

AD-A235 811



040006

2

**Proceedings:
Automated Interpretation
of
Oceanographic Satellite Images Workshop**

February 1991

SP 001:321:91

DTIC
ELECTE
MAY 28 1991
S B D

Matthew Lybanon
Remote Sensing Branch
Ocean Sensing and Prediction Division
Ocean Science Directorate

*Original contains color
pictures. All DTIC reproductions
will be in black and
white.

91-00549



Approved for public release; distribution is unlimited. Naval Oceanographic and Atmospheric Research Laboratory,
Stennis Space Center, Mississippi 39529-5004.

91 5 24 072

Contents

Preface	1
Historical Overview of NOARL Automated Imagery Interpretation Studies Matthew Lybanon and Ronald J. Holyer	3
Edge Detection Applied to Infrared Imagery of the Ocean S. H. Peckinpaugh and R. J. Holyer	9
A Hybrid Technique for Interpreting Mesoscale Analysis System (SAMAS) N. Krishnakumar and S. Sitharama Iyengar	37
Recent Results in Oceanographic Expert Systems Validation Matthew Lybanon	49
Evaluation of the Navy's Semi-Automate Mesoscale Analysis System (SAMAS) S. H. Peckinpaugh and R. J. Holyer	57
OSIRRUS: Symbolic Image Representation, Recognition and Understanding Software Lee A. Atkinson	71
Water-Mass Analysis for Digital Imagery J. Blaha and D. L'Heureux	81
Labeling North Atlantic Mesoscale Features from Satellite Photographs: A New Approach B. P. Buckles and F. E. Petry	95
Progress with Neural Network Gulf Streams Eugene Molinelli, Gregory Muncill, and Kevin Pepe	103
The Marr-Hildreth Operator as an Eddy Detector Carl Szczechowski	153
Approaches to Clouds Detection in AVHRR Images James Kowalski, Peter Comillon, Jean-Francois Cavula, James Gallagher, Daniel Holloway	169
Markov Networks for Knowledge Representation Incorporating Probabilities Michael G. Thomason and Jean R. S. Blair	177
Edge-Region Correlation in Image Segmentation and Analysis Vivien J. Cambridge	183
Finding Ocean Structures Using Mathematical Morphology Suzanna M. Lea	189

Time Coherency of Gulf Stream Normal Mode Coefficients (abstract only)	
Ronald J. Holyer.....	201
Genetic Algorithms and Curve-Fitting with Applications to an Altimeter	
Derived Reference Surface Error	
Kenneth Messa	203
An Automated Technique for Locating the Gulf Stream in Altimeter Profiles	
MIDN Steve L. Cantrell and Ronald J. Holyer	211

Preface

AUTOMATED INTERPRETATION OF OCEANOGRAPHIC SATELLITE DATA

The papers collected in this volume describe work that is potentially of great benefit to the U. S. Navy. The provision of automated or semiautomated tools to aid the oceanographic satellite data interpreter has the potential to make significant contributions to the exploitation of remotely sensed observations to satisfy Navy requirements. The techniques themselves are of considerable scientific interest, since they indicate a high degree of success in dealing with a number of difficult technical problems. The cooperation between Government, academic, and industrial researchers which the papers represent serves as a model for future partnerships.

The environment affects all naval systems, even those designated "all weather." Therefore observations of environmental conditions can provide valuable information to Navy operations. Continuing advances in space technology have generated new, cost-effective techniques for global observation of the ocean and the atmosphere. During the early 1990s the U. S. Navy will receive an unprecedented wealth of new remotely sensed oceanographic data from both U. S. (e. g., TOPEX/Poseidon, GEOSAT Follow-on, GOES-Next) and foreign (European ERS-1, Japanese ERS-1) satellites. Present-day satellites (e. g., DMSP, GOES, NOAA) will continue to supply oceanographic data. The sheer volume of data from these satellites will be overwhelming. Our ability to collect data has exceeded our ability to process, analyze, and interpret satellite data. The integrated multisensor analyses required with advanced systems involve complex procedures that can only be performed by computers. Clearly, automated analysis tools are a critical factor in optimal Navy utilization of remotely sensed environmental data. The papers in this volume describe the beginning of the development of automated tools to carry out the operations required to satisfy the Navy's anticipated needs.

The papers cover a wide range, including image segmentation, feature labeling, empirical orthogonal function techniques, applications of genetic algorithms, nonlinear relaxation methods, mathematical morphology, neural networks, and some others. Sophisticated techniques such as these are necessary because the problem is difficult. For example, interpretation of satellite images of the ocean is hampered by cloud cover, haze and other atmospheric effects, as well as the facts that the features of interest are time-varying and have no well-defined shapes.

Many organizations have contributed to this work. The Remote Sensing Branch of the Naval Oceanographic and Atmospheric Research Laboratory (NOARL) has led the Navy's research efforts. The University of Tennessee (and Perceptics Corporation, a related consulting firm), Louisiana State University, and Tulane University are academic institutions that have provided direct contract support to NOARL. Private companies so involved include Sverdrup Technology, Inc. (the local technical support contractor serving NOARL), Planning Systems Inc., and Consultant's Choice, Inc. Other organizations that have made important contributions to NOARL's effort include the University of North Carolina at Greensboro and Loyola University. The U. S. Naval Oceanographic Office and the University of Rhode Island, working independently, carried out interesting, related work also reported in this volume. The papers collected here give a good overview of the current state of the art in automated interpretation of oceanographic satellite data, and serve as a basis for future research.



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

HISTORICAL OVERVIEW OF NOARL AUTOMATED IMAGERY INTERPRETATION STUDIES

Matthew Lybanon and Ronald J. Holyer
Ocean Sensing and Prediction Division
Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, MS 39529-5004

I. INTRODUCTION

Generation of oceanographic products from the interpretation of satellite data by conventional manual means requires much labor and very high skill levels, both of which are difficult to realize in an operational environment. The work described here deals mostly with the interpretation of image data, specifically satellite infrared (IR) imagery of the ocean. IR images of the ocean are useful because they add the breadth of synoptic coverage to the depth of in situ measurements at a few points. Interpretation of satellite data by human experts via interactive image processing is often successful (Cornillon et al., 1987), but the data from existing and planned satellites will inundate the oceanographic remote sensing analyst with information. For several years, the Remote Sensing Branch of the Naval Oceanographic and Atmospheric Research Laboratory has conducted a research program to address this situation.

Digital image understanding techniques have been successful in a number of applications, and a wide variety of techniques are available. (Gonzalez and Wintz, 1977) However, in the oceanographic image understanding field, there are some problems that are different from those that arise in other applications. This situation is primarily due to the particular dynamical conditions that exist and the users' ultimate objectives. Some of these problems are (Lybanon and McKendrick, 1983):

1. The scenes are naturally time-varying. Features of interest move, change size, etc. Because the scenes are dynamic, single observations are usually not sufficient. In some cases the time dependence of some phenomena is the specific information that is desired. So it is necessary to look through and compare large volumes of data.
2. An area of interest may consist entirely of open ocean. In such a case there are no landmarks to use as "control points" in relating satellite coordinates to geographic coordinates.
3. There have been very few dedicated oceanographic satellites, and there are only a few specifically "oceanographic" sensors on other satellites. SEASAT operated for only a few months, and GEOSAT only carried a microwave altimeter, no imaging sensors. Sensors designed for other purposes may not be well suited for oceanography. As an example, the sensors on a weather satellite are designed to observe clouds and measure atmospheric conditions. But it is necessary to penetrate to the sea's surface to make oceanographic measurements.

The second of these problems can be solved by the use of precise orbit information and a model of the sensor's viewing geometry. (The heavy computational load involved means that in an operational setting the analyst may work with imagery in "natural" satellite coordinates, only transferring results to a map projection.) But the other two problems have a direct bearing on the interpretive process.

The time dependence of the oceanographic processes of interest complicates interpretation in several ways. Oceanographic features do not in general look the same in two images of the same region separated by some time interval, they are generally in different locations, and even the number of features may vary (e. g., a new eddy may be created or an old one absorbed by the Gulf Stream). The inability of the sensors to see the sea surface, either partially because of atmospheric water vapor or totally because of cloud cover, further contributes to the difficulty.

II. PROTOTYPE EXPERT SYSTEM

The observation that human experts can frequently do a good job of interpretation led to the idea that an expert system might be beneficially applied to the problem. Investigators from the University of Tennessee and Perceptics Corporation, under contract to the Naval Ocean Research and Development Activity (NORDA, a predecessor to NOARL), took on the task of applying knowledge-based techniques, beginning in 1984. Their research was directed

toward a system to support the analysis of evolutionary, structural changes in mesoscale features in an area of the ocean by processing a series of registered images and other information from that area, with interactive, semi-automatic processing supported by an expert system. The motion and structural changes of the features in the Gulf Stream area of the north Atlantic Ocean was the focus of the initial work.

The first stage in the development of an expert system is the compilation of a knowledge base. There did not appear to be a well-defined collection of rules about Gulf Stream mesoscale features and their evolution in existence, and consultation with experts to perform the necessary "knowledge engineering" did not appear feasible. Instead, the Tennessee/Perceptics researchers compiled a knowledge base by an exhaustive literature search, supplemented by information gained from contacts with NORDA experts. The knowledge base is listed by Thomason and Blake (1986). Information about the expert system and its use is given by Lybanon, et al. (1986) and by Thomason (1989).

The expert system's rule base represents oceanographic knowledge about the evolution of mesoscale ocean features in the Gulf Stream region of the north Atlantic Ocean. The rules are applied in such a way that the expert system describes the kinematics of that evolution. The expert system uses the rules to evolve an initial "state" to a later time. The new, hypothesized state can then be used as a new initial condition and the process repeated. This procedure can be usefully carried out for several steps.

The NOARL expert system can allow an analyst to fill in during data-sparse periods. In an operational environment, it may be necessary to provide analyses at regular intervals. If there is no observational data for a time period, the analyst must make some assumptions (e. g., features are in the same places as in the previous analysis, or they continue to move in the same directions with the same speeds). The expert system can help the analyst to make an "educated guess" in such a situation.

We have performed validation tests of the expert system's ring motion rules against several sets of mesoscale analyses. Lybanon (1990a) gives more details of the expert system's structure and the results of some of those tests. A later reference (Lybanon, 1990b) gives the results from tests made after rule modifications suggested by the first series of tests. The results indicate that the expert system performs very well for ring motion for periods up to 2 weeks.

III. LOWER-LEVEL PROCESSING

A. The Requirement

It soon became clear that, though the expert system was potentially a valuable tool when used as suggested in the previous section, there was a fundamental mismatch between the source of data and the expert system. A satellite image contains hundreds of thousands, sometimes millions, of pixels, each 8 bits or more; in many cases the satellite sensor provides several channels (different spectral bands of data), as well. On the other hand, the expert system manipulates a representation consisting of a relatively small number of objects, each described by a few numbers. We therefore turned our attention to the interface between the two representations.

The conventional three-level image understanding paradigm proved to be useful. The top level refers to a description in terms of a (problem-dependent) interpretation; that is the level at which the expert system operates. The bottom level refers to operations on individual pixels, or a few at a time, without any domain-specific information. The intermediate level, which may involve some knowledge, is a bridge between the two.

NOARL researchers developed an important tool that works at the pixel level. Holyer and Peckinpaugh (1989) developed an edge-detection algorithm with important advantages compared to other edge detectors. Most edge detectors are essentially high-pass filters, and therefore produce very noisy results when used on satellite images of the ocean. The NOARL edge-detection algorithm is based on the gray level co-occurrence (GLC) matrix, which is commonly used in image texture analysis. A measure known as cluster shade, 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.

The cluster shade edge detector provides a considerable simplification of an input image, but its output is still not what the expert system requires. First, the resulting edges must be labeled, either as boundaries of ocean features or as unimportant "noise" to be discarded. Second, it is found that in most cases the labeled edges consist of disconnected portions of feature boundaries, which must be connected in some fashion to form continuous features. In order to satisfy the requirements imposed by these conditions, several research efforts were begun both by NOARL and by several contractors.

B. The Four-Tiered Approach

As described in the previous sections, the difficulty of the satellite oceanography image understanding problem led to the consideration of four technical issues. Those issues are image segmentation (simplification), feature labeling (assignment of oceanographic identity to image features), feature aggregation (formation of Gulf Stream north wall, eddies, etc.), and mesoscale extrapolation (to deal with temporal data gaps). The concept of a prototype knowledge-based semiautomated interpretation system designed to deal with those issues therefore included a number of modules. Those modules include the cluster shade edge detector, edge labeling algorithms, Gulf Stream and eddy formation modules, and the expert system.

The cluster shade edge detector is described in outline above, and in more detail in the reference (Holyer and Peckinpaugh, 1989). We are investigating several approaches to the feature labeling problem, among them nonlinear relaxation and genetic algorithms. The use of complex empirical orthogonal functions (CEOFs) to interpolate between intermittent observations constructs a physically reasonable north wall, while the Hough transform effectively performs that function for eddies. The expert system described in Section II provides reasonable estimates of feature evolution when new observations are not available because of, e. g., cloud cover. This constitutes our "standard" approach, described in more detail in prior and subsequent sections. We are also supporting research into several other approaches. The details of nearly all of these research efforts are described in other papers in these Proceedings.

C. Feature Labeling

Given the edge detector output from processing an oceanographic satellite image, the problem is to assign to each edge segment a label using certain knowledge. The labels are "north wall," "south wall," "warm eddy," "cold eddy," "shelf front," and "other." The knowledge to be used to direct the labeling includes such things as results of a previous analysis (temporal continuity), temperatures from the original image, bathymetry, rules on the general oceanographic context, and Gulf Stream climatology. Two approaches used are described below. In addition to the references listed, both are described elsewhere in these Proceedings.

Researchers from Louisiana State University applied nonlinear relaxation labeling (Kittler and Illingworth, 1985) to the task. Relaxation labeling blends contextual information with conventional image processing. The goal of the relaxation process is to reduce the uncertainty and improve the consistency in the assignment of one of the labels to each object in a set of related objects. (In the present problem the objects or labels are the oceanographic features.) Details are in Krishnakumar et al. (1990a; 1990b).

Tulane University investigators pursued a different approach, genetic algorithms (Austin, 1990). The ordinary genetic algorithm (GA) is an optimization procedure that works in binary search spaces, i. e., the search space consisting of binary strings, but after some coding it can also be applied to continuous search spaces. Unlike classical hill-climbers it does not evaluate and improve a single solution but, instead, it analyzes and modifies a *population* (that is, a set) of solutions at the same time. In GA a solution (point in the search space) is represented by a finite sequence of 0's and 1's called a *chromosome*. (In this case a chromosome is a labeling of the entire image.) The algorithm manipulates a finite set of chromosomes, the *population*, in a manner that resembles the

mechanism of natural evolution. The chromosomes are allowed to *mate* (by a process called *crossover*) and to *mutate*. The optimization process is carried out in cycles called *generations*. During each generation a set of new chromosomes is created and evaluated. Since the population size is finite, only a predefined number of the (best) chromosomes survives to the next generation. Buckles, et al. (1990) describe the application of GAs to the present problem.

D. Feature Aggregation

Labeled edges may have breaks or be otherwise incomplete for a variety of reasons (e. g., cloud cover, imperfect edge detection or labeling, etc.). The Gulf Stream north wall's description is based on the use of CEOFs. Eddies are specified by applying the circular Hough transform to eddy edges.

The empirical orthogonal function method partitions the temporal variance of the data into orthogonal spatial patterns called eigenvectors. These eigenvectors are ranked in decreasing order according to the percent of the variance they account for. Each eigenvector also has a series of coefficients in time that modulate it. At one time, a linear combination of all the eigenvectors exactly describes the data. An advantage of the representation is that frequently a considerably truncated series (involving those modes that account for most of the variance) represents the data sufficiently well. Only 10 modes are sufficient to describe the Gulf Stream to within 6 nmi. Legler (1983) describes the formulation. While the standard procedure (just described) "reconstructs" the Gulf Stream from complete information, our problem is to reconstruct it from partial information. The procedure developed by Planning Systems Inc. (Molinelli and Flanigan, 1987) is to optimally adjust a set of "first guess" mode coefficients until the series is a least-squares fit to the available Gulf Stream "fixes." It is found that the optimized Gulf Stream is at least as close to the actual Gulf Stream as the analyzed (by a human expert) Gulf Stream is to the actual Gulf Stream, and that the difference between an optimized Gulf Stream and an analyst Gulf Stream is of the same order as the errors in the analyst Gulf Stream itself.

Circle detection in digitized imagery is frequently approached through the use of the Hough transform. In general (for arbitrary curves) the Hough transform consists of remapping edge points from image space onto a parameter space. The Hough transform for circles

$$r^2 = (x - a)^2 + (y - b)^2 \quad (1)$$

projects edge points onto a 3-dimensional parameter space (r, a, b). The parameter space is quantized into discrete bins and is known as the "accumulator array." Each (x, y) edge point "votes" for all those circles of which it may be part by incrementing the appropriate accumulators, i. e., those that satisfy Eq. (1). Local maxima in the accumulator array correspond to the circular shapes within an edge image. This method will find the most prominent eddies and give (r, a, b) parameter sets which permit filling in eddies even if significant parts are missing. Peckinpaugh and Holyer (1990) describe the use of the circular Hough transform in providing circular fits to labeled eddy edges in satellite images.

IV. OTHER APPROACHES

In addition to the "standard" approach described in previous sections, NOARL is also working on, or is collaborating with others, on a number of other topics. V. J. Cambridge of Sverdrup Technology Inc. is developing a segmentation technique which uses both region and edge detection techniques in a mutually augmentory way (Cambridge et al., 1990). S. M. Lea of the University of North Carolina-Greensboro is applying the methods of mathematical morphology (Lea and Kellar, 1989) to eddy detection. Molinelli and others at Planning Systems Inc. are continuing work on the use of a neural network to find Gulf Stream CEOF mode coefficients directly from edge images (Lybanon et al., 1990). K. C. Messa of Loyola University is using genetic algorithms for curve-fitting (Messa and Lybanon, 1990), with application to geoid improvement so as to make the processing of altimeter data more accurate. L. Atkinson of Consultant's Choice, Incorporated is applying symbolic image processing techniques to the ocean feature recognition problem. Several researchers at the U. S. Naval Oceanographic Office

(NAVOCEANO) are working on different approaches; NAVOCEANO's Operational Oceanography Center performs operational interpretations of satellite and surface observations for a number of regions around the world. M. G. Thomason at the University of Tennessee has performed research in knowledge representation by Markov networks (Lauritzen and Spiegelhalter, 1988), with potential application to oceanographic image interpretation.

This is an active area of current research. Aside from those areas of research NOARL has been directly involved in, we have benefitted from collaboration with researchers working independently at other institutions. Other papers in these Proceedings cover virtually all of the topics described in this paper, in substantially more detail.

V. REFERENCES

Cornillon, P., C. Gilman, L. Stramma, O. Brown, R. Evans, and J. Brown (1987). Processing and Analysis of Large Volumes of Satellite-Derived Thermal Infrared Data. *Journal of Geophysical Research* 92(C12):12,993-13,002.

Gonzalez, R. C., and P. Wintz (1977). *Digital Image Processing*. Reading, MA, Addison-Wesley, 431 pp.

Lybanon, M., and J. D. McKendrick (1983). Some Applications of Image Processing in Oceanography. *Proceedings, the Fifteenth Southeastern Symposium on System Theory*, Huntsville, AL, March 28-29, pp. 138-142.

Thomason, M. G., and R. E. Blake (1986). *Development of an Expert System for Interpretation of Oceanographic Images*. Naval Ocean Research and Development Activity, Stennis Space Center, MS, NORDA Report 148.

Lybanon, M., J. D. McKendrick, R. E. Blake, J. R. B. Cockett, and M. G. Thomason (1986). A prototype knowledge-based system to aid the oceanographic image analyst. *SPIE Vol. 635--Applications of Artificial Intelligence III*, 203-206.

Thomason, M. G. (1989). Knowledge-Based Analysis of Satellite Oceanographic Images. *International Journal of Intelligent Systems* 4:143-154.

Lybanon, M. (1990a). *Oceanographic Expert System Validation Using GOAP Mesoscale Products and Gulfcast/DART Validation Test Data*. Naval Oceanographic and Atmospheric Research Laboratory, Stennis Space Center, MS, NOARL Report 5.

Lybanon, M. (1990b). Recent Results in Oceanographic Expert System Validation. These Proceedings.

Holyer, R. J., and S. H. Peckinpaugh (1989). Edge Detection Applied to Satellite Imagery of the Oceans. *IEEE Transactions on Geoscience and Remote Sensing* 27(1):46-56.

Kittler, J., and J. Illingworth (1985). Relaxation labelling algorithms--a review. *Image and Vision Computing* 3(4):206-216.

Krishnakumar, N., S. S. Iyengar, R. Holyer, and M. Lybanon (1990a). Feature labelling in infrared oceanographic images. *Image and Vision Computing* 8(2):141-147.

Krishnakumar, N., S. S. Iyengar, R. Holyer, and M. Lybanon (1990b). An Expert System for Interpreting Mesoscale Features in Oceanographic Satellite Images. *International Journal of Pattern Recognition and Artificial Intelligence* 4(3) (invited paper).

Austin, S. (1990). An Introduction to Genetic Algorithms. *AI Expert* 5(3):48-53.

Buckles, B. P., F. E. Petry, and M. Lybanon (1990). Ocean Feature Labeling Using GA/F³. *Proceedings of*

NAFIPS '90, Toronto, June, pp. 394-397.

Legler, D. M. (1983). Empirical Orthogonal Function Analysis of Wind Vectors over the Tropical Pacific Region. *Bulletin American Meteorological Society* 64(3):234-241.

Molinelli, E. J., and M. J. Flanigan (1987). *Optimized CEOF Interpolation of the Gulf Stream*. Planning Systems Incorporated, McLean, VA, Technical Report TR-392395.

Peckinpaugh, S. H., and R. J. Holyer (1990). Results of the Evaluation of SAMAS 1.0. These Proceedings.

Cambridge, V., M. Lybanon, and S. Peckinpaugh (1990). Mutual Augmentation of Region and Edge Detection Techniques in Oceanographic Images Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (to be published).

Lea, S. M., and L. A. Kellar (1990). An Algorithm to Smooth and Find Objects in Astronomical Images. *The Astronomical Journal* 97(4):1238-1246; 1269-1276 (plates). Erratum (figures reversed) ____ 98:736.

Lybanon, M., E. J. Molinelli, and M. Flanigan (1990). Automatic Description of the Gulf Stream from IR Images Using Neural Networks. *Proceedings of SPIE Conference on Applications of Artificial Neural Networks*, Orlando, FL, April 18-20.

Messa, K., and M. Lybanon (1990). *Curve Fitting Using Genetic Algorithms*. Naval Oceanographic and Atmospheric Research Laboratory, Stennis Space Center, MS, NOARL Report (to be published).

Lauritzen, S. L., and D. J. Spiegelhalter (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B* 50(2):157-224.

EDGE DETECTION APPLIED TO INFRARED IMAGERY OF THE OCEAN

S.H. Peckinpaugh and R.J. Holyer
Ocean Sensing and Prediction Division
Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, Mississippi 39529-5004

1. INTRODUCTION

Infrared (IR) images of the ocean obtained from satellite sensors are widely used for the study of ocean dynamics. The first step in the automation of the interpretation of these images is segmentation. Image segmentation for this application will be in the form of edge detection. The desire is to locate the significant edges within the given IR image. These edges should outline the mesoscale ocean features found in the image; such as the North and South Walls of the Gulf Stream and warm and cold core eddies. The method to be discussed in detail will be based on the grey level co-occurrence (GLC) matrix cluster shade texture measure [1]. Examples of the Sobel and Sigma edge operators will be shown for comparison. The test image is a warmest pixel composite image created from earth located, calibrated Advanced Very High Resolution Radiometer (AVHRR) IR channel 4 images from March 7 and 8, 1987, see figure 1.



Figure 1. Warmest pixel composite image created from two channel 4 AVHRR IR images from March 7 and 8, 1987.

II. SOBEL EDGE OPERATOR

The Sobel edge operator computes the gradient vector of an image by performing two two-dimensional spatial convolutions in the x and y directions. The resultant edge-enhanced image will have pixel values defined by equation 1.

$$P'(X,Y) = (X^2 + Y^2)^{1/2} \quad (1)$$

where X and Y are convolutions using the following 3 x 3 kernels:

$$\begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{array} \quad \begin{array}{ccc} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array}$$

An increment can be set to separate kernel elements. For example, If the values of increment are 2 and 2, then each output pixel is a function of those nine neighboring pixels plus-or-minus a distance of two pixels away in both dimensions, a kernel size of 5 x 5. This effectively subsamples the kernel space. See figures 2 and 3 for examples of the Sobel edge operator.

III SIGMA FILTER EDGE OPERATOR

This edge operator is based on the sigma filter [2]. First the sets A, B, and C must be defined, see equations 2, 3, and 4 below.

$$A = \{ X_{kl} \mid X_{kl} > (X_{ij} + \Delta) \} \quad (2)$$

$$B = \{ X_{kl} \mid (X_{ij} + \Delta) \geq X_{kl} \geq (X_{ij} - \Delta) \} \quad (3)$$

$$C = \{ X_{kl} \mid X_{kl} < (X_{ij} - \Delta) \} \quad (4)$$

X_{ij} is the intensity at pixel (i,j) and X_{kl} are the pixels within the window centered at pixel (i,j). N_A , N_B , N_C are the sums of the number of pixels in the sets A, B, C and Σ_A , Σ_B , Σ_C are the sums of the intensities of the pixels in the sets A, B, C. A and C can be empty sets; B must contain at least one element $X_{i,j}$. The 3 sets must be reduced to just 2 sets of pixels. Either set A or C will be merged with set B. A sequence of decreasing Δ values will be used - $\Delta_1, \Delta_2, \dots, \Delta_k$, where Δ_k defines the finest grey level differential desired. Using Δ_1 , if all pixels are in set B then the process is repeated using Δ_2 . The process is repeated reducing the Δ value until set A or C contains at least 1 element or Δ reaches the Δ_k limit. In the absence of definite knowledge of the Δ sequence, use $\Delta_{i+1} = \Delta_i/2$. Problems noted by the developer of this algorithm are: 1) ramp edges and isolated pixels must be treated as special cases and 2) windows 5 X 5 and larger will cause broad or smeared edges.



Figure 2. Sobel edge image created from the image shown in figure 1 using the regular increment of 1.



Figure 3. Sobel edge image created from the image shown in figure 1 using increment values of 2, 2 (subsampled 5 x 5 kernel).

The algorithm described below has been slightly modified for iterative Δ reduction to allow one element of the window to lie outside the B set. The edge detection procedure with a 3 X 3 window is described in step form as follows:

- (1) Starting with $\Delta = \Delta_1$ in the 3 X 3 sliding window.
- (2) Compute $(X_{ij} - \Delta, X_{ij} + \Delta)$.
- (3) Accumulate N_A, N_B, N_C and $\Sigma_A, \Sigma_B, \Sigma_C$ over all pixels in the 3 X 3 window.
- (4) If $N_B \geq 8$ (i.e., Δ too large), go to step 7.
- (5) If $N_B = 1$ (i.e., isolated pixel or ramp edge), go to step 8.
- (6) If $N_A > N_C$, the edge magnitude S is

$$S = (\Sigma_B + \Sigma_C) / (N_B + N_C) - \Sigma_A / N_A$$

otherwise, $S = (\Sigma_B + \Sigma_A) / (N_B + N_A) - \Sigma_C / N_C$.
Go to step 1, process the next window.
- (7) Replace Δ_i by the smaller Δ_{i+1} in the sequence. If the end of the Δ sequence is reached (i.e., flat area), set the edge magnitude S to zero and go to step 1, process the next window, otherwise, go to step 2.
- (8) If $N_A = 0$ OR $N_C = 0$ (i.e., Isolated pixel), set the edge magnitude S to zero, otherwise (i.e., ramp edge), the magnitude S is

$$S = \Sigma_A / N_A - \Sigma_C / N_C$$
. Go to step 1, process the next window.

See figures 4 and 5 for examples of the sigma edge operator, both examples were produced using the modified algorithm shown in step form above.

IV. CLUSTER SHADE EDGE OPERATOR

Several steps are included in the use of the Cluster Shade texture measure for edge detection [1], see figure 6. The first two steps are absolutely necessary - the computation of the Cluster Shade texture measure and the zero crossing test. The remaining two steps can be omitted or could be used with other edge detectors, such as the two described above.

A. The Grey Level Co-occurrence Matrix

The GLC matrix is defined for a neighborhood or window of the input image. The $(i,j)^{th}$ element of the GLC matrix, $P(i,j|\Delta x, \Delta y)$, is the relative frequency with which two image pixels, separated by the pixel distance



Figure 4. Sigma edge image created from the image shown in figure 1 using $\Delta_1 = 30$ and $\Delta_4 = 5$ ($k = 4$).



Figure 5. Sigma edge image created from the image shown in figure 1 using $\Delta_1 = 30$ and $\Delta_4 = 10$ ($k = 4$).

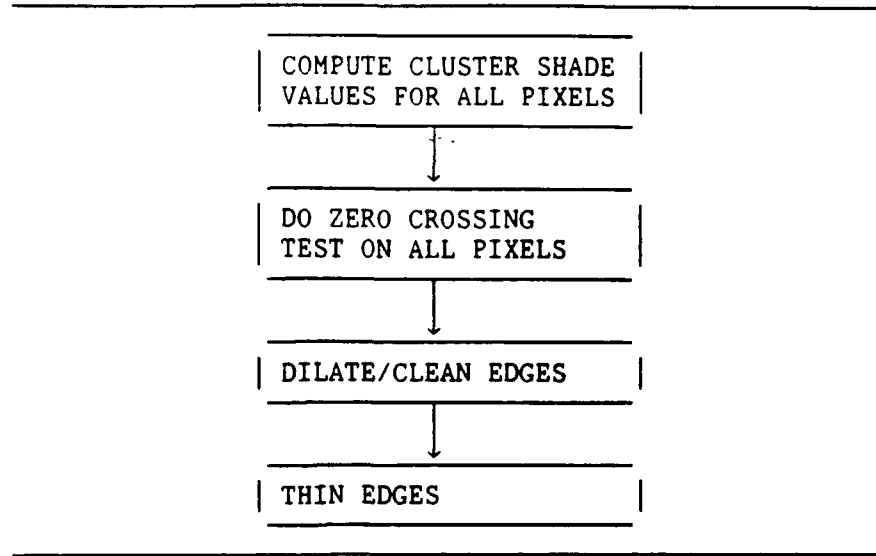


Figure 6. Steps for Cluster Shade edge detection.

$(\Delta x, \Delta y)$, occur within the defined neighborhood, one with intensity i and the other with intensity j . Given an $M \times N$ neighborhood of the input image containing intensity levels ranging from 0 to $(L-1)$, let $f(m,n)$ be the intensity level of the pixel at sample m , line n of the neighborhood. Then

$$P(i,j|\Delta x, \Delta y) = \sum_{n=1}^{N-\Delta y} \sum_{m=1}^{M-\Delta x} A \quad (5)$$

Where:

$$A = \begin{cases} \frac{1}{(M - \Delta x)(N - \Delta y)} & \text{if } f(m,n) = i \text{ and } f(m+\Delta x, n+\Delta y) = j \\ 0 & \text{otherwise} \end{cases}$$

See figure 7 for an example of a GLC matrix computation.

B. The Cluster Shade Texture Measure

The Cluster Shade texture measure is then defined by equation 6 below.

$$S(\Delta x, \Delta y) = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i + j - u_i - u_j)^3 P(i,j|\Delta x, \Delta y) \quad (6)$$

Where:

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

2	2	2	2	2	0
2	2	1	0	1	2
2	1	0	0	2	1
2	1	0	0	1	2
2	1	1	2	1	2
2	1	1	2	2	2

i \ j	0	1	2
0	2/30	2/30	1/30
1	3/30	2/30	5/30
2	1/30	7/30	7/30

Figure 7. The GLC matrix is computed for the 6 x 6 image on the left using the parameters: $\Delta x = 1$; $\Delta y = 0$; $M = 6$; $N = 6$; and 3 grey levels (0, 1, and 2).

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

See figure 8 for an example of Cluster Shade texture measure computation; and see figure 9 for a graph showing the Cluster Shade texture measure values plotted across an edge.

C. Zero Crossing Test

The zero crossing test has two steps. The first step is a simple test given a threshold and a single pass through the Cluster Shade texture image. The user selects an initial threshold to be used for this test. Each 3 x 3 overlapping neighborhood of the computed Cluster Shade texture image is tested. If the center pixel of the 3 x 3 neighborhood has an absolute value greater than or equal to the initial threshold, then the 8 neighbors of that pixel will be tested. If any of the 8 neighbors have an absolute value greater than or equal to the initial threshold and have an opposite sign from the center neighborhood pixel, then that center pixel is considered an edge pixel. Notice that for every zero crossing two pixels will ultimately be set the negative and positive pixels of the crossing. Edge pixels will be set 1, while non-edge pixels will be set to 0 in the output binary edge image. See figure 10 for examples of the first step zero crossing test.

The second step of the zero crossing test is optional. This step is used to extend edges detected in the first zero crossing test. The parameters involved are a second or minimum threshold and maximum allowed passes. Multiple passes are made through the already created edge image and the Cluster Shade texture measure image. The image is reprocessed until the maximum number of passes are used or no more edge pixels are found. For each non-edge pixel of the already created edge image the 3 x 3 neighborhood centered about that pixel are tested. First one of the 8 neighboring pixels must be a predefined edge pixel, either by the initial zero crossing test or a

$$\begin{aligned}
 u_i &= 0 * (2/30 + 2/30 + 1/30) + \\
 &\quad 1 * (3/30 + 2/30 + 5/30) + \\
 &\quad 2 * (1/30 + 7/30 + 7/30) = 40/30 = 1.33\bar{3}
 \end{aligned}$$

$$\begin{aligned}
 u_j &= 0 * (2/30 + 3/30 + 1/30) + \\
 &\quad 1 * (3/30 + 2/30 + 5/30) + \\
 &\quad 2 * (1/30 + 5/30 + 7/30) = 37/30 = 1.23\bar{3}
 \end{aligned}$$

$$u_i + u_j = 2.56\bar{6}$$

$$\begin{aligned}
 S(\Delta x, \Delta y) &= (0 + 0 - 2.56\bar{6})^3 * (2/30) + \\
 &\quad (0 + 1 - 2.56\bar{6})^3 * (2/30) + \\
 &\quad (0 + 2 - 2.56\bar{6})^3 * (1/30) + \\
 &\quad (1 + 0 - 2.56\bar{6})^3 * (3/30) + \\
 &\quad (1 + 1 - 2.56\bar{6})^3 * (2/30) + \\
 &\quad (1 + 2 - 2.56\bar{6})^3 * (5/30) + \\
 &\quad (2 + 0 - 2.56\bar{6})^3 * (1/30) + \\
 &\quad (2 + 1 - 2.56\bar{6})^3 * (7/30) + \\
 &\quad (2 + 2 - 2.56\bar{6})^3 * (7/30) = -1.07
 \end{aligned}$$

Figure 8. Using the GLC matrix computed in figure 7 here is the computation for the corresponding Cluster Shade texture measure.

previous pass of the extended zero crossing test. If one of the 8 neighbor pixels is an edge pixel then a test like that described in the step 1 zero crossing test is performed using the Cluster Shade texture values for that neighborhood. In this case the lesser or minimum threshold is used for the test. If the center pixel of the 3 x 3 neighborhood has an absolute value greater than or equal to the minimum threshold, then the 8 neighbors of that pixel will be tested. If any of the 8 neighbors have an absolute value greater than or equal to the minimum threshold and have an opposite sign from the center neighborhood pixel, then that center pixel is considered an edge pixel. See figure 11 for an example of the second step zero crossing test.

D. Clean Edges

To eliminate isolated pixels or small groups of isolated pixels the edge image is cleaned. The cleaning operation is very simple. The border pixels of each overlapping window of the binary edge image are checked. If all of the border pixels of the neighborhood are 0 then the entire window is set to 0 -

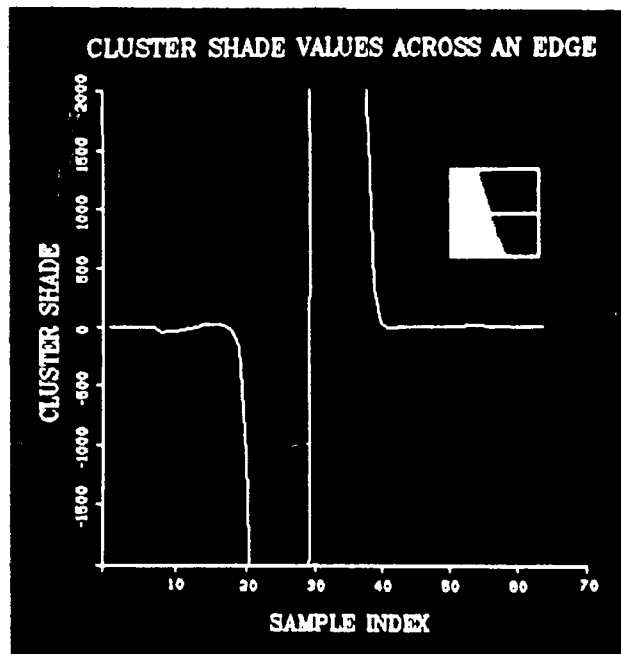


Figure 9. Cluster Shade texture measure values plotted across an image edge.

<table><tr><td>6</td><td>5</td><td>-21</td></tr><tr><td>-9</td><td>20</td><td>20</td></tr><tr><td>-10</td><td>-25</td><td>19</td></tr></table>	6	5	-21	-9	20	20	-10	-25	19	<table><tr><td>X</td><td>X</td><td>-21</td></tr><tr><td>X</td><td>20</td><td>X</td></tr><tr><td>X</td><td>-25</td><td>X</td></tr></table>	X	X	-21	X	20	X	X	-25	X	EDGE
6	5	-21																		
-9	20	20																		
-10	-25	19																		
X	X	-21																		
X	20	X																		
X	-25	X																		
<table><tr><td>6</td><td>5</td><td>21</td></tr><tr><td>-9</td><td>20</td><td>20</td></tr><tr><td>-10</td><td>-15</td><td>19</td></tr></table>	6	5	21	-9	20	20	-10	-15	19	<table><tr><td>X</td><td>X</td><td>X</td></tr><tr><td>X</td><td>20</td><td>X</td></tr><tr><td>X</td><td>X</td><td>X</td></tr></table>	X	X	X	X	20	X	X	X	X	NO EDGE
6	5	21																		
-9	20	20																		
-10	-15	19																		
X	X	X																		
X	20	X																		
X	X	X																		

Figure 10. Example of the zero crossing test step 1, using an initial threshold of 20.

PREVIOUSLY DEFINED EDGES	CLUSTER SHADE VALUES
1 1 0	20 -25 6
0 0 0	10 15 5
0 0 0	-10 6 6
1 1 X	X -25 X
X 0 X	X 15 X
X X X	-10 X X

Figure 11. Shown in this example is an edge to be defined by step 2 of the zero crossing test. A minimum threshold of 5 is used.

cleaned. If any of the border pixels of the neighborhood are 1 then the window will remain unchanged. For an example of this see figure 12.

E. Dilate Edges

Dilation is done to close small 1 or 2 pixel gaps in edges or to thicken edges for further processing modules. For each edge pixel contained within the image set the 8 immediate neighbor pixels are set to be edge pixels. Multiple passes can be made through the image to thicken the edges and in some cases connect edges. See figure 13 for an example of dilation, notice that the two edges in the example original image are connected by pass 2.

F. Line Thinning

The line thinning algorithm used is one described by Pavlidis [3]. The algorithm will not be described here. The reasons for using this particular line thinning algorithm is that it preserves end points and retains connectivity. For an example of the line thinning process see figure 14.

V. RESULTS

The results to be shown are produced by varying the parameters of the Cluster Shade edge detection modules. The edges will be computed using the image shown in figure 1. First the results of changing the window size when computing the Cluster Shade texture values will be shown. These examples will be processed with the Δ values set to 0. To view these as edge images they have been processed through the zero crossing test using an initial threshold of 20, a minimum threshold of 5, and up to 30 allowed extra passes. Figures

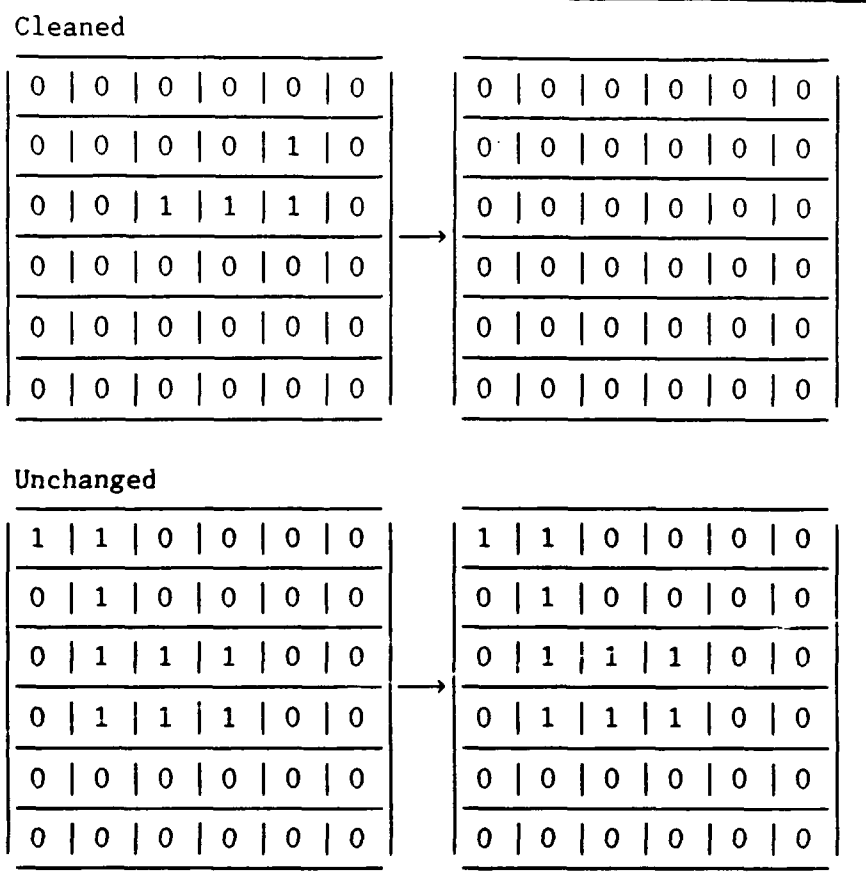


Figure 12. Shown in this example are two 6 x 6 image windows. The first example has no edge pixels in the boundary and, thus, is cleaned. The second example has two edge pixels in the boundary and, thus, remains unchanged.

15, 16, 17, and 18 are examples using a processing window size of 5 x 5, 10 x 10, 16 x 16 and 20 x 20, respectively. The lines in figures 17 and 18 are much more concise than those shown in figures 15 and 16. The other parameters which can be varied in the computation of the Cluster Shade texture measure values are the Δ values. These can be varied together or independently. The Cluster Shade texture measure values for these examples will be computed using the 16 x 16 processing window. The zero crossing test will again be computed using an initial threshold of 20, a minimum threshold of 5, and up to 30 allowed extra passes. For an example with both Δ values set to 0 see figure 17. Figures 19, 20, and 21 use Δ values of 3, 7, and 10, respectively. The larger the Δ values the less concise the edges. Next the Δ values are modified separately. Figure 22 shows the results of using $\Delta x = 0$ and $\Delta y = 7$; and figure 23 shows the results of using $\Delta x = 7$ and $\Delta y = 0$. Figure 24 shows

ORIGINAL	PASS 1	PASS 2
000000000000	000000000000	000000000000
000000000000	000000000000	000111110000
000000000000	000011110000	000111111000
000001000000	000011110000	000111111000
000000100000	000011110000	000111111000
000000100000	000011110000	000111111000
000001100000	000011110000	000111111000
000000000000	000011110000	000111111000
000000000000	000000000000	000111111000
000000000000	000011110000	000111111000
000001000000	000011110000	000111111000
000001000000	000011110000	000111111100
000000100000	000011111000	000111111100
000000110000	000001111000	000111111100
000000000000	000001111000	000011111100
000000000000	000000000000	000011111100
000000000000	000000000000	000000000000
000000000000	000000000000	000000000000
000000000000	000000000000	000000000000
000000000000	000000000000	000000000000
000000000000	000000000000	000000000000

Figure 13. Shown in this example is an original image, the same image after 1 pass of dilation, and then again after 2 passes of dilation.

the results of using $\Delta x = 0$ and $\Delta y = 10$; and figure 25 shows the results of using $\Delta x = 10$ and $\Delta y = 0$. The more horizontal edges seem to thicken or duplicate for larger Δy values, while, the vertical edges seem to thicken or duplicate for larger Δx values.

The next step is to modify the parameters of the zero crossing test or second stage of the Cluster Shade edge detection system. First the initial threshold will be allowed to change and no extra passes used to extend the edges. For these examples the Cluster Shade texture measure will be computed using a processing window of 16×16 and Δ values of 0. Figures 26, 27, 28, and 29 show the results of using an initial threshold of 5, 20, 100, and 200, respectively. The next examples to be shown will also modify the initial threshold, but this time the minimum threshold will be set to 5 and the maximum allowed extra passes set to 40. Figures 30, 31, and 32 show the results of using an initial threshold of 20, 100, and 200, respectively. The larger the initial threshold the fewer edges are defined. Notice that the edges are extended when the extra passes and minimum threshold parameters are used.

The next processing stage is the clean/dilate step. The window size will be set to 16×16 for the clean operation. This parameter will not be modified. Figure 33 shows the results of the clean with no dilation. Figures 34, 35, and 36 show clean with 1, 2, and 3 steps of dilation, respectively. The input image was produced with the Cluster Shade texture measure values

Figure 14. Shown in this example is an original image and a thinned version that same image.

The next examples were produced to further show the results of dilation and the effects of line thinning. There are no tunable parameters for the line thinning operation. Figures 37, 38, 39, and 40 show the results of thinning the images shown in figures 33, 34, 35, and 36, respectively. The more an image has been dilated to the point of running edges together, the more spider web like pattern will be produced in the thinned image.



Figure 15. Cluster Shade edge image created from the image shown in figure 1 using a 5 x 5 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.



Figure 16. Cluster Shade edge image created from the image shown in figure 1 using a 10 x 10 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.

