

# Edge Detection Applied to Satellite Imagery of the Oceans

RONALD J. HOLYER AND SARAH H. PECKINPAUGH

**Abstract**—A computer edge-detection algorithm is developed for automatic delineation of mesoscale structure in digital satellite IR images of the ocean. The popular derivative-based edge operators are shown to be too sensitive to edge fine-structure and to weak gradients to be useful in this application. The new edge-detection algorithm is based on the gray level co-occurrence (GLC) matrix, which is commonly used in image texture analysis. The cluster shade texture measure derived from the GLC matrix is found to be an excellent edge detector that exhibits the characteristic of fine-structure rejection while retaining edge sharpness. This characteristic is highly desirable for analyzing oceanographic satellite images. The method is evaluated on an Advanced Very High Resolution Radiometer (AVHRR) image of the Gulf Stream region.

## I. INTRODUCTION

INFRARED (IR) images of the ocean obtained from satellite sensors are widely used for the study of ocean dynamics. Fig. 1 shows an IR image of the Gulf Stream obtained from the Advanced Very High Resolution Radiometer (AVHRR) aboard the NOAA-7 satellite. The North American continent extending from Florida to Newfoundland is blanked out at the left side of the image, which has been remapped to a Mercator projection. Brightness in this IR image is inversely proportional to the ocean surface temperature (dark areas represent warmer temperatures and light areas represent colder temperatures). Note the presence of the warm (dark) Gulf Stream, the cooler (lighter) waters on the continental shelf from Cape Hatteras northward, and the turbulent flow patterns resulting from the dynamic interaction of the various water masses of contrasting surface temperatures. Vortices (areas of closed circulation) within this turbulent flow pattern are called eddies. Fig. 2 shows the central portion of Fig. 1 at 1.5 km/pixel spatial resolution. Note in Fig. 2 the existence of a Gulf Stream meander at the top of the image, and three cold eddies to the south of the Gulf Stream.

The Gulf Stream and its associated eddies are examples of mesoscale features. ("Mesoscale" is the name commonly applied to features existing on spatial scales on the



Fig. 1. NOAA-7 AVHRR image showing sea surface temperatures in the Gulf Stream region for April 28, 1983. The image is AVHRR channel four with a spectral bandpass of 10.5 to 11.5  $\mu\text{m}$ . The image is subsampled to a Mercator projection at 4 km/pixel resolution.

order of 50 to 300 km.) Mesoscale features are important to the study of ocean dynamics, to fisheries, and to many other diverse interests. Since satellite IR images often depict mesoscale features clearly (in the absence of cloud cover or excessive atmospheric water vapor), the use of AVHRR imagery for various oceanographic applications is expanding rapidly. The National Oceanic and Atmospheric Administration produces operational facsimile products depicting ocean mesoscale features. Satellite imagery is a significant data source for these products. Within the research community AVHRR data analysis is also increasing in volume. For example, Brown *et al.* [1] studied a ten-year set of IR imagery in order to compile statistics on Gulf Stream rings. Thus, with the proliferation of high-volume AVHRR image applications, it becomes highly desirable for certain applications to move from the labor-intensive manual interpretation of IR imagery toward a capability for automated interpretation of these images. The complete automation of the oceanographic image interpretation function is probably not feasible, but one can begin to address certain subsets of the

Manuscript received October 13, 1987; revised August 22, 1988. This work was supported by the Office of Naval Technology, Program Element 62435N.

The authors are with the Computer Science Section, Naval Ocean Research and Development Activity, Department of the Navy, Stennis Space Center, MS 39529-5004.

IEEE Log Number 8824336.

U.S. Government work not protected by U.S. copyright

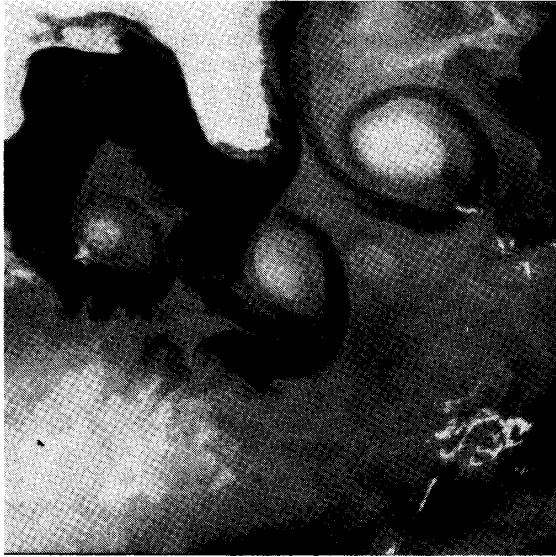


Fig. 2. The central portion of Fig. 1 expanded to show several typical eddies at 1.5 km/pixel spatial resolution.

problem with present-day image processing and artificial intelligence techniques.

Several previous studies have addressed the automation of the analysis of IR satellite imagery for mesoscale features. Gerson and Gaborski [2] and Gerson *et al.* [3] investigated the detection of the Gulf Stream in IR images from the Geostationary Operational Environmental Satellite (GOES). This satellite, with its 0.5°C thermal sensitivity and 8-km spatial resolution, gives a much coarser representation of the ocean surface temperature structure than that available in the present study, where the AVHRR data has 0.1°C thermal sensitivity and 1.5-km spatial resolution. Gerson and Gaborski [2] used a hierarchical approach where  $16 \times 16$  pixel ( $128 \times 128$  km) “frames” within the image were evaluated for the possibility of containing the Gulf Stream. Frames flagged as Gulf Stream possibilities were then further evaluated to determine the exact location of the Stream within the frame by looking at statistics based on  $5 \times 5$  pixel ( $40 \times 40$  km) “local neighborhoods.” A large number of first- and second-order statistics were evaluated in the study, and decisions were based on the Fisher Linear Discriminant function for both the frame and local neighborhood processing. As a result of this study three statistical features—mean, standard deviation, and difference histogram maximum entropy—were shown to give good indication of the presence of the Gulf Stream within a frame. Skewness of the  $5 \times 5$  pixel local neighborhood was found to be the best statistic for locating the front within the frame. Gerson and Gaborski [2] stated that they had clearly demonstrated that a fully automated system for locating fronts in satellite imagery is feasible.

As an outgrowth of the work reported in [2] and [3], Coulter [4] performed automated feature extraction stud-

ies using the higher resolution (1 km) AVHRR data. Mean, standard deviation, and gradient in a  $3 \times 3$  pixel window were combined with *a priori* probabilities based on a large historical data set in order to classify each pixel according to Bayes’ decision theory. Boundaries between water classes then became an indication of edge locations within the image. Promising results were reported for this method for locating the Gulf Stream. However, Coulter [4] reported that eddy classification is more difficult because their historical statistics are less stationary. Indeed, the requirement for *a priori* knowledge and stationary statistics is a limiting factor in the use of this method.

