object Enhancement and Extraction", in Picture Processing and Psychopictorics, B.S. Lipkin and A. Rosenfeld, Eds., Academic Press, New York, 1970.

74. Rosenfeld, A., "Picture processing by Computer", Academic Press, New York, 1969.

75. Shirai, Y., "A context sensitive line finder for recognition of polyhedra", Artificial Intelligence, 4, Summer 1973, 95-119.

76. Somerville, C. and Mundy, J.L., "One pass contouring of images through planar approximation", Proc. of the 3rd International Joint Conference on Pattern Recognition (IJCPR-76), November 1976 (IEEE 76CH1140-3C).

77. Nathan, R., "Picture enhancement for the moon, mars, and man in pictorial pattern recognition", (C.G. Cheng, ed.), 1968, pp. 239-266. Thompson, Washington, D.C.

78. Tukey, J.W., "Exploratory data analysis", 1971, Addison-Wesley, Reading, Massachusetts.

79. Kruger, R.P., Dwyer, S.J., III, and Storvick, T.S., "Techniques for automatic isolation and measurement of shock waves in metals, Univ. of Missouri, College of Engineering Report, Sept. 1970, Univ. of Missouri, Columbia.

80. Wong, R.Y., Hall, E.L., and Rouge, J., "Hierarchical search for image matching", 1977, Proc. IEEE Conf. Decision Control, Clearwater, FL., Dec. 1976.

81. Pratt, W.K., "Correlation techniques of image registration", IEEE Trans. Aeorsp. Electron. Syste. 1974, 10, No. 3, pp. 353-358.

82. Van Vliet, L. and Young, I.T., "A Nonlinear Laplace Operator as Edge Detector in Noisy Images", Computer Vision, Graphics, and Image Processing, Feb. 1989, Vol. 45, No. 2, pp. 167-195.

83. Beckers, A.L., Metingen van Parameters voor Niet-Lineaire Objectgrootte-Filers in Beelden, Ingenieur's Thesis, Dept. of Applied Physics, Delft University of Technology, 1986 [Dutch].

84. Lee, J.S.L., Haralick, R.M. and Shapiro, L.S., "Morphologic Edge Detection in Proceedings", 8th Int. Conf. Pattern Recognition, Paris, 1986, pp. 369-373.

85. Pratt, W.K., "Digital Image Processing, Wiley, New York.

86. McLean, G.F., and jernigan, M.E., "Hierarchical Edge Detection", Computer Vision, Graphics, and Image Processing, Dec. 1988, Vol. 44, No. 3, p. 350-366.

87. Yanowitz, S.D. and Bruckstein, A.M., "A New Method for Image Segmentation", Computer Vision, Graphics, and Image Processing, April 1989, Vol. 46, No. 1, pp. 82-95.

88. Baram, Y. and Margalit, M., "Surface Fitting by Pseudo-Potential Functions", IEEE Trans, Geosci. Remote Sensing GE-22, 1984, pp. 455-461.

89. Dyn, N. and Levin, D., "Bell-Shaped Basis Functions for Surface Fitting in Approximation Theory and Applications", (Z. Zeigler, Ed.) pp. 113-129, Academic Press, New York, 1981.