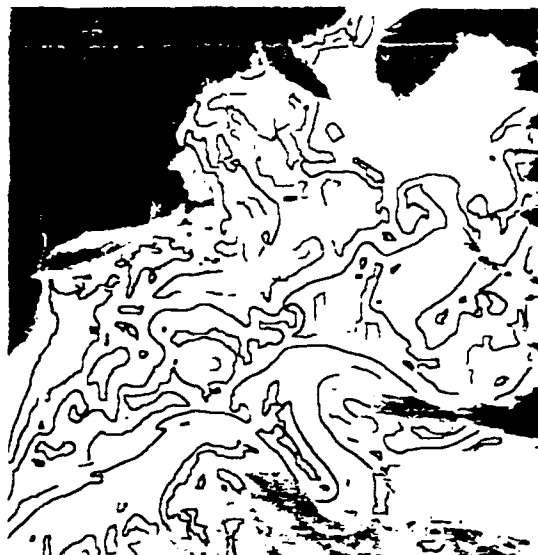


Figure 17. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.

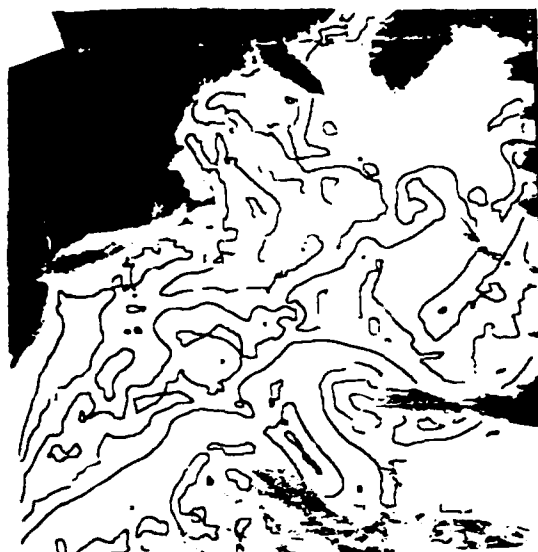


Figure 18. Cluster Shade edge image created from the image shown in figure 1 using a 20 x 20 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.



Figure 19. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 3 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.

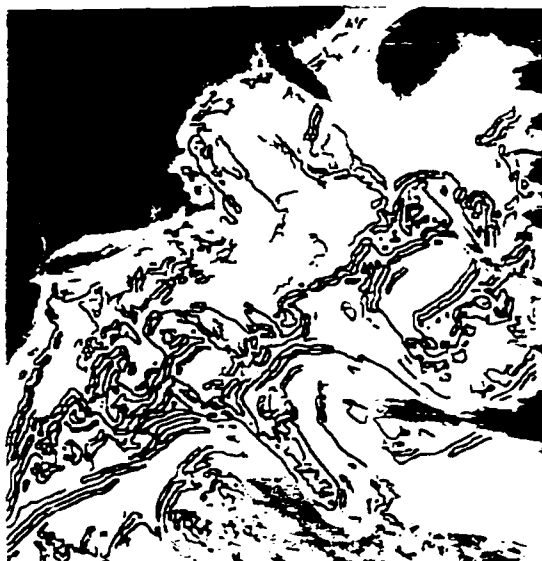


Figure 20. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 7 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.



Figure 21. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 10 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.



Figure 22. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with $\Delta_x = 0$ and $\Delta_y = 7$ to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.



Figure 23. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with $\Delta_x = 7$ $\Delta_y = 0$ to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.



Figure 24. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with $\Delta_x = 0$ and $\Delta_y = 10$ to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.



Figure 25. Cluster Shade edge image created from the image shown in figure 1 using a 16×16 processing window with $\Delta_x = 10$, $\Delta_y = 0$ to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5 and maximum number of passes set to 30 for the zero crossing test.



Figure 26. Cluster Shade edge image created from the image shown in figure 1 using a 16×16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 5 with no extra line extending passes for the zero crossing test.



Figure 27. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20 with no extra line extending passes for the zero crossing test.



Figure 28. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 100 with no extra line extending passes for the zero crossing test.



Figure 29. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 200 with no extra line extending passes for the zero crossing test.



Figure 30. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5, and up to 40 extra line extending passes for the zero crossing test.

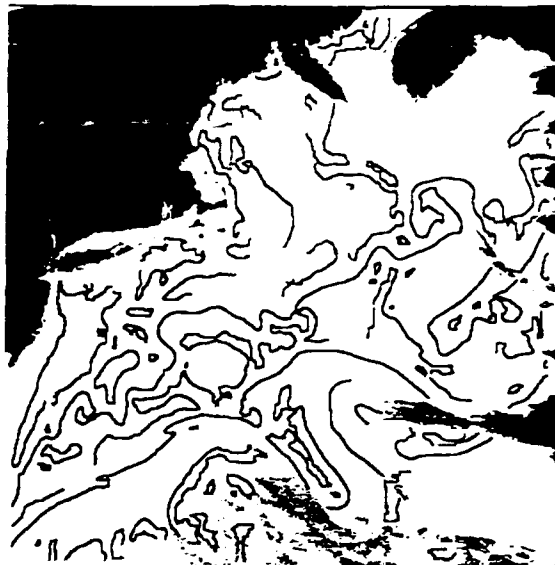


Figure 31. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 100, a minimum threshold of 5, and up to 40 extra line extending passes for the zero crossing test.



Figure 32. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 200, a minimum threshold of 5, and up to 40 extra line extending passes for the zero crossing test.



Figure 33. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5, and up to 40 extra line extending passes for the zero crossing test. The edge image has been cleaned with a 16 x 16 window.



Figure 34. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5, and up to 40 extra line extending passes for the zero crossing test. The edge image has been cleaned with a 16 x 16 window and dilated 1 pass.



Figure 35. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5, and up to 40 extra line extending passes for the zero crossing test. The edge image has been cleaned with a 16 x 16 window and dilated 2 passes.



Figure 36. Cluster Shade edge image created from the image shown in figure 1 using a 16 x 16 processing window with Δ values of 0 to compute the Cluster Shade texture values and an initial threshold of 20, a minimum threshold of 5, and up to 40 extra line extending passes for the zero crossing test. The edge image has been cleaned with a 16 x 16 window and dilated 3 passes.



Figure 37. The edge image from figure 33 after line thinning.



Figure 38. The edge image from figure 34 after line thinning.



Figure 39. The edge image from figure 35 after line thinning.



Figure 40. The edge image from figure 36 after line thinning.

VI. REFERENCES

- [1] R.J. Holyer and S.H. Peckinpaugh, "Edge detection applied to satellite imagery of the oceans," IEEE Transactions on Geoscience and Remote Sensing, Vol. 27, No. 1, pp. 46-56, January 1989.
- [2] 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.
- [3] T. Pavlidis, "A thinning algorithm for discrete binary images," Computer Graphics and Image Processing, Vol. 13, pp. 142-157, 1980.

A HYBRID TECHNIQUE FOR INTERPRETING MESOSCALE FEATURES IN OCEANOGRAPHIC SATELLITE IMAGES

N. Krishnakumar and S.Sitharama Iyengar
Department of Computer Science
Louisiana State University
Baton Rouge, LA 70803.

I. INTRODUCTION

Satellite-borne sensors potentially offer many advantages for the study of oceanic processes. They provide global synoptic measurements of various oceanic surface properties, in contrast to the local measurements, possibly at a range of depths, provided by conventional oceanographic measurement techniques. Thermal infrared (IR) images of the ocean obtained from satellite sensors are widely used for the study of ocean dynamics. Brightness in an infrared image is inversely proportional to the ocean surface temperature (dark areas represent warmer temperatures and light areas represent colder temperatures). Vortices (areas of closed circulation) within this turbulent flow pattern are called *eddies*. The Gulf Stream and its associated eddies are examples of mesoscale features ("mesoscale" is the name commonly applied to the features existing on spatial scales of the order of 50 to 300 km). Mesoscale features are important to the study of ocean dynamics, to the fisheries and to many other diverse interests.

Current image analysis techniques rely on human interpretation of the satellite imagery. Human interpretation is obviously varied in its level of expertise and is highly labor-intensive. With the proliferation of high volume Advanced Very High Resolution Radiometer image applications, it becomes highly desirable for certain applications to move from the labor-intensive manual interpretation of infrared imagery towards 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 problem with the present-day image processing and artificial intelligence techniques. This was the motivation for the work reported by Lybanon et al., [Lybanon86], in the development of a prototype oceanographic expert system.

Several previous studies have addressed the automation of the analysis of infrared satellite imagery for mesoscale features. Gerson and Gabroski [Gerson77] and Gerson et al.[Gerson82] investigated the detection of the Gulf Stream in infrared images from the Geostationary Operational Environmental Satellite (GOES). Gerson and Gabroski [Gerson77] used a hierarchical approach where 16x16 pixel (128x128 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 5x5 pixel "local neighborhoods". As an outgrowth of the work reported in [Gerson77] and [Gerson82], Coulter [Coulter83] performed automated feature extraction studies using the higher resolution Advanced Very High Resolution Radiometer data. Janowitz [Janowitz85] studied the automatic detection of the Gulf Stream eddies using Advanced Very High Resolution Radiometer data. Nichol [Nichol87] used a region adjacency graph to define spatial relationships between elementary connected regions of constant gray level called atoms. Although satisfactory emulation of human extraction of eddy structure is claimed for this method, Nichol[Nichol87] did point out that not all enclosed uniform areas identified by the method will correspond to real ocean structure. Krishnakumar et al[Krishnakumar89-I][Krishnakumar90-I] had earlier approached this problem and shown some preliminary results on the relaxation based feature labeling scheme, the development of a prototype expert system and the interface between these two modules. Krishnakumar et al[Krishnakumar90-II] have developed an hybrid architecture to solve the image interpretation problem for oceanographic satellite

images. The present paper highlights the technique that is described in [Krishnakumar90-II]. For a complete discussion on the technique and the results, please refer [Krishnakumar90-II]. Goodenough et.al [Goodenough87] have developed an expert system for remote sensing applications. In [Goodenough87] the knowledge is presented in the production rules and frame databases. The expert system given in [Goodenough87] is used to study the spatial differences between maps and images.

II. MOTIVATION OF THE PRESENT WORK

Our primary objective is to build a powerful automatic image interpretation system for oceanographic satellite images. In order to make this difficult problem tractable, we divide the problem into two parts: *feature labeling problem* and *the development of an expert system*. It is clear that the performance of the labeling algorithm depends heavily on the low level image processing algorithms. Particularly, the output of an edge detector algorithm plays a major role in the feature labeling process. In view of this, a new efficient edge detector algorithm proposed by Holyer and Peckinpaugh [Holyer89] is employed in our feature labeling technique. It is known that the conventional edge derivative operators are very sensitive to noise and are not suitable for analyzing oceanographic satellite images. The new edge detector algorithm proposed by Holyer and Peckinpaugh [Holyer89] is based on the gray level co-occurrence matrix, which is commonly used in image texture analysis. This algorithm is found to exhibit the characteristics of fine structure rejection while retaining edge sharpness. In this paper we focus on the importance of an expert system in building an automatic oceanographic image interpretation system.

The objective of an oceanographic expert system is to correctly interpret the dynamics of the ocean process with minimal human interaction. Towards this objective, the development of a powerful oceanographic expert system is under way at Remote Sensing Branch, NOARL. We have already developed a prototype expert system for Gulf Stream regional dynamics [Lybanon86]. One of the elements of an expert system is a database of knowledge about the subject matter, with the knowledge represented in a form suitable for manipulation by the "inference engine" of the expert system (i.e., the logical and heuristic procedures for solving problems in the problem domain). With the help of the knowledge gained from discussions at NOARL and from the literature in oceanography, a knowledge base about the nature of the mesoscale features (occurrence, mean lifetime, movement, etc.) was built [Lybanon86].

In this paper, we describe the technique that are utilized in [Krishnakumar90-II] to interpret the features, discuss the features of the technique and possible improvements. The underlying feature labeling technique exploits the advantages of using contextual information in the labeling algorithm. The remainder of the paper is organized as follows: Section III gives an overview of the architecture of the proposed image interpretation system. Section IV introduces the probabilistic relaxation scheme and briefly discusses the application of this scheme to the oceanographic labeling problem. Section V describes the need for an oceanographic expert system and the implementation of the expert system. Section VI concludes with the brief summary on the proposed technique and future extensions to the available scheme.

III. ARCHITECTURE OF THE INTERPRETER

In this section, we briefly describe the various components involved in the image interpretation system. The block diagram shown in figure.1 depicts the underlying architecture of the proposed system. The IR image obtained from the satellite is the input to the feature labeling module. The feature labeling

module consists of two submodules : (a) estimating the initial feature probabilities and (b) implementing the iterative updating scheme. The initial pixel probabilities are assigned with the help of the previous analysis and the ground truth data. These probability values are then iteratively updated using the nonlinear probabilistic relaxation scheme. Finally, the relevant features that are present in the image are labeled with the aid of a decision process.

The labeled features are then fed to the expert system module. This expert system is divided into two submodules. The first submodule consists of a knowledge base and an inference engine. The knowledge base has a thorough collection of facts and rules about the mesoscale features such as gulf stream, warm core rings etc., Based on the positional information obtained from the feature labeling module and the knowledge base, the inference engine attempts to interpret the dynamics of these features. Basically the inference engine is a pattern matching module. The second submodule evaluates the consistency of the labeling process with the help of the knowledge base as well as from the input from the operator. This evaluation results in assigning a confidence factor to each of the features detected. This factor is then "fed back" to the feature labeling module, in addition to the previous data analysis, to improve the consistency of the labeling process in future.

A salient feature of this hybrid architecture is the feedback from the knowledge based system to the labeling module. Due to atmospheric effects like cloud cover, the oceanic features may not be identified by the feature labeling module. The position of the Gulf Stream and eddies may not be determined correctly. In such cases, the expert system provides an approximate position of these features. This helps in interpreting the subsequent images. Another feature is the dynamic interpretation of the features. The hybrid architecture allows the expert system to learn from previous analysis. Also the modular approach allows the user to maintain the system and incorporate changes without much difficulties.

IV. PROBABILISTIC RELAXATION LABELING

An important research area in image analysis and image interpretation technology is the development of methods that blend contextual information with conventional image processing algorithms. A literature survey clearly indicates that such a hybrid approach yields good results. Relaxation labeling is one such process. Relaxation labeling has been applied to a variety of image processing problems e.g., linear feature enhancement [Zucker77], edge enhancement [Schachter77], image enhancement [Davis78], pixel classification [Eklundh80][Davis83]. A recent survey article by Kittler and Illingworth [Kittler85] on relaxation labeling highlights the importance of this area of research. The survey also points out the advantages and possible applications of relaxation methods. More importantly, the relaxation labeling approach was elegantly described by Rosenfeld et al. [Rosenfeld76] who investigated the problem of labeling the sides of a triangle and proposed a set of schemes to solve the problem. The paper [Rosenfeld76] concluded with the result that the nonlinear probabilistic relaxation schemes yield better results than the others. The labeling algorithm presented in this paper is based on the non linear probabilistic relaxation technique.

The goal of the relaxation process is to reduce the uncertainty (and improve the consistency) in the assignment of one of the labels to each object in a set of related objects. In the oceanographic feature classification problem, the classes are the various oceanographic features, namely the north and south walls of the Gulf Stream, cold eddy, warm eddy, shelf front and coastal boundary. The objects are the individual pixels in a set of registered multi-temporal images. The uncertainty could be due to the cloud cover or the overlap of the features, features not belonging to one of the classes, noise in the image, or other factors. In this paper, we are attempting to label mesoscale features, but the ocean exhibits variability on all spatial

scales. Thermal structure on scales smaller than mesoscale will interfere with the mesoscale feature labeling process. The underlying mathematical framework necessary for the relaxation labeling method is described in the next section of the paper.

A. MATHEMATICAL FRAMEWORK FOR RELAXATION LABELING

Let $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$ be the set of possible labels that may be assigned to each pixel x in the IR image. Also we let $p_\lambda^k(x)$ denote the probability that the pixel at $x(i,j)$ belongs to the object λ after k iterations of the relaxation algorithm. Note that the probabilities are functions of time unlike the conventional pixel relaxation labeling schemes where the probability is a function of position alone. This allows the relaxation labeling algorithm to utilize temporal continuity to reduce the ambiguity in labeling. The ambiguity may arise due to noise (cloud cover, for example).

There are two steps in executing the probabilistic relaxation algorithm. In the first step, a priori probabilities are evaluated with the help of ground truth data and / or a previous but recent mesoscale analysis. In the second step, these a priori probabilities are iteratively updated (relaxation) until a consistent labeling is reached. We now discuss these two steps in detail.

Step 1: Estimating the a priori probabilities

Let $p_\lambda^0(x)$ denote the a priori value, that is, the probability that pixel $x(i,j)$ belongs to the object λ at the zeroth iteration. The Bayesian probability equation is used to evaluate this value. The equation (IV.A.1) is used to calculate $p_\lambda^0(x)$.

$$p_\lambda^0(x) = \frac{p(x|\lambda) P(\lambda)}{\sum_{\lambda} p(x|\lambda) P(\lambda)} \quad (\text{IV.A.1})$$

where $p(x|\lambda)$ denotes the conditional density function and $P(\lambda)$ the probability of occurrence of the object λ .

To evaluate the conditional density function $p(x|\lambda)$, a set of parameters is measured at the pixel $x(i,j)$. Let X denote the parameter vector. The following parameters are used to form the vector X :

- (1) vector from origin to pixel $x(i,j)$, both the magnitude and direction.
- (2) gray scale intensity value at the pixel $x(i,j)$.
- (3) the edge magnitude (Section IV.B presents the chosen edge operator algorithm).

For each object, the mean vector μ_λ and the covariance matrix Σ_λ are computed. Also it is assumed that the conditional density function follows a normal distribution. Hence the conditional density function $p(x|\lambda)$ is evaluated using equation (IV.A.2).

$$p(x|\lambda) = (2\pi|\Sigma_\lambda|)^{-1/2} \exp\left\{-\frac{1}{2} (X-\mu_\lambda)' \Sigma_\lambda^{-1} (X-\mu_\lambda)\right\} \quad (\text{IV.A.2})$$

To compute $P(\lambda)$, relative areas of the objects are considered. The number of pixels in the object λ is n_λ . Then $P(\lambda)$ can be calculated using equation (IV.A.3).

$$P(\lambda) = \frac{n_\lambda}{\Sigma_\lambda n_\lambda} \quad (\text{IV.A.3})$$

Step 2: Iterative updating algorithm

We now discuss the probability updating rule. The new estimate of the probability of λ at $x(i,j)$ is given by (IV.A.4).

$$p_\lambda^{k+1}(x) = \frac{p_\lambda^k(x) (1+q_\lambda^k(x))}{\Sigma_\lambda p_\lambda^k(x) (1+q_\lambda^k(x))} \quad (\text{IV.A.4})$$

where $q_\lambda^k(x)$ is called the update factor.

The updating factor for the estimate $p_\lambda^k(x)$ at the k th iteration is given by equation (IV.A.5).

$$q_\lambda^k(x) = \frac{1}{m} \Sigma_y \Sigma_{\lambda'} r_{\lambda\lambda'}(x,y) p_{\lambda'}^k(y) \quad (\text{IV.A.5})$$

where m is the number of objects. In this equation, $r_{\lambda\lambda'}(x,y)$ denote compatibility coefficients. These coefficients are computed as in [Davis83][Rosenfeld76]. According to the relaxation scheme, $r_{\lambda\lambda'}(x,y)$ is a measure of the probabilistic compatibility between label λ on point x and label λ' on point y , and has the following characteristics:

- (1) If λ on x frequently co-occurs with λ' on y , then $r_{\lambda\lambda'}(x,y) > 0$, and if they always co-occur, then $r_{\lambda\lambda'}(x,y) = 1$.
- (2) If λ on x rarely co-occurs with λ' on y , then $r_{\lambda\lambda'}(x,y) < 0$, and if they never co-occur, then $r_{\lambda\lambda'}(x,y) = -1$.
- (3) If λ on x occurs independently of λ' on y , then $r_{\lambda\lambda'}(x,y) = 0$.

B. DISCUSSION OF THE TECHNIQUE

The implementation of the above mentioned technique is carried out in two stages. The flow diagram shown in figure 2 illustrates the steps involved in the implementation of the feature labeling algorithm using the nonlinear probabilistic relaxation scheme.

Stage 1

At the first stage, a priori probabilities are estimated using a manually prepared mesoscale analysis from a time period of two days prior to the test image. The objects present in the oceanographic IR image are identified with the help of previous analysis. A set of pixels on the north and south wall of the Gulf Stream are provided by the previous analysis. The center pixel and radii of the eddies are also given. From the parameter vector X for a pixel $x(i,j)$, the mean vector μ_λ is calculated. Also for each object λ , the covariance matrix Σ_λ is computed. Equation (IV.A.2) is used to compute the conditional density function $p(x|\lambda)$. Finally the initial probability $p_\lambda^0(x)$ is computed using the equation (IV.A.1).

Stage 2

In the second stage, the iterative updating rule is implemented. The compatibility coefficients are calculated using the initial class probability values obtained in the first stage. These are fixed during the update process. As a concluding step in the second stage of the implementation, the iterative updating algorithm is implemented using the equation (IV.A.4). The iterative algorithm terminates when $(p_\lambda^{k+1}(x) - p_\lambda^k(x)) < \epsilon$, where ϵ is a very small quantity.

Now we discuss the features of the edge detection algorithm proposed by Holyer and Peckinpaugh [Holyer89]. The motivation behind the development of such a new edge detector algorithm is to aid the analysis of oceanographic satellite images. The popular derivative-based edge operators viz, Sobel's operator are shown to be too sensitive to edge fine-structure and to weak gradients to be useful in this application. The edge algorithm proposed by Holyer and Peckinpaugh is based on the cluster shade texture measure, which is derived from the gray level co-occurrence (GLC) matrix. Holyer and Peckinpaugh [Holyer89] have suggested that the edge detection technique based on the GLC matrix can be effectively used in automated detection of mesoscale features. The (i,j) 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 . 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 the edges present in the image. Holyer and Peckinpaugh [Holyer89] have used a cluster shade function which is found to be very effective in the edge detection process. The new edge algorithm computes the cluster shade function at each pixel. Then the edges are detected by finding the significant zero crossings in the cluster shade image. The advantages of this new edge algorithm over the conventional derivative-based techniques are discussed in [Holyer89]. It is known that using large windows in derivative-based edge detector algorithms results in poor smoothing. This problem is circumvented in the new algorithm. Because edges are detected by finding zero crossings, precisely positioned lines result, even if the GLC matrix is calculated using a larger window. So, the desired edge detection characteristic of retaining sharp edges while eliminating edge detail is achieved with the help of the new algorithm. As an input to our feature labeling algorithm, we used the image output generated by cluster shade algorithm, with a window size of 16×16 pixels and zero crossing threshold of 50. The edge magnitudes obtained from this new edge detector algorithm are used as an input to the feature labeling algorithm. In particular the edge magnitudes are used to evaluate a priori probability values.

V. OCEANOGRAPHIC EXPERT SYSTEM

The Remote Sensing Branch of the Naval Oceanographic Atmospheric Research Laboratory (NOARL) studies and develops methods to exploit satellite data to provide oceanographic information. The Navy needs to provide such information to the Fleet, so NOARL has an objective to transform research techniques for the oceanographic interpretation of satellite data into operational procedures. From a large volume of satellite data, combined with non-satellite data, the shipboard user will be required to produce operationally useful tactical products. Dramatic improvement will be required in our ability to automate the image analysis process.

Interactive image processing is a powerful tool in the hands of an expert image interpreter. However, human interpretation is both varied in its level of expertise and labor-intensive. Expert judgement is required for oceanographic image interpretation because there is little or no standardization in either image enhancement or interactive techniques. NOARL has conducted research in the use of concepts from the field of artificial intelligence to develop knowledge-based "expert systems" to automate some aspects of the oceanographic image interpretation task. That work is an attempt to remove human-labor-intensive and varying-skill-level elements from the interpretation of oceanographic image and other satellite data. Lybanon, et al. [Lybanon86] describe the motivation and some of the history of NOARL's work in this area.

The rule base of the expert system is derived from an unprecedented compilation of information, and the expert system's mode of operation is innovative. The rule base represents oceanographic knowledge about the evolution of mesoscale ocean features in the Gulf Stream region of the north Atlantic Ocean. The rules are applied in such a way that the expert system describes the kinematics of that evolution. The expert system is presently implemented in a combination of OPS83, C, and FORTRAN running on a VAX 8300. The expert system uses the rules about the features to evolve an initial "state" to a later time. The new, hypothesized state can then be used as a new initial condition and the process is repeated. This process can be usefully carried out for several steps. This mode of operation is different from expert systems which use a (sometimes lengthy) chain of reasoning to reach a single conclusion, such as a system for medical diagnosis.

The domain of the expert system is the Gulf Stream region of the north Atlantic Ocean. The expert system has different rules for ring and Gulf Stream behavior in each of nine geographical regions. Figure 3 shows the operating regions of the expert system. Aside from the lines that separate them, the regions are bounded only by land. A ring's behavior as hypothesized by the expert system depends upon which region its center is in at the beginning of a time step. Basically, the motion has a region-dependent velocity vector. However, a ring that is closer than a certain critical distance from the Gulf Stream undergoes a modification of the basic motion. The Gulf Stream interaction rules are also region-dependent. The details depend upon how close the ring is to the Gulf Stream; the Gulf Stream interaction may result in a deflection or a looping motion, with possible coalescence with the Stream in some cases. Ring sizes decrease with time. The rate of decrease depends on the region and on whether there is Gulf Stream interaction. A ring that shrinks below a certain size disappears. Gulf Stream motion is modeled as downstream propagation of meanders, with region-dependent phase velocities and amplitude factors. While this is only a first-order model, there is justification in the literature for this behavior. Refer [Lybanon86] for a complete description of the expert system.

VI. SUMMARY AND RECOMMENDATIONS FOR FUTURE WORK

In this paper, the need for automatic interpretation of oceanographic images is emphasized. The advantage of exploiting the contextual information in feature labeling is highlighted. An efficient and simple technique for labeling of oceanic features is described. The underlying theoretical framework and the steps involved in estimating the probability functions are explained in detail. The overall architecture of the proposed image interpretation system is presented. The usefulness of an expert system in feature labeling is discussed. The concept of getting feedback from the expert system is a new approach taken to solve the feature labeling problem.

As a future extension to the present work we propose to :

- (i) make the oceanographic expert system "learn" from its past experience in analyzing the satellite imagery data.
- (ii) investigate the possibility of implementing a parallel relaxation labeling algorithm to speed up the labeling process.
- (iii) integrate the useful data available in the altimetry readings to strengthen the feature labeling algorithm.

VII. REFERENCES

- M. Lybanon, J.D. McKendrick, R.E. Blake, J.R.B. Cockett and M.G. Thomason(1986). A prototype knowledge-based system to aid the oceanographic image analyst. SPIE Vol 635- Applications of Artificial Intelligence III 203-206.
- D.J. Gerson and P.Gaborski(1977). Pattern analysis for automatic location of oceanic fronts in digital satellite imagery. Naval Oceanographic Office, TN 3700-65-77.
- D.J. Gerson, E. Khedouri and P. Gaborski (1982). Detecting the gulf stream from digital infrared data pattern recognition. The Belle W. Baruch Library in Marine Science (12):19-39.
- R.E. Coulter(1983). Application of the Bayes decision rule for automatic water mass classification from satellite infrared data. Proceedings in the 17th International Symposium on remote sensing of environment (2):589-597.
- M.F. Janowitz(1985). Automatic detection of gulf stream rings. Technical report TR-J8501, Office of Naval Research.
- D. G. Nichol(1987). Autonomous extraction of eddy-like structure from infrared images of the ocean. IEEE Transactions on Geoscience and Remote Sensing (25):28-34.
- N. Krishnakumar, S.Sitharama Iyengar, Ron Holyer and Matthew Lybanon (1989-I). A Technique for feature labeling in infrared oceanographic images. Fifth International Conference on IIPS, 368-375.
- N. Krishnakumar, S.Sitharama Iyengar, Ron Holyer and Matthew Lybanon (1990-I). A Technique for feature labeling in infrared oceanographic images. Journal of Image and Vision and Computing, 112-119.
- N. Krishnakumar, S.Sitharama Iyengar, Ron Holyer and Matthew Lybanon (1989-II). An Expert System

for Interpreting Mesoscale Features in Oceanographic Satellite Images. Applications of Artificial Intelligence, SPIE, 184-194.

N. Krishnakumar, S.Sitharama Iyengar, Ron Holyer and Matthew Lybanon (1990-II). An Expert System for Interpreting Mesoscale Features in Oceanographic Satellite Images. International Journal of Pattern Recognition and Artificial Intelligence 4(3).

D.G. Goodenough, M. Goldberg, G. Plunkett and J. Zelek (1987). An Expert System for Remote Sensing. IEEE Transactions on GeoScience and Remote Sensing 25(3)349-356.

R.J. Holyer and S.H. Peckinpaugh (1989). Edge detection applied to satellite imagery of the oceans. IEEE Transactions on GeoScience and Remote Sensing 27(1)46-56.

S.W. Zucker, R.A. Hummel and A. Rosenfeld (1977). An application of relaxation labeling to line and curve enhancement. IEEE Transactions on Computers 26,922-929.

B.J. Schachter, A. Lev, S.W. Zucker and A. Rosenfeld (1977). An application of relaxation methods to edge reinforcement. IEEE Transactions on System, Man and Cybernetics. 7, 813-816.

L.S. Davis and A. Rosenfeld (1978). Noise cleaning by iterated local averaging. IEEE Transactions on System, Man and Cybernetics. 8,705-710.

J.O. Eklundh, H. Yamamoto and A. Rosenfeld (1980). A relaxation method in multispectral pixel classification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2):72-75.

L.S. Davis, C.Y. Wang and H.C. Xie (1983). An experiment in multispectral, multitemporal crop classification using relaxation techniques. Computer Vision, Graphics and Image Processing (23):227-235.

J. Kittler and J. Illingworth (1985). Relaxation labeling algorithms - a review. Journal of Image and Vision Computing, 206-216.

A. Rosenfeld, R.A. Hummel and S.W. Zucker (1976). Scene labeling by relaxation operations. IEEE Transactions on System, Man and Cybernetics, 6:420-434.

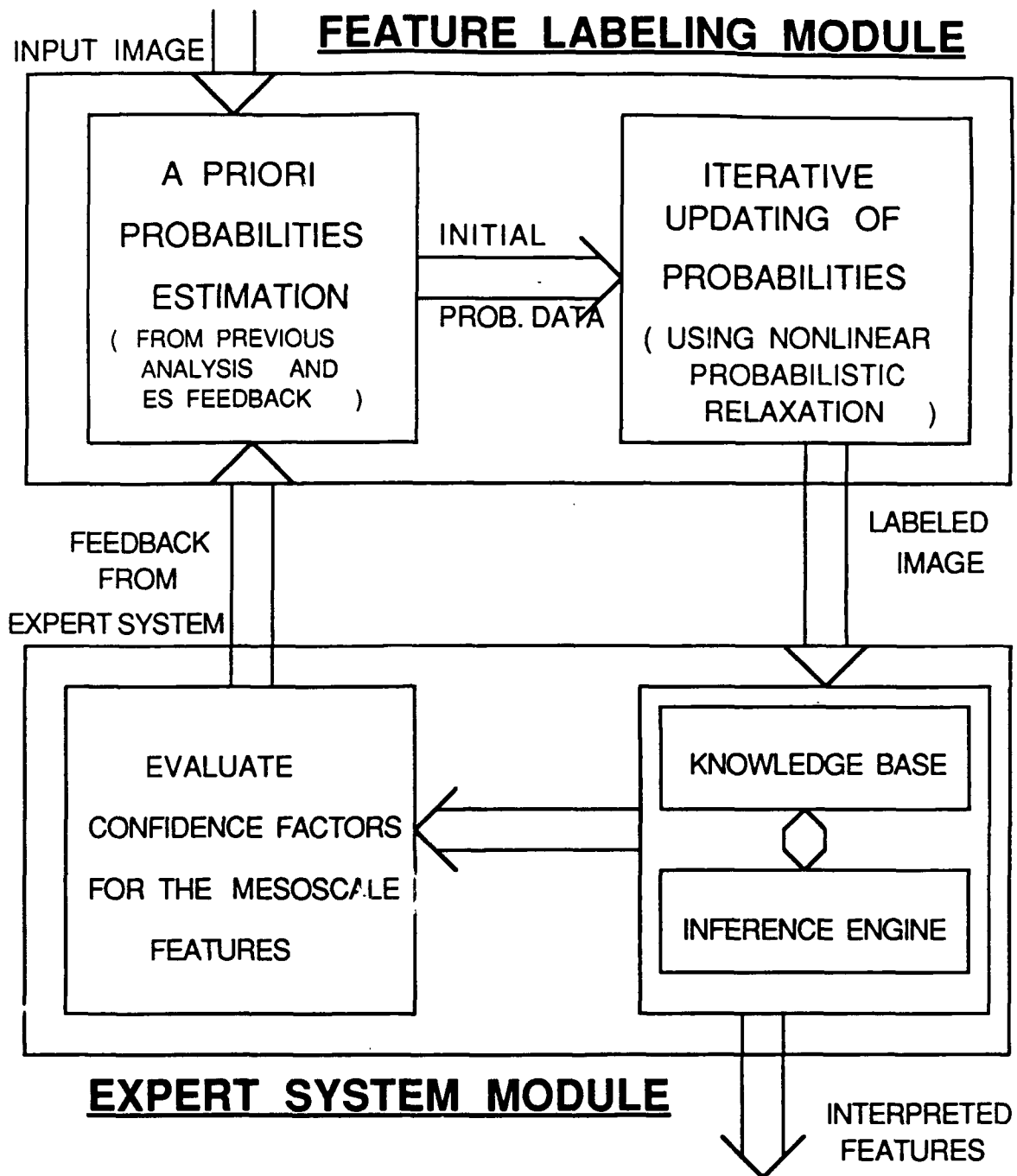


Figure 1. HYBRID ARCHITECTURE

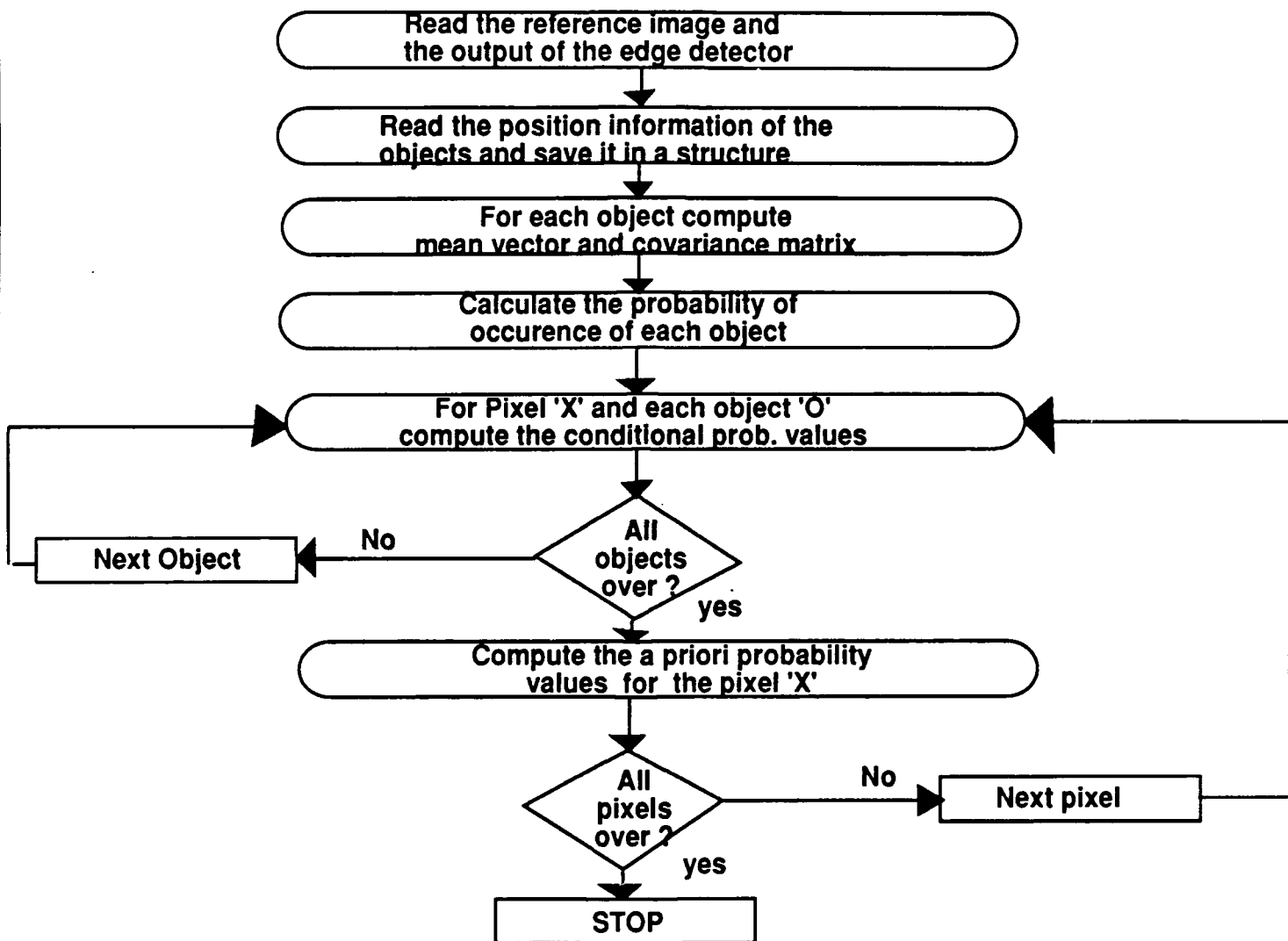


Figure 2. FLOW DIAGRAM FOR LABELING MODULE

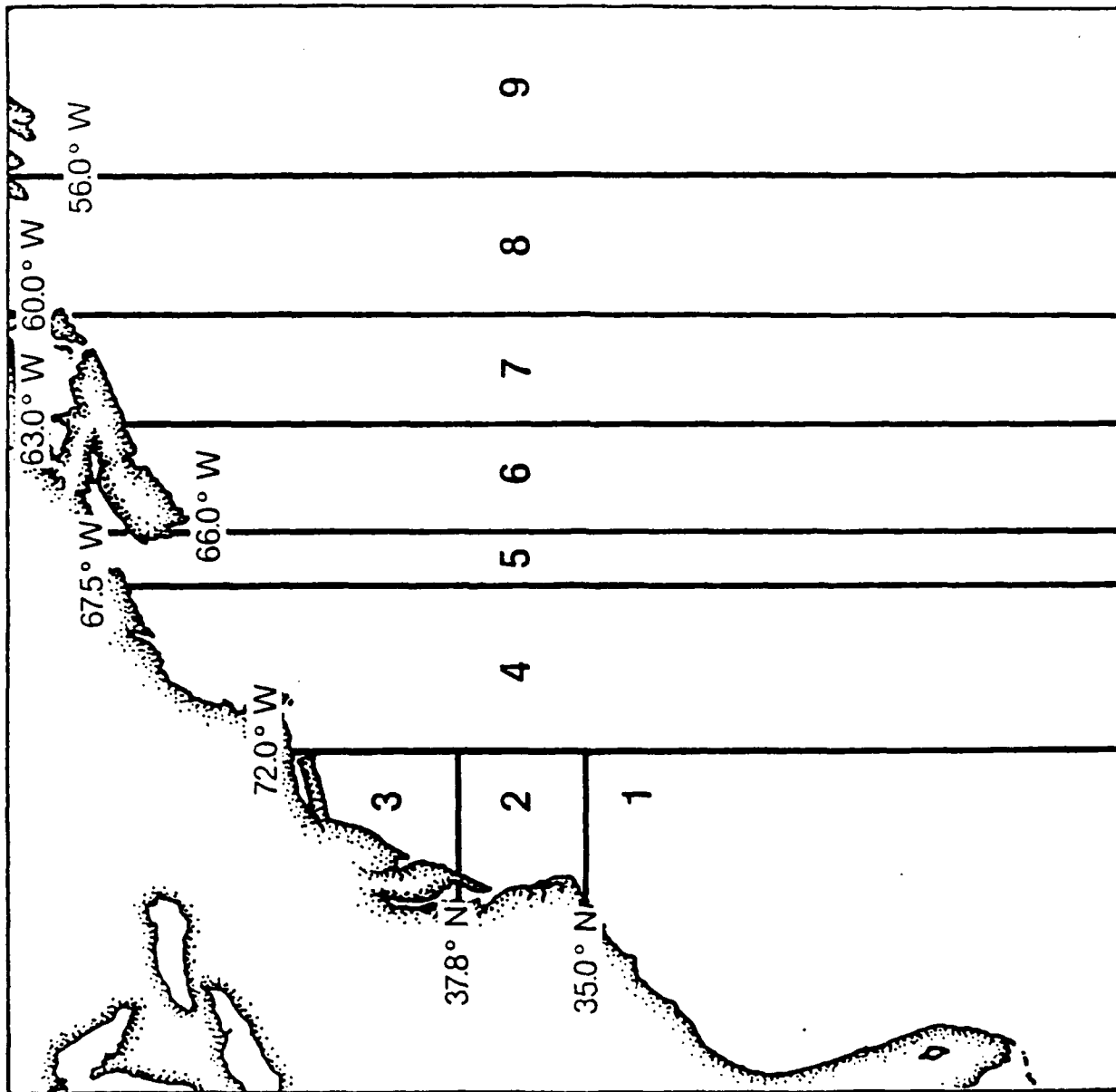


Figure 3. OPERATING REGIONS OF EXPERT SYSTEM

RECENT RESULTS IN OCEANOGRAPHIC EXPERT SYSTEM VALIDATION

Matthew Lybanon
Ocean Sensing and Prediction Division
National Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, MS 39529-5004

I. INTRODUCTION

Satellite imagery of the oceans has become an invaluable tool for the oceanographer, adding the breadth of synoptic coverage to the depth of in situ measurements at a few points. New satellite systems will greatly increase the data volume, increasing the burden imposed by the labor-intensive nature of current processing methods. Consequently, standardization and optimization of interpretative techniques are highly desirable. The Naval Oceanographic and Atmospheric Research Laboratory (NOARL) has sponsored the design of an expert system as part of a system intended to aid the image analyst, since the problem is not amenable to conventional automated image understanding methodology. The expert system's rule base comes from a new, comprehensive compilation of oceanographic information, and the expert system has other unique features. Initial tests of the expert system's ring motion rules were encouraging and indicated some directions for improvements. Both ring position (i. e., motion-induced) and size errors tended to grow linearly with time; size errors were only a few percent after several weeks. While ring motion errors are not negligible, the ring motion rules perform better than an assumption of no motion for periods up to 2 weeks. To a first approximation, the expert system's motion performance could be improved by global corrections to the rules that compensate for the mean errors. A more detailed, case-by-case examination of the errors to identify specific rules in need of modification would be a very labor-intensive manual task and limited in its potential for suggesting rule base improvements, so we are exploring possible automated approaches. This paper presents a summary of the expert system validation tests to date.

II. EXPERT SYSTEM

A. Purpose

Ocean features are time varying; they both move and change their shapes. The frequent presence of cloud cover, atmospheric water vapor, and other types of noise also complicates interpretation. In an operational environment, it may be necessary to provide analyses at regular intervals. An operational analyst who must provide a new analysis when new observations are not available over all or part of the area because of cloud cover or other contamination has two choices: he either assumes the features are in their previous locations, or he must make an "educated guess" about their motion. The NOARL prototype oceanographic expert system facilitates the latter choice, and minimizes the likelihood of significant errors in doing so. Thus, the NOARL expert system has the potential to be a valuable component of a complete automated interpretation system for oceanographic imagery.

The Navy has invested a great deal of research on mathematical models to describe and forecast oceanic events. These models implement the differential equations that describe some aspects of the physics of the oceans. However, the effectiveness of models is limited by the amount and quality of data available to force them. Also, high-resolution mathematical models require very great computing power. This combination of circumstances justifies investigation of other techniques for forecasting in the limited sense required here. And, as described later in this paper, the success of the NOARL oceanographic expert system in hypothesizing the evolution of Gulf Stream rings over time periods of a few weeks justifies this alternate approach.

The expert system has also proved its value in a slightly different application. The system's hypotheses about motion effectively define "regions of interest" near the hypothesized positions, much smaller than the full image, where the features are likely to be found. When no observations are available the hypothesized positions comprise the educated guesses referred to, but when observations are available the regions of interest effectively narrow down

the portions of the image that must be searched for the features. This was used to advantage in relaxation labeling of oceanographic features (Krishnakumar et al., 1990). Relaxation labeling has two steps. In the first step, a priori probabilities are evaluated with the help of ground truth data and/or a previous analysis. In the second step, these probabilities are iteratively updated until a consistent labeling is reached. Use of the expert system to update the previous analysis prior to use in the first step was found to produce a marked improvement in performance.

B. Structure

The NOARL expert system's rule base is derived from an unprecedented compilation of information (Thomason and Blake, 1986), and the expert system's mode of operation is innovative. The rule base represents oceanographic knowledge about the evolution of mesoscale ocean features in the Gulf Stream region of the north Atlantic Ocean. The rules are applied in such a way that the expert system describes the kinematics of that evolution.

The rule base compiled by Thomason and Blake (1986) came from an extensive literature search, supplemented by study of satellite imagery. This was necessary because no existing, formalized compilations of information directly applicable to oceanographic satellite image analyses could be found. Evaluations of the resulting rule base by several outside experts tended to confirm the idea that such a comprehensive compilation of oceanographic information was unique.

The rules which deal with the features' kinematical aspects were incorporated into an expert system which uses the rules to evolve an initial "state" to a later time. The new, hypothesized state can then be used as a new initial condition and the process repeated. This procedure can be usefully carried out for several steps. This mode of operation is different from expert systems which use a (sometimes lengthy) chain of reasoning to reach a single conclusion, such as a system for medical diagnosis.

The NOARL expert system's domain is the Gulf Stream region of the north Atlantic Ocean. It has different rules for ring and Gulf Stream behavior in each of nine geographical regions. A ring's behavior as hypothesized by the expert system depends upon which region its center is in at the beginning of a time step. Basically, the motion has a region-dependent velocity vector (different for warm and cold rings). However, a ring that is closer than a certain critical distance (the critical distance depends on the ring's size) from the Gulf Stream undergoes a modification of the basic motion. The Gulf Stream interaction rules are also region-dependent, and are different for warm and cold rings. The details depend on how close the ring is to the Gulf Stream (there are several interaction regions); the Gulf Stream interaction may result in a deflection or a looping motion, with possible coalescence with the Stream in some cases. Ring sizes decrease with time. The rate of decrease depends on the region and on whether there is Gulf Stream interaction. A ring that shrinks below a certain size disappears. Gulf Stream motion is modeled as downstream propagation of meanders, with region-dependent phase velocities and amplitude factors. While this is only a first-order model, there is justification in the literature for this behavior (Halliwell and Mooers, 1979). The expert system does not currently describe the creation of new rings from pinched-off meanders.