Janowitz [5] studied the automatic detection of the Gulf Stream eddies using AVHRR data. Recognizing that edges in this high-resolution IR imagery are too noisy, i.e., have too much fine-structure detail, for effectively locating eddies, Janowitz started with filtering to smooth the image. Smoothing was followed by an image simplification algorithm described in [6]. The smoothed and simplified image was then transformed into a binary edge image by a Kirsch edge detector (as described in [7]). The last step was ellipse detection on the binary edge image as a means of locating eddies. The centers of all three cold eddies in the test image (same image as shown in Fig. 2) were correctly identified and no eddy false alarms were reported. The Janowitz study lends further support to the feasibility of automated mesoscale feature detection.

Nichol [8] uses a region adjacency graph to define spatial relationships between elementary connected regions of constant grey level called atoms. Eddy-like structure is then identified by searching the graph for isolated atoms of high temperature that are enclosed by atoms of lower temperature (for the case of warm eddies). Although satisfactory emulation of human extraction of eddy structure is claimed for this method, Nichol [8] does point out that not all enclosed uniform areas identified by this method will correspond to real ocean structure. We agree with this conclusion. It seems possible that relatively uniform areas such as the Sargasso Sea that might be devoid of eddies could possibly contain more uniform areas than the region near the Gulf Stream where eddies exist but the spatial patterns of sea surface temperature are very complex. The “false alarm” statistics for Nichol’s method should be examined.

Although these previous investigators have reported some success in their initial efforts to automate the oceanographic analysis of satellite IR images, the present study represents a departure from the previous work. This departure arises from a difference in philosophy between the present work and that previously reported. The previous authors were seeking conventional image processing algorithms that would perform the entire mesoscale feature detection function. In contrast, the present study is part of a larger effort to blend conventional image processing with statistical and artificial intelligence (AI) techniques. As examples of related analysis techniques, Carter [9] and Molinelli and Flanigan [10] have examined the use of complex empirical orthogonal functions for representa-

tion of the North Wall of the Gulf Stream, while Lybanon *et al.* [11], Lybanon [12], and Thomason and Blake [13] have developed an expert system that embodies the mesoscale dynamics of the Gulf Stream region in a rule-based expert system. As a part of this integrated approach we are seeking a low-level image segmenter that will provide useful edge information in a form that can be easily handled by subsequent statistical or AI modules where the feature detection work can be completed. We have therefore developed a new edge-detection algorithm that is an effective preprocessor that transforms satellite IR images into relatively simple line drawings.

We anticipate that the next step after edge detection in the automated analysis of IR imagery of the ocean would be automated labeling of the edge line segments to indicate association with certain mesoscale oceanographic features such as the North Wall of the Gulf Stream or a cold-core ring. The feature labeling step would involve ancillary information, previous analyses, and oceanographic context to assign labels to line segments. The objective of this edge-detection work is to provide clean, simple line representations of thermal gradients for input to such a subsequent feature labeling step.

## II. EDGE DETECTION

Edge detection is one of the most common problems encountered in image analysis. Applications of edge detection include the fields of remote sensing, industrial inspection, optical character recognition, medical imaging, robotics, and many others. Edge detection is often an important first stage in many types of image segmentation such as the present case, where edge detection can be used to simplify complex imagery in preparation for subsequent feature identification. The plethora of edge operators is very difficult to evaluate and compare. Trade-offs between edge detectability, noise sensitivity, and computational efficiency are always involved in selecting the appropriate edge operator for a given application. Numerous authors, among them Davis [14], Levialdi [15], and Kunt [16], have surveyed and compared edge operators. We will not attempt a complete discussion of existing edge-detection techniques here, but will comment briefly on some of the more common approaches in order to show the inadequacy of these methods for finding edges of mesoscale features in satellite IR imagery of the ocean.

The most common and historically earliest image edge operators were based on approximation of the intensity gradient within the image. The two orthogonal components of the gradient vector  $g_x$  and  $g_y$  are normally calculated separately then combined to give the gradient magnitude  $g$

$$g = (g_x^2 + g_y^2)^{1/2} \quad (1)$$

or for increased computational efficiency

$$g = |g_x| + |g_y|. \quad (2)$$

Investigators have proposed many ways to estimate  $g_x$  and  $g_y$  from a discrete image function. Normally these esti-

mation techniques are implemented as a convolution of the image with an appropriate gradient-approximating kernel. Examples of such kernels are given in Table I. Of those shown in the table, the Sobel operator is the most widely used. Davies [22] estimates that the Sobel operator probably accounts for some 70 percent of the usage of all edge-detection schemes.

Another common approach to edge detection arises from the mathematical Laplacian. The discrete Laplacian can also be implemented as a convolution with a  $3 \times 3$  kernel. The Laplacian kernel is also given in Table I. The Laplacian, being an approximation to the second derivative, has the disadvantage that it tends to enhance noise in the image.

The gradient and Laplacian techniques, based on first and second derivatives, respectively, share a common shortcoming in the present application. Namely, they are too sensitive to noise, to fine-structure in the edges, and to weak gradients. To illustrate this problem, the  $g_x$  and  $g_y$  Sobel operators, selected as typical of the  $3 \times 3$  kernel derivative approximation methods, have been applied to Fig. 2, and the gradient magnitude has been calculated using (1). Fig. 3 is an image of Sobel gradient values. The mesoscale features have been captured in the Sobel image, but many areas of the image void of mesoscale structure also exhibit edges comparable to those contained within the mesoscale features. Furthermore, the edges within the mesoscale features are very complex structures that show great detail. In many applications this enhancement of detail by the edge operator is desirable. However, a preferable result in the present application would be an edge operator that resulted in simple, smooth representations of edges within the mesoscale features and no edges at all in the oceanographically bland areas of the image. This desire to eliminate the fine detail of the mesoscale structure is what prompted Janowitz [5] to perform smoothing and simplification prior to edge detection. Of course, the derivative-based edge operators could be made less sensitive to detail and weak edges by increasing the size of the convolution kernel used to estimate the derivatives. However, larger kernels produce a "blurring," which results in edges that are broad bands rather than sharp boundaries.