C. Validation Procedure

NOARL has performed validation tests of the expert system's ring motion rules against several sets of mesoscale analyses. The ring motion rules are more highly developed than the Gulf Stream motion logic, so we chose that portion of the expert system for validation first. Expert system validation requires verification that the system is reaching the correct conclusions based on the input data. Since there is no standard interpretive procedure for oceanographic imagery, the best approach is comparison of the expert system's results with accurate mesoscale analyses. The ideal validation criterion would be performance compared to the real ocean. This is impossible, but the procedure chosen is comparable to a "one-on-one" test between the expert system and a human analyst, if not better.

The tests involve comparison of the expert system's results with "ocean truth," as represented by independently-produced mesoscale analyses. The expert system is initialized with locations along the Gulf Stream's north and

south walls, and ring center positions and sizes, from a mesoscale analysis. (Gulf Stream information was needed even though the tests only involved the ring rules, because ring behavior is influenced by proximity to the Gulf Stream.) The tests compared the hypotheses (concerning ring locations and sizes) generated by the system with the mesoscale analysis for the later time. That is, the expert system initialized with information from one analysis ("day 0") generated hypotheses about the state after 7 days, 14 days, etc. The tests compared the 7-day projections with the "day 7" analysis, the 14-day projections with the "day 14" analysis, etc. By initializing the expert system with each analysis from a sequence and comparing the results with later analyses as described, sets of 7-day, 14-day, ... comparisons were obtained.

The analysis of the test results involves both scalar and vector error measures. Let P_1 label the center position of a ring at the initial time and let r_1 be its size, let P_2 and r_2 be the corresponding quantities at the later time (from ocean truth), and let P_3 and r_3 be the values at the later time as hypothesized by the expert system. Then we define scalar goodness of fit (GOF) measures for motion and size as

$$\text{GOF}_{\text{translation}} = d(P_2, P_3) / d(P_1, P_2) \quad (1)$$

and

$$\text{GOF}_{\text{size}} = |1 - r(P_3) / r(P_2)|. \quad (2)$$

In these equations $d(P_i, P_j)$ is the great-circle distance between P_i and P_j . Both GOF measures are small when the expert system's projections are accurate and large when they are not. It is important to note two things about Eq. (1). If $d(P_1, P_2) = 0$ (i. e., if the ring does not appear to move between the two ocean truth data sets), the denominator is zero and Eq. (1) cannot be applied; the comparison must be discarded (for translation). If the hypothesized position at the later time is the same as the initial position--an assumption of no motion--then the numerator and denominator of Eq. (1) are equal. Thus, one interpretation of Eq. (1) is that $\text{GOF}_{\text{translation}} < 1$ implies that the expert system's result is better than an assumption of no motion, while $\text{GOF}_{\text{translation}} > 1$ means that the expert system's result is worse than an assumption of no motion. A ring may be of order 100 km in diameter, while it typically moves only 3-5 km per day, so a no-motion assumption is fairly reasonable for periods of a few days. This simple interpretation of $\text{GOF}_{\text{translation}}$ allows easy comparison of the expert system's results with this assumption.

The value of $\text{GOF}_{\text{translation}}$ is itself useful information. However, one may criticize the scalar measure of motion error on the grounds that it "discards" all information about the direction of those errors. (The size of a ring is a scalar, so a size error only has magnitude and the scalar error measure given by GOF_{size} is complete. However, a motion error is a vector.) If some of the errors are in one direction and some are in the opposite direction, then one might conclude that, on the average, the expert system is doing fairly well. In this sense, $\text{GOF}_{\text{translation}}$ is an unduly pessimistic error measure. Also, Equation (1)'s denominator is uncertain if the true eddy positions are not precisely known in the "ocean truth." In practice there is always some uncertainty in those positions. Since the denominator is likely to be small for shorter time intervals, the uncertainty in $d(P_1, P_2)$ could be of the order of its size--or even greater. So it is necessary to exercise caution in interpreting the results given by Eq. (1).

A different view of the motion errors is provided by vector error measures. The position error for a specific comparison is resolved into components,

$$\Delta x = d(P_2, P_3) \sin \theta, \quad \Delta y = d(P_2, P_3) \cos \theta, \quad (3)$$

where θ is the angle measured from the y (north-south) axis, Δx is the east-west component, and Δy is the north-south component. Once the vector errors are obtained for each comparison, then mean values of Δx and Δy for an entire data set, the magnitude of the mean vector error, and its direction can be obtained. The scalar and vector motion error measures taken together provide a more complete picture than either one separately.

III. VALIDATION TEST RESULTS

A. Data Sets

Tests so far used two ocean truth data sets, each consisting of several temporal sequences, thus allowing a number of comparisons to be made for each of several time periods. The first set consists of GEOSAT Ocean Applications Program (GOAP) mesoscale products (Lybanon et al., 1990). The second set consists of data also used to validate the Harvard Gulfcast model (Robinson et al., 1988) and an ocean nowcast/forecast system developed in the NOARL Data Assimilation Research and Transition (DART) program. The longest sequence in the latter data set was 14 days, compared to 49 days for the GOAP data set. However, the Gulfcast/DART data set is thought to be particularly "clean," while the GOAP data set is less so. In fact, test results and other information indicate that GOAP data set uncertainties in ring positions are of the order of the distances moved for periods up to about 2 weeks. Lybanon (1990) gives details on the data sets, the tests, and the initial results.

B. Initial Results

Mean fractional size errors (GOF_{size}) were only a few percent even after several weeks, and will not be discussed further here. Table 1 presents the comparison of the expert system results with an assumption of no motion for the Gulfcast/DART data set. WCR is an abbreviation for "warm-core ring," and CCR for "cold-core ring." The values listed show that the ring motion rules perform better than the assumption of no motion for periods up to 2 weeks. This result is notable, because a ring typically requires as much as 7 weeks to move a distance equal to its diameter. Therefore, an assumption of no motion is a reasonable assumption for time periods of 1-2 weeks. For more details on the tests and initial results, the reader is referred to Lybanon (1990).

Table 1. Oceanographic Expert System Ring Motion Results for Gulfcast/DART Data Set

<u>Days</u>	<u>Ring Type</u>	<u>Percent Better Than No Motion</u>
7	WCR	61% (based on 31 comparisons)
	CCR	64% (based on 39 comparisons)
	Total	63% (based on 70 comparisons)
14	WCR	55% (based on 11 comparisons)
	CCR	67% (based on 15 comparisons)
	Total	62% (based on 26 comparisons)

C. Results After Modifications

1. Modification 1

Analysis of the vector error results made using the Gulfcast/DART data set showed that the magnitude of the mean vector error increases linearly with time and the direction is constant, to a first approximation. Similar behavior was noted for the results of the tests with the GOAP data set, with similar numerical values. There were different results for CCRs and WCRs: For CCRs, the error trend was 1.0 km/day, heading -120° . For WCRs the error trend was 1.2 km/day, heading 200° . Results are stated to two significant figures for the magnitude and to the nearest 10° for the direction. A global correction to the ring motion rules to compensate for this mean linear trend should result in an improvement, at least for the mean vector error statistic.

This correction amounts to adding a constant correction vector to each ring velocity described by the rules. The correction is different for warm and cold rings, but the same for all rings of one type, in each of the nine regions of the expert system's domain. The expert system's structure makes it easy to apply such a correction to the rule- for rings that do not interact with the Gulf Stream, so that change was performed as a first-order correction to the expert system's ring motion rule base.

Table 2 summarizes many of the effects of the modification. In the table, "Mean GOF" refers to $GOF_{translation}$, and Δx , Δy , and R are the x-component, y-component, and modulus, respectively, of the mean vector error in km. Table 2 lists the results for the original system (with the rule base prior to modification) and the modified system side-by-side for easy comparison. The top left portion of Table 2 lists some of the same information as Table 1, while the remainder of the table lists new information.

Table 2. Oceanographic Expert System Ring Motion Results Comparison, Modification 1
Gulfcast/DART Data Set

<u>Original Version</u>				<u>Modified Version</u>		
% better than no motion	<u>Days</u>	<u>WCR</u>	<u>CCR</u>	<u>Days</u>	<u>WCR</u>	<u>CCR</u>
	7	61%	64%	7	52%	79%
	14	55%	67%	14	45%	93%
Mean GOF	<u>Days</u>	<u>WCR</u>	<u>CCR</u>	<u>Days</u>	<u>WCR</u>	<u>CCR</u>
	7	1.03	.910	7	1.37	.777
	14	1.01	.891	14	1.34	.756
Mean vector errors	<u>Days</u>	<u>WCR</u>	<u>CCR</u>	<u>Days</u>	<u>WCR</u>	<u>CCR</u>
	7	Δx : -9.49	.161	7	Δx : -11.8	-3.84
		Δy : -6.60	4.22		Δy : -13.0	5.86
		R: 11.6	4.22		R: 17.6	7.01
	14	Δx : -15.9	2.48	14	Δx : -18.2	16.2
		Δy : -10.2	12.6		Δy : -24.5	9.67
		R: 18.9	12.8		R: 30.5	18.9

One interesting result from tests of the original version is that, for WCRs, mean GOF is greater than 1 (recall that in a single comparison, this implies that the expert system performs worse than the no-motion assumption) even though the percent of cases better than an assumption of no motion exceeds 50%. This obviously means that several comparisons had $GOF_{translation}$ significantly greater than 1, although that error measure was less than 1 for more than half of them. The most striking result from tests of the modified system is the marked improvement of the % better than no motion statistic for CCRs--yet this same statistic was noticeably worse for WCRs than in the original version. This puzzling situation also occurs for the mean GOF results. Most surprising of all, however, is the observation that the mean vector errors were larger for the modified system than for the original system for both WCRs and CCRs, for 7 and 14 days, with a significant trend over time. Since the correction was supposedly chosen specifically to reduce or eliminate the linear trend, and presumably also reduce the vector errors (particularly for 14 days), this is a very puzzling situation.

2. Modification 2

It appears that something is wrong. The error vectors listed above were taken from Lybanon (1990). Review of the calculations showed that the error vectors were inadvertently reversed: what was called the CCR error trend was actually the WCR error trend, and vice versa. The actual trends, with all results stated to three significant figures, are as follows. For WCRs, the trend is 1.05 km/day, heading -119°. For CCRs, the error trend is 1.24 km/day,

heading +15.50. The correction to the rules based on these error vectors is called Modification 2 (the correction based on the switched error vectors, that led to the results shown in Table 2, is called Modification 1).

Table 3 lists the results of tests of the expert system after Modification 2. As for Table 2, the left side contains the results for the original system, and the right side contains the results for the modified system.

Table 3. Oceanographic Expert System Ring Motion Results Comparison, Modification 2
Gulfcast/DART Data Set

<u>Original Version</u>				<u>Modified Version</u>			
% better than no motion	<u>Days</u>	<u>WCR</u>	<u>CCR</u>	<u>Days</u>	<u>WCR</u>	<u>CCR</u>	
	7	61%	64%	7	71%	59%	
	14	55%	67%	14	73%	60%	
Mean GOF	<u>Days</u>	<u>WCR</u>	<u>CCR</u>	<u>Days</u>	<u>WCR</u>	<u>CCR</u>	
	7	1.03	.901	7	.811	1.08	
	14	1.01	.891	14	.772	1.12	
Mean vector errors	<u>Days</u>	<u>WCR</u>	<u>CCR</u>	<u>Days</u>	<u>WCR</u>	<u>CCR</u>	
	7	Δx : -9.49	.161	7	Δx : -5.58	-2.08	
		Δy : -6.60	4.22		Δy : -3.37	.011	
		R: 11.6	4.22		R: 6.52	2.08	
	14	Δx : -15.9	2.48	14	Δx : -7.40	.701	
		Δy : -10.2	12.6		Δy : -4.09	11.5	
		R: 18.9	12.8		R: 8.46	11.5	

For the modified system, the striking improvement in the CCR statistics based on $GOF_{translation}$ that were achieved with Modification 1 are gone. In fact, the % better than no motion and mean GOF results for CCRs are slightly worse than for the original system. The corresponding WCR results are better than for either the original system or Modification 1. The vector error results are improved for both WCRs and CCRs (except for 14 days for CCRs), indicating that the vector error trends have been at least partially compensated. They were not completely eliminated, because no corrections were made to the Gulf Stream interaction rules.

3. Interpretation

Modification 1 produced the biggest improvement in the $GOF_{translation}$ statistics for CCRs. Modification 2 actually made these statistics worse for CCRs than the original version of the expert system. However, Modification 2 improved the results for WCRs. Therefore, a "hybrid" modification consisting of the Modification 1 changes for CCRs and the Modification 2 changes for WCRs would improve the expert system's performance as measured by these statistics.

Based on the vector error statistics, Modification 2 produced the best results for both WCRs and CCRs. The magnitudes of Δx and Δy were mostly smaller, and the value of R was always smaller, for both types of rings for both 7 and 14 days. The linear trend calculated from the WCR results has about one-fourth the magnitude of that for the original version, with about the same heading. The trend calculated for the CCR results, however, shows a trend of somewhat greater magnitude than for the original version, with about the same heading, indicating the likelihood of greater vector errors for longer time periods. As is always the case with fits to data, unpredictable results can be obtained outside the domain of the fit. Thus the possibility of big vector errors for CCRs for projections of more than 14 days would exist even if the trends based on the data were small. As pointed out, the fact that these trends were not taken out is presumably because of cases in which rings interacted with the Gulf Stream.

The cases which led to the residual errors were cases for which no modifications were made to the rules that "fired" and for which significant errors existed in the first place. The global correction to the rules was only intended as a first-order correction, and the cases which still have significant error indicate the need for second-order corrections. The most obvious second-order correction--although somewhat more difficult to apply than the correction made so far--is a similar global correction to the Gulf Stream interaction rules. While this may result in some further improvement, it is likely that a detailed analysis of individual cases will be needed to produce a significant improvement in accuracy. The failure of a particular rule to produce the right result in an individual case will indicate a need for improvement of that rule, but it is also necessary to study other cases in which the same rule is applied, to see whether a similar error results. This observation indicates that the analysis to determine needed improvements to individual rules needs to be quite sophisticated.

IV. CONCLUSIONS

Tests of the original version of NOARL's expert system were encouraging and also suggested some improvements. The tests showed that the ring motion rules perform better than an assumption of no motion for periods up to 2 weeks. Both ring position (i. e., motion-induced) and size errors tended to grow linearly with time. Mean ring size errors were only a few percent after several weeks, so it appeared more important improvements could be achieved by concentrating on ring motion.

Tests of the expert system modified in two different ways show that different (combinations of) changes produce improvement in different goodness-of-fit statistics. Also, some significant errors remain. We know how to achieve certain types of improvement in performance, but there is room for further improvement. Those improvements are likely to require notable effort and ingenuity.

The tests described in Lybanon (1990) and in this paper were performed using a limited set of ocean truth data. The quality of the data sets used in testing is a key issue. It is important to differentiate between differences between expert system projections and ocean truth data, and errors in the former. The differences may reflect errors in the ocean truth data as much or more than in the expert system results. Most of the results quoted are from tests made with what is believed to be a high-quality data set. Nonetheless, it is a limited data set. It is desirable to increase the size of the test data set, and simultaneously to ensure the quality of the new data to the extent possible.

There are thus two challenges in continuing the research described in this paper. A detailed, case-by-case examination to identify specific problem areas is indicated, and identification of other high-quality data sets to use in testing the expert system is a priority. Those challenges will help guide the work in the near future.

V. REFERENCES

- Krishnakumar, N., S. S. Iyengar, R. Holyer, and M. Lybanon (1990). Feature Labelling in Infrared Oceanographic Images. *Image and Vision Computing* 8(2):141-147.
- Thomason, M. G., and R. E. Blake (1986). *Development of an Expert System for Interpretation of Oceanographic Images*. Naval Ocean Research and Development Activity, Stennis Space Center, MS, NORDA Report 148.
- Halliwell, G. R., Jr., and C. N. K. Mooers (1979). The Space-Time Structure and Variability of the Shelf Water-Slope Water and Gulf Stream Surface Temperature Fronts and Associated Warm-Core Eddies. *Journal of Geophysical Research* 84(C5):7707-7725.
- Lybanon, M., R. L. Crout, C. H. Johnson, and P. Pistek (1990). Operational Altimeter-Derived Oceanographic Information: The NORDA GEOSAT Ocean Applications Program. *Journal of Atmospheric and Oceanic Technology* 7(3):357-376.
- Robinson, A. R., M. A. Spall, and N. Pinardi (1988). Gulf Stream Simulations and the Dynamics of Ring and

Meander Processes. *Journal of Geophysical Oceanography* 18:1811-1853.

Lybanon, M. (1990). *Oceanographic Expert System Validation Using GOAP Mesoscale Products and Gulfcast/DART Validation Test Data*. Naval Oceanographic and Atmospheric Research Laboratory, Stennis Space Center, MS, NOARL Report 5.

EVALUATION OF THE NAVY'S SEMI-AUTOMATED MESOSCALE ANALYSIS SYSTEM (SAMAS)

S.H. Peckinpaugh and R.J.Holyer
Ocean Sensing and Prediction Division
Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, MS 39529-5004

I. OBJECTIVE

The objective of this project is to automate the analysis and mesoscale interpretation of satellite infrared (IR) imagery of the Gulf Stream Region.

II. APPROACH

The approach is to utilize artificial intelligence and image processing technologies in a multi-level analysis scheme that starts with image segmentation and proceeds through high-level knowledge-based approaches to ocean dynamics, see Figure 1. Imagery from April 17-18, 1989 will be used to illustrate the approach.

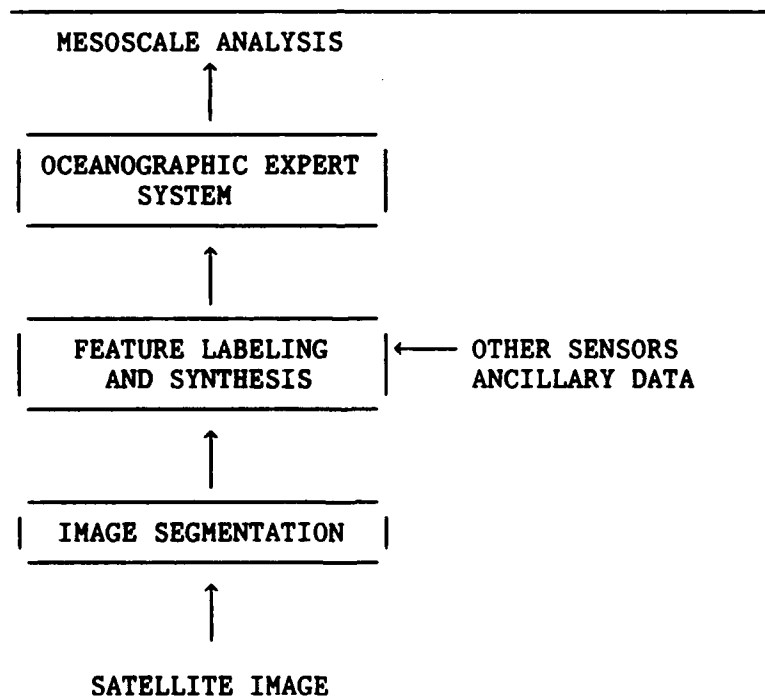


Figure 1. A three-tiered approach to automated oceanographic satellite image analysis.

A. Satellite Image

Three Advanced Very High Resolution Radiometer (AVHRR) IR Channel 4 images from the NOAA-11 satellite, one from April 17 and two from April 18, were each calibrated, earth located, and warped to a mercator projection, see Figures 2, 3 and 4. These images were then composited together to create the selected input satellite image, see Figure 5. For each pixel location within the composite image the warmest pixel was selected from the 3 images. This is done to reduce the clouded areas of the IR image, the clouds are known to be colder than the water in this area.

B. Image Segmentation

The low level image segmentation is accomplished by an edge detection operation. Here the Cluster Shade edge detection method was used. For computing the cluster shade values for the input April 17-18 warmest pixel composite image a window of 16 x 16 was used with delta values of 0. For the zero crossing test of the cluster shade values an initial threshold of 20 was used with these edges being extended for a threshold of 5. The edges were then cleaned using a 16 x 16 window and dilated once. The dilated edges were then thinned to produce the image shown in Figure 6.

C. Feature Labeling and Synthesis

The edge fragments are labeled as warm eddy, cold eddy, North Wall, and South Wall using relaxation techniques. Inputs to the labeling routine are the satellite image, the computed edges, and a previous analysis. For this image the previous analysis was the eddies from the previous image of the data set, April 14-15 combined with an analysis from the Navy's Operational Oceanography Center (OOC) for April 10-12 which includes eddies, North Wall, and South Wall. The eddies were all projected forward in time using a rule based Expert System. The labeled edges are shown in Figure 7.

A circular hough transform is applied to the edges which have been labeled as eddies. The hough transform finds the most prominent circular features represented by the eddy edges. This method is very effective even when parts of the eddy edges are missing. The edge points labeled as part of the North Wall of the Gulf Stream are used to create a continuous Gulf Stream by means of Complex Empirical Orthogonal Functions (CEOF). See Figure 8 for the automated analysis for April 17-18.

D. Oceanographic Expert System

A rule based expert system incorporates knowledge of the time evolution of mesoscale features to forecast the given analysis. This is done for two reasons. The first is to fill gaps caused by cloud cover in future analysis. The second is to provide a better start for the relaxation labeling routine for the next analysis. Figure 9 shows the analysis shown in Figure 8 projected in time 14 days.

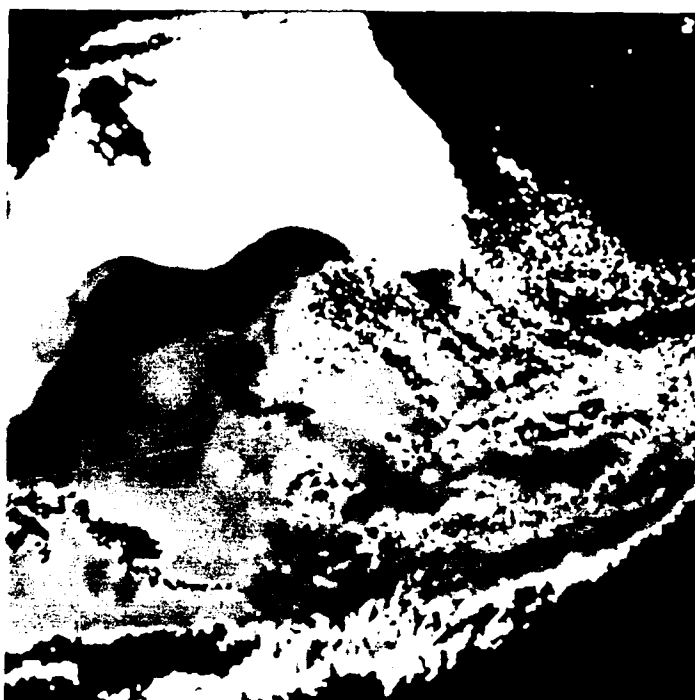


Figure 2. This is an April 17 calibrated AVHRR Channel 4 IR image from the NOAA-11 satellite, with a cloud mask created by thresholding Channel 2.



Figure 3. This is an April 18 calibrated AVHRR Channel 4 IR image from the NOAA-11 satellite, with a cloud mask created by thresholding Channel 2.

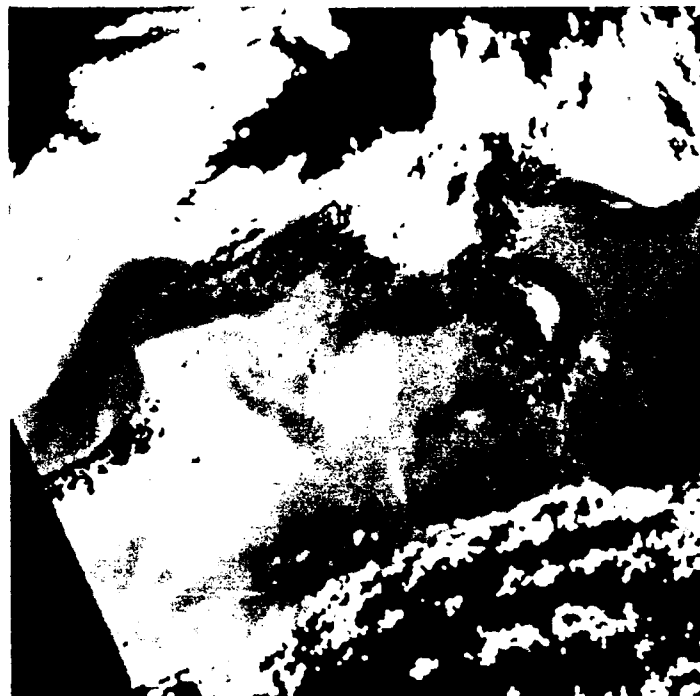


Figure 4. This is an April 18 calibrated AVHRR Channel 4 IR image from the NOAA-11 satellite, with a cloud mask created by thresholding Channel 2.

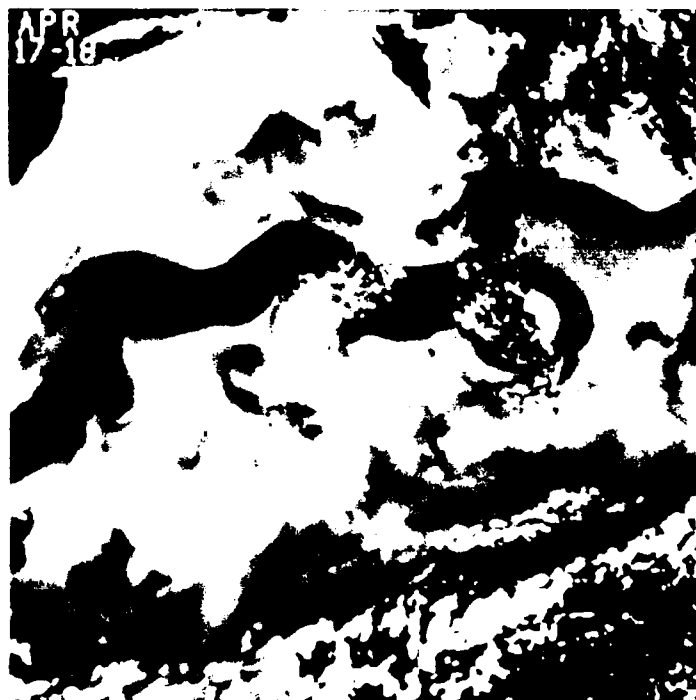


Figure 5. This is the warmest pixel composite image created from the April 17 and the two April 18 images shown in Figures 2, 3, and 4.



Figure 6. The image from Figure 5 with the computed edges overlain.

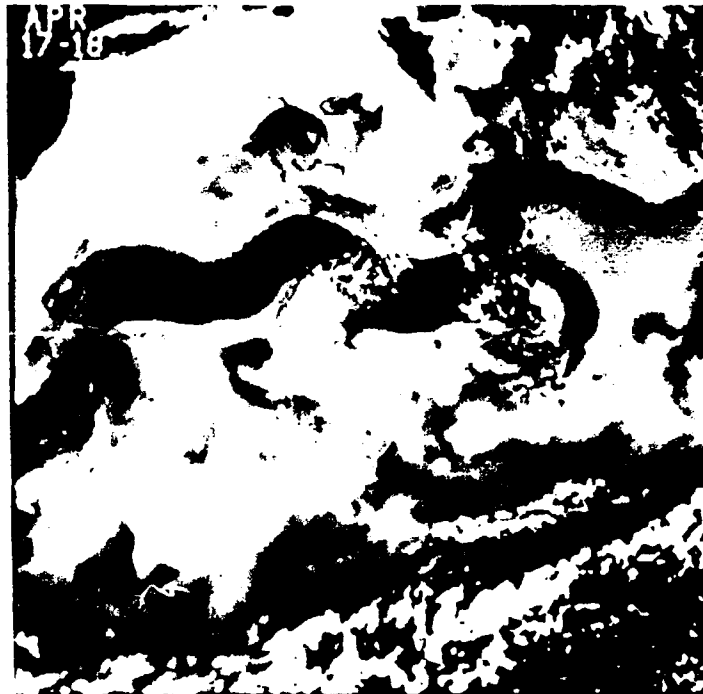


Figure 7. The image from Figure 5 with the relaxation labeled edges overlain.



Figure 8. The image from Figure 5 with the automated analysis of the eddies and North Wall overlain.

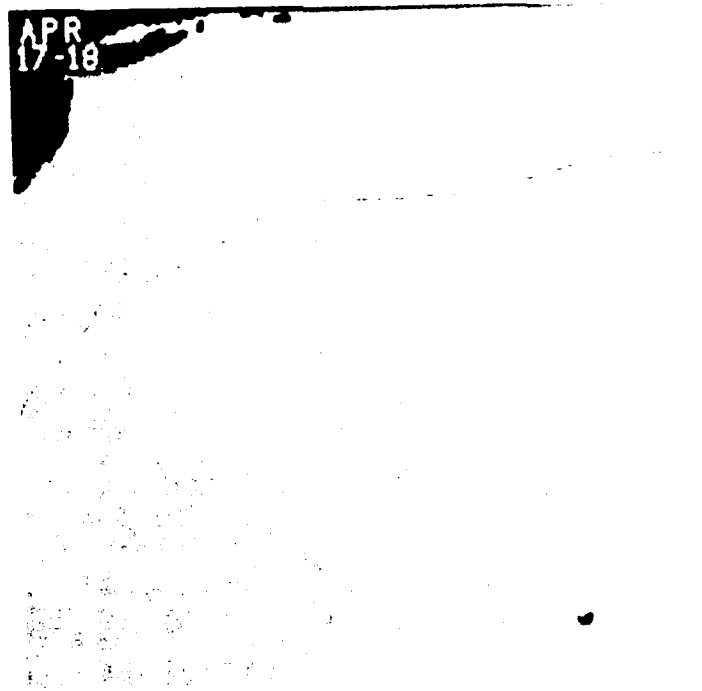


Figure 9. This figure shows the analysis from Figure 8 projected in time 14 days.

III. RESULTS

SAMAS has been applied to twelve test images from April and May of 1989. The test images shown below show the automated analysis (green) and the human interpretation by analyst from the OOC (red). Only the North Wall of the Gulf Stream and selected examples of a warm and cold core ring are shown, see Figures 10-21. A numerical comparison between SAMAS and human interpretation is shown in Table 1.

IV. ACKNOWLEDGMENTS

The authors would like to thank Bobby Grant of Sverdrup Technology for his work in data processing and image photo preparation for this project.

V. REFERENCES

This presentation is taken from a poster by the same name presented at The Fifth Conference on Satellite Meteorology and Oceanography; London, England; September 1990. For more information and references concerning specific modules of the SAMAS system see the conference proceedings, American Meteorological Society; 45 Beacon Street, Boston, USA 02108.

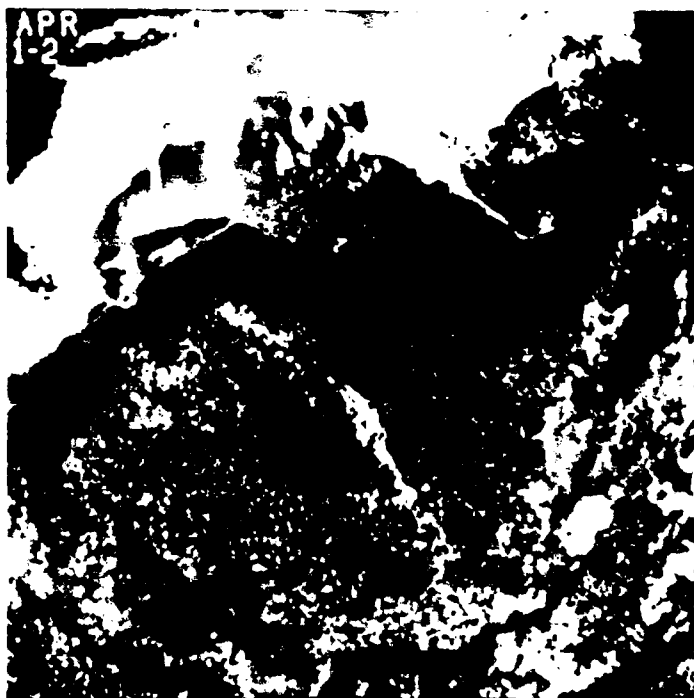


Figure 10. The background image is a warmest pixel composite for April 1-2, 1989, overlaid with the automated analysis for the image.

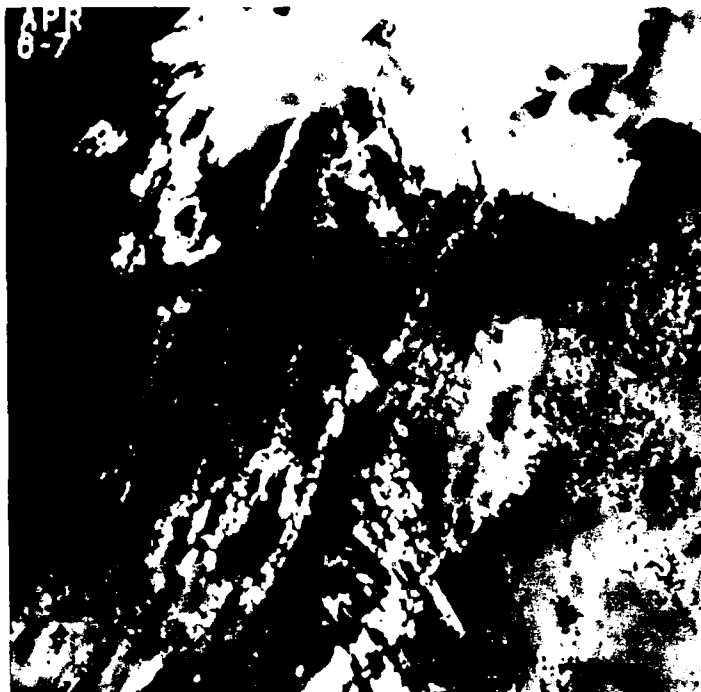


Figure 11. The background image is a warmest pixel composite for April 6-7, 1989, overlaid with the automated analysis for the image.



Figure 12. The background image is a warmest pixel composite for April 10-12, 1989, overlaid with the automated analysis for the image.

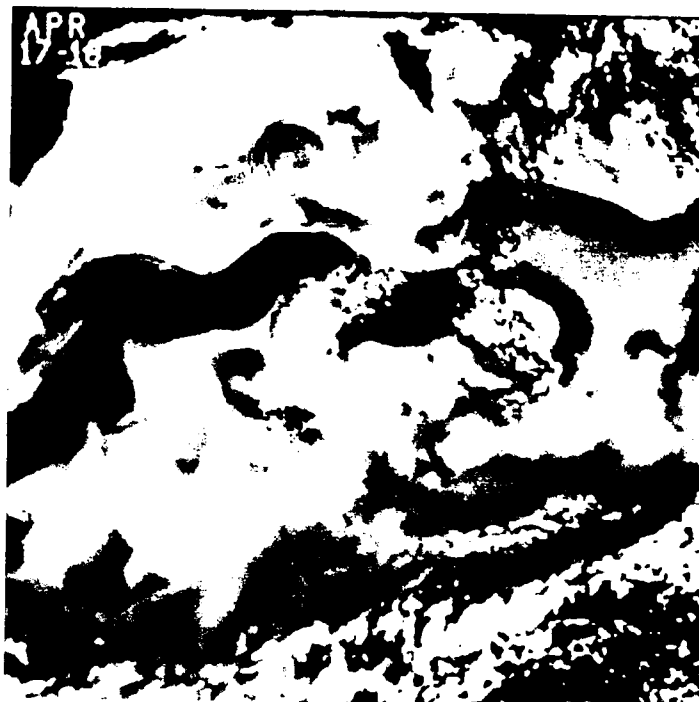


Figure 13. The background image is a warmest pixel composite for April 17-18, 1989, overlaid with the automated analysis for the image.



Figure 14. The background image is a warmest pixel composite for April 21-23, 1989, overlaid with the automated analysis for the image.



Figure 15. The background image is a warmest pixel composite for April 28, 1989, overlaid with the automated analysis for the image.

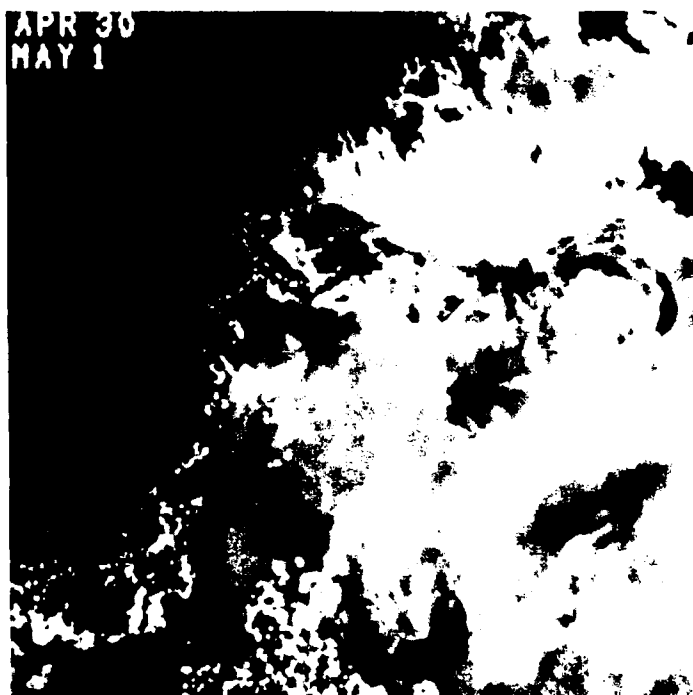


Figure 16. The background image is a warmest pixel composite for April 30-May 1, 1989, overlaid with the automated analysis for the image.



Figure 17. The background image is a warmest pixel composite for May 3-4, 1989, overlaid with the automated analysis for the image.



Figure 18. The background image is a warmest pixel composite for May 8-10, 1989, overlaid with the automated analysis for the image.



Figure 19. The background image is a warmest pixel composite for May 17-19, 1989, overlaid with the automated analysis for the image.



Figure 20. The background image is a warmest pixel composite for May 25-26, 1989, overlaid with the automated analysis for the image.



Figure 21. The background image is a warmest pixel composite for May 30-June 1, 1989, overlaid with the automated analysis for the image.

	WARM	COLD
Mean Scalar Positional Difference for Eddies	27.2 km	35.0 km
Mean Vector Positional Difference for Eddies	13.0 km	15.8 km
Eddy Size Bias	+30%	+04%
Probability of Eddy Detection	0.91	0.80
Mean Scalar Positional Difference for Gulf Stream	33.8	

Table 1. Numerical comparison between SAMAS and human interpretation.

OSIRRUS: OCEANIC SYMBOLIC IMAGE REPRESENTATION, RECOGNITION AND UNDERSTANDING SOFTWARE

Lee A. Atkinson
Consultant's Choice, Inc.
Atlanta, Georgia 30350

ABSTRACT

Symbolic contours representing isotherms are used for shape and feature analysis of infrared oceanic satellite imagery. Techniques for detecting and analyzing oceanic thermal structures are presented. An effective image-processing technique called Image Modulation expedites time-consuming symbolic processing. Eddy-rings are successfully shape-recognized and false alarms are rule-discriminated. Verified eddy-rings are provided with a feature suite analysis useful for an automated image analysis workstation. Potential applications include Gulf Stream detection, symbolic image classification, image repair, and signal projection.

I. INTRODUCTION

Detection and proper labeling of oceanic mesoscale features in remotely sensed data is a mission of NOARL's ocean science directorate. Automated identification of eddy-rings and Gulf Stream walls (two oceanic mesoscale features of specific interest) in infrared satellite imagery using isotherm shape identification is a viable approach. Presented here are the results of an initial effort pursued by Consultant's Choice, Inc. (CCI) concentrating on eddy-ring detection using shape identification.

Oceanic Symbolic Image Representation, Recognition and Understanding Software (OSIRRUS) converts imagery to a "symbolic form" in the sense that data is converted to a list of contiguous pixel points representing a contour of constant temperature on a thermal topology, which is an isotherm. Ideally, isotherms form characteristic shapes that envelop mesoscale features, such as eddy-rings. Shapes identifying potential areas of interest are analyzed with an additional temperature profile to confirm eddy-ring detection and provide automatically extracted measurements useful to a mission specialist.

Since isotherms are acquired easily for all thermal values in the image domain, OSIRRUS has the advantage of avoiding scale-dependent problems usually associated with feature-edge detection. Moreover, isotherms ensure unique labeling of edges (mixed labeling does not occur). OSIRRUS does not attempt to detect and label under one process. The technique of first performing shape extraction and then specific shape detection is more natural in the understanding of information presented, and leads to robust heuristics of detection and analysis. Also, since the techniques are separated, OSIRRUS has the capacity to detect and analyze a wider variety of thermal structures than using a finely-tuned specific detector. Even though shape recognition is of main interest, the isotherm representation still retains thermal information so that intelligent decisions using known physical and oceanographic properties may be made.

Isotherm extraction and identification involves processing of thousands of lists, each with varied lengths of possibly several hundred elements. Although difficult to manage in some programming environments, lists may be combined with data of varied form and type to form complex and powerful data structures. The development of OSIRRUS was performed on an IIM-Inferstar Lisp machine capable of managing complex data structures in an interpretive environment with dynamic memory allocation and automatic variable typing. This platform allowed rapid prototyping of OSIRRUS and, therefore, rapid development of eddy-ring shape identification. However, the algorithms that have been developed are not LISP nor LISP-machine specific; may be ported to the C language or assembly; and will operate on most microcomputers.

II. GENERAL METHOD

Described herein are the general steps (see Table I) for preprocessing, symbolic transformation, eddy-ring detection, and detection analysis performed by OSIRRUS. Improvements concerning speed and quality of identification are described in section IV.

TABLE I
GENERAL METHOD

STEP	OPERATIONS	TYPE	RESULT
1. PREPROCESSING	FILTER	SPATIAL	MEDIAN 5x5
	FILTER	SPATIAL	GAUSSIAN 5x5
2. IMAGE-TO-LIST XFORM	THRESHOLD	DOWN	BINARY
	EDGE	4 CONN	ISOTHERM IMAGE
	CONTOURIZE	RASTER SEARCH	ISOTHERM LIST
	FILTER	RULES	ISOTHERM LIST
	RECONNECT	SEARCH	ISOTHERM LIST
	FILTER	RULES	ISOTHERM LIST
3. DETECT-EDDYS	BOTTLENECK	RULES	BOTTLES
	RING FILTER	RULES	RINGS
4. DETECTION-ANALYSIS	GROUPING	CALC	FEATURE OBJECTS
	FEATURE ASSIGN	CALC	FEATURE OBJECTS

A. Preprocessing

OSIRRUS typically processes 256x256 pixel images with infrared intensity 0 to 255, representing coldest to warmest thermal signature, respectively. The original image is processed using a median 5x5 filter to eliminate shot noise and ensure local continuity. This filter is best since a median 3x3 does not filter typical images enough and a 7x7 does not gain much beyond a 5x5. It was found that, without this filter, many pygmy isotherms are created and large (long) isotherms are contaminated with spurious loops that do not yield general information about the thermal topology. To assist the median 5x5 filter, a Gaussian (binomial 5x5) filter evens out local small-scale features and digitization noise. While these two filters may lose information and gradient detail, feature shapes become much more apparent for recognition, and granularity of information is reduced.

B. Image-to-List Transformation

OSIRRUS transforms the preprocessed image into lists of isotherms by intensity levels. A given level (L) is processed to obtain all usable isotherms, and then the process is repeated for each intensity level. Isotherms from a given level or temperature are grouped together into a temperature labeled list.

Assurance of unbroken isotherms is made by thresholding intensity values (T) to obtain a binary image (B) using the following rule:

$$\begin{array}{lll}
 T(x,y) > L & \rightarrow & B(x,y) = 0 \\
 T(x,y) \leq L & \rightarrow & B(x,y) = 1
 \end{array} \tag{1}$$

Thresholding thus interpolates for areas where large gradients skip over intermediate intensity values. Isotherms are that set of pixels with intensity values not greater than L , which neighbor pixels with intensity levels greater than L (see Figure 1).

Isotherms are formed from the resulting binary image $B(x,y)$ by an edge operator. Edge pixels are only those pixels with a four-connected zero-valued neighbor. The simplest edge operator is the best! At this point, viewing the edge image results in display of image isotherms, revealing the shape of the thermal topography at the processed level (see Figure 2). The edge image is converted to symbolic format which is a list of lists of pixel coordinates, as illustrated in the LISP list below:

```
(( (Xa Ya) (Xb Yb)...(Xn Yn) )
  ( (Xc Yc) (Xd Yd)...(Xm Ym) )
  .
  .
  .
  ( (Xe Ye) (Xf Yf)...(Xp Yp) ) ).
```

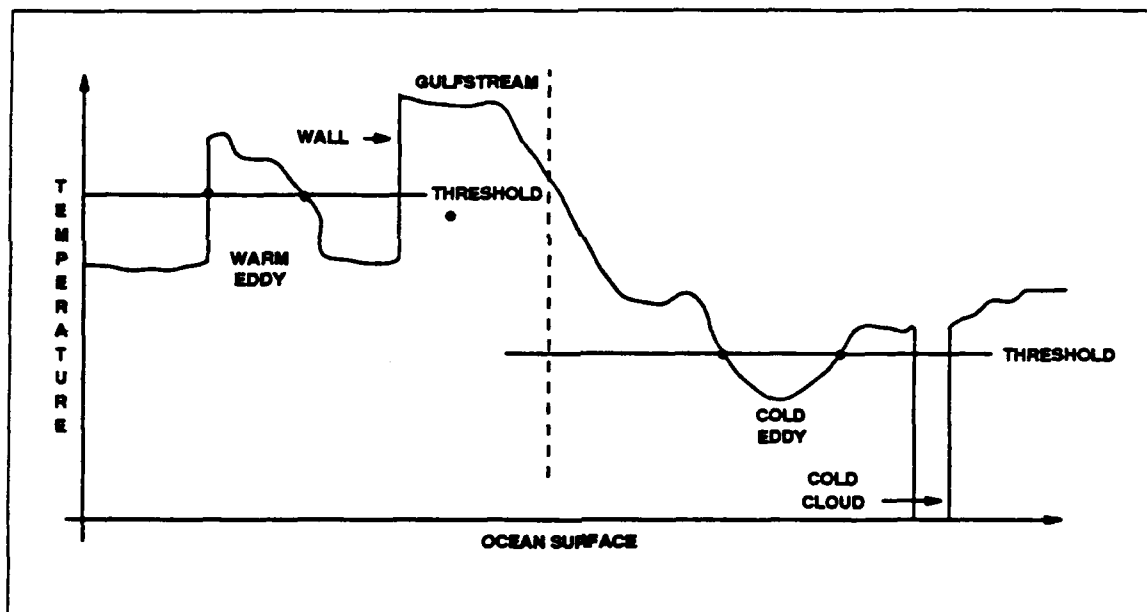


Figure 1. Thresholding for Contours (Isotherms)

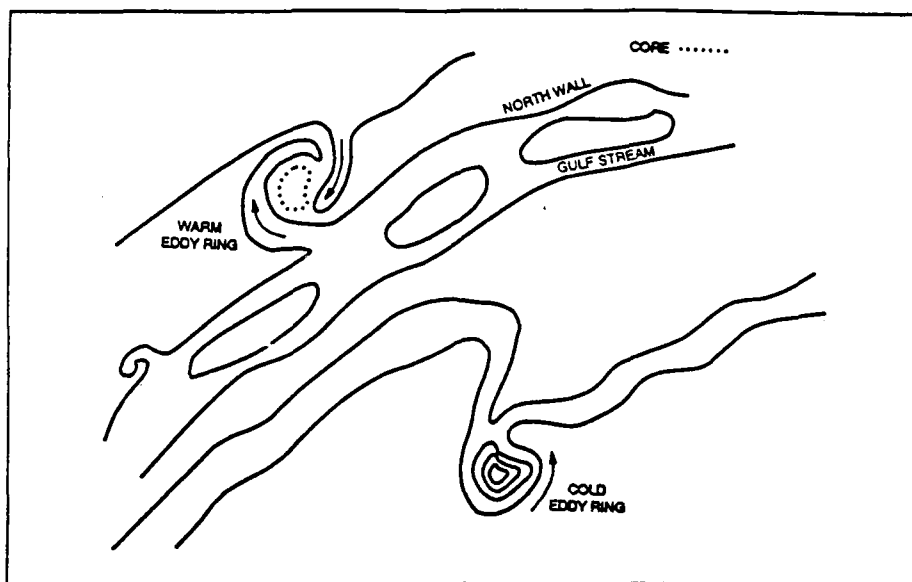


Figure 2. Contours (Isotherms)

Transformation to symbolic format is accomplished by performing a raster search to find at least one point on an edge contour (image isotherm). From this one point, a typical contour-following algorithm collects contiguous pixels into a list and eliminates or marks their extraction from the edge image. The raster search continues until all pixels are extracted.

The resulting collected lists contain numerous short isotherms such as those formed about cloud noise and "loose ends" left from the raster-scanning transformation process. Currently, OSIRRUS uses a rule that eliminates isotherms shorter than four points in length.

Clouds often leave streaks through otherwise uncontaminated thermal signature, breaking isotherms, and forming false termini of isotherms. A reconnection process ties isotherm termini together within a local neighborhood (without interpolating). The search process is on lists rather than contour following on the image, and is, therefore, an N^2 process where N equals number of isotherm termini. For a large number of termini, search becomes a compute intensive process. However, reconnection is not compute intensive since the original contour-following algorithm uses the same neighborhood to recursively trail and join isotherm termini. Reconnection, however, ensures unbroken isotherms and thus preserves whole shapes.

Finally, all collected isotherms are filtered a second time for length greater than or equal to 18 points. Generally, short isotherms will not embody enough shape to be recognized by the remaining detection scheme. Should additional algorithms be developed, groups of short isotherms (with adjacency) may be useful to detect partially occluded features. However, for the present time, this rule serves as a dividing line and data reduction limit.

C. Eddy-Shape Detection

Once isotherms are acquired in a symbolic format, shape recognition may begin. Eddy-rings apparently have two major isotherm shape structures. The first of these structures is due to a turbulent swirling and mixing of cold and warmer waters. In such a case, isotherms tend to form a crested spiral or hook shape. During the development of OSIRRUS, a hook shape detector was developed and tested. However, simple schemes of hook and spiral detection failed to produce robust results. The reason for difficulty is probably that there are many subclasses of hook and spiral shapes. For example, warm eddy-rings near the Gulf Stream pull colder slope water around from one side while pulling warmer water from near the north wall to the other side (see Figure 2). The shifting of waters in such fashion often creates apparent irregularities in the sensed gradient field.

The second major structure occurs when eddy-rings no longer pull neighboring waters into shallow spirals. In this case, the core appears homogeneous and thermal gradients converge (or diverge) evenly from the ring center. One expects that isotherms would form concentric contours about the eddy-ring. In general, the second structure is typified by concentric open-ended bottles containing the eddy-ring (see Figure 3). Currently, OSIRRUS uses only a bottle shape detector to filter isotherms with this type of major structure.

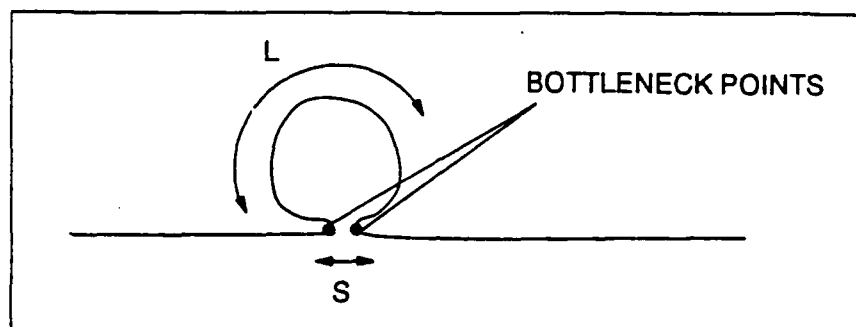


Figure 3. Bottleneck Filter

Currently, OSIRRUS employs an algorithm similar to a hill-climbing procedure which is faster than N^2 search. The algorithm finds the closest points (with euclidean separation S) that have at least a minimum arc length, L , between them. The two closest points generally mark the "neck" of the bottle or isotherm structure. Points between these bottleneck points are retained if local curvature is less than some threshold, $S/L < C_{Th}$. Thresholds governing the hill-climbing search serve as rules for detection.

Eddy-rings are not always blessed with circular or ellipsoidal symmetry. Another filter involved in detection eliminates "bottles" that are considered extremely eccentric. The "ring filter" rule decides if a bottle-shaped isotherm is a possible component of an eddy-ring:

$$(R_{max} - R_{min})/RMS < \epsilon \quad (2)$$

where R_{max} (R_{min}) is the maximum (minimum) distance from the centroid of the bottle and RMS is the root-mean-square of all radii from the centroid of the bottle. Epsilon is proportional to the maximum allowable eccentricity of the bottle. A summary of detection rules is in Table II.

TABLE II
CURRENT DETECTION RULES

ISOTHERM FILTER	- MINIMUM ISOTHERM LENGTH (1ST) = 4 - MINIMUM ISOTHERM LENGTH (2ND) = 18
BOTTLENECK FILTER	- MAXIMUM BOTTLE SIZE = 80 - MAXIMUM NECKSIZE = 16 - MINIMUM BOTTLE LENGTH = 32 - MAXIMUM BOTTLE LENGTH = 256
RING FILTER	- $(R_{max}-R_{min})/RMS < .75$
GROUPING	- $\bar{R}_{GROUP} < 16$

D. Detection-Analysis

Once bottles have been detected at all intensity (thermal) levels, OSIRRUS performs a grouping of concentric isotherms to strengthen or rank detection. Observations have indicated that centroids of isotherms about an eddy-ring are not co-located. Therefore, bottles are grouped if within a neighborhood of radius = 16. Each group is then considered as one detection for analysis.

Analysis serves to label features of each detection that might: (a) help eliminate false detections; and (b) be of interest to an analyst reviewing the automatic interpretation. The feature suite is listed in Table III. Research currently involves using a multi-discriminant analysis program called GOPAD to determine the manner in which features can be combined to form identification rules.¹ GOPAD will process features of each detection together with ground truth (i.e., known eddy-rings). The result will be a statistical model from a database of over 100 images that will provide rules to distinguish actual eddy-rings from false alarms.

TABLE III
FEATURE ASSIGNMENT FOR EACH
EDDY-RING DETECTION
(ISOTHERM GROUP)

● CENTROID X, Y
● MAXIMUM ISOTHERM LENGTH
● MINIMUM ISOTHERM LENGTH
● MAXIMUM EPSILON (OVALNESS)
● MINIMUM EPSILON (OVALNESS)
● MAXIMUM RADIUS
● MINIMUM RADIUS
● NUMBER OF ISOTHERMS IN GROUP
● THERMAL PROFILE (56 PARAMETERS)

III. DETECTION RESULTS

Table IV shows results of detection prior to analysis (after grouping). The new method refers to results after optimizations and fine-tuning of rules, thresholds, etc., had been implemented. Four images are warmest-pixel composite, (designated with an A---) and six are single images (designated with an M---). When comparing results, there are indications that combined results over a few days would be better than processing composite images where isotherms apparently broaden and smear.

¹ Goal Oriented Pattern Detection, ThinkNet, Inc.

TABLE IV
CURRENT RESULTS SHOWING BEFORE AND AFTER OPTIMIZATION
AND IMPROVEMENT OF ALGORITHMS

IMAGE NAME	BEFORE					AFTER				
	A	D	F	M	E	A	D	F	M	E
A1T2	2	5	3	0		2	17	15	0	
A6Y7	1	5	4	0		1	11	10	0	
A17T18	2	5	4	1		2	7	5	0	
A21T23	1	6	6	1	clouds	1	15	15	1	
M09	3	4	3	2	clouds	3	18	17	2	clouds
M10B	4	6	4	2		4	15	11	0	
M10D	4	6	3	1		4	11	7	0	
M11A	4	7	5	2	clouds	4	11	8	1	clouds
M11C	3	5	2	0		3	11	8	4	
M12	1	8	7	0		1	13	12	0	
	25	57	41	9		25	129	108	4	

HITS= D-F=16
 POD= HITS/ACTUAL=64%
 FAR= F/D=72%

HITS= D-F=21
 POD= HITS/ACTUAL=84%
 FAR=F/D=84%

LEGEND

A - ACTUAL
 D - DETECTED
 F - FALSE ALARM
 M - MISS
 E - POSSIBLE REASON FOR MISSES
 POD - PROBABILITY OF DETECTION
 FAR - FALSE ALARM RATE

These results have high false alarm rates since, to date, rules and methods have not been employed to eliminate detections due to noise. Many detections are too small to be eddy-rings or have inconsistent thermal signature. The continuing effort to find elimination rules from a database of 100 images should bring false alarm rates down significantly.

IV. IMPROVEMENTS

Processing each intensity level to acquire isotherms is very time consuming. Yet, peculiar eddy-rings may have domains over most of the 256 intensity levels. In order not to skip over possible detections and at the same time expedite processing, techniques were devised to process all intensity levels while clarifying information content that led to two major improvements in the speed of obtaining isotherms and improving isotherm representation for better eddy-ring detection.

A. Modulation

The first technique may improve use of symbolic contours for many applications other than eddy-ring detection. The process of "Image Modulation" for isotherm contour processing is based upon the following premise: "If the image topology is smooth and there are very few strong gradients over a base distance, B, then isotherms at a thresholding of t_1 are spatially separated from isotherms at a thresholding of $t_2 = t_1 + B$ at distances most like B." Thus, if B is large enough, the two isotherms at t_1 and at t_2 are almost always spatially separated.

This separation allows the processing of more than one thresholding of an image at once without the usual problems of adjacent and overlapping contours (isotherms). A pixel of intensity $I(x,y)$ becomes:

$$I(X,Y) = \text{MOD}_B [I(X,Y)] + 1 \quad (3)$$

where B is the base. After this modulation has been accomplished, B levels may be processed instead of 256! In order to ensure that the image is smooth, a specialized binomial filter is applied (see section B below) before the modulation processing is completed.

Prior to implementing the image modulation scheme, the time to process ten levels of an image was 20 to 40 minutes. Thus, the expected time to process all 256 levels was about 8 to 17 hours. After implementation, images have been processed in full range (256 levels - 12 modulated levels) within an hour. Using this powerful approach, eddy-rings are detected (both warm and cold rings), regardless of their internal or background temperature. Combined with other optimizations (not discussed herein), OSIRRUS currently can process 48 modulated levels in 15 minutes.

B. Special Gaussian

The second technique is required for the image modulation and further reduces noise and data representation. The method begins with a Gaussian filter (binomial 5×5) and normalizes the sum of weights corresponding to non-zero pixels. Using only non-zero pixels prevents cloud contaminated areas from becoming too smooth (like eddy-rings) or blending with neighboring features. Isotherm representation is benefited because isotherms are smoother and represented with fewer points. Finally, isotherms are well separated before symbolic transformation takes place, resulting in less ambiguous shapes.

V. CONCLUSION

Continued research to create and test new shape detections could bring probability of detection to near 100% for eddy-rings in the data provided. False detections will be eliminated via rules currently being determined using over 100 (256x256) images. Current results indicate that OSIRRUS is more likely to detect cold eddy-rings than warm ones. This capability complements analysts who are prone to detect warm eddy-rings more often than cold ones.

Symbolic isotherm shape recognition may be extended to locate features within the Gulf Stream as an assist to temporal interpolation schemes. Thermal features currently not explored by NOARL can be extracted if characterized by shape, size, and thermal profile. Furthermore, CCI suggests that isotherm shape recognition has potential in the areas of image classification and image repair.

VI. ACKNOWLEDGMENTS

This project was funded under contract to NOARL, Contract No. N00014-89-C-6027. The author wishes to express thanks to Matthew Lybanon for continued interest and support of OSIRRUS. Acknowledgments also go to Bennett Teates and Paul Lampru for acquisition and guidance, and to Kathie Speas for the manuscript and graphics.

REFERENCES

- Goel, Ashok and Tom Bylander, "Computational Feasibility of Structured Matching", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. II, Dec. 89, pp. 1312-1316.
- Gonzalez, R. C., and P. Wintz, *Digital Image Processing*, Addison-Wesley, 1987, 2nd Edition.
- Hoyer, R. J., and S. H. Peckinpaugh, "Edge Detection Applied to Satellite Imagery of the Oceans", *IEEE Trans. Geo. Remote Sensing*, Vol. 27, No. 1, Jan. 89.
- Young, Kenneth C. and Robert L. Gall, "A Streamflow Forecast Model for Central Arizona", *Mon. Wea. Rev.*, to be published.

WATER-MASS ANALYSIS FOR DIGITAL IMAGERY

J. Blaha and D. L'Heureux
U. S. Naval Oceanographic Office
Stennis Space Center, MS 39522-5001

I. ABSTRACT

Oceanographers have often used the concept of "water masses" to show how water properties are distributed geographically. We use water masses here to automate the detection of ocean fronts, which are the boundaries between water masses. Water masses that have strong surface signatures, for example in temperature, are visible in satellite images. Four water masses in the western North Atlantic are used to locate the major fronts. Shelf water, slope water, Gulf Stream warm core water, and Sargasso water appear as distinct thermal features in satellite digital images. Texture analysis, clustering, and maximum likelihood classification are used to locate cloud and water-mass segments on a standard grid. The standard grid allows the user to build separate, co-registered levels of image interpretation and to link the results of image processing to data bases. We show a composite of several images classified under this methodology, and suggest how such a composite can be combined with multiple climatologies to produce grids of water properties.

II. INTRODUCTION

Satellites scan the ocean surface globally, at high spatial resolution and on a continual basis. This advancement over traditional ocean measurements has led to new products that display the continual change of ocean water properties. Temperatures from the Advanced Very High Resolution Radiometer (AVHRR), for example, are used successfully to monitor the lateral movement of ocean fronts and eddies. Presenting satellite data in an image format is central to this application and a major reason for its success. By nature, we tend to interpret pictures more easily than other forms of information. Especially when specific features are sought, large volumes of data are readily processed using images. Additional automation will be needed to process the projected enormous increases of satellite data. The image format may be a valuable and necessary device toward achieving this goal.

The experienced data analyst brings knowledge about ocean features to image processing. This knowledge generates expectations of how ocean features should look. Striking thermal contrasts in AVHRR IR images indicate the locations of

ocean fronts and eddies, but equally important is the framework of regional thermal patterns in which the fronts and eddies reside. These larger patterns recur from image to image, and the analyst uses knowledge of them to interpret the many highly variable and smaller features. Oceanographers know these regional patterns as the surface manifestations of ocean water masses. The boundaries between these water masses are the ocean fronts. The eddies are segments of one water mass embedded in another. In the western North Atlantic, four water masses are sufficient to locate the major fronts: a) cold continental shelf waters near the east coasts of the U.S. and Canada, b) slope water between the shelf and Gulf Stream, c) warm core water of the Gulf Stream, and d) Sargasso water seaward of the Gulf Stream. We will show that we can devise computer algorithms around the concept of these water masses to emulate manual processing of digital satellite data.

Water masses have characteristics that expand the usefulness of image processing in practical ways. The climatological properties of water masses tend to be less variable, both at the surface and subsurface, than those of the region as a whole. This means that climatologies of water masses approximate synoptic conditions, as might exist in an individual image, better than ordinary climatologies that do not distinguish between water masses. Therefore, water-mass climatologies are inherently better suited to products which intend to depict real conditions. Moreover, water masses appear in different kinds of data. Researchers have identified the same water-mass segments, for example, in images of sea surface temperature and ocean color. Products based on composites of water-mass locations can take advantage of a combination of sensors that have different strengths.

We describe a methodology to identify cloud and water masses in digital satellite data using common techniques of pattern recognition. We then show an example of water-mass classification. Finally, we discuss briefly an application of classified images and an important aspect for future work.

III. METHODOLOGY

The software, called Water-Mass Analysis for Digital Imagery (WADI), is divided into five processing functions that: 1) load pixels of multi-channel sea surface temperature (MCSST) onto a geo-referenced grid, 2) mask clouds, 3) classify water masses, 4) update statistical data bases of SST, and 5) build composites of classified images. The software is modular; each function may be executed independently, although the intention is to achieve an automated "batch" runstream as suggested in

Fig. 1. Currently, WADI is limited to AVHRR IR data and the western North Atlantic, but it can be configured to handle different kinds of scanner data from different regions.

A. Load Data Onto A Standard Grid

The user must initialize a standard grid by specifying the origin, dimensions, and resolution. This configuration remains fixed throughout the processing and is the default for follow-on images. A landmask generated from the World Data Base II coastline is mapped to the grid as an assigned value of "2". Local Area Coverage (LAC) data are loaded on this grid using the 51 earth location points provided in the level 1b data record. Our choice of projection for the grid is Lambert conformal conic.

Sea surface temperature is computed using a multi-channel algorithm (see, McClain et al., 1985). Pixel brightness is scaled to capture the temperature range -1°C to 34°C , with a resolution of $.14^{\circ}\text{C}$ per brightness level. This range is suitable for the western North Atlantic, but it can be changed by the user as desired. Pixels are adjusted for the atmospheric attenuation that darkens the margins of the scan. The satellite data are subsampled if the chosen mesh size of the grid is larger than the pixel size of the scanner.

Pixels of temperatures less than -1°C are taken to signify cloud and are assigned a "1". This simple determination almost certainly underestimates the amount of cloud in the image. It provides, however, a useful initial mask for a sequence of more complex evaluations that follow.

B. Locate All Cloud

The technique of comparing pixel temperature to a reference temperature, and then thresholding the difference to detect cloud, can be refined using climatologies of sea surface temperature. These climatologies are grids of monthly means and variances. They are the same data bases used during water-mass classification and are described later in that section. Using the means and variances, we compute statistical tests on individual pixel temperatures. If a pixel is significantly cooler than the temperature of the coolest water mass that could occupy the pixel location, the pixel is labeled "cloud". This scheme reduces the tendency to label cold water falsely as cloud.

Generally, thresholding tends to underestimate cloud because thin cloud is not sufficiently cold to be detected. Some of the thin cloud exists at the periphery of very cold cloud that is already masked. Therefore, we usually expand the cloud mask one or more pixels, depending upon the mesh size.

Two properties of cloud are used to maximize the detection of thin cloud. First, cloud appears in clusters. Cloud pixels tend to be contiguous or in close proximity, so thin cloud is likely to appear near cloud that is already labeled. Second, cloud temperatures, within a small neighborhood of pixels, tend to vary greatly in all directions. By contrast, ocean fronts orient the temperature variability in particular directions. There is large change across the front and small change along the front. This contrast is readily apparent in images of temperature differences: the texture of cloud is grainy, whereas the texture in the vicinity of ocean fronts is dominated by lines.

To represent image texture, we create a grid of the minimum temperature differences. These differences are computed in all directions allowed in a 5-pixel x 3-pixel window. The minimum difference is selected from among these directions. Minimum differences are low for fronts, but are high for clouds. It is relatively easy to find a threshold that satisfactorily separates these categories. Pixels centered in windows of large minimum differences are candidates for cloud. If these candidates are within roughly 5 km of the cloud mask or are part of a contiguous sequence of candidates, they are labeled cloud.

These techniques do not accurately detect heavy fog and other low lying cloud of smooth appearance over cold waters in the northern regions.

C. Locate Water Masses

The cloud-free portion of the image appears as a collection of adjoining patches of homogeneous temperature. The patches are separated by ocean fronts or surrounded by cloud. We refer to the patches as "segments" of water masses. We use cluster techniques to define the spatial bounds of a segment and maximum likelihood techniques to classify a segment as one of the four water masses -- shelf water, slope water, Gulf Stream warm core, or Sargasso water.

A cluster is simply a temperature band. If the pixels of a cluster are highly fragmented, or in some other way show a high perimeter to area ratio on the image, the dispersion of that cluster is high. Clusters that have higher dispersion are added to adjacent clusters that have lower dispersion. The reduction in number of clusters is accompanied by increases in band-width

of the lower dispersion clusters. The process is iterated until the band-width of one cluster exceeds a user-specified estimate of the temperature variability within a single water mass.

Segments are constructed from the remaining clusters at the end of the consolidation process. A segment is built by finding contiguous pixels of the same cluster and assigning a unique label to these pixels. Generally, several hundred segments are found in an image that is mostly cloud-free.

The segments are located but not yet classified as parts of water masses. We begin by computing the probabilities of temperatures exhibited by the water masses. This computation requires the frequency distribution, or probability density function, of surface temperatures within each water mass (see Coulter, 1983). We assume that the frequency distributions are normal, so that we need only supply the mean and variance of temperature for each water mass to compute probabilities. Each mean or variance represents the average over a 50 km x 50 km block, to allow for the geographical variations of water masses and temperature. The blocks are co-registered to the standard grid. Collectively, the means and variances make up a data base of files grouped according to water mass.

We build "look-up" tables relating pixel temperatures to one of the four water masses. One table must be built for each 50 km x 50 km block because the means and variances change. Using the data base, we compute the probabilities of a pixel temperature occurring in each of the water masses. The greatest of these probabilities indicates the water mass that is most likely to correspond to that temperature. The maximum probability must exceed a chi-square threshold; otherwise, the temperature is assigned the "unknown" classification. The full temperature range is classified in this way for each block. Following the construction of the tables, each non-cloud pixel on the image is assigned a water-mass label.

We transfer a water-mass label to a cluster segment by separately tallying the number of pixels of each water mass in the segment. The label with the largest tally is assigned to all pixels of the segment. Each segment on the image is classified in this way.

D. Update Data Bases

The classified image shows the correspondence between water mass and pixel temperature. Using this correspondence, the means and variances of temperature are computed for each water mass. The computation is made separately in each 50 km x 50 km block.

This new information is entered into both an accumulated statistics file, which keeps a running average of temperature on a monthly basis, and a recent statistics file, which keeps only statistics from the most recent image. Separate files exist for each water mass. The files for Sargasso water are updated with the statistics of the pixels that are classified as Sargasso water, and so on. There are also historical statistics files of long-term averages, which are not updated. The user has the option of employing either recent statistics, accumulated statistics, or historical statistics to classify the next image.

Fig. 2 shows an example of a statistics file as a graphic overlay. Nonzero values specify the mean temperature of the Gulf Stream water mass in terms of brightness level. Zeros indicate that Gulf Stream water is not found in that region.

E. Composite Several Images

A composite is the result of combining several classified images on a single grid. A sliding window of several days limits the age span of information in the composite. As the most recent classifications are added, classifications falling outside the window are dropped. New, non-cloud pixels are entered on the composite by simply overwriting the existing classifications. Otherwise, the composite pixels retain the classification of the older images in the window. Over time and several images, the cloud tends to be removed by this process. The result is an up-to-date digital chart of water-mass features that can be edited and annotated for follow-on applications.

Adding more images would eventually remove all of the cloud, but the movement of the Gulf Stream and related features causes distortion and intolerable error in long-term composites.

IV. EXAMPLES

Fig. 3 demonstrates the effect of water-mass classification. It is an image from a NOAA-11 scene of the western North Atlantic on May 11, 1990. Fig. 3a is the result of loading AVHRR LAC multichannel temperature on a Lambert grid and then labeling the cloud (white). The grid is 1500 km x 3000 km, roughly north-south by east-west. The mesh size is 3 km square; so the grid array is 500 rows and 1000 columns. A brown landmask covers the eastern U.S. and Canada. A color table has been applied to the sea surface temperatures; brighter intensities represent warmer temperatures.

Fig. 3b is Fig. 3a with the water masses classified. The image was segmented using clusters, and then the segments were classified. Red signifies the warm core of the Gulf Stream. Blue signifies Sargasso water and is generally seaward of the

stream. Gray signifies slope water, which has the most complex of the surface patterns. Often the algorithm is unable to decide which class is appropriate in the slope water region. When this happens, the "unknown" segments are shown as black.

This classification used files of historical statistics of surface temperature. Frequently we will find segments labeled "Sargasso" shoreward of the stream and segments labeled "slope" seaward of the stream. When these segments are not the surface effects of rings, the labels are oceanographically incorrect. We have programs that search and change the incorrect labels based on rules, but these are not always effective and some manual editing may be warranted.

Fig. 4 shows a composite of three classified images from May 9 to May 11, 1990. It includes the image in Fig. 3. The addition of the two other images has removed much of the cloud in Fig. 3.

V. DISCUSSION AND FUTURE WORK

A methodology that automates digital image processing based on water masses has advantages. For example, it avoids high wave number noise common to most edge detection techniques. It preserves the contextual framework needed for rule-based decision-making and tends to model the largely successful manual process of image interpretation. Since properties other than temperature also correlate with water masses, the water-mass locations from an image of sea surface temperature can be extended to these other properties. On the same basis, water masses provide a vehicle for building composites from different kinds of data.

Fig. 5 shows how this methodology would support subsurface ocean products in a simple way. We suppose that an analyst takes recent satellite data and locates the water masses on a digital chart, using WADI software or manual procedures. By showing where the water masses are found, the chart shows where other properties of those particular water masses also are found. By independently supplying those water properties as data bases, we can automatically create new grids from the feature chart. These grids could be three-dimensional and would represent up-to-date conditions at both surface and subsurface levels. This figure shows several proposed data bases of water properties.

The greatest impediment to this approach is the bias in IR measurements from satellites. The classification algorithms assume that the pixel and data base surface temperatures are standardized. Atmospheric water vapor, or other contaminants, cause the the satellite scanner to report erroneous surface temperatures. If not properly compensated, this error

propagates into the water-mass classifications when the pixel and data base temperatures are compared. A similar problem arises if ocean skin temperatures exhibit large fluctuations which are independent of temperatures a meter or so below the surface. Future work needs to find better ways of adjusting AVHRR IR temperatures to reflect true water temperatures, on an image-to-image basis. One way may be to adjust daily the IR temperatures to in-situ temperatures from bathythermographs, surface buoys, and permanent coastal stations. Another way may be to find water masses which historically show little change of surface temperature. The adjustment needed to correct the pixel temperatures of these water masses may turn out to apply to large portions of the image.

VI. ACKNOWLEDGEMENTS

This work is in support of ongoing ocean products at the U.S. Naval Oceanographic Office. Dr. Melvin Janowitz, from University of Massachusetts, Amherst, provided algorithms used in clustering and cloud detection and called to our attention several potential avenues of pattern recognition. Numerous special contributions were made to this work by Maria Kalcic (NOARL) and Isaac Traxler (University of Southern Mississippi). Ron Vaughan (Sverdrup Technology, Inc.) did the majority of software design and programming required for WADI.

VII. REFERENCES

McClain, E. P., W. G. Pichel, and C. C. Walton (1985). Comparative performance of AVHRR-based multichannel sea surface temperatures. *J. Geophys. Res.*, 90 (C6), 11587-11601.

Coulter, R. E. (1983). Application of the Bayes Decision Rule for automatic water mass classification from satellite infrared data. *Proceedings Seventeenth International Symposium on Remote Sensing of the Environment*, Ann Arbor, Michigan, 589-597.

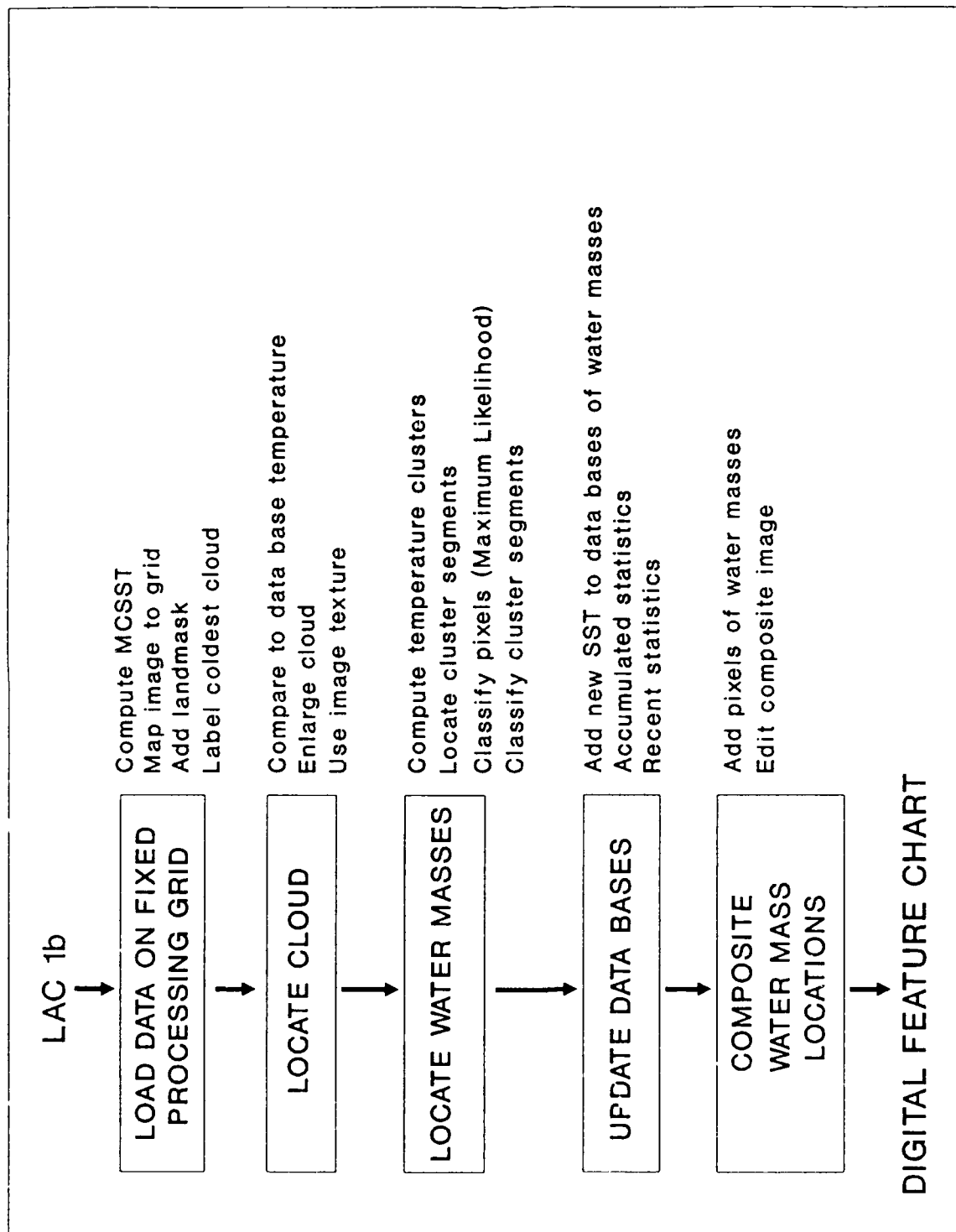


Figure 1. Order of image processing using WADI. The five main processing functions are shown enclosed in boxes. The specific tasks performed by each function are listed adjacent to the boxes.

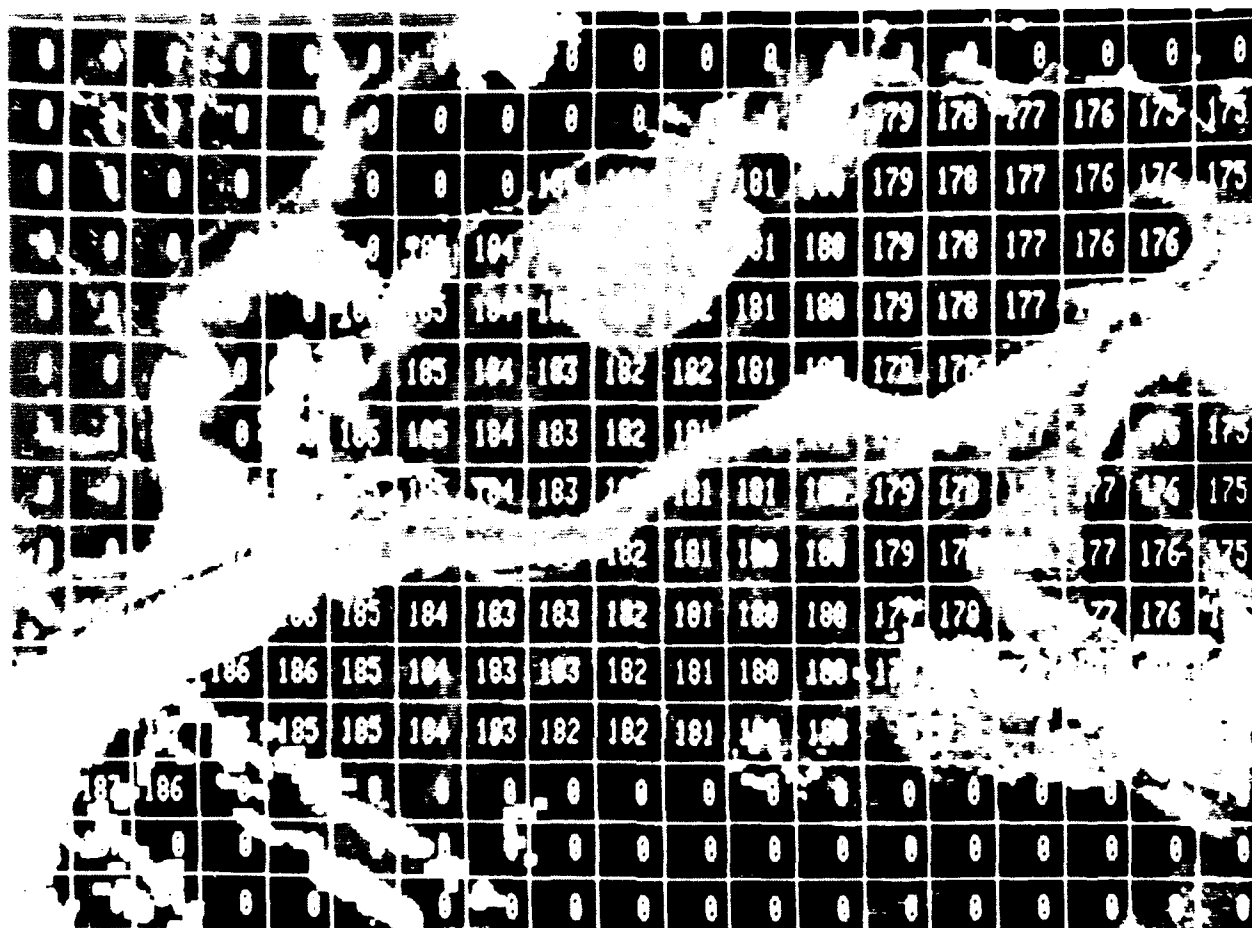


Figure 2 Data base of historical sea surface temperature of the Gulf Stream for May given in brightness levels. The grid cells are 50 km square. Entries are provided where Gulf Stream water has been found historically.

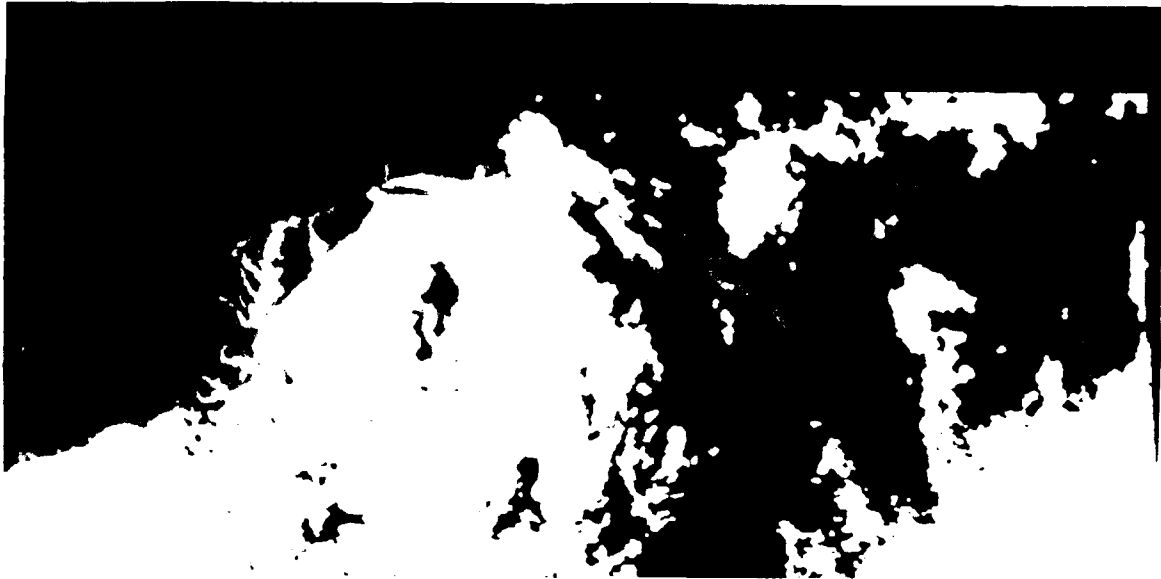


Figure 3 a) (bottom) AVHRR scene of the western North Atlantic on May 11 1990, mapped to a Lambert grid. A color table has been applied to the image. The landmask is brown and detected clouds are shown white. Brighter colors signify warmer ocean temperatures

b) (top) Same as in "a", but with classified water masses. Dark blue signifies Sargasso water, red signifies warm core water of the Gulf Stream, light blue signifies slope water, purple signifies shelf water, and black signifies unknown features



Figure 4. Composite of three classified AVHRR scenes from May 9-11, 1990. One of the three classified scenes is shown in Fig. 3. Color legend is as in Fig. 3.

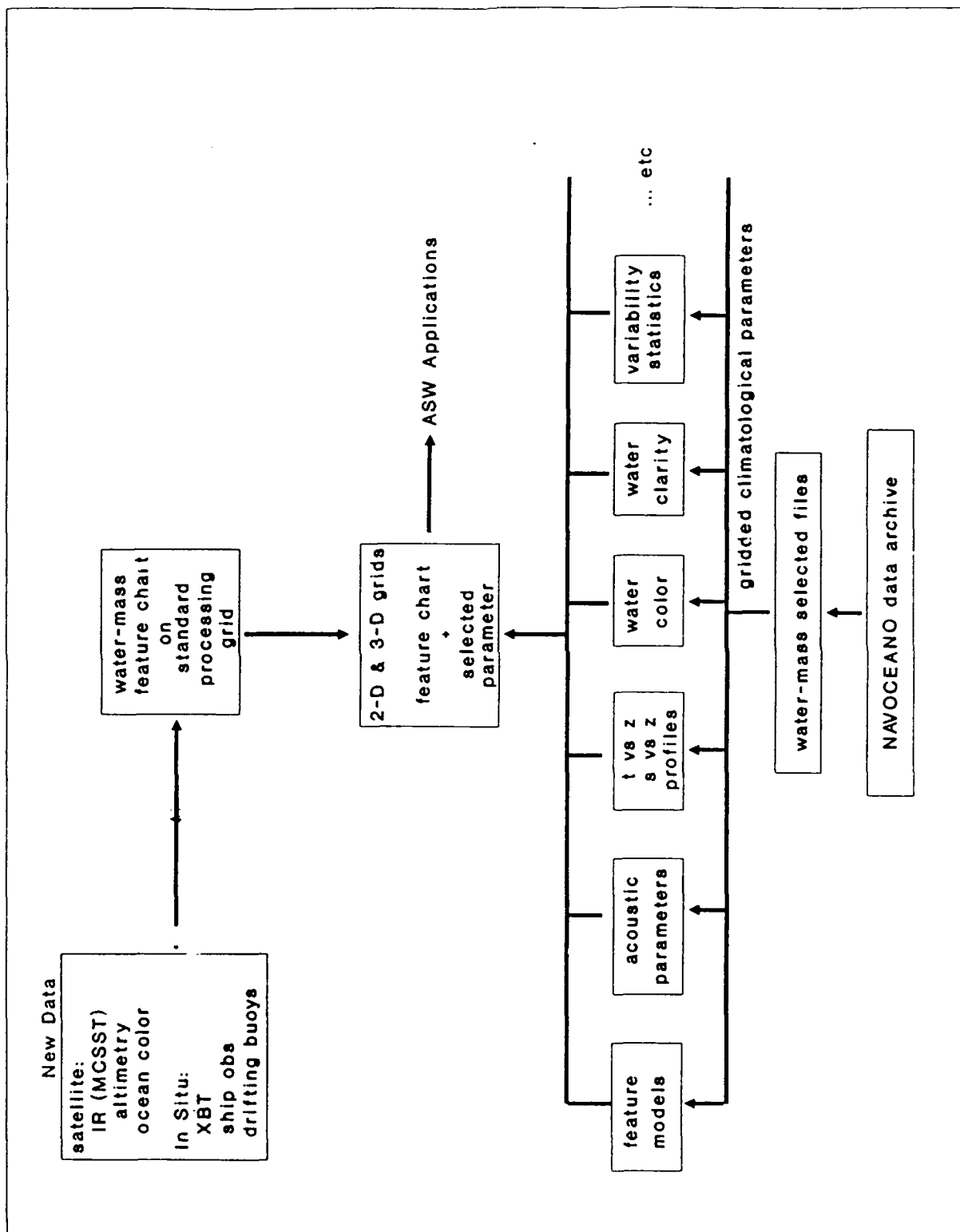


Figure 5. Integrating satellite image processing and ocean climatologies. A feature chart of water-mass locations is combined with water-mass climatologies to produce grids of water properties. These grids can be either stand-alone products or utilized in follow-on applications.

LABELING NORTH ATLANTIC MESOSCALE FEATURES FROM SATELLITE PHOTOGRAPHS: A NEW APPROACH

by

B.P. Buckles and F.E. Petry
Department of Computer Science
Center for Intelligent and Knowledge-based Systems
Tulane University
New Orleans, LA 70118

I. INTRODUCTION

For several years, NOARL has been engaged in a multi-faceted attack on the problem of automating the labeling of satellite images of ocean regions [Lybanon, 1986]. Several important advances have been achieved including a highly flexible algorithm for segmenting infrared images of the ocean [Holyer, 1989] and the development of an expert system for predicting future positions of mesoscale features [Thomason, 1989]. Work is currently in progress to tie these two together as well as improve both. Tied together, the fleet could acquire an image and predict the state of the ocean seven days hence without the aid of oceanographic specialists.

The work reported here is one approach to that process. Beginning with a segmented infrared image, the features are labeled using a new search paradigm known as genetic algorithms. Success will allow the process to be automated from a "cold start." That is, images can be labeled without knowledge of the ocean state during previous time frames.

The testbed for experimentation is based on images from the North Atlantic. The Gulf Stream is the main component of a complex system of oceanic features in the North Atlantic. Other components include cold core rings, warm core rings, continental shelves, and other mesoscale features. The predominant technique for "now-casting" the condition of the Gulf Stream and other currents in its vicinity is via human expert interpretation of satellite photos.

II. GENETIC ALGORITHMS

Genetic algorithms (GAs) are search procedures modeled after the mechanics of natural selection. Domain knowledge is embedded in the abstract representation of a candidate solution, termed an organism. A search over the decision space is performed via an analog of natural selection rather than a simulated reasoning process. Sets of organisms are grouped together into populations. Successive populations are called generations. The GA creates an initial generation, $G(0)$, and for each generation, $G(t)$, generates a new one, $G(t+1)$. A abstract view of a genetic algorithm is given by:

```
generate initial population,  $G(0)$ ;  
evaluate  $G(0)$ ;  
 $t := 0$ ;
```

```

repeat
    t := t + 1
    generate G(t) using G(t-1);
    evaluate G(t);
until solution is found.

```

An *organism*, or chromosome, of length n consists of a vector of the form $\langle x_1, x_2, \dots, x_n \rangle$ where each x_i is an *allele*, or gene. The domain of values from which x_i is chosen is called the *alphabet* of the problem. Alleles from two organisms called *parents* are mixed via an operator called a *crossover* rule of which there are many. Simple one point crossover of two parents at a random point, j , is illustrated by

$$\begin{array}{c}
 \langle x_1 \ x_2 \ \dots \ x_j \ x_{j+1} \ x_{j+2} \ \dots \ x_n \rangle \\
 + \\
 \langle y_1 \ y_2 \ \dots \ y_j \ y_{j+1} \ y_{j+2} \ \dots \ y_n \rangle \\
 = \\
 \langle x_1 \ x_2 \ \dots \ x_j \ y_{j+1} \ y_{j+2} \ \dots \ y_n \rangle
 \end{array}$$

where the resultant organism is called the *offspring* and is placed in the next generation. A *mutation* is the random change of an allele from one alphabet value to another.

The mechanics of GAs are discussed in [Goldberg, 1989a]. GA/F³ in particular is discussed in [Ankenbrandt, 1989]. A separate application, feature extraction, of GAs in satellite photograph analysis is given in [Siedlecki, 1989]. From a mechanistic view, GAs are an iterative process where each iteration has two steps, generate and evaluate. The generate step includes a selection operator and two modification operators. In the selection operation, domain information in the form of a fitness function is used to guide the selection of parents for the following iterations. The fitness of a candidate is a measure of its quality. High quality candidates are given more emphasis in following iterations because they are selected more often. In the modification operation, crossover rules combine candidates selected to form offspring for the next generation. The mutation operator is applied to a small number of offspring. It forces an arbitrary change in one allele. The GA converges when there is no change in the candidates from iteration to iteration. At convergence, often the candidates are all identical. Stopping rules other than convergence are sometimes used.