Edge detectors other than those consisting of convolution masks have also been developed. One example in this category is the sigma filter edge detector [23]. This edge detector is particularly noteworthy here since it is based on the sigma filter [24], which is purported to have excellent noise smoothing characteristics. The design objective for the sigma filter appears to be quite similar to the smoothing objective in the present study. In the sigma filter edge detector, pixels in the local  $3 \times 3$  neighborhood are divided into three groups. Pixels with intensity values exceeding the intensity value of the center pixel by more than some threshold, called delta, are put into group A. Pixels with intensity values less than the center intensity minus delta are put into group C. All remaining pixels that have intensities lying between the center pixel plus

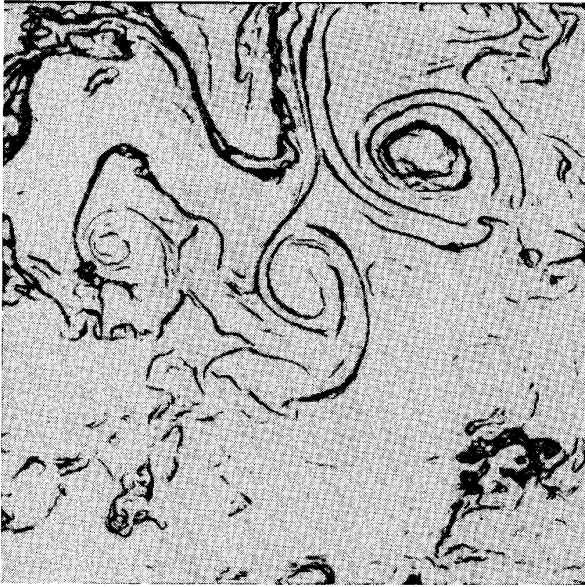


Fig. 3. Result of applying a Sobel edge operator to Fig. 2.

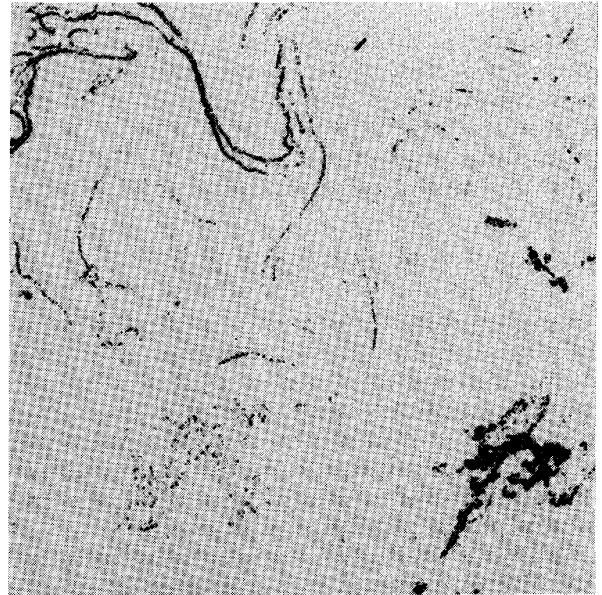


Fig. 4. Result of applying a sigma filter edge detector to Fig. 2.

 TABLE I  
 KERNELS OF WELL-KNOWN IMAGE EDGE OPERATORS

	$g_x$	$g_y$
Roberts [17]	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Sobel [18]	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Prewitt [19]	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
Frei-Chen [20]	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix}$
Laplacian [21]	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	

delta and the center pixel minus delta are placed in category B.

Either group A or group C, whichever has fewer members, will be merged with group B. The difference between the mean values of the two remaining groups is interpreted as the edge magnitude. In the event that all pixels fall within group B, the delta value is lowered (typically halved) and the process is repeated until an edge is detected or a minimum delta is reached. Lee [23] notes that ramp edges and isolated pixel edges cause problems for the sigma filter edge detector. Also, Lee [23] warns that care should be exercised when the window size is larger than  $5 \times 5$ . This makes it difficult to add additional smoothing to the sigma filter by increasing window size.

Fig. 4 is an example of the sigma filter edge detector applied to the IR image of Fig. 2. The window size in this

case is  $3 \times 3$  and the minimum delta permitted is five. The sigma filter produces smoother edges and has better noise rejection than the Sobel operator, but the degree of smoothing is still not sufficient for the present application and many edge detections result that are not associated with mesoscale oceanographic structure.

It is desirable to develop an edge-detection technique that would reject detail in the edges to a higher degree than shown in Figs. 3 and 4 while maintaining sharp edge definition and accurate localization of the edge. This objective is difficult because edges are characterized by much high-frequency content; henceforth, an edge detector must be, in effect, a high-pass filter. Passing high frequencies almost guarantees that edge fine structure will be preserved. This paper presents a new edge-detection method that possesses these desired smoothing characteristics to a degree that is not obtainable from small convolution operators such as those shown in Table I.

### III. CLUSTER SHADE EDGE ALGORITHM

The edge algorithm developed here is based on the cluster shade texture measure, which is derived from the gray level co-occurrence (GLC) matrix. The GLC matrix has been widely used as a texture analysis tool, e.g., Haralick *et al.* [25] and Ballard and Brown [26]. Connors *et al.* [27] specifically point out that the GLC matrix contains edge information. Harlow *et al.* [28] used contrast and symmetry measures calculated from the GLC matrix to investigate edge detection on synthetic bar images. However, previous attempts at automated edge detection in real imagery based on the GLC matrix, or use of the cluster shade measure for edge detection, are not known to the authors.

The  $(i, j)^{\text{th}}$  element of the GLC matrix,  $P(i, j | \Delta x,$

$\Delta y$ ), is the relative frequency with which two image elements, separated by distance  $(\Delta x, \Delta y)$ , occur in the image, one with intensity level  $i$  and the other with intensity level  $j$ . Mathematically speaking, consider an  $M \times N$  pixel image or local neighborhood within an image with  $L$  intensity levels ranging from 0 to  $(L - 1)$ . Let  $f(m, n)$  denote the intensity level of the pixel at sample  $m$ , line  $n$ . Then

$$P(i, j | \Delta x, \Delta y) = \sum_{m,n} A \quad (3)$$

where  $A = 1 / [(M - \Delta x) \times (N - \Delta y)]$  if  $f(m, n) = i$  and  $f(m + \Delta x, n + \Delta y) = j$ . Otherwise,  $A = 0$ . Both pixels at locations  $(m, n)$  and  $(m + \Delta x, n + \Delta y)$  must lie within the  $M \times N$  image space.  $P(i, j | \Delta x, \Delta y)$  is therefore an  $L \times L$  matrix of second-order probabilities. In many texture problems several  $\Delta x, \Delta y$  combinations are analyzed so that one actually works with a set of GLC matrices, one for each  $\Delta x, \Delta y$  combination.