GAs differ from traditional search techniques in several ways. First, genetic algorithms optimize the tradeoff between exploring new points and exploiting the information discovered so far. This was proved using an analogy with the k -armed bandit problem [Holland, 1975, pp. 75-88]. Second, GAs have the property of implicit parallelism. Implicit parallelism means that the effect of the genetic algorithm is equivalent to an extensive search of the hyperplanes of the given space, without directly testing all hyperplane values [Goldberg, 1989b]. Third, genetic algorithms are randomized algorithms, in that they use operators whose results are governed by probability. The results for any given operation are based on the value of a random number. Fourth, GAs operate on several solutions simultaneously, gathering information from current search points to direct the subsequent search. Maintaining multiple solutions concurrently makes GAs less susceptible to the problems of local maxima and noise.

Relationships between segments in an image consist of linguistic terms such as "near" or "curved". Imprecision is inherent in the language used to define the relationships. Predicates that evaluate to truth values in the interval $[0,1]$ naturally express this imprecision. Given a labelling of the image segments, the degree each relationship fits each pair of segments can be measured. These measures are combined into an overall figure of merit. In the genetic algorithm context, this is a fitness function. The label list is an organism. Organisms are collected into sets called generations. The fitness function determines the probability that some subset of the labelling will propagate to the next generation.

As noted earlier, the labeling problem discussed here is part of a larger goal of not only recognizing but predicting the future positions of mesoscale features in the North Atlantic [Thomason, 1989]. A previous description of the GA labelling of oceanic features is in [Ankenbrandt, 1990]. The work described is implemented within a GA testbed at Tulane University.

III. SOLUTION MODEL

Scene recognition is an application for which the GA model we use is suited. For example, Figure 1 is a segmented image of the North Atlantic. The lines (referred to here as segments, $s_1, s_2, \dots$) represent boundaries between warm and cold regions of sea water. The problem is to classify the segments as Gulf Stream North Wall (NW), Gulf Stream South Wall (SW), cold eddies (CE), warm eddies (WE), continental shelf (CS), and "other" (O).

Relationships which can be expressed as fuzzy truth functions are known to exist within or between classifications. Principal among these are (1) the average width of the Gulf Stream is 50 kilometers, (2) the average diameter of an eddy is 100 kilometers, (3) cold eddies are usually south of the Gulf Stream, and (4) warm eddies are usually north of the Gulf Stream. To these one must add the trivial (yet necessary) relationships such as the south wall is at a lower latitude than the north wall and the known geophysical coordinates of continental shelves.

Starting with infra-red photos, a scene is segmented by finding the borders between warm and cold bodies of water. An organism is composed of a series of labels, one for each segment. For example,

<CE CE SW NW WR>

means that segments 1-5 are, respectively, cold eddy, cold eddy, Gulf Stream south wall, Gulf Stream north wall, and warm ring.

A scene consisting of classification categories ($cat_1, cat_2, \dots, cat_n$) and relationships expressed as truth functions ($P^{(1)}_{ij}, P^{(2)}_{ij}, \dots$) between categories can be modelled as a semantic net (or, more precisely, an association list). A generic one is shown in Figure 2. Segments are attached to the categories via the INST (instance) relation. An allele (or gene) is a

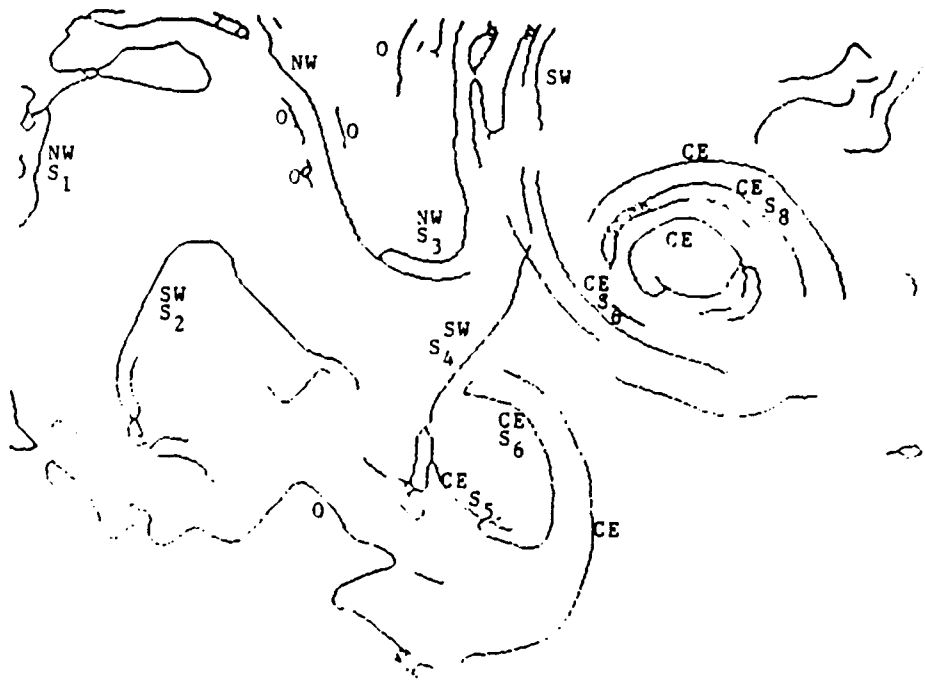


Fig. 1. Segmented Image of the North Atlantic

category name. An organism is a list of categories, one allele for each segment as earlier illustrated.

Formally, let an association list be defined as $A = \langle V, P \rangle$ where $V = \{cat_1, cat_2, \dots, cat_m\}$ is a set of categories, and $P = \{P_{ij}(g) \mid i, j \leq m, g = 1, 2, \dots, r_{ij}\}$ is a set of binary predicates. These predicates describe the relationships between categories and the ideal relationships between categories and ideal relationships between segments assigned to these categories. Let an organism for spatial labeling is defined as $Q = \langle S, INST \rangle$, where $S = \{s_1, s_2, \dots, s_n\}$ is a set of segments, and $INST: S \rightarrow V$ is a function.

The lowest level fuzzy predicates, called derived predicates, are used to evaluate conditions that exist between segments. They are of the form "is north of", "is near", "is 50 km from", and others. The chief form given these predicates is $\exp(-b|a|)$. Where b is a constant and a is a measure of an inter-segment relationship. For example, if d_{ij} is the distance in kilometers between segments i and j , then the predicate "is 50 km from" is

$$F_{50km}(s_i, s_j) = \exp(-0.5 \times |d_{ij} - 50|)$$

These predicates are used to determine how closely the labels in an organism match ideal inter-segment relationships in a given image of the North Atlantic. The ones with higher values after combining and aggregating all relationships, as described in [Messa, 1990a], are given higher fitness function values.

Organisms with higher values are given increased opportunities to "reproduce" and contribute offspring to the next generation of organisms. When all organisms achieve nearly equal fitness values, the algorithm stops. The final labeling is equivalent to the organism with the highest fitness function value.

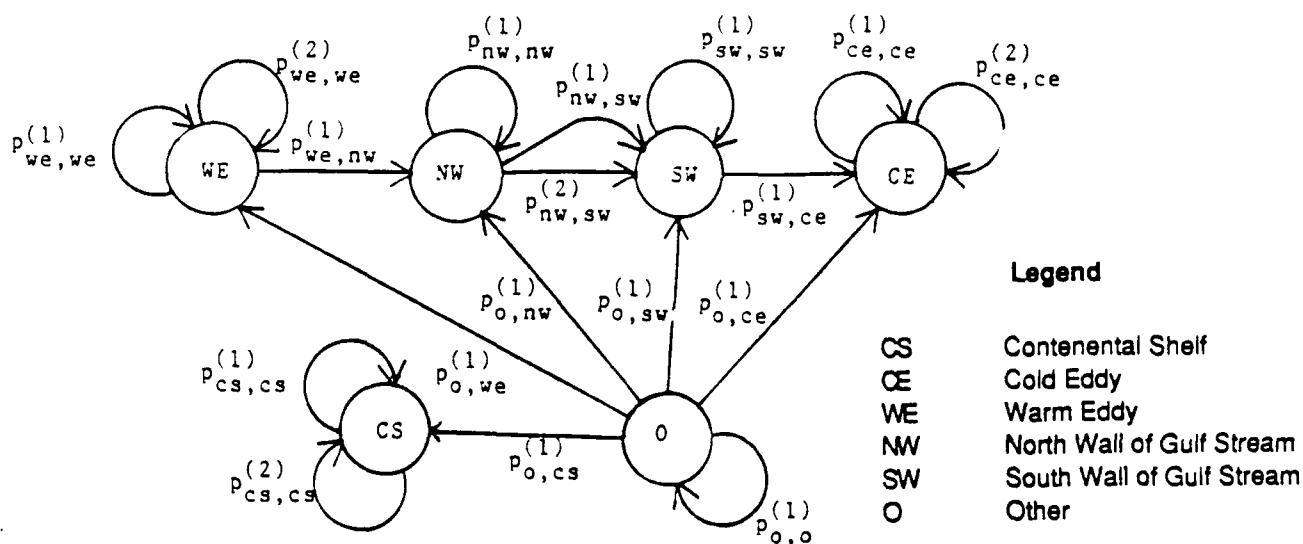


Fig. 2. Generic Semantic Net for Oceanic Features

IV. EXAMPLE

Figure 1 has eight segments labeled (correctly). The segments are denoted as $s_1, s_2, \dots, s_8$ where the subscript denotes the position of a candidate label in an organism. Table 1 lists and defines all predicates and derived predicates required for the segments of Figure 1. The value 0.5 is chosen arbitrarily for β in all derived predicates. In all cases, x stands for the distance between a pair of segments and X stands for the latitude of the center of a segment. The exponential form of derived predicates is used for all relationships except "north of" where the subformula $X_k > X_l$ means 1 (one) if segment k has a greater latitude than segment l and 0 (zero) otherwise. The default value for any predicate or derived predicate is zero should a denominator evaluate to zero. MAX and AVG are the dominant combinators and aggregators.

For this example, we need only the geophysical coordinates, the distances between segment centroids, and the distances between the closest points of segments. A larger, more complete description might also contain the length and degree of curvature of each segment.

Table 2 lists six organisms together with their fitness function values which are computed using the predicates in Table 1 and the geophysical values from the segments in Figure 1. Organism O_1 has no segments labeled incorrectly. O_2 has two segments labeled incorrectly. O_3 through O_6 have 3, 3, 5, and 8 incorrectly labeled segments, respectively. The fitness function values correspond roughly to the correctness of the labeling. Additional predicates (i.e., a more complex semantic net) would improve upon the ordering and separation in most cases.

V. CONCLUSION

A model for labeling complex scenes via genetic algorithms with fuzzy fitness functions evaluated over semantic nets and GAs is possible. Truth functionals indicating

the degree to which specific interfeature relationships are fulfilled are combined at the segment level then aggregated at the category level using fuzzy set operators.

Table 1. Predicate Descriptions

Analytical Form	Description
$P^{(1)}_{we,we} = \text{AVG}_{k \text{ in } we} [\text{MAX}_{l \text{ in } we} [\exp(-0.5 100-x)]]$	The diameter of a warm eddy is about 100 km. x is distance between segs.
$P^{(2)}_{we,we} = \text{AVG}_{k \text{ in } we} [\text{MAX}_{l \text{ in } we} [\exp(-0.5x)]]$	A warm eddy segment is close to a we segment. x is distance between.
$P^{(1)}_{we,nw} = \text{AVG}_{k \text{ in } we} [\text{MAX}_{l \text{ in } nw} [X_k > X_l]]$	A warm eddy is north of north wall. X is latitude value.
$P^{(1)}_{nw,nw} = \text{AVG}_{k \text{ in } nw} [\text{MAX}_{l \text{ in } nw} [\exp(-0.5x)]]$	A north wall segment is close to a nw segment. x is distance between them.
$P^{(1)}_{nw,sw} = \text{AVG}_{k \text{ in } nw} [\text{MAX}_{l \text{ in } sw} [\exp(-0.5 50-x)]]$	The north wall is about 50 km from south wall. x is a distance
$P^{(2)}_{nw,sw} = \text{AVG}_{k \text{ in } nw} [\text{MAX}_{l \text{ in } sw} [X_k > X_l]]$	The north wall is north of south wall. X is a latitude value.
$P^{(1)}_{sw,sw} = \text{AVG}_{k \text{ in } sw} [\text{MAX}_{l \text{ in } sw} [\exp(-0.5x)]]$	A south wall segment is close to a sw segment. x is a distance.
$P^{(1)}_{sw,ce} = \text{AVG}_{k \text{ in } sw} [\text{MAX}_{l \text{ in } ce} [X_k > X_l]]$	The south wall is north of cold eddies. X is a latitude value.
$P^{(1)}_{ce,ce} = \text{AVG}_{k \text{ in } ce} [\text{MAX}_{l \text{ in } ce} [\exp(-0.5 100-x)]]$	The diameter of a cold eddy is about 100 km. x is a distance.
$P^{(2)}_{ce,ce} = \text{AVG}_{k \text{ in } ce} [\text{MAX}_{l \text{ in } ce} [\exp(-0.5x)]]$	A cold eddy segment is close to a ce segment. x is a distance.

We are currently investigating such issues as the effect of many predicates clustered on one or two categories, alternate forms for the truth functionals themselves, and the crossover rules. Our image set consists of six segmented infrared photographs of the North Atlantic, each photograph having a different degree of occlusion. Our testbed consists of a GA algorithm capable of manipulating the alleles' correspondence to the semantic net.

Table 2. Fitness Function Values for Selected Organisms

Organism	Fitness Value
$O_1 = \langle \text{NW SW NW SW CE CE CE CE} \rangle$	$E(O_1) = 2.2098$
$O_2 = \langle \text{SW SW NW NW CE CE CE CE} \rangle$	$E(O_2) = 2.2511$
$O_3 = \langle \text{NW SW NW NW CE CE NW SW} \rangle$	$E(O_3) = 2.1251$
$O_4 = \langle \text{SW SW NW CE NW CE CE CE} \rangle$	$E(O_4) = 1.4731$
$O_5 = \langle \text{NW NW CE CE SW NW SW CE} \rangle$	$E(O_5) = 1.6757$
$O_6 = \langle \text{SW CE SW CE SW NW SW NW} \rangle$	$E(O_6) = 0.9235$

VI. ACKNOWLEDGEMENT

This research was partially supported by a grant from the Naval Oceanographic and Atmospheric Research Lab, Grant #N00014-89-J-6003.

VII. REFERENCES

- Messa, K. (1990a). A Generalization of the Fitness Function and Crossover Operator Used in Genetic Algorithms. Thesis. Tulane University.
- Messa, K. (1990b). Genetic Algorithms and Curve Fitting with Applications to an Altimeter Derived Reference Surface Error. this proceedings.
- Siedlecki, W. and J. Sklansky. (1989). A Note on Genetic Algorithms for Large-scale Feature Selection. Pattern Recognition Letters 10(11):335-347.
- Thomason, M. (1989). Knowledge-based Analysis of Satellite Oceanographic Images. Int. J. of Intelligent Systems 4(1):143-154.
- Ankenbrandt, C., B. Buckles, and F. Petry (1989). Ocean Feature Recognition Using Genetic Algorithms with Fuzzy Fitness Functions. 3rd Workshop on Space Applications, Automation, and Robotics. 679-685.
- Ankenbrandt, C., B. Buckles, and F. Petry. (1990). Scene Recognition Using Genetic Algorithms with Semantic Nets. Pattern Recognition Letters 11(4):285-293.
- Lybanon, M., J.D. McKendrick, et al. (1986). A Prototype Knowledge-based System to Aid the Oceanographic Image Analysis. Applications of Artificial Intelligence III, SPIE 635. 203-206.
- Holyer, R.J., and S.H. Peckinpaugh. (1989). Edge Detection Applied to Satellite Imagery of the Oceans. IEEE Trans. on Geoscience and Remote Sensing 27 (1):46-56.

Goldberg, D.E. (1989a). Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley, Reading MA.

Goldberg, D.E. (1989b). Sizing Populations for Serial and Parallel Genetic Algorithms. 3rd Intern. Conf. on Genetic Algorithms. 70-79.

Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. Univ. of Michigan Press, Ann Arbor, MI.

PROGRESS WITH NEURAL NETWORK GULF STREAMS

Eugene Molinelli, Gregory Muncill, and Kevin Pepe
Planning Systems Incorporated
7925 Westpark Drive
McLean, Virginia 22102

I. INTRODUCTION

An expert human analyst is capable of examining an infrared image of sea surface temperature in the western North Atlantic Ocean offshore of Cape Hatteras, and identifying a Gulf Stream north wall where the sea surface is not obscured by clouds. The north wall of the Gulf Stream is a spatially continuous but temporally meandering narrow boundary between warm waters carried swiftly from the south and cold waters indigenous to the continental slope regime off the east coast of the United States. The Gulf Stream as depicted by a Mesoscale Product of the Remote Sensing Branch of the Naval Oceanographic and Atmospheric Laboratory (NOARL), formerly NORDA, is provided in Figure 1.

The north wall is closely related to a boundary between two ocean regions in which acoustic propagation conditions are dramatically different (e.g., Molinelli and Hanna, 1988). The position of the north wall of the Gulf Stream is therefore important to acoustic antisubmarine warfare (ASW) tactics and hence is important to Navy operations. Gulf Stream analyses are performed by expert analysts at the Operational Oceanography Center (OOC) of the Naval Oceanographic Office (NAVOCEANO) and the Naval Eastern Oceanography Center (NEOC) and transmitted to Fleet units.

The current analyses are time consuming, tedious, qualitative, and subjective, i.e. dependent on the person performing the analysis. Researchers at NAVOCEANO and at NOARL are developing more objective, numerical procedures that can be implemented on modern computer systems and that can, at least, assist the analyses performed by humans.

This article reports on our progress in NOARL sponsored efforts to utilize the emerging technology of neural networks to automatically interpret infrared imagery to detect Gulf Stream north wall positions.

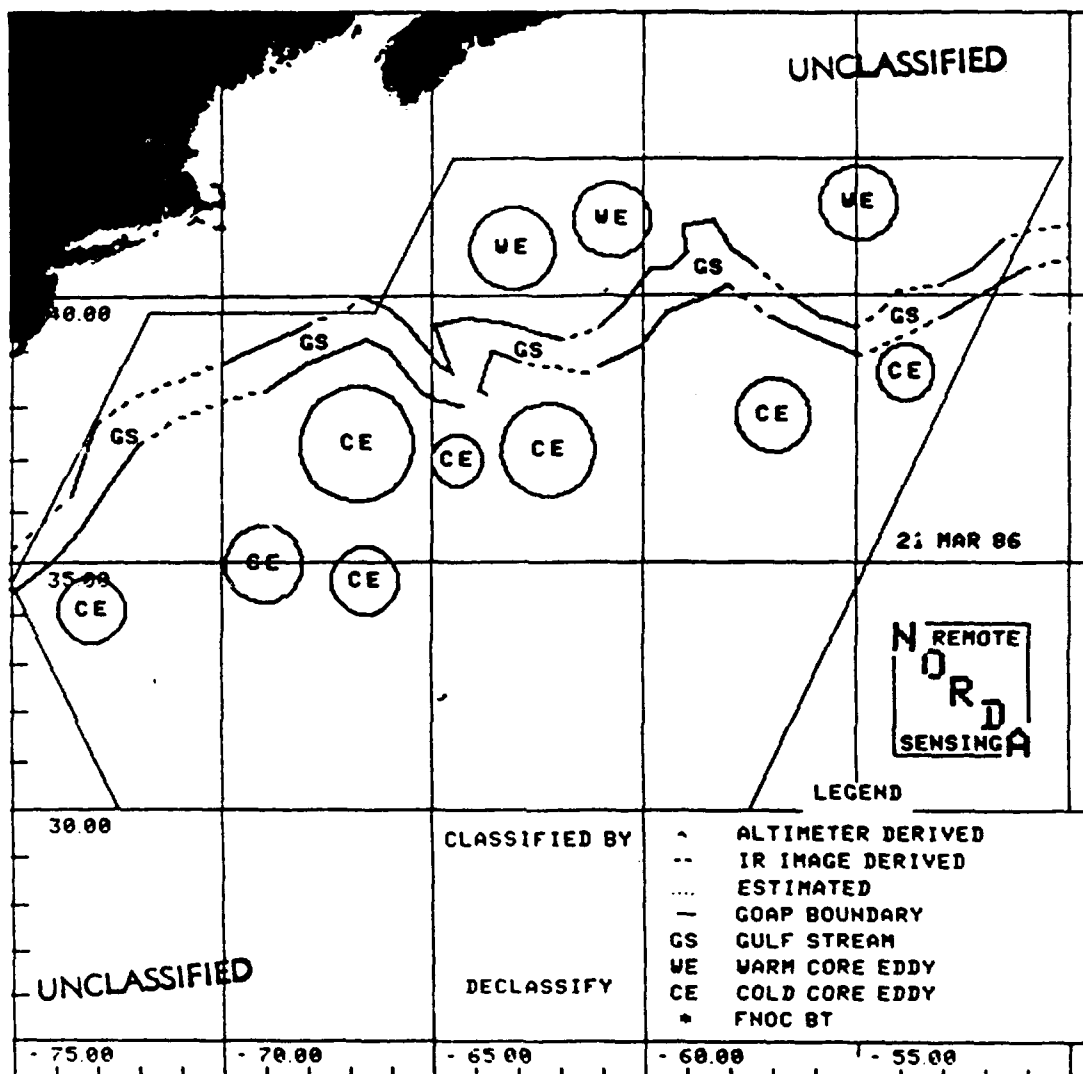


Figure 1. Gulf Stream Region Showing Coastal and Oceanographic Features of Interest on a Mercator Projection. The Gulf Stream is represented by the parallel curves. The north wall of the Gulf Stream is the curve closest to the solid area representing the North American land mass; it has strong sea surface temperature (SST) gradients. The other curve represents the offshore limit of the warmest waters in the scene, and has been called a south wall, though the SST gradients associated with it are an order of magnitude less than those of the north wall. Warm core (WE) and cold core (CE) eddies are represented by circles. This Mesoscale Product of the NOARL was produced on a semiweekly basis from September 1985 through June 1987. The GOAP boundary outlines the region of interest to NOARL's GEOSAT Ocean Applications Program.

II. BACKGROUND

A. Complex Empirical Orthogonal Functions (CEOFs)

To employ numerical procedures, the Gulf Stream must be represented numerically. A natural but inefficient representation of the Gulf Stream is as a set of values arranged in a latitude longitude grid. At grid points where the values are above some threshold, the Gulf Stream is considered to occupy that location. The Gulf Stream is absent at the other grid points. To resolve Gulf Stream positions with tactically significant resolution requires grid points about 10 n.mi. apart; so, because the Gulf Stream region off Cape Hatteras is about a 2000-n.mi. square, 40,000 grid points, many of them without a Gulf Stream, would be required. And a new grid would be needed with every significant Gulf Stream position shift (i.e., every two to three days). One year of Gulf Stream positions would require at least 4.5 million values.

A much more efficient numerical representation of the Gulf Stream is as a complex vector. A single position along the Gulf Stream is designated by a latitude, longitude pair which is combined into a single complex number—the longitude is the real part and the latitude is the imaginary part. A continuous Gulf Stream is made up of a sequence of these complex numbers, one number every 10 n.mi. along the Stream, assembled into one vector. A Gulf Stream segment 1310 n.mi. long is then represented by one vector with 132 complex elements (Figure 2). Now the 40,000 values of the previous method are replaced by 264 values (132 real parts and 132 imaginary parts)—a 150-fold improvement in efficiency. Again, however, a new representation is needed every few days. Even with this method, one year of Gulf Stream positions requires about 30,000 values.

Another increase in efficiency can be obtained by better representing the fluctuations in Gulf Stream positions so that entirely new vectors are not needed every few days. A set of complex vectors, representing an historical collection of Gulf Streams, can be decomposed into principal components of variation, i.e. into eigenvectors of an expectation covariance matrix (Preisendorfer, et al. 1981; Carter, 1985; Molinelli and Flanigan, 1987). These vectors have also been called complex empirical orthogonal functions (CEOFs). Theoretically, for 132 vector elements, 132 different eigenvectors each with 132 elements can be defined. For each eigenvector there is an eigenvalue that indicates the percentage of the total covariance that is accounted for by the eigenvector. The eigenvectors can be ranked by their eigenvalues and only the few most significant vectors used. These most significant ranked vectors are called modes of the variability. Sample modes are illustrated in Figure 3.

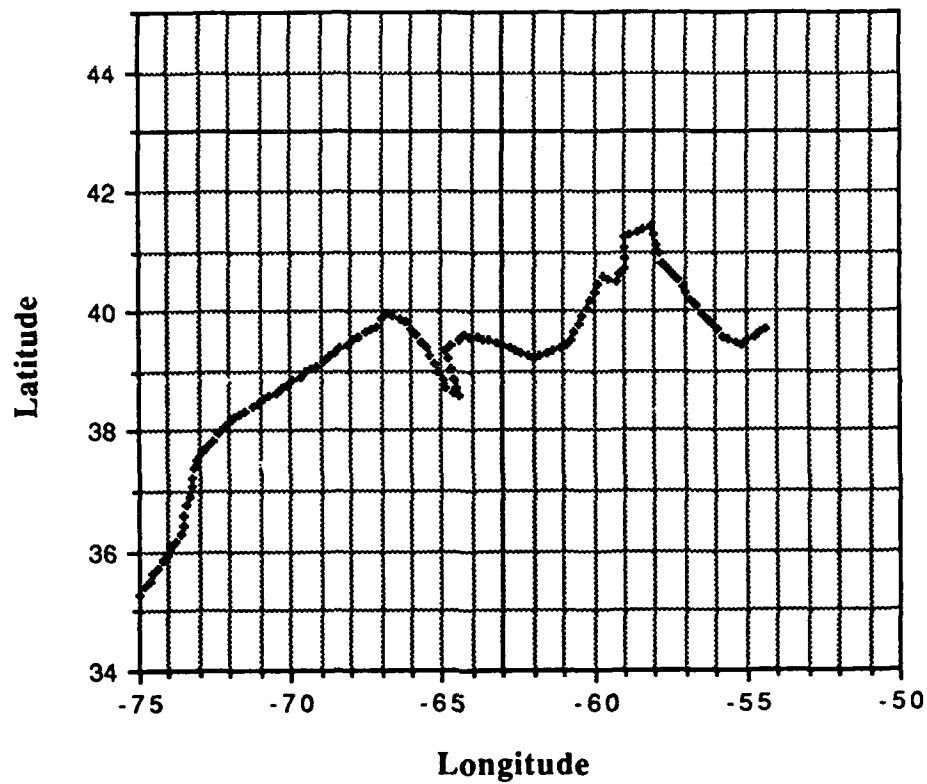


Figure 2. A Gulf Stream 132 Element Vector. A 1310 n.mi. long Gulf Stream can be represented by 132 complex elements in a downstream sequence. Here such a Gulf Stream is mapped on a Cartesian plane with a latitude to longitude distance ratio approximating the local tangent plane.

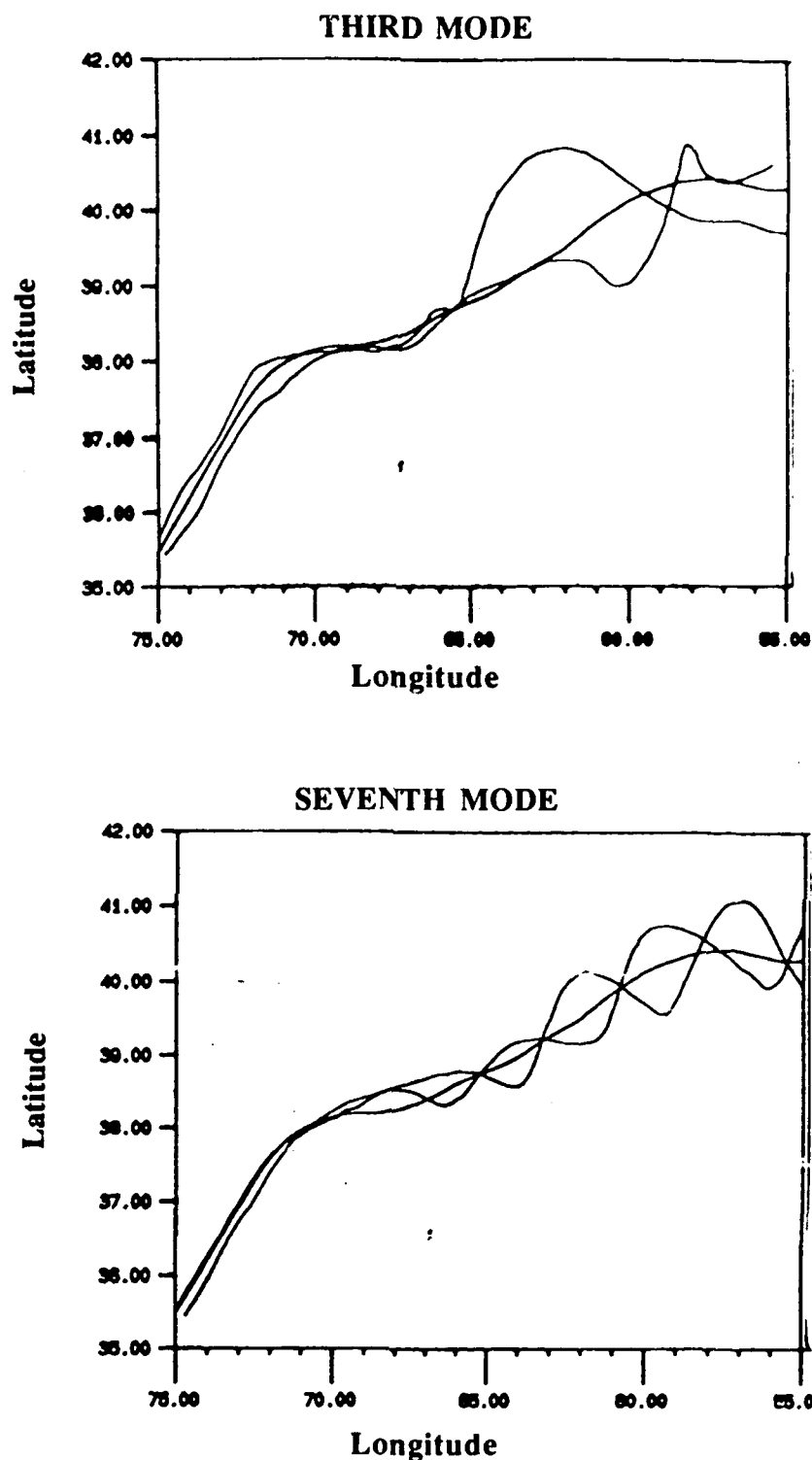


Figure 3. Sample Modes of Gulf Stream Variability. This figure illustrates the difference between the Third Mode and the Seventh Mode. For each mode, the mean Gulf Stream is plotted along with two other Gulf Streams that result from adding contributions from two different coefficients multiplied by the mode's eigenvector. The two chosen coefficients for each mode represent the opposite extremes of the coefficients derived from all the available Mesoscale Products.

A single Gulf Stream vector can be reconstructed from a linear combination of the modes, each multiplied by a complex coefficient. Because the modes are orthogonal vectors, the needed coefficients are easily found from a given Gulf Stream vector and the known modes. Molinelli and Flanigan (1987) found ten modes sufficient to describe Gulf Stream north wall positions within about 10 to 30 n.mi. (Figure 4). Using ten modes, it is possible to describe all Gulf Streams for a year to the stated accuracy with 10 mode vectors (2640 values) and just 20 new values (10 complex coefficients) every few days (about 2400 additional values). The total number of values is about 5,000—at least a six-fold improvement over previous cases.

B. Optimal Interpolation using CEOFs.

The first application of CEOFs of Gulf Stream variability to an operational scenario was as an interpolator to connect Gulf Stream segments inferred from sea surface temperature (SST) gradients in infrared imagery and to connect Gulf Stream points detected by altimeter. Molinelli and Flanigan (1987) demonstrated that reasonable analyses of Gulf Stream positions could be produced from spotty observations. Average Gulf Stream position errors in gaps between observations were about 30 n.mi.

In the procedure developed for interpolation, the human analyst must provide Gulf Stream segments inferred from infrared satellite imagery and Gulf Stream points deduced from satellite altimetry. The optimal estimation technique then adjusts the coefficients of the first ten CEOFs from some initial state until the distance between the observations provided by the analyst and the Gulf Stream reconstructed from the ten modes is minimized. Figure 5 illustrates this process by depicting the Gulf stream of the initial coefficients, the infrared segment data, the altimeter point data, and the final Gulf stream generated from the optimal coefficients.

C. Neural Networks

Our next step is to employ the emerging technology of neural networks to provide CEOF coefficients directly from SST imagery. We begin operating on imagery that has already been processed through an edge detector, so that the neural network is dealing only with SST gradients.

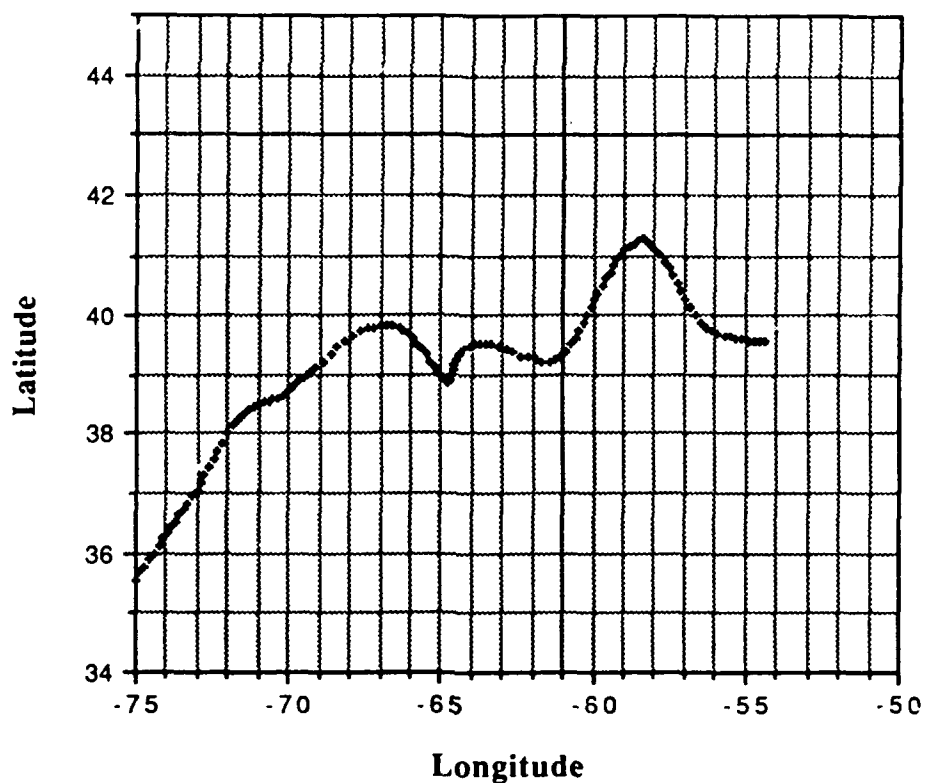


Figure 4. Ten Mode Reconstruction of a Gulf Stream Vector. This figure illustrates how well an actual Mesoscale Product is approximated by a ten mode reconstruction. This representation and the one shown in Figure 2 of the same Gulf Stream are separated by about a 10 n.mi. rms difference.

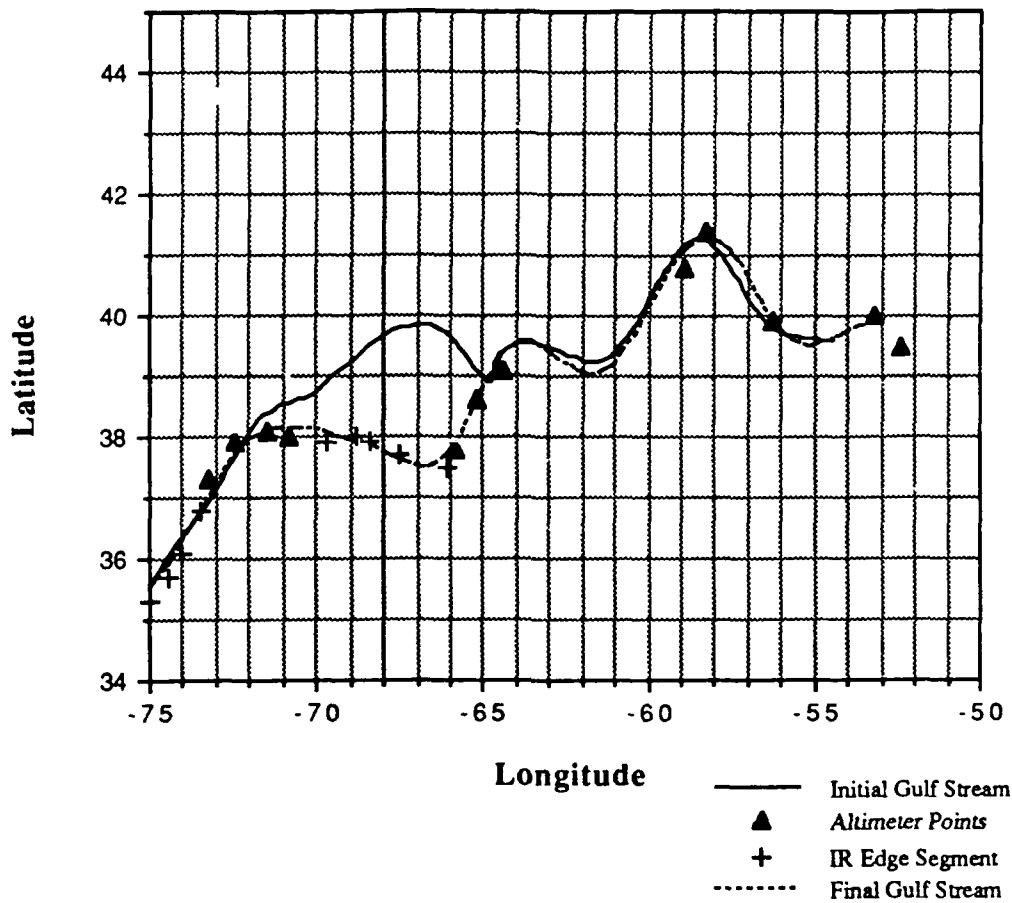


Figure 5. Optimal Estimation. This figure shows how the CEOFs are used in an interpolation scheme. The coefficients of the initial Gulf Stream (the solid line representing the previous analysis) are modified using a gradient search least squares optimization procedure until the resulting Gulf Stream (dashed line) differs as little as possible from the available observations (symbols representing altimeter crossings and infrared image edges).

The coefficients that might be produced from neural networks then have two operational uses:

1. to define a Gulf Stream neighborhood so that automatically detected SST gradients in imagery can be associated with or excluded from association with the Gulf Stream (an associated segment can then serve as an observation of a Gulf Stream segment), and
2. to serve as the initial state of the optimal interpolation procedure.

The first use of CEOF coefficients is necessary because there are a plethora of SST gradients apparent in infrared imagery, only a fraction of which are associated with the Gulf Stream.

Neural networks are composed of many simple computational units (corresponding to biological neurons) that work in unison to process information. Computational elements, or nodes, used in neural network models usually sum N weighted inputs and pass the result through a nonlinearity. The node is characterized by an internal threshold or offset and by the type of nonlinearity. The node weights are not programmed in the traditional sense; rather the weights that connect the neurons are "learned" by the network through repeated applications of input data for which the desired result is known. Neural networks use their massively parallel architecture to explore many competing hypotheses simultaneously. This gives neural networks great potential for application in areas such as speech and image recognition where many hypotheses are pursued in parallel and high computation rates are required. In particular, a network can be presented with an infrared image, and it can process that data based on the information presented during training. The network is tolerant of noise since it has learned the salient features of an image during training, and the noise is in some sense averaged out. A number of neural models for pattern recognition have recently produced promising results and are now being employed on larger applications.

The potential benefits of neural networks extend beyond the high computation rates provided by massive parallelism. For one, neural networks typically provide a greater degree of robustness or fault tolerance due to the large number of computational elements. Also, compared to traditional statistical techniques, neural network classifiers are non-parametric and make weaker assumptions concerning the shapes of the underlying distributions than traditional statistical classifiers. They may thus prove to be more robust when distributions are generated by nonlinear processes and are strongly non-Gaussian.

Neural network models are specified by the network topology, node characteristics, and training or learning rules. These rules specify an initial set of weights and indicate how weights should be adapted during use to improve performance. The nodes of a network are generally

grouped into layers. Nodes within a layer have common activation rules and transfer functions. In addition, each layer may have a unique learning rule for modifying connection weights to other layers.

The particular network architecture used for this study was a two-layer perceptron (shown in Figure 6). Multi-layer perceptrons are feed-forward networks with one or more layers of nodes between the input and output layers. These additional layers contain hidden units or nodes that are not directly connected to both the input and output nodes. A two-layer perceptron can form convex open or closed regions and a three-layer perceptron (two hidden layers) can form arbitrarily complex decision regions. Furthermore, it has been demonstrated (Cybenko, 1988) that continuous nonlinear functions can be approximated to arbitrary precision using three-layer perceptrons with sigmoidal nonlinearities (see Figure 7).

The training algorithm used for the previous work was the gradient descent or back-propagation algorithm (described in Rumelhart and McClelland, 1986 or see Lippman, 1987 for a more general discussion of neural networks). The back-propagation algorithm uses a gradient search technique to minimize a cost function equal to the mean square difference between the desired and the actual network outputs. The network is trained by initially selecting small random weights and internal thresholds and then presenting all training data repeatedly. Weights are adjusted after every trial using information specifying the desired output until weights converge and the cost function is reduced to an acceptable value. The back-propagation algorithm determines how much to adjust each weight by assuming that all processing elements and connections are somewhat responsible for an erroneous response. Responsibility for the error is affixed by propagating the output error backward through the connections to the previous layer. This process is repeated until the input layer is reached.

Although it cannot be proven that the back-propagation algorithm must converge, it has been shown to be successful for many problems of interest. Multi-layer perceptrons trained with back-propagation have been used successfully in such diverse applications as forming letter-to-phoneme rules, classifying submarine signatures, and nonlinear signal processing.

In this application we use the technology of neural networks to connect the gradients of infrared imagery to the complex coefficients of the modes of a continuous, tactically significant Gulf Stream. Ultimately our neural networks will have to identify the Gulf Stream (by means of its CEOF coefficients) in imagery with extraneous edges and missing Gulf Stream segments. It will be sufficient if the neural network can operate well enough to label edges as Gulf Stream edges

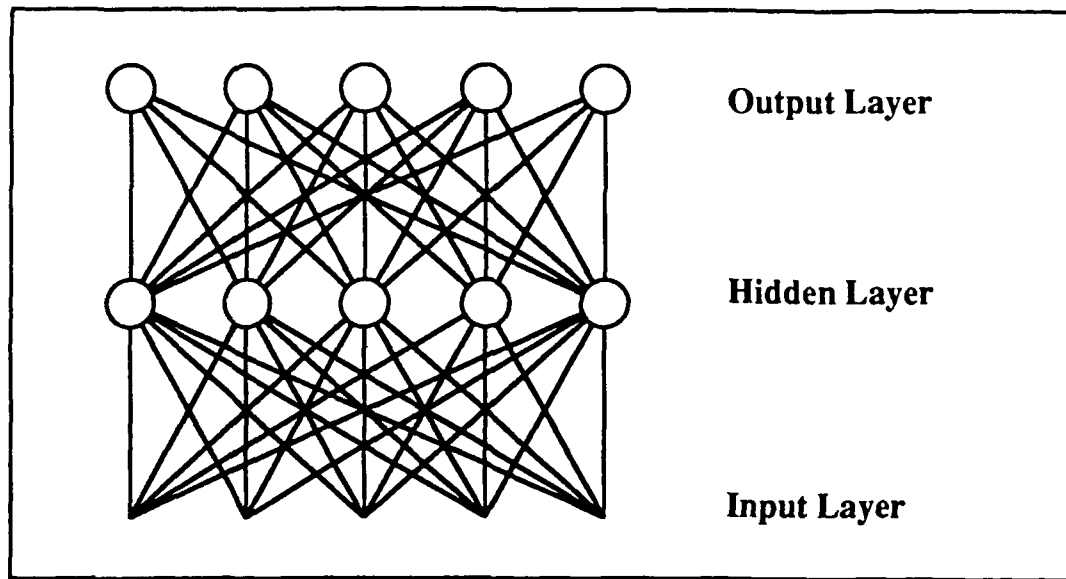


Figure 6. Two-Layer Perceptron. This diagram shows the scheme of connections between an input layer and a hidden layer, and those between the hidden layer and the output layer that constitute the framework of a neural network. Different feed forward weights are assigned to each connection through some learning algorithm.

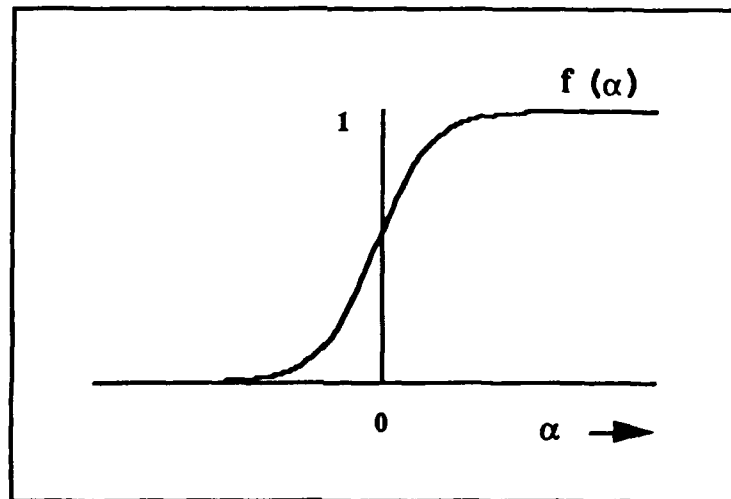


Figure 7. Sigmoid Transfer Function. This plot shows the nonlinear function used to combine the contributions at a node via various connections from a previous layer. This nonlinear transfer function gives this type of network many of its distinctive features.

so that they can be used as hard data in the optimal interpolation scheme and if, in addition, the neural network Gulf Stream can serve as an initial state for the optimal interpolation scheme. There is reason to expect that neural networks can perform in this situation because of their ability to generalize on the training data presented and to degrade gracefully. Thus, when a neural network is presented with partial data it gives the closest answer (in a least squares sense for back-propagation training) to the correct answer.

D. Previous Work

Our first attempts to employ neural networks established the feasibility of the concept (Molinelli and Flanigan, 1989). In that work we used 86 Gulf Streams generated as Mesoscale Products (Lybanon and Crout, 1987) and defined the ten CEOFs that describe the variability, and then calculated the 10 complex coefficients for each of the 86 Gulf Streams. We then established a neural network to output the first three coefficients using the 132 element Gulf Stream vectors as inputs. The network was trained using the back-propagation algorithm with the learning coefficient set to 0.9 and the momentum coefficient set to 0.6. Only 77 Gulf Streams were used in the training set, nine were withheld at random for testing. About 100,000 iterations were required for the network to converge; at that juncture the network CEOF coefficients were highly correlated with the correct CEOF coefficients (the correlation coefficient was 0.98). Though training is a lengthy process, after the network is trained the coefficients can be calculated from the inputs in a fraction of a second. This first step established the ability of a neural network to produce meaningful CEOF coefficients for completely defined Gulf Stream vectors as inputs. We were then ready to attempt a network whose inputs would be analogous to imagery.

As a first attempt to operate on image type input we established a 50x50 grid, i.e. a low resolution (20 n.mi.) image, to serve as the input to a new neural network. We then mapped our 86 Gulf Streams onto this grid; Figure 8 shows what a typical Gulf Stream looks like on this grid. Despite errors in this mapping which added substantial amounts of noise and clutter to the image (errors not discovered until the present work), the network still converged after 150,000 iterations and produced CEOF coefficients correlated to the correct CEOF coefficients with a correlation coefficient of 0.87 for the 77 Streams of the training set. However, network CEOF coefficients for the 9 Gulf Streams of the test set were less well correlated to the correct coefficients, achieving significant correlation coefficients (greater than 0.75) only for the real or imaginary part of the coefficients, but never both.

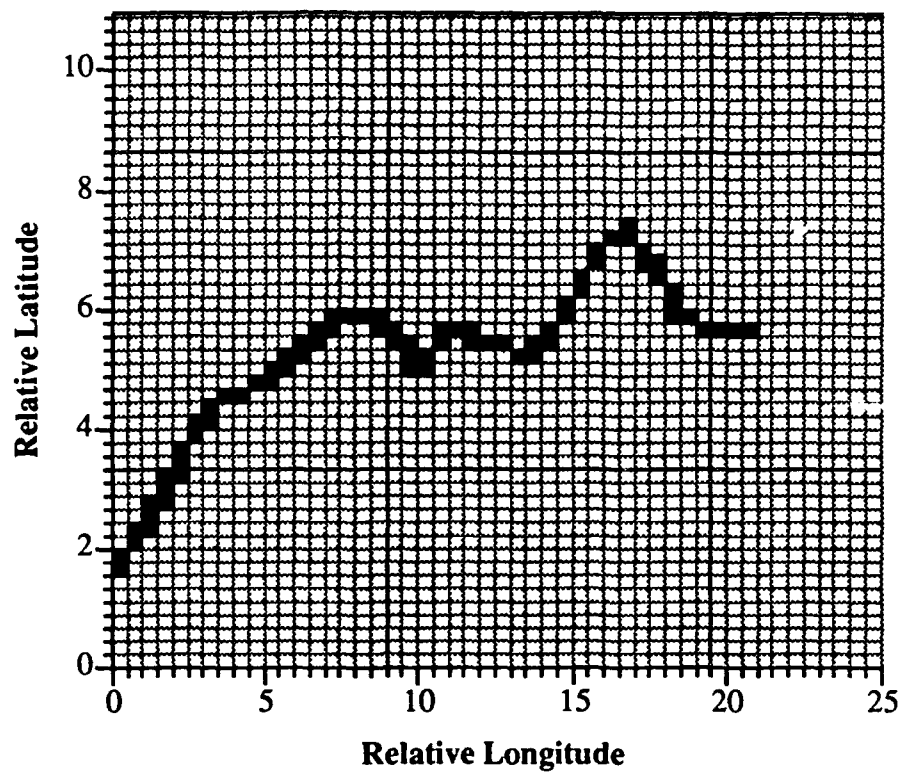


Figure 8. Gridded Gulf Stream on 50x50 Grid. This figure shows how a Gulf Stream looks on the input layer to a network trying to produce coefficients from the 50x50 gridded data with 20 nmi resolution. The appearance is less smooth than that of the original vector (Figure 2)

Given the initial promising results we believed it possible to improve upon the disappointing results for the gridded test cases. The present work is aimed at improving the performance achieved on test cases of gridded, imagery-like, inputs.

III. APPROACH

The encouraging but still inadequate performance with test cases in the previous work indicated that more experiments with training and testing the network would be required. Because so many iterations were required for convergence, and because we were operating on microcomputers, our network training phase was taking several days of computer time. Further experimentation would require faster training, thus one of the first steps of our approach was to upgrade the computer equipment hardware and software.

In addition to better performance on the test set, we wanted the Gulf Streams put out by the network to be more realistic, at least for the training set, so that we could demonstrate the capability of the network to ultimately produce operationally useful analyses; and thus justify the effort required to improve the performance on the test set. Therefore the other steps of our approach included increasing both the resolution of the input grid and the number of modes output by the network. Our approach consisted of the following steps.

1. Recompute CEOFs and coefficients,
2. Increase training speed with hardware upgrade,
3. Rerun 20 n.mi. grid as baseline,
4. Define a measure of performance,
5. Increase the resolution,
6. Increase the number of output modes, and
7. Evaluate the strengths and weaknesses of the approach.

IV. RESULTS

In Table 1, repeated from Molinelli and Flanigan (1989), we list the 86 Mesoscale Product Gulf Streams that make up our data set and identify by asterisk the nine to be withheld from the training set for testing purposes. We had not stored the CEOFs of the 4th through 10th modes, nor the corresponding coefficients during our previous work; and, we found an initialization error in the computer program that originally computed the first 3 CEOFs. Therefore we recomputed the one set of 10 CEOFs for all 86 Gulf Streams, and then computed the 86 sets of 10 complex

Table 1. List of 86 Gulf Stream Pairs at Inflection Points
Stored in Computer Files

FILE NO.	NAME	DATE	FILE NO.	NAME	DATE
1	MESO002.DAT	JAN 10, 1986	44	MESO057.DAT	AUG 29, 1986
2	MESO003.DAT	MAR 10, 1986	45	MESO058.DAT	SEP 3, 1986
3*	MESO004.DAT	FEB 21, 1986	46	MESO059.DAT	SEP 5, 1986
4	MESO005.DAT	FEB 12, 1986	47	MESO060.DAT	SEP 10, 1985
5	MESO006.DAT	FEB 14, 1986	48	MESO061.DAT	SEP 12, 1986
6	MESO007.DAT	FEB 19, 1986	49*	MESO062.DAT	NOV 24, 1986
7	MESO008.DAT	FEB 21, 1986	50	MESO063.DAT	NOV 28, 1986
8	MESO009.DAT	FEB 26, 1986	51	MESO064.DAT	DEC 3, 1986
9	MESO010.DAT	FEB 28, 1986	52*	MESO067.DAT	DEC 12, 1986
10	MESO011.DAT	MAR 5, 1986	53	MESO068.DAT	DEC 17, 1986
11	MESO012.DAT	MAR 7, 1986	54	MESO069.DAT	DEC 19, 1986
12	MESO013.DAT	MAR 12, 1986	55	MESO070.DAT	DEC 23, 1986
13	MESO014.DAT	MAR 14, 1986	56	MESO072.DAT	MAY 30, 1986
14	MESO015.DAT	MAR 19, 1986	57	MESO074.DAT	JAN 9, 1987
15	MESO016.DAT	MAR 21, 1986	58	MESO076.DAT	JAN 16, 1987
16	MESO017.DAT	MAR 26, 1986	59	MESO078.DAT	JAN 30, 1987
17	MESO019.DAT	MAR 28, 1986	60	MESO079.DAT	FEB 6, 1987
18*	MESO020.DAT	APR 2, 1986	61	MESO080.DAT	FEB 11, 1987
19	MESO021.DAT	APR 4, 1986	62	MESO081.DAT	FEB 11, 1987
20	MESO022.DAT	APR 9, 1986	63	MESO082.DAT	FEB 13, 1987
21	MESO024.DAT	APR 16, 1986	64	MESO086.DAT	FEB 27, 1987
22	MESO025.DAT	APR 18, 1986	65	MESO087.DAT	MAR 4, 1987
23	MESO026.DAT	APR 23, 1986	66	MESO088.DAT	MAR 6, 1987
24	MESO027.DAT	APR 25, 1986	67	MESO089.DAT	MAR 11, 1987
25*	MESO029.DAT	MAY 7, 1986	68	MESO091.DAT	MAR 23, 1987
26	MESO032.DAT	MAY 16, 1986	69	MESO092.DAT	MAR 25, 1987
27	MESO033.DAT	MAY 21, 1986	70	MESO093.DAT	MAR 27, 1987
28*	MESO034.DAT	MAY 23, 1986	71	MESO094.DAT	APR 1, 1987
29	MESO040.DAT	JUN 18, 1986	72*	MESO095.DAT	APR 3, 1986
30	MESO041.DAT	JUN 20, 1986	73	MESO097.DAT	APR 10, 1987
31	MESO042.DAT	JUN 25, 1986	74	MESO098.DAT	APR 14, 1987
32	MESO043.DAT	JUN 27, 1986	75	MESO099.DAT	APR 16, 1987
33	MESO044.DAT	JUL 2, 1986	76	MESO101.DAT	APR 24, 1987
34*	MESO045.DAT	JUL 9, 1986	77	MESO102.DAT	APR 28, 1987
35	MESO046.DAT	JUL 11, 1986	78	MESO103.DAT	MAY 1, 1987
36	MESO047.DAT	JUL 16, 1986	79	MESO106.DAT	MAY 12, 1987
37	MESO049.DAT	JUL 30, 1986	80	MESO107.DAT	MAY 15, 1987
38	MESO050.DAT	AUG 1, 1986	81	MESO108.DAT	MAY 19, 1987
39	MESO051.DAT	AUG 6, 1986	82	MESO109.DAT	MAY 22, 1987
40*	MESO052.DAT	AUG 8, 1986	83	MESO110.DAT	MAY 27, 1987
41	MESO053.DAT	AUG 13, 1986	84	MESO113.DAT	JUN 5, 1987
42	MESO055.DAT	AUG 22, 1986	85	MESO114.DAT	JUN 9, 1987
43	MESO056.DAT	AUG 27, 1986	86	MESO115.DAT	JUN 12, 1987

*WITHHELD FROM TRAINING SET FOR TESTING

coefficients. The new CEOFs are listed in an Appendix and are available from the authors on request. Also listed there are the 86 sets of 10 complex coefficients.

Our previous work was accomplished on a 20 MegaHertz (MHz) Intel 80386 based microcomputer with an Intel 80387 coprocessor. The 100,000 iterations required for convergence of the back-propagation C language computer program we wrote took about four days of run time. To reduce this time we incorporated a WEITEK math coprocessor into a 33 MHz 80386 microcomputer. In the previous study the neural network program was run under XENIX (a UNIX operating system for the Intel microcomputer) as a large amount of memory was required for the large data arrays of network connections (over 100,000 connections for our 50x50 input grid with 40 nodes in the network's hidden layer). For this study we used a 32 bit, 80386 specific C compiler (NDP C) which operates under the Phar Lap DOS extender. We could thus access the full RAM capability of the microcomputer (8 Megabytes for our current configuration). An added benefit is a two fold increase in computation speed over standard MSDOS (Microsoft Disk Operating System) 8088 compilers. These modifications to our computing platform decreased our training time from about one hour per 1000 iterations to about 10 minutes—a six fold increase in speed.

Also to improve efficiency (i.e., to increase the resolution without increasing the number of network connections to train) we doubled the resolution to 100x100 but confined our input to just that portion within the Gulf Stream's historical envelope. For each longitude we defined a minimum and maximum latitude that encompassed all 86 Gulf Streams at our disposal to produce an historical envelope. This procedure reduces the number of input nodes by 59 percent. Figure 9 shows the 86 Gulf Streams mapped on a local planar projection with a superposed 20-pixel (nominally 20 n.mi.), 50x50 grid. The variability here is typical of other Gulf Stream analyses (Cornillon, 1985; Halliwell and Mooers, 1983). The number of input nodes for the 100x100 resolution grid is reduced from 10,000 to 4,100 grid points. Thus, there is only a 64-percent increase in input nodes for a 100-percent increase in resolution.

The correlation coefficients we used as a measure of performance in our previous work do not provide a sense of the error in position which is, of course, the concern of the Naval operational units for whom the procedure is being developed. We will therefore employ as our measure of performance the root mean square (rms) distance between the Gulf Stream as provided by the Mesoscale Product, and the Gulf Stream constructed from the CEOF coefficients produced

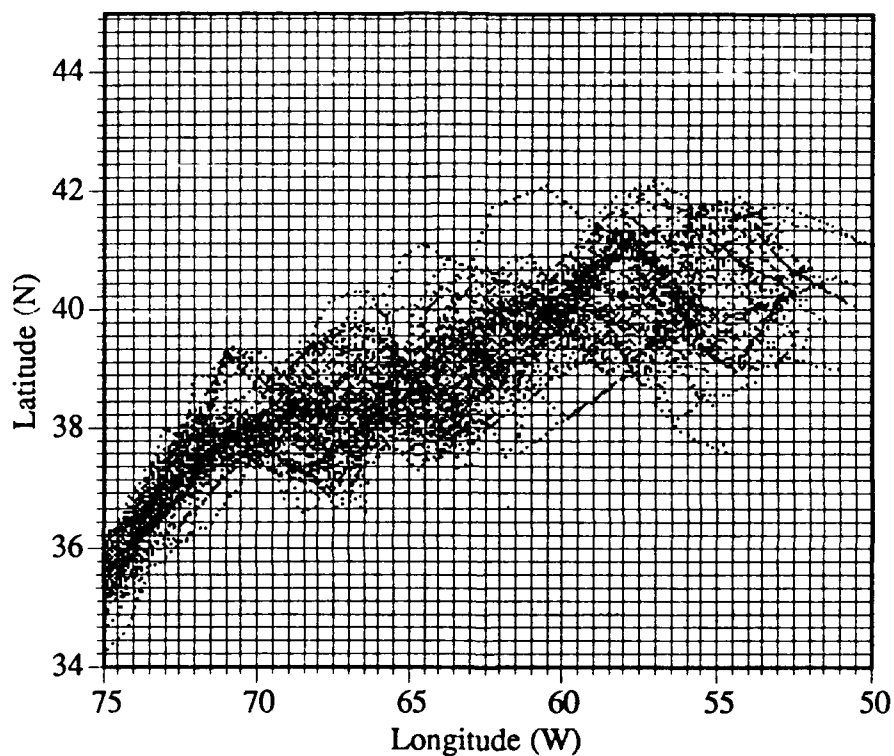


Figure 9. All 86 Gulf Streams on 20 n.mi. Grid. This plot shows the combination of all Gulf Stream positions superimposed on a 50x50 element grid representing 20 n.mi. resolution. The historical envelope is defined by all the grid points between the northernmost and southernmost positions, inclusive, of the Gulf Streams at each longitudinal position.

by the network. Since 10 n.mi. accuracy is only achieved via CEOF coefficients if ten modes are used, the fact that our network only provides three or seven modes in the cases we run precludes us from achieving 10 n.mi. accuracy. However, this measure is useful as a single standard for all networks outputting any number of mode coefficients.

As a secondary standard, we can use the rms distance between a Gulf Stream produced from network CEOF coefficients and that produced from the correct CEOF coefficients *using only the same number of modes*. This secondary measure can show how close the network gets to the theoretical limit of its performance.

We relate the secondary performance measure to the correlation coefficient performance measure for only our first case.

For our first case we trained a network with 2500 input nodes, one at each grid point with a nominal 20 n.mi. resolution, and three complex coefficients at the output nodes. We used 40 nodes in the one hidden layer. At nodes corresponding to grid points where the Mesoscale Product showed a Gulf Stream position, a value of 1.0 was assigned; at other nodes a value of 0.0 was assigned. This corresponds to the network used last by Molinelli and Flanigan (1989).

There are two differences, however, between the present case and the last one run during the previous work. First, we are employing new CEOFs and therefore new CEOF coefficients at the output nodes. There were three reasons for this—because the original computation software had an initialization error, the original Mesoscale Product set included a bad Gulf Stream vector, and the original coefficients for the 4th through 10th modes were lost.

The second difference between the present case and the last one is that the software that maps the Gulf Stream vectors to a grid also had an initialization error, and later grids contained not only the acceptable Gulf Stream vector but also all previous vectors. Table 2 shows that correlations have improved in the present work for both the training set and the test set of Gulf Streams.

Table 3 lists all the cases we ran to assess the factors that can improve the performance of the test cases. These cases showed that the CEOF coefficients of the training set were easily obtained to arbitrary accuracy depending on the number of iterations chosen. Only when fewer than 500 iterations of the back propagation algorithm were employed (case 9) did the training set CEOF coefficients differ from the correct ones by more than one percent. The performance on the

Table 2. Comparison of Base Case Correlation Coefficients

Mode Coefficient	Previous 2500-40-6 Training Set	Present 2500-40-6 Training Set (rms distance =0.1 n.mi.)	Previous 2500-40-6 Test Set	Present 2500-40-6 Test Set (rms distance =13.8 n.mi.)
1 Real	.9598	.9970	.3874	.4981
1 Imaginary	.9832	.9998	.7499	.9551
2 Real	.9584	.9975	.5866	.6491
2 Imaginary	.9675	.9997	.6433	.9758
3 Real	.9265	.9992	.7828	.0289
3 Imaginary	.8788	.9996	-.1000	.6944

The network is defined by the number of input/hidden/output nodes; hence the networks compared here have 2500 input nodes, 40 hidden nodes, and 6 output nodes (three complex coefficients). The measure of performance here is the correlation coefficient between correct CEOF coefficients and those output by the network. Our base case training set has improved from correlations of .96 to .99 just due to improved inputs and outputs, as discussed in the text. The correlation of the test set has also improved. The rms distances associated with our present network are from our secondary standard, in this case the Gulf Stream constructed with the three correct coefficients.

Table 3. Summary of Cases

Case	Resolution	Iterations	Hidden Layer Nodes	No. of Output Modes	Values of "Off" Pixels	"On" Pixels Displaced
1	20 n.mi.	20,000	40	3	0.0	No
2	20 n.mi.	5,000	40	3	0.0	No
3	10 n.mi.	5,000	40	3	0.0	No
4	10 n.mi.	5,000	80	3	0.0	No
5	10 n.mi.	3,000	80	3	0.0	No
6	10 n.mi.	2,000	80	3	0.0	No
7	10 n.mi.	5,000	80	7	0.0	No
8	10 n.mi.	1,000	80	7	0.0	No
9	10 n.mi.	500	80	7	0.0	No
10	10 n.mi.	5,000	180	7	0.0	No
11	10 n.mi.	5,000	80	7	0.1	No
12	10 n.mi.	5,000	80	7	0.1	Yes

test set by the networks of these various cases are listed in Table 4. There was little improvement obtained by adjusting these parameters of the network. Rms differences from the Mesoscale Product were about 27 n.mi., ranging from 15 to 36 n.mi., when 5000 iterations were involved during the training.

For cases with 7 coefficients at the output nodes the network performed about 2 n.mi. better than similar cases with only three output coefficients. The best performance, only about 1 n.mi. better than the next best, is obtained for case 7 which trained for 5000 iterations with 80 hidden layer nodes and with 10 n.mi. nominal resolution on the input grid (case 10 has identical results to case 7, apparently adding 100 extra nodes on the hidden layer makes no difference). Subsequent attempts (cases 11 and 12) to make the network less sensitive to clean training sets and more forgiving of slight differences with the test set did not improve performance but degraded it a little.

Table 5 shows the differences between the Gulf Streams produced by the network CEOF coefficients and those by the correct coefficients. The agreement is closer, with about 20 n.mi. rms differences, but again show not much improvement among the 12 cases. Here the low grid resolution, three mode coefficients cases (1 and 2) lie closer to their theoretical limit than do the seven mode coefficients cases.

In order to determine the degree to which the training data covered the space represented by the test Gulf Streams several statistics were computed. First, the mean, standard deviation, maximum, and minimum of each CEOF mode coefficient over all training data were computed (see Table 6, columns 2 through 5). Using these values the CEOF mode coefficients for the test Gulf Streams were subtracted from the mean and divided by the standard deviation in order to gain a measure of the normalized deviation of each mode coefficient (see Table 6, columns 6 through 14). In Table 7, the sum of the absolute value of the normalized deviations was computed for the first 6 and then the first 14 (3 and 7 complex) mode coefficients in order to obtain an overall measure of the deviation of a test Gulf Stream from the training data. Comparing the overall measurement of deviation to the results presented in Table 5 we see a strong association between a high measure of deviation and poor Gulf Stream position determination. For test cases 1 through 6 in Table 5 (which trained to 3 complex mode coefficients) we see that test Gulf Streams 5, 6, and 8 did poorly. This corresponds with the deviations for the first 6 coefficients being the largest for test sets 5, 6, and 8. Similarly, test sets 1, 2, 4, and 9 did quite well and they have correspondingly small measures of deviation. Similar results can be seen for test cases 7 through 12 in Table 5 which trained to 7 mode coefficients. This gives a strong indication that the variability of the

**Table 4. RMS Difference Between Network Gulf Stream
and Mesoscale Product (in n.mi.)**

Case	Test Gulf Stream											
	1	2	3	4	5	6	7	8	9	Min	Max	Mean
1	18.2	23.0	25.4	26.2	28.8	26.1	18.7	21.9	20.9	18.2	28.8	23.3
2	17.9	23.2	25.4	25.9	28.6	26.1	18.9	22.3	20.9	17.9	28.6	23.2
3	17.8	32.3	35.5	23.7	30.7	33.4	23.7	28.0	22.7	17.8	35.5	27.5
4	17.8	32.3	35.5	23.7	30.7	33.4	23.7	28.0	22.7	17.8	35.5	27.5
5	17.4	32.3	35.4	23.6	31.0	33.5	23.8	28.1	22.8	17.4	35.4	27.6
6	17.2	31.9	35.3	23.4	31.6	33.6	24.0	28.2	23.2	17.2	35.3	27.6
7	15.4	30.3	33.9	14.0	33.3	31.7	20.5	27.4	23.4	14.0	33.9	25.6
8	20.9	31.6	36.5	16.6	34.5	31.5	22.3	28.7	26.2	16.6	36.5	27.6
9	24.5	27.5	38.2	23.2	36.2	33.7	24.2	32.3	26.6	23.2	38.2	29.6
10	15.4	30.3	33.9	14.0	33.3	31.7	20.5	27.4	23.4	14.0	33.9	25.6
11	15.2	30.7	34.0	15.0	33.0	32.5	20.4	27.0	23.7	15.0	34.0	25.7
12	23.7	32.6	32.5	17.7	33.6	36.5	23.6	27.8	23.0	17.7	36.5	27.9

**Table 5. RMS Difference Between Network Gulf Stream
and Gulf Stream With Correct Coefficients (in n.mi.)**

Case	Test Gulf Stream											
	1	2	3	4	5	6	7	8	9	Min	Max	Mean
1	4.5	6.7	14.5	7.1	33.5	23.4	11.5	16.5	6.6	4.5	33.5	13.8
2	5.6	6.8	14.0	6.7	33.2	23.7	11.6	16.7	6.4	5.6	33.2	13.9
3	6.4	16.7	28.0	6.8	30.6	33.8	18.1	22.5	14.9	6.4	33.8	19.7
4	6.4	16.7	28.0	6.8	30.6	33.8	18.1	22.5	14.9	6.4	33.8	19.8
5	7.5	16.6	27.8	7.0	31.1	34.0	18.2	22.5	15.1	7.0	34.0	20.0
6	9.2	16.3	27.5	7.3	31.8	34.3	18.4	22.6	15.3	7.3	34.3	20.3
7	10.4	24.8	33.4	14.6	29.9	32.0	18.3	27.4	17.9	10.4	33.4	23.2
8	20.2	26.3	35.9	17.2	30.7	32.1	20.5	28.6	19.4	17.2	35.9	25.7
9	25.1	21.9	37.7	24.4	32.8	34.3	23.1	32.1	20.0	20.0	37.7	27.9
10	10.4	24.8	33.4	14.6	29.9	32.0	18.3	27.4	17.9	10.4	33.4	23.2
11	10.2	25.3	33.5	16.0	29.5	32.6	18.2	26.9	18.2	10.2	33.5	23.4
12	24.8	27.0	32.1	19.3	31.2	38.6	22.7	26.7	14.0	14.0	38.6	26.3

Table 6. CEOF Mode Coefficient Statistics*

Coeff #	$\mu - \sigma$	$\mu + \sigma$	Min	Max	TEST STREAM								
					1	2	3	4	5	6	7	8	9
1	690.93	711.37	685.21	735.29	0.21	0.09	-0.40	-1.12	1.07	-0.12	-1.05	-0.97	-0.50
2	509.09	515.12	503.79	518.88	0.78	0.30	1.05	-0.22	-0.89	-0.71	0.37	1.48	-0.27
3	-9.99	10.43	-15.72	33.39	-0.38	0.19	-0.18	-0.86	0.87	0.43	-1.09	-1.31	-0.16
4	-3.23	3.08	-6.27	9.02	<u>-2.07</u>	0.30	1.44	-0.56	-0.92	-0.66	-0.09	1.93	1.48
5	-3.86	3.76	-9.73	10.96	-0.53	1.45	0.60	0.63	1.21	-1.21	-0.21	-0.90	0.35
6	-2.39	2.62	-7.25	6.03	-0.29	-0.23	1.88	0.35	-0.94	<u>-2.95</u>	<u>-2.17</u>	1.01	0.41
7	-2.31	2.45	-6.52	6.23	-1.75	-1.16	-0.82	-0.09	-0.30	0.96	0.33	-0.25	-0.26
8	-2.66	2.63	-6.91	6.01	-0.81	0.07	-0.58	-0.35	1.57	1.37	-1.24	0.39	0.67
9	-2.21	2.35	-7.66	5.53	0.82	0.55	-1.24	0.37	-0.64	0.03	-0.37	-0.82	-0.12
10	-2.67	2.75	-6.37	6.89	-0.12	-0.35	0.14	1.67	-1.16	-0.52	0.24	-0.42	0.75
11	-1.45	1.40	-3.87	4.00	-1.07	-0.78	2.82	0.33	-1.41	-0.30	1.42	0.31	-0.26
12	-2.13	2.09	-5.81	3.72	-0.25	1.13	0.15	0.87	-1.54	1.94	0.63	-0.15	-0.70
13	-1.17	1.24	-2.49	3.29	-0.40	1.33	0.47	1.18	-1.81	-1.58	-0.42	-0.26	-0.95
14	-1.54	1.55	-4.69	3.08	-1.55	1.19	0.28	-1.20	0.61	-1.09	0.74	-0.12	0.54

* Mean +/- one Standard Deviation, Minimum and Maximum over the set of training data. Columns 6-14 list the value of each Test Gulf Stream coefficient minus the mean divided by the standard deviation. Deviations greater than 1 are bolded and deviations greater than 2 are underlined.

Table 7. Sum of Test Gulf Stream Deviations*

	TEST STREAM								
	1	2	3	4	5	6	7	8	9
Sum of $ \mu - \text{Coeff} /\sigma$ (1-6)	4.26	2.56	5.55	3.74	5.90	6.08	4.98	7.60	3.17
Sum of $ \mu - \text{Coeff} /\sigma$ (1-14)	11.03	9.12	12.05	9.80	14.94	13.87	10.37	10.32	7.42

* Sum of the absolute value of the deviations for CEOF coefficients 1-6 and 1-14.

training data affects the ability of the network to classify data it has not seen before and argues that a more comprehensive training set would significantly improve performance. Interestingly, this also shows that the network degrades gracefully since, when presented with data approaching 3 standard deviations from anything it has been trained on, it still provides a reasonable solution.

V. CONCLUSIONS AND RECOMMENDATIONS

The most significant improvement obtained by the present work is the increased efficiency in training the network achieved by removing obscuring clutter in the input image-like gridded Gulf Streams. After making that improvement, the network can be trained to within one percent of the correct CEOF coefficients in under 5000 iterations. This is a 30 fold improvement over the 150,000 iterations formerly needed, plus the 1-percent agreement is better. The Gulf Streams produced by the network for the training set lie within 1 n.mi. rms of the correct Gulf Streams with the same number of modes.

While the network so trained easily discriminates among the Gulf Streams in both the training set and the test set, the CEOF coefficients for the test set differ from the network by about 20 % from the correct values. This translates into Gulf Stream positions with 14 to 26 n.mi. rms differences from the correct Gulf Streams constructed from the same number of modes, and 23 to 30 n.mi. rms differences from the Mesoscale Product Gulf Streams.

The poorer performance of the seven mode network Gulf Stream (case 7) than the three mode network Gulf Stream (case 3) when differenced from their respective theoretical limits, may indicate that the network has not incorporated the characteristics of the higher modes as well as the lower modes (e.g., the sixth and seventh modes are not as well modeled as the second and third). This observation argues against seeking improvement by introducing more modes—at least until better agreement is obtained with the correct 7 mode Gulf Streams.

When we examine the test Gulf Streams that do most poorly compared to the correct coefficients (i.e., test Streams 5 and 6), we note that these have coefficients outside the envelope of the training set coefficients. This indicates that the training set may have too few members to adequately describe the full range of Gulf Stream shapes and positions. This suggests that the next degree of improvement will hinge on training the network with a more comprehensive set of Gulf Stream shapes. Since a more comprehensive set of Mesoscale Products are not available for the modes already defined, the extended training set can not be made of more observations. However, an arbitrary number of simulated Gulf Streams can be generated from ten modes multiplied by

complex coefficients sampled from the range of observed coefficients. An expanded set of simulated Gulf Streams could provide adequate information to train a network to better determine the appropriate coefficients.

In summary, this work has shown that neural networks can produce coefficients from which realistic Gulf Streams can be produced. The output promises to be accurate enough to use with edge labeling and optimal interpolation schemes.

VI. REFERENCES

Carter, E.F., "The Structure of the Gulf Stream as derived from an EOF analysis," Gulf Stream Workshop Proceedings, University of Rhode Island, Kingston, RI, 1985.

Cornillon, P., "Gulf Stream Envelope and Mean Path Between 75°W and 58°W," Gulf Stream Workshop Proceedings, University of Rhode Island, Kingston, RI, 1985.

Cybenko, G., "Continuous valued neural networks with two hidden layers are sufficient," Tech. Rep., Department of Computer Science, Tufts University, March 1988.

Fukushima, K., "Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition," Neural Networks, 1, 2, Pergamon Press, New York, NY, 1988.

Holyer, R.J., and S.J. Peckinpaugh, "Edge Detection Applied to Satellite Imagery of the Oceans," IEEE Trans. Geosc. Rem. Sens., 27, 1, January 1989, pp. 46-56.

Halliwell, G.R. and C.N.K. Mooers, 1983, "Meanders of the Gulf Stream downstream from Cape Hatteras 1975-1978," J. Phys. Oceanogr., 13, 7, 1275-1292.

Lybanon, M. and R. Crout, "The NORDA GEOSAT Ocean Applications Program," Johns Hopkins APL Technical Digest, 8, 2, Laurel MD, pp 212-218, June 1987.

Lippman, Richard P., "Introduction to Computing with Neural Nets," IEEE ASSP Magazine, April 1987.

Molinelli, E. and J. Hanna, "Environmental Satellites for ASW," Planning Systems Inc. Technical Report #403403, May 1988.

Molinelli, E. and M. Flanigan, "Optimized CEOF Interpolation of the Gulf Stream," Planning Systems Inc. Technical Report #392395, October 1987.

Molinelli, E. and M. Flanigan, "Preliminary Neural Network Detection of a Gulf Stream in Images of Sea Surface Temperature Gradients," Planning Systems Inc. Technical Report #477421, April 1989.

Preisendorfer, Rudolph, W., Francis W. Zwiers, and Tim P. Barnett, "Foundations of Principal Component Selection Rules," University of California, San Diego, Scripps Institution of Oceanography, La Jolla, California, SIO Reference Series #81-4, May 1984.

Rumelhart, D. and J. McClelland, Parallel Distributed Processing: Explorations in the microstructure of Cognition, Boston, The MIT Press/Bradford Books, 1986.

APPENDIX A

10 EIGENVECTORS

EIGENVECTOR NUMBER: 1

(.093856, -.017961)	(.088966, -.010408)	(.083803, -.004348)
(.093784, -.017762)	(.088828, -.010268)	(.083686, -.004194)
(.093712, -.017562)	(.088690, -.010138)	(.083566, -.004042)
(.093645, -.017360)	(.088557, -.010005)	(.083444, -.003893)
(.093578, -.017160)	(.088432, -.009874)	(.083324, -.003744)
(.093510, -.016959)	(.088308, -.009732)	(.083211, -.003597)
(.093445, -.016761)	(.088185, -.009589)	(.083089, -.003458)
(.093389, -.016562)	(.088064, -.009448)	(.082963, -.003328)
(.093331, -.016362)	(.087944, -.009312)	(.082845, -.003201)
(.093272, -.016163)	(.087833, -.009166)	(.082725, -.003070)
(.093213, -.015962)	(.087717, -.009029)	(.082598, -.002939)
(.093151, -.015761)	(.087596, -.008887)	(.082467, -.002802)
(.093092, -.015559)	(.087468, -.008753)	(.082336, -.002668)
(.093022, -.015358)	(.087339, -.008616)	(.082200, -.002533)
(.092947, -.015160)	(.087202, -.008481)	(.082071, -.002399)
(.092880, -.014962)	(.087072, -.008339)	(.081934, -.002268)
(.092810, -.014765)	(.086937, -.008205)	(.081795, -.002138)
(.092743, -.014570)	(.086805, -.008071)	(.081648, -.002011)
(.092664, -.014378)	(.086674, -.007938)	(.081499, -.001893)
(.092586, -.014184)	(.086545, -.007807)	(.081359, -.001773)
(.092502, -.013993)	(.086420, -.007682)	(.081217, -.001660)
(.092405, -.013807)	(.086305, -.007553)	(.081068, -.001548)
(.092301, -.013624)	(.086186, -.007423)	(.080920, -.001441)
(.092194, -.013444)	(.086070, -.007292)	(.080770, -.001330)
(.092078, -.013269)	(.085960, -.007160)	(.080611, -.001226)
(.091949, -.013097)	(.085853, -.007030)	(.080453, -.001133)
(.091808, -.012931)	(.085749, -.006889)	(.080300, -.001046)
(.091671, -.012767)	(.085640, -.006749)	(.080140, -.000956)
(.091527, -.012609)	(.085532, -.006612)	(.079986, -.000860)
(.091387, -.012448)	(.085426, -.006480)	(.079835, -.000770)
(.091235, -.012291)	(.085322, -.006356)	(.079679, -.000676)
(.091077, -.012142)	(.085218, -.006228)	(.079522, -.000584)
(.090918, -.011994)	(.085109, -.006105)	(.079366, -.000489)
(.090754, -.011852)	(.084999, -.005979)	(.079211, -.000389)
(.090588, -.011717)	(.084885, -.005844)	(.079053, -.000279)
(.090425, -.011585)	(.084770, -.005704)	(.078898, -.000159)
(.090256, -.011459)	(.084651, -.005562)	(.078743, -.000038)
(.090084, -.011333)	(.084541, -.005416)	(.078584, .000088)
(.089913, -.011203)	(.084440, -.005263)	(.078429, .000210)
(.089742, -.011076)	(.084340, -.005112)	(.078276, .000337)
(.089582, -.010944)	(.084233, -.004963)	(.078126, .000463)
(.089426, -.010810)	(.084133, -.004810)	(.077973, .000588)
(.089264, -.010675)	(.084032, -.004651)	(.077824, .000714)
(.089111, -.010541)	(.083918, -.004499)	(.077679, .000826)

EIGENVECTOR NUMBER: 2

(-.083393, .020494)	(-.083331, -.002402)	(.072856, -.010556)
(-.083966, .019548)	(-.081411, -.001848)	(.075704, -.009119)
(-.084451, .018679)	(-.078717, -.001184)	(.078513, -.007948)
(-.084867, .017612)	(-.075933, -.000996)	(.081381, -.007333)
(-.085260, .016376)	(-.072228, -.000201)	(.083826, -.006760)
(-.085342, .015316)	(-.068329, .002274)	(.086490, -.006532)
(-.085838, .014216)	(-.064867, .004848)	(.089222, -.006516)
(-.086123, .013290)	(-.061554, .006883)	(.091823, -.006275)
(-.086313, .012369)	(-.057234, .009291)	(.095476, -.006869)
(-.086455, .011670)	(-.053055, .012387)	(.098795, -.007524)
(-.086216, .011248)	(-.049353, .014419)	(.101635, -.008147)
(-.085665, .011307)	(-.045671, .015937)	(.103459, -.008292)
(-.084643, .011805)	(-.042209, .017487)	(.105380, -.008419)
(-.084196, .011814)	(-.038874, .018368)	(.107057, -.008702)
(-.083856, .011894)	(-.034938, .019960)	(.108521, -.008890)
(-.083587, .012374)	(-.030698, .021364)	(.109720, -.009075)
(-.083331, .013060)	(-.026865, .021412)	(.111142, -.008541)
(-.083653, .013473)	(-.023081, .021025)	(.112268, -.008297)
(-.084161, .013667)	(-.018976, .020070)	(.113363, -.008059)
(-.084563, .013911)	(-.015280, .019146)	(.114585, -.007655)
(-.085189, .014099)	(-.011735, .018209)	(.115392, -.006318)
(-.086200, .014135)	(-.008447, .017367)	(.116088, -.004956)
(-.087475, .014030)	(-.004699, .015969)	(.116525, -.003779)
(-.088932, .013776)	(-.001844, .014082)	(.117236, -.002417)
(-.090403, .012645)	(.001501, .011699)	(.118104, -.000924)
(-.091347, .011679)	(.004532, .008708)	(.118741, .000831)
(-.091879, .010146)	(.007792, .006563)	(.119942, .002524)
(-.092559, .008137)	(.011209, .004660)	(.120769, .004109)
(-.093174, .006375)	(.013792, .002405)	(.121904, .005419)
(-.093679, .004987)	(.016940, .001186)	(.123377, .006320)
(-.093327, .004329)	(.021128, -.000514)	(.124652, .007276)
(-.092442, .003526)	(.025243, -.002851)	(.125790, .008237)
(-.091649, .002571)	(.029593, -.004938)	(.126721, .008852)
(-.090684, .001998)	(.032836, -.007547)	(.126656, .010263)
(-.089664, .001103)	(.036041, -.009219)	(.126407, .011690)
(-.088694, .000534)	(.039276, -.010759)	(.126265, .012399)
(-.087644, .000202)	(.042611, -.012869)	(.126144, .013006)
(-.086624, -.000503)	(.045743, -.014137)	(.125691, .013329)
(-.086252, -.000990)	(.049178, -.015510)	(.125303, .013695)
(-.086123, -.001234)	(.052939, -.016067)	(.124786, .014551)
(-.086640, -.002254)	(.056927, -.016109)	(.124342, .014759)
(-.086829, -.002704)	(.061732, -.014699)	(.125526, .015535)
(-.086580, -.002880)	(.065777, -.013376)	(.126673, .015981)
(-.085377, -.002752)	(.069227, -.012011)	(.128026, .016623)

EIGENVECTOR NUMBER: 3

(.017483, -.008260)	(.036274, -.025097)	(-.136134, .006910)
(.018108, -.006787)	(.035302, -.027762)	(-.132177, .007275)
(.018798, -.005432)	(.033249, -.030300)	(-.128037, .008145)
(.019342, -.004181)	(.031659, -.031996)	(-.122826, .010050)
(.020887, -.003235)	(.028556, -.032280)	(-.117578, .012742)
(.022658, -.002388)	(.023014, -.031568)	(-.112981, .014623)
(.024430, -.001855)	(.017285, -.030057)	(-.104280, .016904)
(.025366, -.002926)	(.013455, -.027435)	(-.095370, .017517)
(.025347, -.004148)	(.010378, -.022776)	(-.085143, .020006)
(.024452, -.005959)	(.008151, -.017805)	(-.075401, .023108)
(.023356, -.008194)	(.005529, -.012826)	(-.062774, .026035)
(.022757, -.009929)	(.002322, -.007322)	(-.051276, .028904)
(.022672, -.011140)	(-.001406, -.000897)	(-.040065, .031649)
(.023643, -.012001)	(-.004017, .005490)	(-.029108, .033261)
(.023936, -.013153)	(-.007602, .010911)	(-.017249, .035204)
(.024094, -.014915)	(-.010469, .015559)	(-.005997, .035144)
(.025910, -.015429)	(-.015718, .019320)	(.004722, .034542)
(.027222, -.015218)	(-.021548, .023612)	(.015537, .033850)
(.028404, -.014742)	(-.029027, .027557)	(.025687, .031761)
(.029204, -.013914)	(-.037738, .032119)	(.036508, .028336)
(.030605, -.012061)	(-.047722, .037818)	(.043905, .023637)
(.032573, -.009458)	(-.056926, .043444)	(.050624, .018044)
(.033823, -.006701)	(-.065629, .047657)	(.057150, .012476)
(.034258, -.005384)	(-.073130, .051522)	(.063108, .006965)
(.033421, -.006377)	(-.080022, .055204)	(.070458, .001306)
(.031609, -.007934)	(-.088017, .056884)	(.080293, -.004659)
(.031105, -.008919)	(-.096703, .057049)	(.092103, -.010221)
(.030844, -.010843)	(-.105748, .056911)	(.102498, -.015587)
(.030527, -.013105)	(-.114230, .055918)	(.110691, -.021689)
(.030889, -.014122)	(-.121157, .054883)	(.119062, -.027459)
(.031829, -.015327)	(-.128130, .052538)	(.126337, -.031945)
(.032668, -.017694)	(-.134732, .047984)	(.132924, -.036684)
(.033841, -.019170)	(-.142147, .041506)	(.138319, -.038797)
(.035139, -.020364)	(-.147705, .035029)	(.142384, -.041225)
(.036187, -.020751)	(-.151170, .028037)	(.144623, -.043468)
(.037099, -.021359)	(-.151917, .022280)	(.144947, -.044628)
(.037246, -.021426)	(-.152285, .017925)	(.145206, -.045358)
(.037579, -.020034)	(-.153267, .015740)	(.145936, -.044074)
(.037544, -.018216)	(-.153544, .012881)	(.146022, -.040627)
(.038262, -.018418)	(-.153362, .009967)	(.144984, -.038461)
(.038074, -.020248)	(-.150167, .008140)	(.142915, -.034089)
(.037556, -.021536)	(-.146412, .006576)	(.138364, -.030047)
(.037479, -.022832)	(-.142617, .007192)	(.134003, -.025526)
(.037095, -.023091)	(-.139663, .007627)	(.127063, -.018257)

EIGENVECTOR NUMBER: 4

(.064731, -.060577)	(.006570, .047547)	(.117114, .008807)
(.067691, -.060092)	(.004161, .055856)	(.111370, .010579)
(.070773, -.059813)	(.000658, .066588)	(.105000, .011938)
(.073147, -.059606)	(-.003830, .077061)	(.097921, .013876)
(.074956, -.058840)	(-.012575, .085489)	(.090098, .015294)
(.076206, -.058178)	(-.022724, .091803)	(.082512, .016257)
(.076197, -.057282)	(-.033067, .097581)	(.071483, .016582)
(.073706, -.055399)	(-.045185, .102360)	(.059069, .016910)
(.071632, -.053810)	(-.060523, .105727)	(.043548, .015177)
(.069783, -.052024)	(-.074542, .107442)	(.027837, .012685)
(.069892, -.050859)	(-.088566, .108888)	(.012917, .008500)
(.070017, -.049562)	(-.101995, .109456)	(.000104, .005192)
(.070429, -.049468)	(-.115440, .107032)	(-.011803, .001718)
(.071848, -.050063)	(-.126483, .103112)	(-.021137, -.002488)
(.072975, -.050759)	(-.133805, .098292)	(-.030200, -.008118)
(.072756, -.050343)	(-.141319, .092291)	(-.038259, -.013272)
(.069714, -.050653)	(-.146942, .083569)	(-.047378, -.019400)
(.065803, -.050543)	(-.152566, .076569)	(-.053953, -.025554)
(.061893, -.050382)	(-.157978, .069035)	(-.060370, -.030763)
(.056795, -.049859)	(-.161723, .062529)	(-.063553, -.037722)
(.051921, -.049298)	(-.164062, .054315)	(-.063753, -.041730)
(.045275, -.047650)	(-.165453, .046431)	(-.063627, -.046055)
(.038350, -.045727)	(-.165715, .038912)	(-.061084, -.050525)
(.031260, -.043019)	(-.161632, .032577)	(-.057492, -.052467)
(.022824, -.040150)	(-.155970, .027524)	(-.052417, -.051374)
(.015440, -.036986)	(-.146993, .022133)	(-.046175, -.050441)
(.009794, -.034133)	(-.133026, .014457)	(-.038779, -.048242)
(.002592, -.032621)	(-.119112, .010972)	(-.031583, -.043844)
(-.002619, -.031465)	(-.102238, .007282)	(-.023680, -.038864)
(-.007040, -.029905)	(-.080626, .004214)	(-.012651, -.034209)
(-.008270, -.027616)	(-.058573, .000345)	(-.001393, -.029179)
(-.009005, -.024766)	(-.037229, -.002802)	(.009813, -.023795)
(-.009782, -.021715)	(-.015801, -.004970)	(.021232, -.019129)
(-.010220, -.018511)	(.006022, -.007268)	(.029919, -.015132)
(-.008282, -.015308)	(.024249, -.006920)	(.037011, -.010868)
(-.006280, -.011341)	(.041318, -.006067)	(.042436, -.007286)
(-.002327, -.007225)	(.057160, -.004472)	(.044611, -.003447)
(.000895, -.003586)	(.071734, -.002554)	(.044489, -.001509)
(.002998, .001180)	(.086156, .000093)	(.043194, -.000444)
(.006775, .008753)	(.100450, .001742)	(.041512, .002646)
(.006543, .015012)	(.111654, .002952)	(.039971, .002140)
(.006464, .022187)	(.116362, .003711)	(.036361, .002618)
(.006754, .029324)	(.118788, .005510)	(.031355, .001516)
(.007158, .038386)	(.119859, .007644)	(.027384, -.003096)