The elements of the GLC matrix could be combined in many different ways to give a single numerical value that would be a measure of edginess. One such measure, called cluster shade,  $S(\Delta x, \Delta y)$  [27], has been found here to be an especially good edge measure for the present problem.

$$S(\Delta x, \Delta y) = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i + j - \mu_i - \mu_j)^3 \cdot P(i, j | \Delta x, \Delta y) \quad (4)$$

where

$$\mu_i = \sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} P(i, j | \Delta x, \Delta y)$$

$$\mu_j = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} j P(i, j | \Delta x, \Delta y).$$

The quantities  $\mu_i$  and  $\mu_j$  are estimates of mean intensity based on weighted summations of rows and columns within the GLC matrix. We have chosen to detect edges in the present study by computing  $P(i, j | \Delta x, \Delta y)$  and subsequently computing  $S(\Delta x, \Delta y)$  in overlapping local neighborhoods within an image. The center point of the neighborhood is then replaced by the  $S(\Delta x, \Delta y)$  value computed from its neighborhood, thus creating a "cluster shade image" of  $S(\Delta x, \Delta y)$  values. Fig. 5(b) shows one such cluster shade image for a  $64 \times 64$  pixel portion of Fig. 2, which spans the north wall of the Gulf Stream (Fig. 5(a)). The intensity in Fig. 5(b) is given by cluster shade,  $S(\Delta x = 0, \Delta y = 0)$ . (A later section will point out that the  $\Delta x$  and  $\Delta y$  values have very little effect in the resulting edge detection. We therefore use  $\Delta x = 0, \Delta y = 0$  because of simplification of the mathematics, and the reduced computation time that results.) Note that (a)  $S(\Delta x, \Delta y)$  can assume either positive or negative (black or white) values, (b) the values are the largest in the vicinity of the north wall of the Gulf Stream, (c) the values are negative to one side of the wall and positive to the other side, and (d) the transition point from large positive

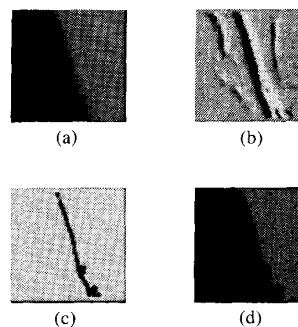


Fig. 5. Cluster shade detection algorithm applied to a  $64 \times 64$  pixel test image spanning the north wall of the Gulf Stream. (a) The original sub-image containing the north wall. (b) cluster shade image where the intensity of each pixel is proportional to cluster shade calculated over a  $5 \times 5$  pixel local neighborhood in (a), (c) result of applying the thresholded zero crossing algorithm to (b), (d) the edge from (c) overlain on (a).

to large negative values coincides with the exact location of the edge of the Gulf Stream. These observations suggest that one could detect edges by finding the significant zero crossings in the cluster shade image.

Significant zero crossings in the cluster shade image are found as follows. For each  $3 \times 3$  pixel neighborhood in the cluster shade image (Fig. 5(b)) the absolute value of the center pixel is tested against a defined threshold. If this value does not exceed the threshold, then a "0" will be output to the binary edge image. If, however, the absolute value of the center pixel exceeds the threshold, then a test of the neighboring pixels is performed. If the absolute value of any of the eight neighbors also exceeds the threshold but is opposite in sign from the center pixel, a "1" will be output to indicate the presence of an edge. This algorithm has the effect of marking the major zero crossings in the cluster shade image with 1's in the resulting binary edge image. Fig. 5(c) shows the result of applying this algorithm to the north wall test image.

Because edges are detected by finding zero crossings, precisely positioned lines result even if the GLC matrix is calculated over a large window. (The lines are formed two pixels wide because the zero crossing algorithm we use detects both the large positive value with a large negative neighbor and the large negative value with a large positive neighbor.) The cluster shade edge detector, therefore, does not give blurred or positionally uncertain edges when large windows are used. The desired edge-detection characteristics of retaining sharp edges while eliminating edge detail are thereby achieved. In addition to setting the window size to reject edge detail, the threshold used to determine which zero crossings are significant can also be adjusted to eliminate detection of weak edges. Therefore, two major tunable parameters are associated with this algorithm: window size and zero crossing threshold. The length and angle of the displacement vector could also be changed in an attempt to tune the performance of this edge detector. However, we have found that the displacement vector is not a significant contribu-

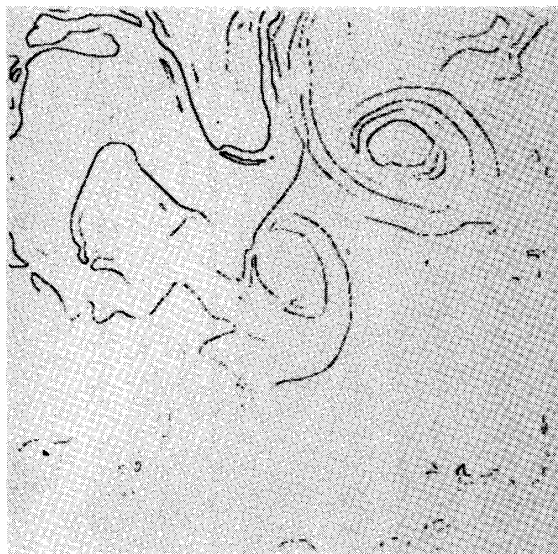


Fig. 6. Edges detected in Fig. 2 by the CSED algorithm with window =  $16 \times 16$  pixels,  $\Delta x$  and  $\Delta y$  displacements = 0 pixels, and zero crossing threshold = 20.



Fig. 7. Edges shown in Fig. 6 overlaid on Fig. 2.

tor to performance. The effects of changing parameter values are illustrated in a subsequent section.

To illustrate the cluster shade edge-detection (CSED) algorithm, the AVHRR image shown in Fig. 2 has been processed with a displacement vector of zero length, window size (local neighborhood) of  $16 \times 16$  pixels, and zero crossing threshold of 20. The result is shown in Fig. 6 as a binary edge image. The edge image is shown overlaid on the original image in Fig. 7. Note the delineation of mesoscale features in Fig. 6 is much "cleaner"

than that generated by the Sobel operator (Fig. 3) or sigma edge detector (Fig. 4).

#### IV. DEALING WITH CLOUD COVER

The test image in Fig. 2 is not typical of IR images of the ocean because it is almost completely cloud free. In practical application one must consider the fact that partial cloud cover will exist in nearly every image analyzed. Therefore, any useful edge-detection algorithm must be able to locate edges of mesoscale features in the image while ignoring cloud edges. If we can generate a mask that consists of all cloud-covered pixels within the image, the CSED algorithm can be made to ignore cloud edges by simply assigning a value of zero to the cluster shade for any case where one or more of the pixels within the GLC matrix local neighborhood are covered by the mask.