EIGENVECTOR NUMBER: 5

(-.011816, -.027653)	(-.006751, .046473)	(-.007234, .029198)
(-.012576, -.028097)	(-.011498, .049020)	(-.024416, .031511)
(-.013213, -.028661)	(-.015433, .049876)	(-.041690, .032473)
(-.013465, -.029054)	(-.019265, .051808)	(-.059655, .032841)
(-.015466, -.030012)	(-.021179, .052452)	(-.079197, .034556)
(-.017740, -.030430)	(-.022351, .053292)	(-.097251, .036797)
(-.020932, -.030867)	(-.024071, .051061)	(-.115482, .036892)
(-.022504, -.032589)	(-.023044, .046761)	(-.134483, .035611)
(-.021927, -.034300)	(-.022753, .041257)	(-.149725, .032809)
(-.021689, -.035708)	(-.021124, .035560)	(-.161495, .028846)
(-.021309, -.037779)	(-.020358, .026716)	(-.172332, .025523)
(-.021605, -.039813)	(-.019635, .017859)	(-.180195, .020656)
(-.021604, -.042019)	(-.017294, .010677)	(-.186458, .016255)
(-.022631, -.043999)	(-.013118, .004228)	(-.186305, .012557)
(-.026081, -.045215)	(-.005991, -.003275)	(-.184844, .008008)
(-.028353, -.046821)	(.002571, -.011282)	(-.180548, .002444)
(-.029630, -.046840)	(.013039, -.020423)	(-.174242, -.003976)
(-.032468, -.045840)	(.025928, -.028392)	(-.166046, -.008744)
(-.035849, -.044699)	(.040051, -.034132)	(-.155211, -.012767)
(-.036809, -.042672)	(.053739, -.039850)	(-.142641, -.015903)
(-.038799, -.038880)	(.067111, -.043138)	(-.121079, -.018409)
(-.040424, -.033969)	(.081055, -.044915)	(-.099731, -.020779)
(-.040150, -.027961)	(.093807, -.045752)	(-.078505, -.021575)
(-.039597, -.021384)	(.104551, -.043414)	(-.058972, -.021790)
(-.037355, -.014326)	(.115430, -.038957)	(-.039377, -.020828)
(-.034331, -.008830)	(.124089, -.035354)	(-.019960, -.020394)
(-.030941, -.003297)	(.126351, -.032532)	(.000349, -.019485)
(-.027107, .002682)	(.126530, -.027938)	(.018929, -.017601)
(-.022395, .008040)	(.125871, -.021389)	(.037779, -.014759)
(-.018057, .012379)	(.124484, -.015076)	(.055816, -.012684)
(-.014939, .016170)	(.122345, -.008604)	(.073104, -.010196)
(-.010759, .019209)	(.119623, -.001562)	(.088572, -.005957)
(-.005250, .022066)	(.117181, .004724)	(.102421, -.002304)
(-.000341, .024753)	(.112308, .009937)	(.112706, .000180)
(.005088, .028504)	(.106182, .014217)	(.120435, .002387)
(.008500, .031024)	(.100894, .016386)	(.122708, .003980)
(.011469, .033391)	(.092949, .016723)	(.123879, .005969)
(.013939, .035789)	(.083847, .015363)	(.123818, .008860)
(.014973, .037851)	(.074410, .015922)	(.121511, .012400)
(.014501, .038501)	(.067208, .017075)	(.119508, .015211)
(.012051, .040564)	(.057429, .018565)	(.117631, .018547)
(.007214, .043093)	(.042167, .020377)	(.115212, .017659)
(.003199, .044235)	(.026732, .022528)	(.112371, .016751)
(-.002569, .044310)	(.009971, .025418)	(.110161, .017947)

EIGENVECTOR NUMBER: 6

(.051579, -.056853)	(-.107313, .056706)	(-.092220, -.036195)
(.051495, -.055340)	(-.118101, .054122)	(-.095141, -.028335)
(.052934, -.054522)	(-.127222, .051853)	(-.096063, -.023685)
(.053187, -.053104)	(-.133893, .049364)	(-.096461, -.020193)
(.056367, -.052194)	(-.138149, .047837)	(-.093001, -.017486)
(.058524, -.050274)	(-.137099, .046835)	(-.090073, -.013617)
(.060890, -.048038)	(-.134541, .046398)	(-.081344, -.012281)
(.063464, -.046366)	(-.127904, .047320)	(-.071597, -.010611)
(.066055, -.045940)	(-.118209, .048492)	(-.059918, -.006901)
(.067512, -.045557)	(-.108840, .047948)	(-.048818, -.001209)
(.069438, -.045257)	(-.097335, .048461)	(-.034971, .006086)
(.071755, -.044475)	(-.085518, .048371)	(-.019190, .015857)
(.073030, -.043591)	(-.072496, .046944)	(-.001117, .025869)
(.076073, -.042518)	(-.060652, .044744)	(.017535, .036432)
(.080233, -.041752)	(-.048752, .043648)	(.035252, .048326)
(.082024, -.040248)	(-.035785, .041327)	(.052461, .059293)
(.084161, -.038730)	(-.023717, .039685)	(.068894, .070395)
(.084194, -.035768)	(-.009992, .038131)	(.080546, .079821)
(.084466, -.034300)	(.001119, .037684)	(.089575, .089074)
(.081880, -.032772)	(.010279, .036445)	(.094677, .096934)
(.077686, -.030727)	(.019980, .031069)	(.101336, .099496)
(.070886, -.026918)	(.031581, .028426)	(.103630, .098436)
(.062147, -.021358)	(.042214, .026774)	(.103589, .093081)
(.052559, -.015747)	(.054424, .027859)	(.101521, .087721)
(.042646, -.012435)	(.065123, .023398)	(.097121, .081209)
(.031922, -.006646)	(.074098, .015787)	(.091967, .076145)
(.022396, -.004906)	(.082652, .003019)	(.085204, .068863)
(.013910, -.003825)	(.089178, -.011904)	(.078010, .057704)
(.002434, -.005437)	(.092528, -.024499)	(.065699, .045103)
(-.007346, -.007592)	(.090400, -.036563)	(.051011, .029763)
(-.013415, -.006505)	(.085514, -.047823)	(.035842, .014843)
(-.019493, -.003615)	(.081056, -.057747)	(.019798, .000663)
(-.024564, -.001411)	(.071707, -.063492)	(.003070, -.013932)
(-.029520, .003901)	(.061545, -.069376)	(-.019412, -.026145)
(-.034328, .010243)	(.047871, -.071011)	(-.037564, -.035556)
(-.041075, .020524)	(.032977, -.073324)	(-.051146, -.043031)
(-.047667, .030596)	(.019063, -.074349)	(-.061459, -.051906)
(-.051977, .039386)	(.004724, -.072967)	(-.069281, -.058394)
(-.056318, .048416)	(-.011196, -.068495)	(-.074235, -.063206)
(-.061966, .053526)	(-.026281, -.065235)	(-.078333, -.073420)
(-.069744, .055862)	(-.043574, -.061172)	(-.083719, -.071453)
(-.078952, .057577)	(-.057831, -.055830)	(-.088967, -.073866)
(-.088130, .057781)	(-.071617, -.050637)	(-.092054, -.073746)
(-.098846, .057810)	(-.082285, -.043762)	(-.093600, -.079899)

EIGENVECTOR NUMBER: 7

(.019474, -.057302)	(-.060970, .103662)	(-.020501, -.012037)
(.020782, -.056370)	(-.064068, .089856)	(.003179, -.013432)
(.022905, -.055737)	(-.062868, .076631)	(.026622, -.016489)
(.023331, -.054092)	(-.059545, .063299)	(.047946, -.017825)
(.022739, -.052494)	(-.057985, .046011)	(.068536, -.016829)
(.021997, -.051646)	(-.055661, .032755)	(.088034, -.018951)
(.021019, -.050478)	(-.054735, .016856)	(.104508, -.014532)
(.018099, -.045841)	(-.048157, .000706)	(.118791, -.009046)
(.016872, -.044117)	(-.041402, -.013833)	(.132578, -.005252)
(.014312, -.043825)	(-.036820, -.025511)	(.137861, .002095)
(.012227, -.043677)	(-.030558, -.036980)	(.142168, .005849)
(.011848, -.042010)	(-.020810, -.047089)	(.132445, .012550)
(.014321, -.040916)	(-.014320, -.055655)	(.118693, .019625)
(.015147, -.041672)	(-.006342, -.061478)	(.100976, .023897)
(.017934, -.043445)	(.007733, -.064348)	(.079565, .028775)
(.020431, -.043112)	(.021477, -.065716)	(.055222, .031572)
(.020871, -.041076)	(.033780, -.065763)	(.029795, .034497)
(.021189, -.040616)	(.045926, -.065115)	(.003062, .036288)
(.020900, -.039285)	(.056572, -.063109)	(-.023720, .036453)
(.020508, -.035716)	(.065939, -.060735)	(-.052809, .036273)
(.021492, -.035339)	(.072912, -.057833)	(-.074455, .036587)
(.019384, -.034511)	(.076610, -.051594)	(-.096947, .033220)
(.017524, -.031370)	(.080424, -.042862)	(-.114628, .029214)
(.013493, -.028157)	(.080443, -.034919)	(-.130153, .028478)
(.011841, -.026442)	(.074448, -.027743)	(-.138389, .023914)
(.012731, -.025452)	(.064859, -.022260)	(-.142774, .013704)
(.011096, -.022343)	(.050112, -.017455)	(-.146762, .004374)
(.008180, -.020873)	(.031570, -.009608)	(-.145808, .000371)
(.008236, -.017112)	(.012423, .000349)	(-.136933, .000396)
(.009444, -.009444)	(-.008996, .001523)	(-.122575, .002165)
(.008339, .000552)	(-.031188, .009849)	(-.108452, .004694)
(.004850, .015916)	(-.049142, .008987)	(-.089333, .009992)
(.003428, .030362)	(-.068542, .014349)	(-.066009, .011699)
(.000166, .043836)	(-.088075, .022509)	(-.036682, .013821)
(-.006510, .057758)	(-.102368, .025378)	(-.009615, .015511)
(-.015321, .075188)	(-.112640, .030287)	(.019455, .012640)
(-.021830, .090469)	(-.120957, .031070)	(.047466, .008911)
(-.027441, .105591)	(-.122218, .027375)	(.074162, .004353)
(-.030379, .117511)	(-.120799, .024346)	(.102538, -.000179)
(-.034159, .124829)	(-.111866, .020607)	(.126677, -.014664)
(-.041849, .128336)	(-.099333, .017644)	(.149238, -.012099)
(-.049702, .128315)	(-.081533, .009574)	(.170157, -.018941)
(-.055357, .124973)	(-.062790, -.000081)	(.188620, -.022316)
(-.059413, .116382)	(-.042322, -.006409)	(.200931, -.030895)

EIGENVECTOR NUMBER: 8

(-.001306, .064666)	(-.039870, -.084123)	(-.048965, .085583)
(-.000959, .065701)	(-.035274, -.071757)	(-.046181, .077833)
(.000526, .065756)	(-.032243, -.046792)	(-.045036, .066306)
(.000630, .070305)	(-.027743, -.020497)	(-.043347, .055067)
(.001888, .073878)	(-.018966, .011203)	(-.042848, .040607)
(.003414, .078794)	(-.006353, .035699)	(-.040558, .021500)
(.007308, .080312)	(.005151, .062683)	(-.036146, .001281)
(.008802, .079494)	(.018041, .086538)	(-.032121, -.018715)
(.007772, .077080)	(.030405, .105917)	(-.020304, -.038345)
(.007083, .074695)	(.038211, .122989)	(-.009007, -.056615)
(.004773, .072932)	(.042794, .137899)	(-.001451, -.067139)
(.003344, .067615)	(.046669, .149685)	(.009875, -.069128)
(.003640, .061234)	(.048706, .158395)	(.018689, -.071661)
(.007582, .055583)	(.050962, .163259)	(.023895, -.071056)
(.011282, .051829)	(.053345, .158409)	(.025047, -.067990)
(.014320, .043463)	(.051245, .153194)	(.025771, -.059424)
(.019929, .033727)	(.050170, .137111)	(.026263, -.051389)
(.023131, .021587)	(.047340, .118498)	(.025364, -.042431)
(.029355, .006217)	(.041935, .101404)	(.021534, -.032426)
(.033288, -.008158)	(.034760, .080033)	(.014527, -.017698)
(.036956, -.022573)	(.029112, .053591)	(-.000486, -.006570)
(.038500, -.034259)	(.024309, .018186)	(-.014547, .007380)
(.037499, -.044426)	(.018839, -.017030)	(-.032189, .016100)
(.036889, -.054231)	(.013964, -.050959)	(-.051097, .026112)
(.034592, -.063646)	(.019526, -.083207)	(-.066254, .034878)
(.028719, -.065667)	(.023672, -.112294)	(-.081689, .041491)
(.023948, -.069766)	(.026453, -.123903)	(-.091167, .047509)
(.018822, -.077202)	(.027525, -.134251)	(-.092352, .050219)
(.010794, -.088299)	(.023796, -.134893)	(-.090502, .052807)
(.003618, -.099036)	(.024654, -.129696)	(-.084929, .053485)
(-.006922, -.099353)	(.022043, -.121169)	(-.078745, .054564)
(-.016812, -.099969)	(.020822, -.108900)	(-.066893, .052686)
(-.026624, -.101224)	(.017168, -.092426)	(-.047923, .049764)
(-.032540, -.098651)	(.011003, -.074032)	(-.029535, .041842)
(-.038539, -.099799)	(.002201, -.050217)	(-.011374, .027252)
(-.043971, -.100150)	(-.005297, -.024503)	(.010042, .016501)
(-.047680, -.099867)	(-.013124, .002019)	(.034203, .002409)
(-.051147, -.098352)	(-.018836, .025099)	(.054308, -.009063)
(-.051786, -.100119)	(-.024566, .043975)	(.072392, -.020694)
(-.053782, -.099894)	(-.029931, .059390)	(.087177, -.035159)
(-.054560, -.098474)	(-.035794, .069277)	(.099972, -.034566)
(-.052121, -.096045)	(-.041545, .079763)	(.108495, -.041832)
(-.046454, -.096643)	(-.044320, .083980)	(.116710, -.050395)
(-.044175, -.093734)	(-.047973, .087198)	(.121526, -.052667)

EIGENVECTOR NUMBER: 9

(.043120, -.035494)	(.066308, .033613)	(-.102169, .072284)
(.049612, -.036416)	(.090224, .014556)	(-.114021, .061025)
(.054492, -.035698)	(.109092, -.009752)	(-.119147, .048417)
(.059321, -.037057)	(.127447, -.034083)	(-.115866, .030272)
(.064233, -.036990)	(.138493, -.056128)	(-.106753, .014119)
(.067627, -.039419)	(.143033, -.070906)	(-.091566, -.003231)
(.068532, -.041447)	(.140527, -.080817)	(-.070761, -.016811)
(.058805, -.034700)	(.135075, -.090840)	(-.047589, -.031258)
(.048164, -.026205)	(.121593, -.095450)	(-.020646, -.040146)
(.037085, -.016105)	(.101118, -.093343)	(.002079, -.043290)
(.027686, -.010985)	(.073185, -.089966)	(.015571, -.045163)
(.020223, -.004507)	(.046745, -.086590)	(.021309, -.041417)
(.012520, .001906)	(.016143, -.077889)	(.031747, -.034080)
(.010173, .000141)	(-.014607, -.064548)	(.044654, -.028362)
(.003562, -.002248)	(-.039109, -.045837)	(.054103, -.022731)
(-.005745, -.001706)	(-.060265, -.019417)	(.060114, -.014647)
(-.015833, .001576)	(-.082591, .001548)	(.061556, -.004900)
(-.024964, .002103)	(-.098274, .018950)	(.061514, .005745)
(-.033382, .003806)	(-.108494, .034820)	(.059613, .018857)
(-.036950, .005740)	(-.115031, .051174)	(.060776, .026603)
(-.038149, .004525)	(-.118806, .062936)	(.054851, .024225)
(-.036295, .001740)	(-.116104, .068868)	(.043508, .026471)
(-.037676, .001440)	(-.105973, .065913)	(.028113, .021866)
(-.041588, .000731)	(-.086323, .058355)	(.009193, .018103)
(-.044437, .001242)	(-.066307, .044143)	(-.009711, .012351)
(-.046699, .002261)	(-.038276, .024734)	(-.022874, .000009)
(-.052930, .008233)	(-.009638, .006570)	(-.033499, -.015570)
(-.056951, .014633)	(.015040, -.014981)	(-.045585, -.026304)
(-.064765, .020128)	(.042105, -.036381)	(-.056048, -.031911)
(-.071957, .027783)	(.066936, -.049580)	(-.063337, -.036506)
(-.079560, .031875)	(.093550, -.065979)	(-.067859, -.042575)
(-.087964, .036666)	(.116098, -.076834)	(-.074472, -.046834)
(-.096016, .041970)	(.130180, -.077858)	(-.076910, -.046217)
(-.094868, .050655)	(.138819, -.076745)	(-.072096, -.044260)
(-.091302, .058621)	(.131608, -.063052)	(-.062897, -.036141)
(-.088493, .066239)	(.119494, -.041707)	(-.045533, -.025256)
(-.080274, .073333)	(.100690, -.025001)	(-.027516, -.011799)
(-.072356, .079917)	(.075610, -.003047)	(-.005595, -.000594)
(-.063728, .082333)	(.049752, .021965)	(.021344, .013132)
(-.054444, .082653)	(.014389, .046109)	(.046093, .033366)
(-.038360, .078518)	(-.017416, .068658)	(.076092, .036832)
(-.014666, .070319)	(-.045929, .081495)	(.102189, .052998)
(.008829, .064961)	(-.066146, .085669)	(.125918, .070402)
(.035582, .051039)	(-.087552, .081566)	(.149906, .070418)

EIGENVECTOR NUMBER: 10

(.047127, -.011672)	(.116632, -.022848)	(-.031713, .006423)
(.045099, -.007962)	(.127844, -.037065)	(.001305, .001042)
(.046014, -.005851)	(.128542, -.049250)	(.030741, -.001971)
(.042833, .000240)	(.119404, -.059159)	(.063365, -.002941)
(.042030, .005240)	(.107388, -.062998)	(.089987, -.002353)
(.043292, .010544)	(.092509, -.058262)	(.115396, .001809)
(.042170, .016406)	(.066378, -.050675)	(.128148, .003145)
(.043199, .021632)	(.037755, -.044465)	(.128987, .005434)
(.044403, .021208)	(.005690, -.039585)	(.116300, .014425)
(.043493, .019932)	(-.023152, -.028596)	(.100610, .020982)
(.045357, .015491)	(-.049586, -.019162)	(.077433, .023480)
(.047550, .010134)	(-.072409, -.000710)	(.054019, .023711)
(.046209, .008726)	(-.088439, .018639)	(.026523, .021811)
(.039092, .012773)	(-.105493, .039733)	(-.003075, .014152)
(.038494, .015504)	(-.105801, .050094)	(-.033580, .010158)
(.037990, .015232)	(-.101256, .058136)	(-.056386, -.000961)
(.035597, .012245)	(-.088412, .063801)	(-.071821, -.010430)
(.032281, .009114)	(-.066027, .061998)	(-.092916, -.017201)
(.027647, .006002)	(-.044460, .060663)	(-.105173, -.025562)
(.014246, .004661)	(-.015077, .057449)	(-.110397, -.033885)
(.001273, .002898)	(.018441, .048882)	(-.106660, -.039421)
(-.014319, -.002327)	(.048199, .033282)	(-.094750, -.035310)
(-.031742, -.003147)	(.078873, .016116)	(-.073367, -.027071)
(-.041619, -.003111)	(.102634, -.009339)	(-.041917, -.016052)
(-.057513, -.001363)	(.117191, -.031889)	(-.010689, -.007757)
(-.066041, .004300)	(.132306, -.051955)	(.017496, .000778)
(-.074001, .006723)	(.125709, -.055864)	(.040041, .009825)
(-.081590, .002977)	(.117322, -.057585)	(.057380, .018822)
(-.091120, -.000355)	(.104902, -.062115)	(.074699, .026786)
(-.102086, -.001821)	(.082121, -.055513)	(.087264, .034324)
(-.112656, .002804)	(.048831, -.050594)	(.100389, .036074)
(-.116355, .006192)	(.018471, -.041394)	(.109741, .039588)
(-.119823, .002892)	(-.007319, -.031490)	(.114935, .043106)
(-.119742, -.002787)	(-.031094, -.017902)	(.112084, .041073)
(-.113821, -.000617)	(-.055425, -.006828)	(.106142, .032982)
(-.102548, -.001114)	(-.082210, .001817)	(.080168, .018794)
(-.088980, -.000025)	(-.105476, .008713)	(.052121, .009317)
(-.069007, .002986)	(-.119195, .016111)	(.021112, .000208)
(-.042741, .004290)	(-.135190, .023729)	(-.012535, -.005892)
(-.014804, .002929)	(-.133236, .030378)	(-.051214, -.051555)
(.020461, .001090)	(-.124189, .033076)	(-.093046, -.022714)
(.048170, -.001899)	(-.109505, .032567)	(-.131389, -.036902)
(.073175, -.006505)	(-.089706, .023568)	(-.167073, -.043773)
(.093896, -.011618)	(-.063133, .015920)	(-.194168, -.055250)

APPENDIX B

86 SETS OF COEFFICIENTS

MODE COEFFICIENTS GULF STREAM 1

(702.963000,518.883700)
 (-3.496401,9.023292)
 (-2.336997,-2.153740)
 (3.317367,3.069364)
 (-1.700231,2.251121E-02)
 (-5.839274E-01,-1.298463)
 (-1.821355E-02,-8.541310E-01)
 (-9.145637E-01,-1.064214)
 (-1.405977E-01,-9.664470E-01)
 (-7.707310E-01,1.401969E-01)

MODE COEFFICIENTS GULF STREAM 2

(713.593200,507.799800)
 (6.966372,1.170524)
 (-4.081405,6.239426E-01)
 (-4.730963,-2.100406)
 (2.353327,1.859110)
 (1.494451,3.082035)
 (8.538086E-01,-1.091997)
 (-1.255160,2.107807E-01)
 (1.705115E-01,-2.943341)
 (2.297228E-01,5.431842E-01)

MODE COEFFICIENTS GULF STREAM 3

(703.255600,514.455600)
 (-3.662902,-6.589464)
 (-2.069695,-6.166406E-01)
 (-4.097052,-2.170532)
 (1.946534,-2.890532E-01)
 (-1.548069,-5.525108E-01)
 (-4.434157E-01,-2.391598)
 (-3.176118E-01,-8.493607E-01)
 (5.278947E-01,4.030538E-02)
 (-9.189860E-01,-8.280309E-01)

MODE COEFFICIENTS GULF STREAM 4

(703.225200,515.460800)
 (-7.443419,-2.854774)
 (-1.605353,-4.540330)
 (-1.780602,-3.856049)
 (1.844313,-3.001653)
 (-9.422248E-01,-1.147489)
 (-1.247351,-2.972438E-01)
 (-3.869081E-01,-6.852542E-02)
 (-1.982626E-01,-1.234355)
 (2.972766E-01,-1.844132)

MODE COEFFICIENTS GULF STREAM 5

(701.217600,515.916000)
 (-6.854622,-2.563072)
 (-1.092398,-3.595206)
 (-1.924155,-3.070938)
 (1.759655,-3.046493)
 (-1.799974,-1.352205)
 (-1.113620,1.231153)
 (-9.362885E-02,3.166465E-01)
 (3.444421E-01,-1.101833)
 (1.410776E-01,-2.170311)

MODE COEFFICIENTS GULF STREAM 6

(696.961400,516.654400)
 (-8.710243,-4.252588)
 (6.843429E-01,-2.050183)
 (3.713719E-02,-2.580526)
 (1.670005,-2.071541)
 (-5.073778E-01,-3.113592)
 (-1.100102E-01,9.127513E-01)
 (-1.233184,1.309060)
 (-1.229050E-01,1.005036)
 (-6.391854E-01,-2.332311E-01)

MODE COEFFICIENTS GULF STREAM 7

(695.887000,514.952400)
 (-8.493741,-6.273344)
 (4.452821E-01,-2.953732E-02)
 (-1.866338,-1.783902)
 (6.452515E-01,1.343513E-02)
 (-1.051840,-8.554311E-01)
 (-1.870130E-01,-8.440110E-01)
 (-1.863574,1.815927)
 (3.778395E-01,1.177284)
 (-1.297496,-9.661433E-01)

MODE COEFFICIENTS GULF STREAM 8

(691.428400,517.491400)
 (-10.640840,-3.873661)
 (3.480093E-01,-1.741350)
 (8.771643E-02,-3.212570E-01)
 (1.106180,-5.120583E-01)
 (-2.035252,-2.954460)
 (-4.044437E-01,1.791818)
 (-1.555961,1.375545)
 (1.700096E-01,-3.567387E-01)
 (-6.385556E-01,1.351258E-01)

MODE COEFFICIENTS GULF STREAM 9

(693.958900,517.174900)
 (-10.076790,-3.392074)
 (-1.186375,-2.875787)
 (-1.192077,-2.041348)
 (1.891879,-9.573478E-01)
 (-2.307045,-2.558914)
 (-9.215226E-01,1.616150)
 (-1.047550,9.959757E-01)
 (2.126311E-01,4.320209E-01)
 (-6.568421E-01,-3.660082E-02)

MODE COEFFICIENTS GULF STREAM 10

(693.974200,516.704700)
 (-8.658381,-3.605646)
 (1.644310,-2.493514)
 (-3.228227E-01,-1.223820)
 (4.439794E-01,-2.322437)
 (-1.160945,-1.967237)
 (-5.437487E-01,1.558061)
 (-1.411393,1.764392)
 (-2.653246E-02,-8.440588E-02)
 (-6.195560E-01,4.478160E-01)

MODE COEFFICIENTS GULF STREAM 11

(695.229000,517.224500)
 (-8.073400,-2.514964)
 (1.206995E-01,-4.352563E-01)
 (1.667611E-01,-1.266325)
 (3.578352E-01,-2.008255)
 (-4.327064E-01,-2.949644)
 (-3.177943E-01,1.523971)
 (-1.760741,1.433018)
 (-9.215638E-02,-4.186174E-01)
 (-4.268465E-01,-3.353425E-02)

MODE COEFFICIENTS GULF STREAM 12

(718.357800,510.497500)
 (12.459800,5.252330E-01)
 (-3.604871,7.145603E-01)
 (-2.202813,-5.822809E-01)
 (1.077577,3.698484)
 (-5.054855E-01,4.561712E-01)
 (1.137718,4.681557E-02)
 (-2.410167,8.106937E-01)
 (-5.972977E-01,-3.215318)
 (5.281357E-01,1.141639)

MODE COEFFICIENTS GULF STREAM 13

(715.725800,514.733700)
 (3.814767,-1.156456)
 (-1.985308,2.215222)
 (-4.970636E-01,-5.336769)
 (6.800669E-01,-2.131605E-01)
 (-1.195981,-2.855513)
 (-5.673249E-01,-4.687914)
 (4.512575E-01,-1.694313E-01)
 (1.445743E-01,-1.986873)
 (-1.153278,-1.190070)

MODE COEFFICIENTS GULF STREAM 14

(703.720200,513.471000)
 (-2.204317,-3.235945)
 (-1.186400,-1.354938)
 (1.943700,-2.459894)
 (1.872913,-1.664177)
 (2.351117,-1.957529)
 (1.618403,-2.515518E-01)
 (-1.469769,2.235325)
 (-4.790648E-01,1.871080)
 (-4.787088E-01,3.410404E-01)

MODE COEFFICIENTS GULF STREAM 15

(704.199400,515.467700)
 (9.419603E-01,-2.658927)
 (4.487805E-01,-5.809051E-04)
 (8.037076E-01,-1.491255)
 (1.924602,-2.331084)
 (3.997525,-7.310619E-01)
 (1.186678,-2.204529)
 (-1.352274,1.225559)
 (-7.787593E-01,1.268262)
 (-5.646523E-01,6.780322E-01)

MODE COEFFICIENTS GULF STREAM 16

(698.803600,511.239000)
 (-2.924368,2.932875E-01)
 (1.599858,-1.807583)
 (-5.715560E-01,-3.293107E-01)
 (-4.935038E-01,-1.996184)
 (1.963712,2.719178)
 (-4.444437E-01,2.509575)
 (-1.129547,9.789498E-01)
 (-3.994774E-01,-1.201008)
 (5.837190E-01,-8.186629E-02)

MODE COEFFICIENTS GULF STREAM 17

(697.277800,512.916100)
 (-3.929834,2.617091)
 (1.603153,-2.439452E-01)
 (-1.474545,-8.704750E-01)
 (-3.257977,7.359341E-01)
 (2.050131,3.662152)
 (3.094008E-01,-3.277218E-01)
 (-1.085316,6.898721E-01)
 (-1.078940,3.123060E-02)
 (1.101965E-01,-7.064739E-02)

MODE COEFFICIENTS GULF STREAM 18

(702.034900,513.018700)
 (2.147135,8.619984E-01)
 (5.464461,-4.631912E-01)
 (-2.692777,1.762569E-01)
 (1.316316,-8.991231E-01)
 (-1.125248,2.364334)
 (1.648538,1.836605)
 (-1.380277,8.108210E-01)
 (2.013586,8.820873E-01)
 (2.222187E-01,-1.265042)

MODE COEFFICIENTS GULF STREAM 19

(698.856000,512.992600)
 (2.649778,1.359969E-01)
 (7.839282,-1.256100)
 (-9.415748E-01,2.250745E-01)
 (-4.943928E-01,-1.817579)
 (1.491516,1.937367)
 (1.748783,8.562062E-01)
 (-1.492578,6.522793E-01)
 (9.469903E-01,1.155311)
 (4.498497E-01,-2.774667E-01)

MODE COEFFICIENTS GULF STREAM 20

(698.972800,514.009800)
 (3.120184,-5.106960E-01)
 (7.976817,1.445238)
 (-1.021337,1.654828)
 (-3.825404E-02,-3.142187E-01)
 (6.368481E-01,3.249652)
 (1.765069,-7.772012E-01)
 (1.895193,-2.690281E-01)
 (1.098053,3.789251E-01)
 (4.661914E-01,-8.489091E-01)

MODE COEFFICIENTS GULF STREAM 21

(700.222900,514.266800)
 (2.056959,2.554005)
 (6.083383,2.517537)
 (3.538621E-01,-9.231187E-02)
 (-9.575809E-01,-8.535779E-01)
 (1.569757,1.503455)
 (4.595125E-01,1.255505)
 (1.794445,5.333387E-01)
 (9.818558E-01,-9.983448E-01)
 (-1.012964,-1.366582)

MODE COEFFICIENTS GULF STREAM 22

(702.836600,514.526200)
 (5.847373,3.067340)
 (2.659712,2.960908)
 (8.118296E-01,5.730330E-01)
 (-2.318163,1.349104)
 (8.241981E-01,1.395159)
 (-1.187805,-3.841493E-01)
 (7.001916E-01,-1.576865)
 (-5.771477E-01,-2.270688)
 (-7.163742E-01,-7.727909E-01)

MODE COEFFICIENTS GULF STREAM 23

(701.970400,509.638100)
 (-2.629277,-4.515778)
 (-1.116685,4.908396)
 (-1.413741,-3.344043)
 (-1.482585,4.686128)
 (-2.118003,8.682452E-01)
 (-5.788318E-01,-8.497199E-02)
 (-1.954239,1.607635)
 (-9.517231E-01,-1.669015)
 (-9.198095E-02,-4.436314E-01)

MODE COEFFICIENTS GULF STREAM 24

(689.771700,510.536100)
 (-10.398020,-2.179247)
 (3.690616E-01,3.469826)
 (-3.111818E-01,-3.575656)
 (-1.509536,1.334688)
 (2.179791,2.354146)
 (-4.955988E-01,-1.517240E-01)
 (9.511297E-01,4.558479E-01)
 (-8.392387E-01,-1.271620)
 (-1.910008E-01,2.306689E-01)

MODE COEFFICIENTS GULF STREAM 25

(697.113800,515.279400)
 (-1.668125,4.469181)
 (2.227590,4.827870)
 (-1.870748,-1.563329)
 (-2.769660,4.250980E-01)
 (4.000587,2.867771E-01)
 (6.036326E-01,4.307971E-01)
 (-9.429684E-01,1.815333E-01)
 (6.271465E-01,1.054876E-02)
 (2.644609E-01,1.697690E-01)

MODE COEFFICIENTS GULF STREAM 26

(697.848400,517.661300)
 (-2.878277,3.073571)
 (4.709021,-8.591979E-01)
 (6.920397E-01,-1.826941)
 (2.263727,-2.091653)
 (9.555296E-01,3.531625)
 (1.992159,1.086834E-01)
 (1.058721E-01,-7.705902E-01)
 (-1.553268E-01,3.488333E-01)
 (5.673878E-01,1.134120)

MODE COEFFICIENTS GULF STREAM 27

(696.353900,513.604200)
 (-6.236855,-1.172528)
 (-6.187150E-01,-2.217064E-01)
 (-4.625672E-01,-4.179912E-01)
 (1.214875,4.964334)
 (-1.002205,7.913394E-01)
 (-2.341897E-02,5.240785E-01)
 (-7.186674E-01,1.041613)
 (1.325209,1.292488E-01)
 (-6.420894E-01,7.036077E-01)

MODE COEFFICIENTS GULF STREAM 28

(689.710200,511.452700)
 (-8.548587,-1.827275)
 (2.352555,1.006042)
 (-1.428154E-01,-9.398879E-01)
 (9.046755E-01,4.566540)
 (4.556282E-01,1.818422)
 (1.456315,-1.855497)
 (3.036347E-01,-4.700607E-01)
 (1.349062,-1.288794)
 (-2.269298E-01,8.645772E-01)

MODE COEFFICIENTS GULF STREAM 29

(692.812700,511.046500)
 (-6.293259,-6.872703E-01)
 (-2.990704,2.665602E-01)
 (1.667260,3.828961)
 (-5.880973E-01,2.019611)
 (3.197948E-01,1.291883)
 (9.919962E-02,-1.257711)
 (-1.344852E-01,3.350193E-01)
 (-2.926847E-01,7.170061E-01)
 (5.628832E-01,1.936106E-01)

MODE COEFFICIENTS GULF STREAM 30

(689.420500,512.816400)
 (-2.441973E-01,2.019613E-01)
 (4.841470E-02,3.842167)
 (3.155338E-01,4.867586)
 (3.204657,5.004608)
 (-1.960217,2.118466)
 (-2.234735E-01,-2.279065E-01)
 (4.588203E-01,6.897264E-01)
 (4.102154E-01,2.201217)
 (8.572603E-01,1.072692)

MODE COEFFICIENTS GULF STREAM 31

(704.579800,512.690200)
 (11.548210,1.212454E-02)
 (-1.318980E-01,2.068617)
 (6.228910,6.011047)
 (-2.717779,-7.721015E-01)
 (-3.046266,-7.216001E-01)
 (1.030523,3.313755E-01)
 (-1.451847,-8.842844E-01)
 (-1.222867,-5.742679E-01)
 (-1.341015,-4.477703E-01)

MODE COEFFICIENTS GULF STREAM 32

(721.177400,510.009900)
 (21.607380,-2.271178)
 (8.674854,3.188779)
 (-1.291466,3.063144)
 (-7.663263,-2.827212)
 (1.531752,-2.184958)
 (2.456755,-2.683471)
 (-1.217685E-01,1.085632)
 (2.625549E-01,-8.419424E-01)
 (1.028227,1.457402E-01)

MODE COEFFICIENTS GULF STREAM 33

(709.247900,510.074000)
 (14.807880,-3.037763)
 (6.596721,5.096736)
 (1.028500,3.465273)
 (-4.211142,-2.793808)
 (-1.773778,-9.763409E-01)
 (3.288317,3.067185)
 (-2.295747E-01,2.123661)
 (7.564377E-01,-6.061194E-01)
 (4.131922E-01,-2.849678E-01)

MODE COEFFICIENTS GULF STREAM 34

(712.114700,509.420300)
 (9.060073,-2.986905)
 (4.575575,-2.243097)
 (-6.467623E-01,4.126770)
 (-1.399709,-3.107337)
 (-2.030876,-3.276294)
 (-2.145780,9.435312E-01)
 (4.892911E-01,-1.648965)
 (5.710474E-02,-5.408979E-01)
 (3.351813E-01,-3.201642E-01)

MODE COEFFICIENTS GULF STREAM 35

(700.374500,505.439100)
 (-5.199073,-4.131497)
 (-3.112542,8.284205E-01)
 (-2.641723,-1.096364)
 (-1.322947,-2.136815)
 (-1.652065,-1.437625)
 (-2.073020,-1.570388E-01)
 (1.914986,-9.505325E-02)
 (1.588643E-02,-3.785586E-03)
 (1.527875,2.784980E-01)

MODE COEFFICIENTS GULF STREAM 36

(711.593700,507.765500)
 (5.438057,-4.184016)
 (-4.186124,-7.895800E-01)
 (-4.724351,-2.780774)
 (1.708372E-01,-2.375581)
 (-7.176620E-01,2.098064)
 (-1.363842,1.891117)
 (2.717853,-2.160302)
 (-2.577707E-01,-1.754656)
 (1.225200,6.980090E-01)

MODE COEFFICIENTS GULF STREAM 37

(716.136100,512.467700)
 (18.816370,-4.570937)
 (1.428503E-02,6.046445E-01)
 (-1.938717,-6.913051)
 (-5.567921E-01,-1.939239)
 (1.693995,1.411026)
 (-7.112700E-01,1.572717)
 (2.180555,-1.707846)
 (-5.256014E-01,1.884084)
 (1.415787E-01,1.838800E-01)

MODE COEFFICIENTS GULF STREAM 38

(721.197700,511.381400)
 (19.379900,-6.173073)
 (-2.754895,3.080118)
 (1.478106,-5.687934)
 (-3.895838E-01,-2.187170)
 (3.538652,-3.720997E-01)
 (-2.243895,-1.455086)
 (1.671247,1.157744)
 (9.166304E-01,2.079062)
 (6.127146E-02,9.129584E-01)

MODE COEFFICIENTS GULF STREAM 39

(726.541100,512.781600)
 (24.019540,-7.802523E-01)
 (-1.594685,-1.035735)
 (-2.067937,-6.282386)
 (-3.735966,-1.962984)
 (5.454220E-01,3.676263E-01)
 (-6.660357E-01,1.382978)
 (6.199601E-01,-1.831076)
 (-5.125505E-01,2.033197)
 (-2.208401E-02,1.372778)

MODE COEFFICIENTS GULF STREAM 40

(699.967100,509.961800)
 (4.577512,-2.155299)
 (-4.654486,-7.278334)
 (2.344633,3.591352)
 (1.432634E-01,-1.360885)
 (-4.424835E-01,4.069405)
 (-1.871453,-1.679634)
 (-5.294562E-01,1.325809)
 (7.927814E-01,1.727675)
 (5.333507E-01,8.512930E-01)

MODE COEFFICIENTS GULF STREAM 41

(718.121500,511.312900)
 (23.310350,2.176324)
 (-2.754539,-2.304881)
 (3.168965,3.304664)
 (-5.398859,-6.367819)
 (-3.867254,1.022754)
 (-2.489500,-9.443452E-01)
 (-4.243271E-01,-2.545360)
 (-2.345482E-01,4.865802E-01)
 (3.286265E-02,6.611174E-01)

MODE COEFFICIENTS GULF STREAM 42

(688.653000,511.554700)
 (-6.454943,1.593673)
 (1.477229,-5.360279E-01)
 (-5.159518E-01,2.220857)
 (-1.079189,2.568550)
 (-3.665636E-01,2.587675)
 (-1.225558,-1.990501)
 (9.030418E-01,-9.851233E-01)
 (2.042507E-01,1.311367)
 (7.264438E-01,9.767978E-02)

MODE COEFFICIENTS GULF STREAM 43

(693.352300,509.149100)
 (-6.244148,2.683531)
 (4.744523E-01,1.254606)
 (-2.153130,1.808316)
 (2.931555E-01,2.467328)
 (-1.018987,3.716818)
 (-1.314593,-1.408618)
 (1.832754,-1.766397)
 (-2.816620E-02,1.712604E-01)
 (4.263536E-01,1.856109E-01)

MODE COEFFICIENTS GULF STREAM 44

(696.429100,509.336400)
 (-1.881572,3.058028)
 (-4.512824E-01,2.451509E-01)
 (-1.099368E-01,4.709879)
 (-1.342872E-01,2.794807)
 (-1.871724,1.271370)
 (-1.384781,7.078860E-01)
 (2.779800E-01,3.928949E-03)
 (-1.388120E-01,8.285105E-01)
 (5.660871E-01,9.818660E-01)

MODE COEFFICIENTS GULF STREAM 45

(689.003200,510.653400)
 (-7.718359,3.156654)
 (-1.098261E-01,-6.420212E-01)
 (2.876114E-01,3.011690)
 (1.728564E-01,4.217030)
 (-7.175345E-01,3.044680)
 (-6.925988E-01,-9.085957E-01)
 (1.279833,-6.945995E-01)
 (1.262789E-01,-1.171221E-01)
 (7.896444E-01,1.244981)

MODE COEFFICIENTS GULF STREAM 46

(702.155100,510.982100)
 (2.457781,-3.713036E-01)
 (-5.030961,1.647718)
 (2.769666,1.049632)
 (1.190403,-2.598406E-01)
 (2.550363E-02,-2.006456)
 (-2.078405E-01,2.007698)
 (2.154979,1.004232)
 (-5.564111E-01,5.549840E-01)
 (4.014928E-01,-1.027147)

MODE COEFFICIENTS GULF STREAM 47

(707.609100,511.616000)
 (11.636840,-2.000851)
 (-2.971110,-5.421029E-01)
 (3.836617,-2.698543)
 (-2.948940,-8.967403E-01)
 (9.119661E-01,2.377815E-01)
 (-8.622344E-01,2.810008)
 (5.205429E-01,7.747980E-01)
 (-3.780875E-01,-1.082129)
 (-5.235347E-01,-2.141468E-01)

MODE COEFFICIENTS GULF STREAM 48

(699.686800,510.335300)
 (3.203900,7.507204E-01)
 (-2.604559,1.930095)
 (4.245921,2.653404)
 (-2.582318E-01,-5.397973E-02)
 (-1.588234,-1.265844)
 (-2.977435E-01,1.854681)
 (1.236969,8.209280E-01)
 (4.502397E-01,-1.079552)
 (6.111686E-02,-1.284795)

MODE COEFFICIENTS GULF STREAM 49

(690.411100,513.215500)
 (-10.959310,-3.513042E-01)
 (-8.655737E-01,-5.317354)
 (8.671882E-01,-3.293595)
 (-7.759424E-01,6.944557E-01)
 (1.998562,1.311899)
 (-4.640138E-01,1.137169)
 (7.042738E-01,-5.853336E-01)
 (-5.099288E-01,-1.248639)
 (6.471673E-01,-2.093506E-02)

MODE COEFFICIENTS GULF STREAM 50

(693.356100,516.189900)
 (-6.032840,4.284920)
 (-3.161356,1.524983E-01)
 (5.032875E-01,1.988444E-01)
 (5.739289E-01,2.952763)
 (2.359052,1.673235)
 (2.286534E-01,-1.495840)
 (-1.287505,4.938924E-01)
 (-3.092938E-01,1.344021E-01)
 (2.051438E-01,1.315951)

MODE COEFFICIENTS GULF STREAM 51

(696.959700,516.520600)
 (1.006232,3.569976)
 (-3.797921,1.089379)
 (3.966903,3.367511)
 (-1.849016,1.973852)
 (1.405733E-01,1.492035)
 (-5.818437E-01,-2.207985)
 (-1.689368,-2.860075E-01)
 (1.028906,8.027918E-01)
 (-4.027671E-01,-3.746769E-01)

MODE COEFFICIENTS GULF STREAM 52

(691.189500,516.578900)
 (-13.188560,6.023227)
 (-3.480597,2.642459)
 (-5.254615E-01,1.014598)
 (-1.793994,-1.104474)
 (4.230989E-01,-3.381321E-01)
 (-2.807890E-01,-1.828747E-01)
 (6.888667E-01,-1.928161E-01)
 (-7.282932E-01,6.644180E-02)
 (4.466559E-01,2.884842E-01)

MODE COEFFICIENTS GULF STREAM 53

(689.644200,515.898600)
 (-12.659550,3.162520)
 (-2.138563,8.789328E-01)
 (-1.008340,8.018392E-01)
 (-1.526047,-1.727914)
 (8.346485E-01,1.040164)
 (6.531850E-01,-1.148997E-01)
 (1.903109E-01,-1.372569)
 (-3.601877E-01,-3.682868E-02)
 (5.325193E-01,1.215985)

MODE COEFFICIENTS GULF STREAM 54

(686.849800,516.136100)
 (-13.860230,3.176141)
 (-1.741922,2.802396E-01)
 (-4.781947E-01,9.183332E-01)
 (-1.338098,-1.696519)
 (1.018361,1.269917)
 (7.237759E-01,8.648073E-04)
 (7.724047E-02,-1.644003)
 (-5.446035E-02,-1.334971E-01)
 (8.086445E-01,8.367401E-01)

MODE COEFFICIENTS GULF STREAM 55

(687.447400,515.934300)
 (-15.723040,5.842304)
 (-3.138865,1.147199)
 (-1.986583E-01,1.521503)
 (-2.072695,1.520039E-02)
 (8.818707E-01,3.170019E-01)
 (3.024432E-01,8.735221E-01)
 (-3.932930E-01,-9.368233E-01)
 (-3.625658E-01,-8.013393E-01)
 (4.659436E-01,1.583523E-01)

MODE COEFFICIENTS GULF STREAM 56

(694.962900,511.569300)
 (-2.117012,3.948773E-02)
 (3.149429,2.879222)
 (5.427880E-01,2.099094)
 (3.136863,6.890540)
 (-1.459060,9.550983E-01)
 (-4.632482E-01,-6.272370E-02)
 (-1.994404E-01,-3.544829E-01)
 (-1.469082E-01,5.561220E-01)
 (5.915532E-01,2.197934E-02)

MODE COEFFICIENTS GULF STREAM 57

(693.326800,512.324700)
 (-6.393505,-4.185627E-01)
 (-3.679273,-4.812334E-01)
 (3.616325,3.219023)
 (-1.787180,-3.549844E-01)
 (3.846209E-01,9.528198E-03)
 (-1.217869,-2.702366E-01)
 (1.762053E-01,-1.154209E-01)
 (4.615309E-02,9.673475E-01)
 (-5.477868E-01,-1.087796)

MODE COEFFICIENTS GULF STREAM 58

(691.293100,512.980800)
 (-8.247775,-4.889322E-01)
 (-4.396955,-6.796810E-01)
 (3.059243,3.251696)
 (-1.929517,3.756847E-01)
 (1.982546E-02,1.120274)
 (-1.733222,4.009512E-02)
 (6.077687E-01,4.230346E-01)
 (1.194257E-02,1.583710)
 (-2.418810E-01,-5.356544E-01)

MODE COEFFICIENTS GULF STREAM 59

(696.126400,510.509900)
 (-6.608427,-7.863241E-01)
 (4.505509E-01,-2.520097)
 (9.337415E-01,-3.997869E-01)
 (-2.090952,-1.881544)
 (6.949063E-01,1.025533)
 (-1.108817,2.478730)
 (-4.969724E-01,7.464414E-01)
 (-1.038006,-7.073680E-01)
 (-1.472931E-01,4.181948E-01)