Cloud masks suitable for this purpose could be generated by several methods. The two most common are described below. First, one can threshold on the near-IR ( $0.72$  to  $1.0 \mu\text{m}$ ) channel of the AVHRR. In this spectral band the clouds, which are highly reflective, are always brighter than the highly absorptive ocean and cloud/water discrimination is therefore an easy task. The disadvantage of near-IR thresholding is that the method can be used only for daytime images. A second possibility for generating a cloud mask is to threshold on the IR image. Clouds are normally radiometrically colder than the sea surface, so this method will work in many cases on either daytime or nighttime imagery. The disadvantage is that clouds will not always be colder than the sea surface. This difficulty is especially serious when the image under analysis covers a large latitude range. Over a wide latitude range the clouds in the lower latitude portion of the image will often be warmer than the water at the higher latitudes.

Therefore, no single temperature threshold would mask out clouds over the entire image. Assuming one can produce a valid cloud mask by one of the above or by any other method, the CSED algorithm with mask testing in each local neighborhood will not detect cloud edges.

The fact that cloud rejection is implemented via masks opens up other possibilities. For example, masks could also be produced from bathymetry data, which could be used to force omission of edges in shallow water. Or masks could be generated on the basis of the mean position or last known position of the Gulf Stream in order to force detection on only those features within a certain distance from these predetermined positions.

To illustrate the detection of mesoscale features in the presence of partial cloud cover, a second subsection of Fig. 1 is selected. This area, to the east of that shown in Fig. 2, is shown in Fig. 8. A cloud mask generated by thresholding the near-IR image corresponding to Fig. 8 at an albedo value of 0.05 is shown in Fig. 9. Fig. 10 shows the mesoscale edges detected by the CSED algorithm applied to Fig. 8. Fig. 11 is Fig. 10 overlaid on the original image in Fig. 8. The CSED algorithm clearly detects those portions of major mesoscale features that are not cloud covered while ignoring the cloud-contaminated areas.

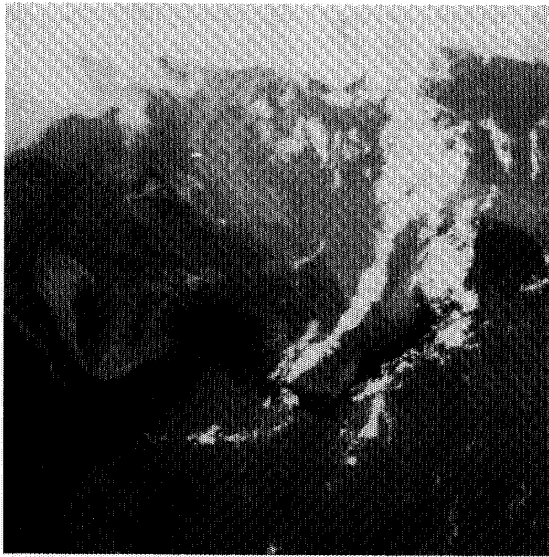


Fig. 8. A subsection from the upper right portion of Fig. 1 expanded to show typical cloud cover over mesoscale features at 1.5 km/pixel spatial resolution.



Fig. 10. Edges detected in Fig. 8 by the CSED algorithm with window =  $16 \times 16$  pixels,  $\Delta x$  and  $\Delta y$  displacements = 0 pixels, and zero crossing threshold = 20.

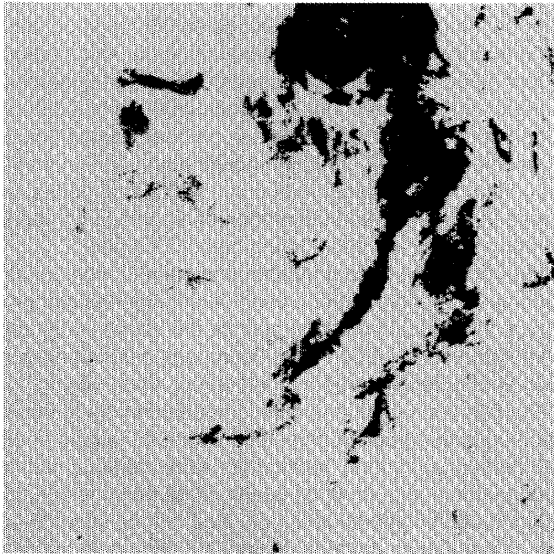


Fig. 9. Cloud mask for Fig. 8 generated by thresholding AVHRR channel 2 data at the 5-percent albedo level.



Fig. 11. Edges shown in Fig. 10 overlaid on Fig. 8.

## V. DISCUSSION

The CSED algorithm presented here involves these parameters: size of the local neighborhood over which cluster shade is calculated, length and orientation of the displacement vector used for GLC matrix calculations, and a cluster shade threshold used to determine which zero crossings in the cluster shade image are significant. For the oceanographic test image analyzed here, excellent results were obtained with a  $16 \times 16$  pixel local neighborhood, a zero length displacement vector, and a cluster

shade threshold of 20. A systematic study of the effects of changing each of these parameters has not been performed, but a discussion of their effects follows. Examples of changing these parameters are also included.

First, let us consider the choice of length and orientation of the displacement vector used for GLC matrix calculations. Several combinations of  $\Delta x$  and  $\Delta y$  were tried. Our observation (see Fig. 12 where various displacement vectors are applied to the same image) is that a displacement vector makes only a small difference in the edge de-

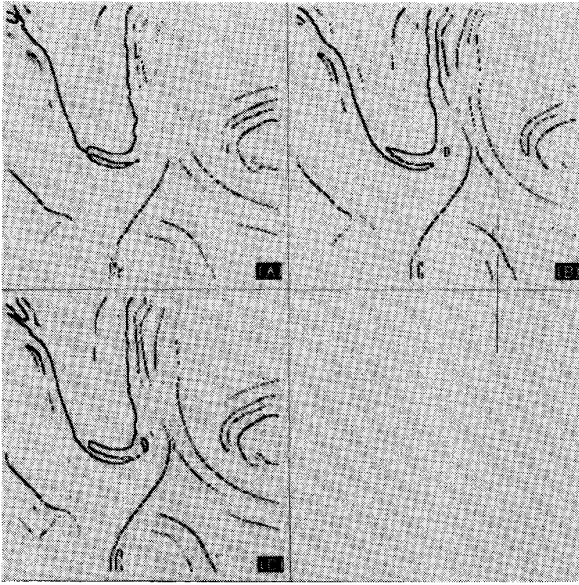


Fig. 12. Application of the CSED algorithm to a subsection of Fig. 2 with various values of the displacement vector. (a)  $\Delta x = 5$ ,  $\Delta y = 0$ . (b)  $\Delta x = 0$ ,  $\Delta y = 5$ . (c)  $\Delta x = 3$ ,  $\Delta y = 2$ .

tection result. If this is true then a zero length displacement vector can be selected to gain considerable computational advantage. For a zero length displacement vector, the following simplifications apply to the variables occurring in (4).

$$P(i, j | \Delta x, \Delta y) = 0, \text{ for } i \neq j \quad (5a)$$

$$\mu_i = \mu_j = \mu \quad (5b)$$

$$P(i, i | \Delta x, \Delta y) = H(i) \quad (5c)$$

where  $H(i)$  is the histogram of intensity values and  $\mu$  is the mean intensity within the local neighborhood.

Substituting (5a)–(5c) into (4) results in a simplified expression for cluster shade.

$$S(\Delta x, \Delta y) = 2 \sum_{i=0}^{L-1} (i - \mu)^3 H(i). \quad (6)$$

From (6) it is obvious that cluster shade with a zero length displacement vector is actually a measure of the asymmetry in the gray-level histogram of the local neighborhood. That is, if  $H(i)$  is symmetrical about the mean, then the cluster shade is zero. However, if  $H(i)$  is skewed to the negative side of the mean, the cluster shade value will be negative. Likewise, a histogram skewed to the positive side of the mean will result in a positive cluster shade. Thus, the CSED algorithm with a zero length displacement vector should produce results similar to those that are obtained from finding zero crossings in the skew values for local neighborhoods within the image. It is interesting to note that Gerson and Gaborski [2] have reported zero crossings in local histogram skew as the best way to locate the North Wall of the Gulf Stream. Although the CSED algorithm resembles previous work with

skew, CSED contains a major difference that is important to its performance. The equation for skew of a distribution is

$$\text{SKEW} = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N \frac{(f(m, n) - \mu)^3}{\sigma^3} \quad (7)$$

where  $\sigma$  is the standard deviation of  $f(m, n)$ . The numerator of (7) is similar in form to the cluster shade expression (6). However, the denominator in (7) normalizes the skew value by the standard deviation in the local neighborhood. This normalization by standard deviation means that based on skew, a small gradient in a low contrast area would be equally likely to be detected as would a large gradient in a high contrast area. This fact would cause many edges to appear in nearly isothermal areas of the IR image if skew were used as the basis for edge detection. This is an undesirable result for the present application where rejection of small gradients is required to reduce noise in the edge image. The CSED algorithm, by contrast, because it does not normalize its asymmetry measurement by standard deviation, results in an asymmetry parameter (cluster shade) that is proportional to gradient magnitude in an absolute (unnormalized) sense. This fact results in detection of the large gradients associated with the major features while ignoring the smaller gradients such as those in the Sargasso Sea at the bottom of Fig. 10.

To summarize the discussion of selection of displacement vector values, we have shown that this is not an important parameter, and furthermore, selection of zero as the displacement vector length eliminates the requirement for calculation of the GLC matrix in order to get cluster shade. Cluster shade is reduced to a histogram calculation (6) or alternatively

$$S(\Delta x, \Delta y) = \sum_{m=1}^M \sum_{n=1}^N (f(m, n) - \mu)^3. \quad (8)$$

Therefore, a zero length displacement vector is the obvious choice for most applications.

The question of optimal window or local neighborhood size is highly dependent upon the application. No strict rules can be stated in this regard. However, the effect of changing window size can be demonstrated. Fig. 13 shows the CSED results for a subsection of our test image processed with window sizes of  $3 \times 3$ ,  $9 \times 9$ , and  $16 \times 16$  pixels. The increase in window size has the effect of reducing fine detail and small spot noise in the edge representation. For our application the  $16 \times 16$  pixel window gives the most suitable result. However, the scale of the features to be delineated and the desired degree of edge smoothing would lead to other choices for other applications.

The effect of the last CSED parameter, cluster shade threshold, is demonstrated in Fig. 14. As the threshold is increased the number of edges detected is reduced. Therefore, if only a small number of strong edges is desired a large threshold is employed; if many edges including rel-



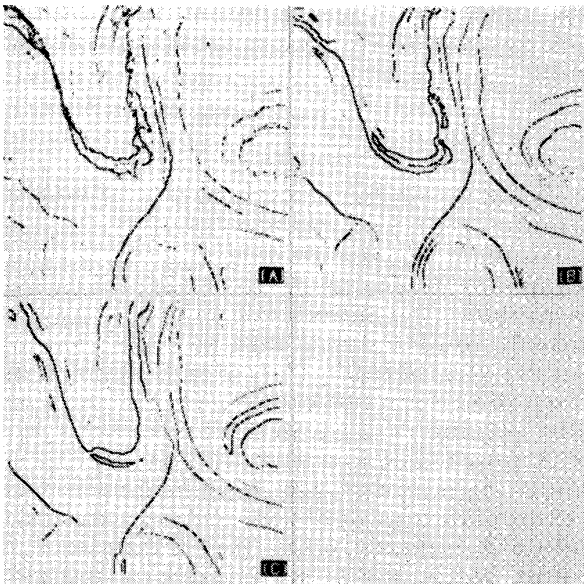


Fig. 13. Application of the CSED algorithm to a subsection of Fig. 2 with various values of local neighborhood size. (a)  $3 \times 3$ , (b)  $9 \times 9$ , (c)  $16 \times 16$ .

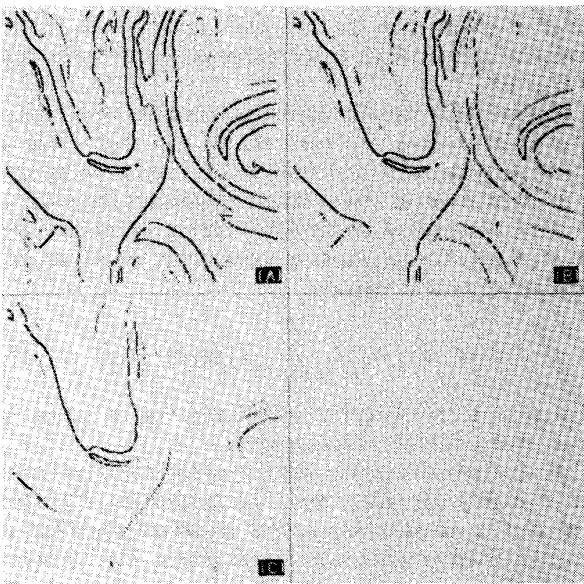


Fig. 14. Application of the CSED algorithm to a subsection of Fig. 2 with various values of the cluster shade zero crossing threshold. (a) 5, (b) 20, (c) 100.

actively weak ones are desired, the threshold is lowered. Cluster shade threshold like neighborhood size must therefore be determined for each application.

The edges detected by the CSED algorithm can contain many one- or two-pixel-wide breaks in the lines (see Fig. 6). These broken lines can be connected to form solid lines by dilation/erosion operations. Fig. 15 shows such a dilated/eroded result. The dilation (line thickening) algorithm, which also includes a test to remove spatially iso-

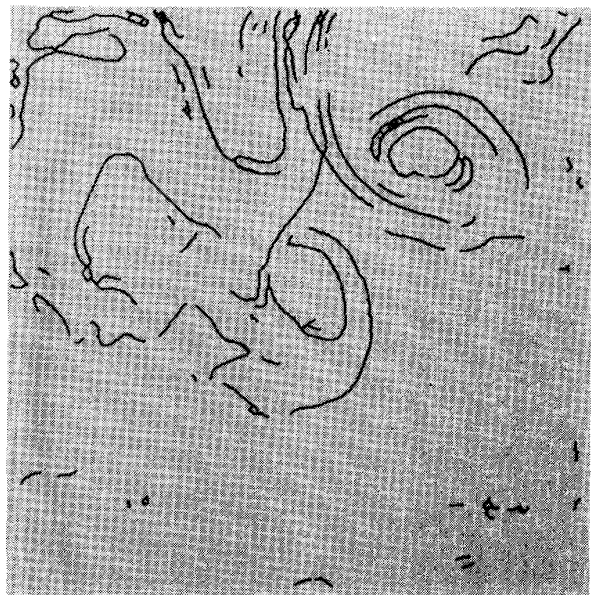


Fig. 15. Edges from Fig. 6 after cleaning/dilation/erosion operations.

lated points, operates as follows. First, a check for isolated points is performed by examining overlapping windows (called the cleaning window) in the binary edge image. The boundary pixels of each cleaning window are checked for "1" values. If a "1" is found anywhere on the boundary the window is left unchanged. If, however, no 1's are present on the boundary, all pixels within the window are set to "0." This cleaning operation removes isolated edge points that are not connected to edge structure of sufficient spatial extent to span the cleaning window. Fig. 15 was produced using a cleaning window size of  $16 \times 16$ . Next, the dilation step checks each pixel in the binary edge image for a value of "1." For every "1" pixel, all eight of its immediate neighbors are also set to "1." An edge image may be subjected to more than one iteration of the dilation algorithm, with each iteration making the lines thicker. To produce the result shown in Fig. 15 only one iteration was used. Following dilation and cleaning operations the lines are eroded (thinned) back to single pixel width. The thinned lines are nearly identical to the original edge lines, except that small breaks in the lines are now connected. The thinning algorithm used in the generation of Fig. 15 was taken from Pavlidis [29].

The dilation/erosion operations are applied here because the output of our simple algorithm to detect zero crossings in the cluster shade image results in broken lines of up to three pixels in width. If a more complex zero crossing detector were used, connected, single-pixel-wide lines, such as those shown in Fig. 15, could probably be derived directly from the CSED algorithm and would eliminate the need for any subsequent processing steps to connect and thin. Such an improved zero crossing detection algorithm would be a worthwhile enhancement to the CSED algorithm.

The question can be raised as to whether the CSED should be applied to the raw imagery data, or if some preprocessing step should be included to smooth the image prior to application of the CSED algorithm. Since smooth edges are the objective, some prior smoothing of the image would seem like a logical step. Several smoothing preprocessors, among them median filtering and sigma filtering [24], have been applied to our test images prior to CSED processing. The output of the edge detector was found to be virtually unchanged as the result of presmoothing of the imagery. We conclude therefore that the smoothing characteristics of the CSED method are powerful enough that presmoothing is redundant to that which occurs within CSED. Our test images indicate that the computational requirements associated with preprocessing are not justified by improved edge-detection results in this case.

The effects of atmospheric attenuation play a role in edge detection on IR imagery of the ocean. Atmospheric constituents, principally water vapor, absorb and emit energy in the wavelength bands sensed by the IR channels of the AVHRR instrument. The result of this atmospheric effect is typically a lowering of the apparent sea surface temperature (SST) by 1 to 3°C. An overall lowering of temperature values is of no consequence to an edge detector. However, the second result of atmospheric attenuation is a reduction in the magnitude of SST gradients. Under some conditions the atmosphere can completely obscure SST patterns on the ocean surface. The severity of the atmospheric degradation of SST gradients depends on the amount and temperature of the atmospheric water vapor. Warm, humid atmospheres cause the greatest degradation of SST gradients; cool, dry atmospheres the least degradation.

Multispectral techniques are frequently used to correct for atmospheric effects [30]. However, these correction techniques have been shown to introduce considerable noise into the resulting corrected SST image. The multispectral correction techniques also do not appear to restore SST gradient magnitude as they should [31]. Therefore, it is not clear whether one would get better results from edge detection on raw imagery or on atmospherically corrected imagery. In the present study we have used raw data from channel 4 of the AVHRR. Useful edge representations of the mesoscale features were obtained in this case because SST gradients are so large in the Gulf Stream region that they are easily discernable even when viewed through the atmosphere. In different locations, different seasons, or under different atmospheric conditions, edge detection on IR imagery may not be possible without some type of atmospheric correction. The type of atmospheric correction that would best restore SST gradients to the imagery is an area requiring further investigation.

Edges in IR imagery of the ocean are of varying magnitude and of varying widths. Fig. 14 has shown how the CSED algorithm can be tuned via the cluster shade zero crossing threshold to detect edges of greater or lesser magnitude. However, the issue of width of an edge has

not been addressed here. Rather large gradients may not be detected if they occur over distances that are large with respect to the local neighborhood size. Clearly, it is a combination of edge magnitude and width that determines edge detectability for the CSED algorithm, or for any other edge operator. The interrelationship in the CSED algorithm between edge magnitude, edge width, and detectability, is a matter for further work.

## VI. CONCLUSIONS

The GLC matrix based edge-detection algorithm presented here has done an excellent job of delineating mesoscale ocean structure in the test images. The algorithm has the desirable characteristics of smoothing the fine structure in the edges while preserving edge sharpness and accurate localization. Edge detectors with these characteristics show promise as tools for segmentation of satellite IR imagery of the ocean. The CSED algorithm should be useful in other image processing applications where these characteristics are required.

An improved algorithm for detecting zero crossings in the cluster shade image is recommended for future development. An improved zero crossing algorithm that would eliminate the small holes and double width of the edge lines would be useful in order to avoid the need for the dilation/erosion step required here to fill in the gaps.

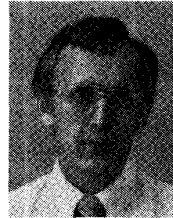
The relationship between edge-detection performance and atmospheric attenuation effects should be investigated. In particular, an atmospheric correction technique that restores the atmospherically diminished SST gradients to their true surface values would be useful.

## REFERENCES

- [1] O. B. Brown, P. C. Cornillon, S. R. Emmerson, and H. M. Carle, "Gulf stream warm rings: a statistical study of their behavior," *Deep Sea Research*, vol. 33, no. 11/12, pp. 1459-1473, 1986.
- [2] D. J. Gerson and P. Gaborski, "Pattern analysis for automatic location of oceanic fronts in digital satellite imagery," Naval Oceanographic Office, TN 3700-65-77, Oct. 1977.
- [3] D. J. Gerson, E. Khedouri, and P. Gaborski, "Detecting the Gulf Stream from digital infrared data pattern recognition," in *The Belle W. Baruch Library in Marine Science: No. 12-Processes in Marine Remote Sensing*. Columbia, SC: Univ. South Carolina Press, 1982, pp. 19-39.
- [4] R. E. Coulter, "Application of the Bayes decision rule for automatic water mass classification from satellite infrared data," in *Proc. 17th Int. Symp. Remote Sensing Environ.*, vol. II, pp. 589-597, 1983.
- [5] M. F. Janowitz, "Automatic detection of Gulf Stream rings," Office of Naval Research, Tech. Rep. TR-18501, Contr. N-00014-79-C-0629, June 1985.
- [6] —, "A cluster analysis program for image segmentation," in *Statistical Image Processing*, E. J. Wegman and J. G. Smith, Eds. New York: Dekker, 1984, pp. 399-410.
- [7] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1978, p. 489.
- [8] D. G. Nichol, "Autonomous extraction of an eddy-like structure from infrared images of the ocean," *IEEE Trans. Geosci. Remote Sensing*, vol. GE-25, no. 1, pp. 28-34, 1987.
- [9] E. F. Carter, "The structure of the Gulf Stream as derived from an EOF analysis," in *Gulf Stream Workshop Proc.* (Univ. Rhode Island), 1985.
- [10] E. J. Molinelli and M. J. Flanigan, "Optimized CEOF interpretation of the Gulf Stream," Planning Systems Inc., Contract N66604-86-D-0120, Tech. Rep. #TR-392395, 1987.
- [11] M. Lybanon, J. D. McKendrick, R. E. Blake, J. R. B. Cockett, and

- M. G. Thomason, "A prototype knowledge-based system to aid the oceanographic image analyst," *Proc. SPIE*, vol. 635, Appl. of Artificial Intelligence III, pp. 203-206, 1986.
- [12] M. Lybanon, "Knowledge-based aids to the oceanographic image analyst," *MTS Journal*, vol. 20, no. 2, pp. 28-32, 1986.
- [13] M. G. Thomason and R. E. Blake, "Development of an expert system for interpretation of oceanographic images," Naval Ocean Research and Development Activity, NORDA Rep. 148, June 1986.
- [14] L. S. Davis, "A survey of edge detection techniques," *Comput. Graph. Image Proc.*, vol. 4, pp. 248-270, 1975.
- [15] S. Levialdi, "Finding the edge," in *Digital Image Processing*, J. C. Simon and R. M. Haralick, Eds. Reidel Publishing Company, 1981, pp. 105-148.
- [16] M. Kunt, "Edge detection: a tutorial review," in *Proc. Int. Conf. Acoustic, Speech, Signal Processing* (Palais des Congres, Paris, France), pp. 1172-1175, May 3-5, 1982.
- [17] L. G. Roberts, "Machine perception of three-dimensional solids," in *Optical and Electro-optical Information Processing*, J. Tippett et al., Eds. Cambridge, MA: MIT Press, 1965, pp. 159-197.
- [18] R. C. Gonzalez and P. A. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1977, pp. 337-338.
- [19] J. M. S. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld, Eds. New York, Academic, 1970, pp. 75-149.
- [20] W. Frei and C. C. Chen, "On the encoding of arbitrary geometric configurations," *IEEE Trans. Computers*, vol. C-26, no. 10, pp. 988-998, 1977.
- [21] R. C. Gonzalez and P. A. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1977, pp. 211-213.
- [22] E. R. Davies, "Circularity—a new principle underlying the design of accurate edge orientation operators," *Image and Vision Computing*, vol. 2, no. 3, pp. 134-142, Aug. 1984.
- [23] J. S. Lee, "Edge detection by partitioning," in *Statistical Image Processing*, E. J. Wegman and J. G. Smith, Eds. New York: Dekker, 1984, pp. 59-69.
- [24] —, "The sigma filter and its application to speckle smoothing of synthetic aperture radar images," *IEEE Trans. Syst., Man, Cyber.*, vol. SCM-13, no. 1, pp. 85-89, 1983.
- [25] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cyber.*, vol. SMC-3, no. 6, pp. 610-621, Nov. 1973.
- [26] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982, pp. 186-188.
- [27] R. W. Conners, M. M. Trivedi, and C. A. Harlow, "Segmentation of a high-resolution urban scene using texture operators," *Computer Vision, Graphics, and Image Processing*, vol. 25, pp. 273-310, 1984.
- [28] C. A. Harlow, M. M. Trivedi, and R. W. Conners, "Use of texture operators in segmentation," *Optical Engineering*, vol. 25, no. 11, pp. 1200-1206, Nov. 1986.
- [29] T. Pavlidis, "A thinning algorithm for discrete binary images," *Computer Graphics and Image Processing*, vol. 13, pp. 142-157, 1980.
- [30] E. P. McClain, W. G. Pichel, C. C. Walton, Z. Ahmad, and J. Sutton, "Multichannel improvements to satellite-derived global sea surface temperatures," *Adv. Space Res.*, vol. 2, pp. 43-47, 1983.
- [31] P. E. La Violette and R. J. Holyer, "Noise and temperature gradients in multichannel sea surface temperature imagery of the ocean," *Remote Sensing Environ.*, in press, 1988.

\*



**Ronald J. Holyer** received the B.A. degree in physics and mathematics from Augustana College, Sioux Falls, SD, in 1964, and the M.S. degree in physics from South Dakota School of Mines and Technology in 1966.

He has conducted remote-sensing and image processing research at Texas Instruments, Inc., Dallas, TX, and Lockheed Electronics Co., Bay St. Louis, MS. He is presently with the Naval Ocean Research and Development Activity, Stennis Space Center, MS, where he is a principal investigator in the field of automated image analysis and pattern recognition techniques.

\*



**Sarah H. Peckinpaugh** received the B.S. degree in computer science with a minor in mathematics from the University of Southern Mississippi in 1981.

She is currently employed at the Naval Ocean Research and Development Activity, Stennis Space Center, MS. Her current research interests are image segmentation and classification applied to remotely sensed oceanographic data.