MODE COEFFICIENTS GULF STREAM 60

(703.428000,509.754000)
 (-4.790624,5.644373E-01)
 (5.594577E-01,4.579922E-01)
 (-1.017352,-4.369150E-01)
 (-2.592566,-2.818565)
 (2.557158E-01,1.611486)
 (-1.043577,2.508664E-01)
 (-6.704720E-01,1.861838)
 (-1.685629,9.547769E-02)
 (-1.889590E-01,-2.848328E-01)

MODE COEFFICIENTS GULF STREAM 61

(704.625300,508.439400)
 (-5.896623E-01,-1.381029)
 (4.420723,4.633703E-01)
 (-1.495755,-1.239017)
 (2.432138E-01,-3.562410)
 (-5.981513E-01,2.708923)
 (-2.142830E-01,-9.002106E-01)
 (-2.583823E-01,2.192016)
 (-8.581938E-01,5.381820E-01)
 (-6.235462E-01,-7.005246E-01)

MODE COEFFICIENTS GULF STREAM 62

(704.625300,508.439400)
 (-5.896623E-01,-1.381029)
 (4.420723,4.633703E-01)
 (-1.495755,-1.239017)
 (2.432138E-01,-3.562410)
 (-5.981513E-01,2.708923)
 (-2.142830E-01,-9.002106E-01)
 (-2.583823E-01,2.192016)
 (-8.581938E-01,5.381820E-01)
 (-6.235462E-01,-7.005246E-01)

MODE COEFFICIENTS GULF STREAM 63

(704.320700,509.605700)
 (6.760390,-1.570900E-01)
 (10.961130,4.235375)
 (-2.821133,1.875071)
 (-1.732570,1.661709)
 (2.060144,-4.955516E-01)
 (-2.122898,-1.657725)
 (1.220582,1.389749)
 (4.532455E-01,-8.108236E-01)
 (-1.610963,1.386140)

MODE COEFFICIENTS GULF STREAM 64

(685.206700,511.148900)
 (-12.569920,-1.200575)
 (-9.315946E-01,1.744381)
 (7.710291E-01,1.623600)
 (-1.789025,4.320840)
 (3.932981E-02,-9.485951E-01)
 (5.263388E-01,-1.357170)
 (-5.124022E-01,-8.022326E-01)
 (4.744093E-01,1.140774E-01)
 (6.414865E-01,-5.427095E-01)

MODE COEFFICIENTS GULF STREAM 65

(691.671800,510.443400)
 (-2.018059,-4.340871)
 (8.551056,-3.727866)
 (2.001661,3.696286)
 (3.758660,-6.644680E-01)
 (-1.400769E-01,-1.865161)
 (7.977470E-01,2.356659)
 (-7.647635E-01,-9.527429E-02)
 (1.326384,2.867075E-01)
 (-1.153428,1.012704)

MODE COEFFICIENTS GULF STREAM 66

(697.852500,509.740300)
 (-1.772127,-4.886454)
 (5.606143,-4.112952)
 (5.394695E-01,1.001919)
 (3.947412,-5.338181)
 (5.594749E-01,5.473462E-01)
 (6.305885E-02,-7.373373E-01)
 (-1.316642E-01,-1.467809)
 (-5.464839E-01,1.480456)
 (8.894063E-01,9.295660E-01)

MODE COEFFICIENTS GULF STREAM 67

(691.332300,512.286500)
 (-9.982033,2.719455E-01)
 (-2.809774,7.094106E-01)
 (2.984497E-01,8.353264E-01)
 (6.318518E-01,4.068292E-01)
 (5.244652E-01,-1.330480)
 (7.243118E-01,-1.628671)
 (-5.028397E-02,-1.727535)
 (-3.593268E-01,9.913721E-01)
 (3.343371E-01,1.071706)

MODE COEFFICIENTS GULF STREAM 68

(698.021900,513.649800)
 (-3.004955,9.792199E-01)
 (-2.335829,1.641225)
 (1.994602,4.302980E-01)
 (1.090761,1.577730E-02)
 (5.207107E-01,-4.920012)
 (9.425412E-01,-1.482443)
 (5.345916E-01,-1.818415)
 (-1.041439,-5.660668E-02)
 (2.726426E-02,7.635444E-01)

MODE COEFFICIENTS GULF STREAM 69

(705.728000,511.023900)
 (-3.428928E-01,2.683789)
 (1.110708E-01,3.635587)
 (2.065750,-1.083198)
 (-5.926057E-01,-1.626019)
 (3.958386E-01,-5.807369)
 (2.666304,5.986003E-02)
 (1.697979E-01,1.978474E-01)
 (-1.346872,-3.040643E-02)
 (5.005058E-01,8.116337E-01)

MODE COEFFICIENTS GULF STREAM 70

(735.294700,503.786400)
 (33.390150,-4.286970)
 (-9.731042,-7.248786)
 (-3.991103,2.468803)
 (5.525724,4.102843)
 (-1.373652E-01,2.021789)
 (-4.145531E-01,6.837333E-01)
 (3.558155E-01,2.099859)
 (6.206024E-01,-8.010718E-01)
 (4.211870E-01,-6.796951E-01)

MODE COEFFICIENTS GULF STREAM 71

(731.469500,503.802000)
 (29.751530,-4.165913)
 (-6.399043,-4.364828)
 (-6.517061,1.083517)
 (4.350976,6.598176)
 (-1.542082,9.567368E-01)
 (7.837381E-01,1.111328)
 (8.039503E-01,1.093135)
 (-6.624457E-02,8.925886E-01)
 (-3.794785E-01,-1.526369)

MODE COEFFICIENTS GULF STREAM 72

(696.001300,511.288900)
 (-1.418428,4.578149)
 (1.291311,1.137439)
 (-5.392514E-01,1.764550)
 (-2.007778E-01,2.083289)
 (-3.908184E-01,-1.485563)
 (-1.104059,8.284492E-01)
 (2.638110,-1.031272)
 (-1.268100,-5.685430E-01)
 (8.542603E-01,-1.311862E-01)

MODE COEFFICIENTS GULF STREAM 73

(702.339900,508.800400)
 (-7.168324,4.594784)
 (-1.216614,-1.544007)
 (-1.857434E-01,-2.896730)
 (3.208241E-01,-1.513599E-01)
 (3.089426E-01,1.588204E-01)
 (1.798639E-01,-1.807674)
 (1.481420,-1.395460)
 (-5.736849E-01,-6.062310E-01)
 (-3.842419E-02,-1.357884)

MODE COEFFICIENTS GULF STREAM 74

(698.087800,510.029500)
 (-3.655498,5.970052)
 (3.079345,3.338156)
 (-3.002485E-01,-1.997541E-01)
 (-3.884063E-01,4.188681)
 (3.891334E-01,-8.848245E-01)
 (-3.533927E-01,-7.247279E-01)
 (1.550780,4.576984E-01)
 (2.866970E-02,6.145939E-01)
 (-7.739978E-02,-1.471354)

MODE COEFFICIENTS GULF STREAM 75

(703.170200,510.039600)
 (-2.389590,6.350429)
 (2.887696,3.962485)
 (-1.908565,-1.291783)
 (-2.087858E-01,4.375535)
 (5.148883E-01,-2.487374)
 (-7.578807E-01,-1.092645)
 (3.883190E-01,-2.152266E-01)
 (6.482091E-01,-9.605448E-01)
 (-3.328543E-01,2.362094E-01)
 (-4.845420E-01,-9.990865E-01)

MODE COEFFICIENTS GULF STREAM 76

(693.020800,510.932600)
 (-6.443829,4.663389)
 (2.511999,-3.356363)
 (-8.831861E-01,-1.454345)
 (9.002008E-01,-1.696190)
 (3.553440E-01,-5.802572E-01)
 (-5.813751E-01,3.077696)
 (3.534124E-01,-5.055571E-01)
 (4.785475E-01,-9.634930E-01)

MODE COEFFICIENTS GULF STREAM 77

(695.704000,511.503900)
 (-3.642094,4.285524)
 (4.527302,-2.421233)
 (-9.700813E-01,-2.986470)
 (4.105972,-3.261497)
 (-3.912188E-01,1.751071)
 (1.767669,1.094575)
 (8.530706E-01,-1.809992)
 (3.450517E-01,-3.652894E-01)
 (8.965814E-01,-1.255611)

MODE COEFFICIENTS GULF STREAM 78

(711.507800,509.153100)
 (6.750412,2.581955)
 (-6.224189,3.063689)
 (-6.059482,-1.321211)
 (4.223364,1.042850)
 (-1.299715,-5.222246E-01)
 (2.767099,-1.758723)
 (-1.666622,-3.341007)
 (1.034076,1.508454)
 (-9.515814E-01,1.646403)

MODE COEFFICIENTS GULF STREAM 79

(689.995500,512.190400)
 (-9.362789,-1.970121E-01)
 (-1.132105,-1.238698)
 (1.023467,-1.044299)
 (1.839922,2.399164)
 (-8.200145E-01,-3.199054)
 (1.064038,5.624490E-01)
 (6.477774E-01,-6.683685E-01)
 (7.284215E-01,1.640703)
 (1.712292E-01,7.556151E-01)

MODE COEFFICIENTS GULF STREAM 80

(691.715100,511.859000)
 (-9.109613,5.239965E-01)
 (-2.442183,-1.086524)
 (2.171441E-01,-5.543244E-01)
 (1.492610,2.295257)
 (2.144332E-01,-2.655033)
 (1.046658,9.487199E-01)
 (-6.307831E-01,2.571775E-01)
 (2.903532E-01,3.971664E-02)
 (-4.079652E-01,6.425191E-01)

MODE COEFFICIENTS GULF STREAM 81

(698.801500,510.443100)
 (-7.962111E-01,8.483861E-02)
 (-5.146618,-1.587457)
 (1.599951,1.054140)
 (1.102484,-8.190283E-01)
 (1.073182,-1.165346)
 (-3.436277E-01,-1.253969E-01)
 (-7.609149E-01,5.948664E-01)
 (1.382580E-01,1.903398)
 (-8.212179E-01,-5.478792E-01)

MODE COEFFICIENTS GULF STREAM 82

(708.527800,510.968000)
 (7.986053,-5.530685E-01)
 (-3.883810,1.264815)
 (3.723278,3.996322E-01)
 (1.248596E-02,-1.632993)
 (5.079728E-02,-2.829165)
 (-4.101861E-02,-3.594601)
 (9.958949E-01,-1.303069)
 (-9.124732E-01,-7.033086E-01)
 (7.276225E-02,-2.891198E-02)

MODE COEFFICIENTS GULF STREAM 83

(704.365700,512.292800)
 (10.039200,2.305541)
 (-7.828640E-01,6.026075)
 (5.519813,3.560106)
 (2.508489,-9.697664E-02)
 (-1.386046,-1.728037)
 (1.848640,1.945500E-01)
 (1.773806,2.669034E-02)
 (3.665176E-01,-4.098931E-01)
 (1.867504E-01,-6.133246E-02)

MODE COEFFICIENTS GULF STREAM 84

(696.705300,512.539600)
 (-1.947327,4.447982E-01)
 (5.999065E-02,-2.601480)
 (1.451325,-1.717963)
 (2.219032,-4.494566E-01)
 (-2.204287E-01,1.704015)
 (1.791008,-1.949455)
 (-9.418183E-02,-1.900000)
 (1.200469,1.160032)
 (-3.061041E-01,-8.391240E-01)

MODE COEFFICIENTS GULF STREAM 85

(703.581500,512.161900)
 (8.491824,-1.486138E-01)
 (9.186464E-01,-5.833287E-01)
 (3.028009,4.474676E-01)
 (-1.160125,4.124387E-01)
 (-5.327655E-01,-2.366274)
 (4.969417E-01,1.919589)
 (-4.038183E-01,-1.493148)
 (2.665846E-01,-2.557502)
 (6.539689E-01,1.538824)

MODE COEFFICIENTS GULF STREAM 86

(706.209800,511.842400)
 (8.072547,-7.039294E-01)
 (2.637011,-2.121590)
 (8.139099E-01,1.204669)
 (1.317737,2.314915)
 (6.744333E-02,-2.819751)
 (6.776850E-01,1.792463)
 (-1.959733,2.978929E-01)
 (1.555206E-02,-1.925999E-02)
 (9.498680E-02,2.657493)

THE MARR-HILDRETH OPERATOR AS AN EDDY DETECTOR

Carl Szczechowski
Scientific and Technical Staff
Naval Oceanographic Office
Stennis Space Center, MS 39522-5001

I. INTRODUCTION

Edges correspond to localized intensity gradients in digital images. Automated image interpretation systems often rely on some sort of edge detector as one of the initial processing steps. Traditionally, after significant edges have been located, they are linked together using some sort of edge-tracking algorithm. A number of different edge detectors have been introduced but each possesses an inherent weakness or weaknesses. The Marr-Hildreth (Laplacian of Gaussian) operator addresses some of the major weaknesses inherent in typical edge detection schemes. A concise comparison of edge detectors in general versus the Marr-Hildreth operator is shown in table 1.

This paper describes the Marr-Hildreth operator and its properties (section II) and then examines how this operator may be used in the detection of eddies in AVHRR infrared (IR) imagery of the sea surface (section III). Section IV presents results from sample images followed by discussion (section V) and conclusions (section VI).

II. BACKGROUND

A. The Marr-Hildreth Operator

Based on the premise that no single operator can optimally detect edges at all scales simultaneously, Marr and Hildreth (1980) gave a heuristic argument in support of an operator which would deal separately with intensity changes at different scales. The basic idea was to smooth the image to varying degrees and then to detect the intensity changes at each smoothing level. Since the smoothing filter is to reduce the range of scales over which intensity changes can take place, a general requirement for the filter is that it should be band-limited in the frequency domain. Furthermore, since intensity changes with respect to a given scale are spatially localized, the filter should be localized in the spatial domain. However, these spatial and frequency domain requirements are conflicting in that localization in space reduces localization in frequency and vice versa. The filter which optimizes the tradeoff between spatial and frequency localization is the Gaussian.

Table 1. Comparison of edge detectors in general versus the Marr-Hildreth operator.

<u>Edge detectors (general)</u>	<u>Marr-Hildreth Operator</u>
Susceptible to noise (especially small operators); edge-preserving pre-filtering needed	Attenuates high frequencies
A single detector cannot respond to gradients at different scales; pyramid schemes necessary	Deal separately with intensity changes at different scales; straightforward relationships exist between scales
Threshold and parameter optimization needed	Only one free parameter; this parameter controls scale
Tracking algorithms must deal with uneven edge strength, orientation anomalies, gap-jumping, etc.	No (initial) thresholds needed for edge strength; edges form continuous (closed) curves
Techniques are typically ad hoc and task specific	Operator is supported by physiological evidence from the human visual system

Once the smoothing is accomplished, a means of assigning representative positions to the intensity changes must be implemented. Intensity changes are characterized by a peak in the first directional derivative or a zero-crossing in the second derivative. For an exact zero-crossing location one should calculate the second derivative along the gradient but a computationally expedient, isotropic two-dimensional second derivative operator is the Laplacian. Marr and Hildreth (1980) and Torre and Poggio (1986) demonstrate the conditions when the two operators are equivalent.

Based upon the arguments above, edge detection at a given scale simply involves finding the zero-crossings in

$$E(x, y) = \nabla^2 [G(r) \otimes I(x, y)] \quad (1)$$

where ∇^2 represents the Laplacian operator, $G(r)$ is the two-dimensional Gaussian in polar coordinates, $\otimes$ represents the convolution operator, and $I(x, y)$ is the image. The derivative rule for convolutions allows us to form one two-dimensional operator, the Laplacian of Gaussian, which reduces (1) to

$$E(x, y) = \nabla^2 G(r) \otimes I(x, y) \quad (2)$$

where

$$\nabla^2 G(r) = -\frac{1}{\pi \sigma^4} \left[1 - \frac{r^2}{2\sigma^2} \right] e^{-\frac{r^2}{2\sigma^2}}. \quad (3)$$

The beauty of the technique is that the standard deviation of the Gaussian, σ , is the operator's sole free parameter and acts as the scale parameter. At a given scale, the operator has two other desirable properties. The first is that the slope of the convolution output across a zero-crossing (hereafter referred to as the "slope of the zero crossing") is proportional to the intensity gradient magnitude (Marr and Hildreth, 1980; Hildreth, 1983). Secondly, due to the Laplacian, the output of this operator has a desirable topological aspect which lends itself to edge tracking. The zero-crossings are guaranteed to form continuous closed curves (unless the zero-crossing contour intersects with the image boundary).

A perspective view of the two-dimensional operator in the space domain is shown in figure 1a. The Fourier transform magnitude for the one-dimensional equivalent of the operator is shown in figure 1b for three values of σ . Note that the operator attenuates high frequencies, is band-pass, and as the scale parameter (σ) is changed, different frequency bands are passed.

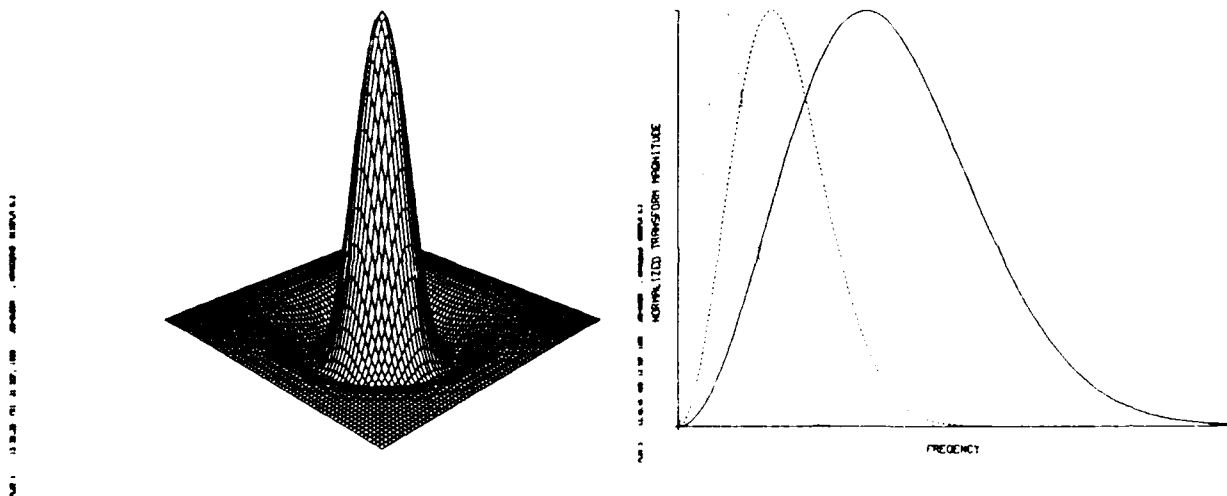


Figure 1. a. Two-dimensional Marr-Hildreth operator in the spatial domain (left). b. Fourier transform magnitude of the one-dimensional equivalent of the Marr-Hildreth operator for $\sigma = 2$ (solid), 4 (dashed), and 8 (dotted) pixels (right).

B. Analysis Across Scales

The exposition so far primarily has dealt with analyses at individual scales. Various questions can be raised about comparing or combining analyses at various scales. Is there a scale or set of scales that is optimum for detecting eddies and other significant features (e.g., fronts) while rejecting fine-scale phenomena? How can information from the various scales be combined? The first question can be addressed empirically. By implementing the operator for various values of σ , it may be possible to subjectively determine which scale or scales are optimum for the detection of significant features in the imagery. The second question is more problematic and remains unsolved within the Marr-Hildreth operator framework (Hildreth, 1983). The remainder of this section will address the latter question.

Marr and Hildreth (1980) proposed that zero-crossings which are spatially coincident at two contiguous scales are representative of "physical" edges (i.e., real boundaries as viewed by a human). This is due to the fact that a spatially localized intensity change is broad-band in the frequency domain and therefore appears in the pass band of the operator for a number of contiguous scales (compare with figure 1b).

This "spatial coincidence assumption" was more formally addressed within the framework of "scale-space filtering". The Marr-Hildreth operator evolved into the concept of scale-space filtering when it was shown that the Gaussian (in one dimension) is unique in that no new information is created as the scale (σ) of the filter increases. That is, zero-crossings are not created as the scale increases and in general, a zero-crossing theoretically may be traced continuously from smaller scales to larger scales within scale-space until it reaches an apex at some scale. This is relevant to image analysis in that one could locate an edge at some relatively large scale (where the smoothing has displaced the edge somewhat) and then work down into the smaller scales where the edge can be more finely localized. However, this tracing of zero-crossings across scale is straightforward in one dimension but difficulties can arise in two dimensions (cf. Babaud et al., 1986).

Recently, Bischof and Caelli (1988) introduced a parameter called the "spatial stability index" derived from scale-space. Assuming that spatially coincident zero-crossings from adjacent scales are indicative of a physical edge, they further proposed that the range of scales over which a zero-crossing exists at a given position is an appropriate indicator of the importance of that edge in describing the image. The spatial stability index can be thought of as a measure of "narrowness" because the more narrow a feature is, the more broad band it will be, and consequently it will appear at more contiguous scales. The same authors note that requiring the zero-crossing locations to be *identical* across scales is a stringent requirement. Allowing the zero-crossing locations to be *nearly* identical takes into account that a zero-crossing may shift position at different scales, the uncertainty in position being proportional to the scale of the operator.

C. Weaknesses of the Operator

The Marr-Hildreth operator has some shortcomings. An aspect of concern is the size of the operator. As the scale σ increases, so does the size of the operator. The largest operator of practical value used in the present study is about 60×60 . Since the size of the operator dictates the amount of computation necessary for the convolution, various speedup techniques have been proposed and a simple one will be demonstrated in section III.A.

Berzins (1984) pointed out that the operator has problems localizing some types of edges, particularly those associated with corners, sharp angles in general, and curves with large curvature. Sharp angles and corners should not be prevalent in the imagery used in this study. The effect of curvature is of some concern and will be addressed later.

The same author also demonstrated the occurrence of false edges caused by non-linearities in the image intensity field. Since these are artifacts of the operator and are "non-physical" edges, they should not be spatially coincident at contiguous scales. Thus, they should be effectively dealt with within a multi-scale framework.

D. Summary

Before moving on to the implementation section, it may be helpful to summarize the aspects of the Marr-Hildreth operator which make it a potentially effective tool in detecting significant features in sea surface IR imagery.

1. To assign edge positions, one only has to find the zero-crossings in the convolution output of the operator.
2. The standard deviation of the Gaussian, σ , is the operator's sole free parameter and acts as the scale parameter. By simply changing the scale value, the appropriate scale(s) for detection of significant thermal features can be subjectively determined.
3. At every zero-crossing position, the magnitude of the gradient is proportional to the slope of the zero-crossing. Therefore, at every zero-crossing position we can assign an "edge strength" value.
4. Edge tracking is simplified by the fact that the zero-crossings form continuous closed curves.
5. The "spatial coincidence" of zero-crossings at contiguous scales, and in particular the spatial stability index, can be used to flag potential "non-physical" edges as well as indicate the narrowness of "physical" edges.

III. IMPLEMENTATION

The goal of this study is to accurately locate eddies within AVHRR IR imagery of the sea surface. Eddies are swirling masses of water which normally have physical characteristics distinct from their surroundings. Temperature gradients across the eddy boundaries along with their elliptical shape and large size make them distinguishable in IR imagery.

It is hypothesized that the Marr-Hildreth operator may prove to be a good eddy detector because not only will it localize the thermal gradients along the boundaries, but the eddy will be encircled by a continuous closed curve of zero-crossings. In addition, the intensity gradients estimated from the slope of the zero-crossings and the spatial stability indices can be used to assign relative importance to the edges detected. An analysis across various scales is important because eddies in imagery do not always appear as compact disks on a background of different intensity. When an eddy is "isolated" (i.e., detached from the main flow and contrasting with the "background" water mass about its total perimeter), the operator should delineate it with no problem. However, eddies often entrain filaments of water about them making the delineation of their boundaries tenuous. The Marr-Hildreth operator at larger scales may be able to "blur" the image enough to make an eddy such as this appear more as a disc on a contrasting background.

The analysis was done in three stages (figure 2). The first stage implements the Marr-Hildreth operator and outputs the convolution results for all desired scales. This is followed by a process which inputs the convolution results for all the scales and localizes the zero-crossings, forms the scale-space description, and calculates the spatial stability indices. The last stage extracts, from the scale-space description, the closed curves at each individual scale, and calculates eddy parameters for each closed curve.

A. Stage 1

The direct spatial domain convolution of any two-dimensional operator of size $M \times M$ with an image requires M^2 multiplications and M^2 additions for each image pixel (neglecting effects at the image margins). However, the isotropic nature of the Marr-Hildreth operator allows for a significant decrease in the number of computations. Following the discussion of Wiejak et al. (1985), the polar coordinate form of (3) can be expanded in Cartesian coordinates as

$$\nabla^2 G(x, y) = -\frac{1}{2\pi\sigma^4} \left(1 - \frac{x^2}{\sigma^2} \right) e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} - \frac{1}{2\pi\sigma^4} e^{-\frac{x^2}{2\sigma^2}} \left(1 - \frac{y^2}{\sigma^2} \right) e^{-\frac{y^2}{2\sigma^2}}$$

The two terms above each correspond to the product of a one-dimensional Gaussian and a one-dimensional second derivative of a Gaussian. Letting g be the one-dimensional Gaussian distribution, the above becomes

$$\nabla^2 G(x, y) = g''(x)g(y) + g(x)g''(y). \quad (4)$$

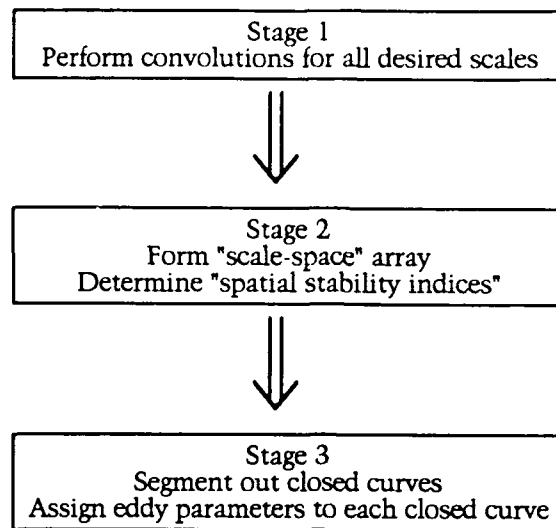


Figure 2. Stages in processing for the extraction of, and assignment of parameters to, closed curves using the Marr-Hildreth operator.

Using (4), (2) expressed as a discrete convolution becomes

$$E(x, y) = \sum_{x'} \sum_{y'} g''(x - x') g(y - y') I(x', y') + \sum_{x'} \sum_{y'} g(x - x') g''(y - y') I(x', y')$$

or

$$E(x, y) = \sum_{x'} g''(x - x') \sum_{y'} g(y - y') I(x', y') + \sum_{x'} g(x - x') \sum_{y'} g''(y - y') I(x', y'). \quad (5)$$

This equation demonstrates that the two-dimensional convolution is now reduced to four one-dimensional convolutions each requiring M additions and multiplications per image pixel (disregarding margin effects). The number of computations is thus reduced by about a factor of $M/4$ which presents a significant savings for the large operators used.

To construct the discrete versions of the continuous functions g and g'' , the continuous functions were discretely sampled at unit intervals, the values were scaled to a maximum value of 256, and the values were rounded to the nearest integer. In order for the operator to produce a zero response to a constant image, the sum of the values of the discrete version of g'' must sum to zero. This was accomplished by testing maximum scale values in the range 255.5 to 256.5 exclusive until g'' summed to zero. The resulting width (i.e., extent of non-zero values) of the operators was about 8.5σ pixels. (See Lunscher and Beddoes, 1986a and 1986b, for a complete treatment of the discrete operator design.)

The calculations in (5) are performed exclusively in integer*4 arithmetic. Assuming 8-bit imagery, using 256 as a maximum operator coefficient assures that integer*4 overflow will not occur for $\sigma \leq 4.0$ pixels. In practice, most images did not result in an overflow up to $\sigma = 7.0$ pixels.

B. Stage 2

In order to assign zero-crossing locations on a discrete grid, the "crack" edge representation of Prager (1980) is utilized. This edge representation divides the image into horizontal and vertical edge locations (figure 3). In this study, a zero-crossing is assigned to the common side of (i.e., crack between) two pixels if the convolution output values at those pixels have different signs or if only one of the values is zero.

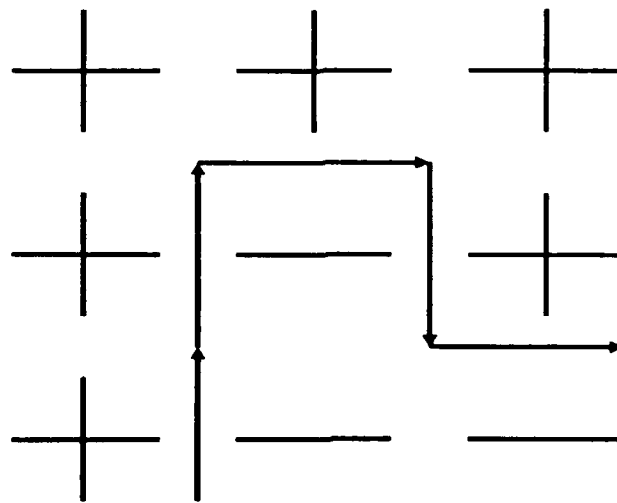


Figure 3. Crack edge representation for zero-crossing localization. A plus sign indicates a position where the convolution output is positive-valued; negative signs indicate negative values. A zero-crossing is positioned at the common side of two contiguous pixels which differ in sign. Arrowheads indicate the edge-tracking direction.

Since a zero-crossing either exists or does not exist at any one location, the zero-crossings for a given scale can be represented as a binary map. To form a compact scale-space description, then, the bit map from the smallest scale was assigned to the first bit plane of a scale-space array, the bit map from the next to the smallest scale was assigned to the second bit plane, etc. Consequently, spatial stability index values are easily derived from the scale-space array through analysis of contiguous bit values at each edge location. In deriving the spatial stability indices in this manner, the zero-crossing locations are required to be identical at contiguous scales. This is an exacting condition but it proved to be adequate in practice.

C. Stage 3

The final stage of the sequence accomplishes two things, at each scale the closed curves are segmented out of scale-space and each closed curve is assigned a set of eddy parameters. Closed curves are segmented out quite easily due to the representation of zero-crossings as crack edges (see figure 3). When tracking along a zero-crossing contour, one only has to consider the conditions at each vertex where two horizontal and two vertical crack edge positions meet. For example, suppose that we start at a vertical zero-crossing location and wish to track the contour upward. Since we are assured continuous closed curves, the zero-crossing contour must be continuable at the vertex at the top of the vertical crack (unless we are at the edge of the image). Furthermore, since zero-crossing contours normally do not intersect, there exists only one possible edge location which continues the contour. The process is repeated from this location. If the contour does not intersect the edge of the image, the process is terminated when the search returns to the original location. If at any time the edge of the image is met, the zero-crossings tracked up to that point are removed from further consideration as being part of a "non-closed" curve.

In almost all cases, not all of the closed curves found by the preceding process will correspond to what a human analyst would call an eddy. Therefore, a set of descriptive parameters, which may be useful in objectively ascertaining whether a curve represents an eddy boundary, is determined for each closed curve. Directly computable parameters include eddy type (warm/cold), area enclosed, two radius estimates, and center coordinates. Since eddies in IR imagery are characterized by the thermal gradients associated with their boundaries and by their shape, heuristics were devised which quantize these characteristics - overall edge strength and circularity. A listing of parameter values for one of the images to be discussed in section IV is shown in table 2.

Table 2. Example of calculated parameters for closed curves.

Eddy Label	Eddy Type	R _A	R _p	I _{CTR}	J _{CTR}	Overall Edge Strength	Percent of Edges Unused	Area	Circ	Rank ₃	Rank ₄
2	Cold	10	17	137	41	70	67	285	.55	1.16	1.59
6	Cold	8	10	180	45	49	64	179	.74	1.39	1.85
7	Cold	10	15	106	54	64	64	317	.69	1.60	2.06
9	Cold	6	8	78	61	200	25	131	.78	2.18	3.18
10	Cold	13	23	214	77	60	77	521	.55	1.59	1.88

Eddy type. The sign of the slope of the zero-crossing is consistent with the sign of the contrast change across the edge. Furthermore, the sign change is the same (relative to the search direction) for all crack edges along a closed curve. In addition to the sign of the contrast change, one also needs to know which side of the closed curve corresponds to the interior in order to determine whether the interior has higher or lower pixel intensities than the surroundings. Due to the search pattern employed by the edge-tracking algorithm, the sign of the contrast change and interior/exterior distinction can be determined at the first search location. Thus, the type of eddy is quickly ascertained with no recourse to the original image intensities.

Area enclosed. Once a closed curve is segmented out, the determination of the area (i.e., number of pixels) enclosed is readily accomplished by knowing the direction of the interior with respect to the crack edges.

Radius estimates. Two radius estimates are made, one based on the perimeter of the curve and the other based on the area enclosed by the curve. The radius based on perimeter is approximated via the formula for a circle:

$$R_p = \frac{N}{2\pi} \quad (6)$$

where N is the number of crack edges comprising the curve. The area-based radius also is estimated via the formula for a circle:

$$R_A = \sqrt{\frac{A}{\pi}} \quad (7)$$

where A is the area enclosed by the curve. The rationale for using formulas based on a circle soon will be made clear when the circularity heuristic is discussed.

Center coordinates. The coordinates for the center of the closed curve are given by the center of gravity of the pixels enclosed by the curve:

$$I_{CTR} = \frac{\sum_{k=1}^A I_k}{A}$$

and

$$J_{CTR} = \frac{\sum_{k=1}^A J_k}{A}$$

where I_k is the sample number and J_k the row number for the k 'th pixel enclosed by the curve and A is the area enclosed by the curve.

Overall edge strength. This heuristic is a weighted average of the slope across the zero-crossing values computed at each zero-crossing location comprising the closed curve. The weighting factor for each edge location is the spatial stability index. Thus, the narrowest gradients are given more weight. This heuristic is defined as

$$C_o = \frac{\sum_{k=1}^N SS_k \Delta E_{k\sigma}}{\sum_{k=1}^N SS_k} \quad (8)$$

where k corresponds to an edge location along the zero-crossing contour, SS_k is the spatial stability index for edge location k , $\Delta E_{k\sigma}$ is the absolute value of the slope of the zero-crossing for edge location k at the scale σ , and N is as previously defined. For simplicity, $\Delta E_{k\sigma}$ is taken as the absolute difference of the pair of convolution output values to either side of the zero-crossing. In order to reduce the possibility of including a false edge, whenever SS_k equals one a value of zero is substituted (i.e., the edge location is not used in evaluating the overall edge strength). The percentage of unused edges is included in the set of derived parameters for each closed curve. A value near 100% may indicate a "non-physical" closed curve.

Circularity. This heuristic is estimated using R_p and R_A as defined in (6) and (7):

$$circularity = \frac{R_A}{R_p}$$

A maximum value of one holds for an idealized circle, but even with area defined in square pixel units and perimeter given in crack edge units, it can be asserted that this heuristic will be larger (closer to one) for contours that are more circular.

With automated selection of the most "eddy-like" closed curves in mind, two heuristics, "Rank3" and "Rank4", were experimented with. Since eddies are identifiable due to their thermal contrast, circularity, and size, for a given scale the Rank3 parameter can be used to rank the closed curves, in a relative sense, based on the overall edge strength, circularity, and area parameters. Based on the maximum and minimum values over all curves, the three parameters for each closed curve are assigned normalized values. For example, a normalized area value is calculated as

$$area_{norm} = \frac{area - area_{min}}{area_{max} - area_{min}}$$

Rank3 is the unweighted sum of the normalized values for the three parameters. Rank4 is the Rank3 value plus a normalized value for "percentage of edges used" (i.e., $100 - \text{"percentage of edges unused"}$). The highest values of the ranking parameters should correspond to the most "eddy-like" closed curves.

IV. APPLICATION TO AVHRR IMAGERY

Nine AVHRR HRPT channel 4 images were input to the process described in section III. Eight were 512x512 pixel images from the Gulf Stream area of the North Atlantic Ocean. Both warm and cold eddies (also referred to as rings) are evident in these images. A 350x350 pixel image from the marginal ice zone area off the southeast coast of Greenland also was analyzed. Evident in this image are five cold eddies.

In their spatial stability study, Bischof and Caelli (1988) obtained satisfactory results with a 1/4 octave sampling of scale-space. Following their lead, at Stage 1 five of the images were convolved with Marr-Hildreth operators with σ ranging from 2 to 16 pixels in 1/4 octave increments - 2.0, 2.5, 3.0, 3.5, 4.0, 5.0, 6.0, 7.0, 8.0, 10.0, 12.0, 14.0, and 16.0 pixels. Based on a subjective analysis of the results from these five initial images, the remaining four images were not convolved with operators with σ less than 5.0 pixels. The results from six of the nine images will be discussed in detail.

An image containing two warm eddies is shown in the top left corner of figure 4a. The convolution output for operators with σ equal to two, four, and eight pixels are shown in the top right, bottom left, and bottom right, respectively. Positive values are shown in white, negative in black. This figure graphically demonstrates the reduction in the number of zero-crossings as the scale increases, a property noted in section II.B. The convolution output for σ equal to two pixels is striking in terms of "flow visualization". However, because of their attachment to the main flow of the Gulf Stream, the eddies are not completely encircled by curves of zero-crossings at any scale.

The birth of a cold-core ring is shown in figure 4b. In this figure, all of the zero-crossings are shown, not just those which form closed curves. At this scale (σ equal to five pixels) and for σ equal to six and seven pixels, the cold-core is totally encircled by a zero-crossing curve. Although somewhat elongated (i.e., low circularity), its size and overall edge strength make this feature the most "eddy-like" of all the closed curves based on the ranking parameters. The low contrast, eddy-like feature in the bottom right corner is not sufficiently isolated from its surroundings and therefore is not surrounded by a zero-crossing curve.

Two cold-core rings, one of which is newly-formed, are shown in figure 4c. For this figure and all subsequent figures, only the closed curves are shown, closed curves with 100% "unused edges" are eliminated and not shown, zero-crossings around warm features are displayed in green, and zero-crossings around cold features are displayed in red. The newly-formed cold-core ring has a slight attachment to the Sargasso Sea water (at least partly due to clouds) and therefore is not fully enclosed by a zero-crossing curve. The larger cold-core ring has its outer warm ring delineated as well as the cold core within for all σ equal to four pixels and larger. Out of the twenty closed curves shown in the image, the warm outer ring and cold core have the highest and fourth highest Rank₄ values, respectively. As evidenced in this image and others, cloud patches, due to their high contrast, often are associated with ranking parameter values that are relatively large.

The image in figure 4d shows a warm ring (top left), a newly-forming cold-core ring, and a cold-core ring partially occluded by clouds (bottom right). The latter ring is the only one totally circumscribed by zero-crossings, and is so delineated for σ equal to five, six, and seven pixels. The other two features are not sufficiently isolated due to trailing filaments. The partially occluded ring had the highest Rank₄ value for all closed curves not associated with cloud patches.

An eddy dipole can be seen in figure 4e. The cold-core of one of the eddies is encircled by zero-crossings for σ greater than or equal to six pixels. An extensive trailing filament presented a problem with the other half of the dipole. Except for a high contrast shingle spinoff (not discernible with the enhancement used in this figure), the cold-core feature detected was the most "eddy-like" of all the closed curves based on the ranking parameters.

An image from the marginal ice zone southeast of Greenland is shown in figure 4f. Eddies such as the ones seen in this image are believed to form under the ice and are almost always cold-core (cf. Skagseth et al., 1988). Therefore, only cold features are shown. For σ equal to six pixels (bottom image), the closed curves remaining, after elimination of those with 100% "unused edges", encircle each of the eddies discernable in the image. It is important to note that these eddies are very low contrast and are visually predominant only because of a histogram equalization enhancement based on the open-water portion of the image. Note also that the zero-crossings encircling the smallest eddies in the image are displaced a great deal from where one would subjectively place the eddy boundary. This probably is due to the effect of large curvature on the Marr-Hildreth operator noted in section III.C.

V. DISCUSSION

In terms of automated eddy detection, the simple scheme presented here works well when the eddies are sufficiently isolated. Trailing filaments and attachment to the main flow, as well as the presence of clouds, degrade performance. In the cases where eddies were completely enclosed by zero-crossing curves, the performance of the ad hoc ranking parameters, especially Rank₄, indicates potential for objectively specifying the most "eddy-like" closed curves based on easily derivable descriptive parameters.

In almost all of the cases where the process failed due to the eddies having trailing filaments or some attachment to the main flow, the zero-crossing curves would come very close to closing on themselves. In order to complete the closure, it may be possible to treat these cases as "bottlenecks" (see the paper by L. Atkinson in this proceedings). An alternative could be to input the zero-crossing locations to a circular Hough Transform algorithm such as that presently used by NOARL to automatically extract eddies from edge descriptions of images (see the overview by Lybanon and Holyer in this proceedings).

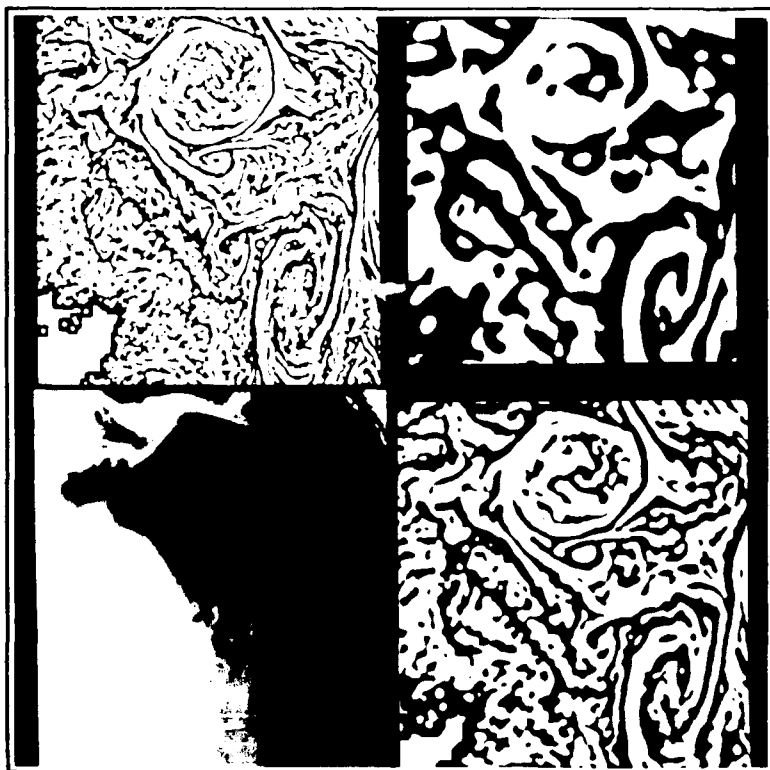


Figure 4a. Mosaic of an image with its convolution output for Marr-Hildreth operators with $\sigma = 2, 4,$ and 8 pixels. Positive values are white; negative are black.



Figure 4b. A Gulf Stream image with all zero-crossings superimposed;
 $\sigma = 5$



Figure 4c. A Gulf Stream image showing only the zero-crossings which form closed curves, cold features are red, warm features are green; $\sigma = 0$.



Figure 4d. A Gulf Stream image showing only the zero-crossings which form closed curves, cold features are red, warm features are green; $\sigma = 6$.



Figure 4e. A Gulf Stream image showing only the zero-crossings which form closed curves; cold features are red, warm features are green; $\sigma = 6$.

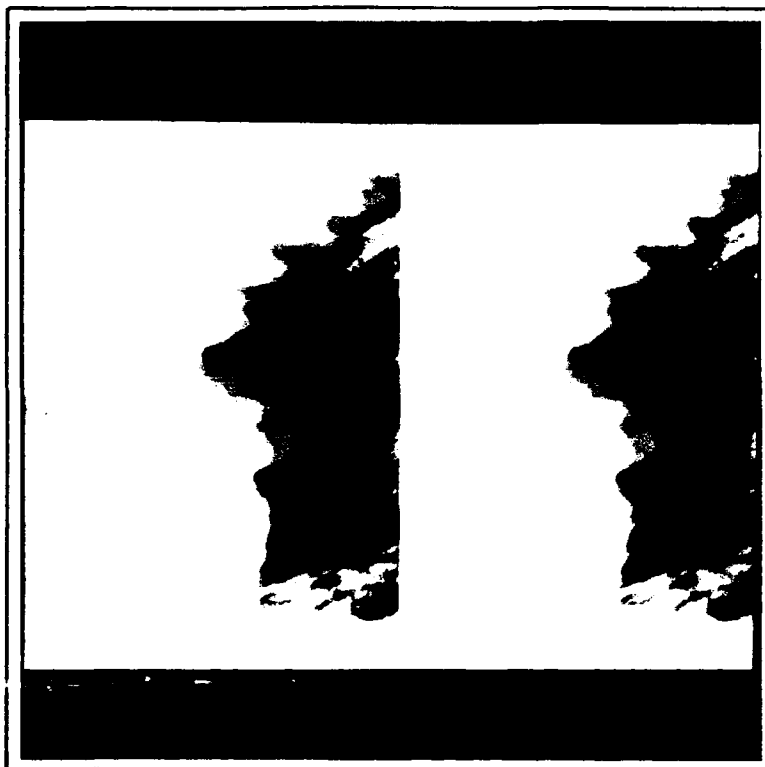


Figure 4f. Image of an area south of Greenland's ice edge with closed curves superimposed; top image used $\sigma = 5$, bottom used $\sigma = 6$.

The cloud contamination problem could be most easily addressed by using a cloud mask independently derived by one of the many cloud detection algorithms available. The mask could be applied to the zero-crossing description of the image and the zero-crossings in and near (perhaps within half the width of the Marr-Hildreth operator) cloudy areas could be eliminated. Alternatively, a "special" Marr-Hildreth operator, analogous to the "special Gaussian" referred to by L. Atkinson in this proceedings, possibly could be formed which would exclude cloud pixels from the convolution.

Emphasis in the present study is put on automated eddy detection. However, the Marr-Hildreth operator could be implemented as a general detector of significant thermal features (e.g., fronts and eddies). Note the delineation, as continuous segments, of the North Wall (top left) and South Wall of the Gulf Stream in figure 4b. In an automated sense, how can these significant features be separated from the rest of the zero-crossing segments evident in the image? Experiments were performed with a module that in effect replaces Stage 3 (figure 2) of the eddy detection scheme. Instead of segmenting out only the closed curves, all zero-crossing contours are segmented out. Each contour is assigned an overall edge strength based on (8), where N is now the total length of the contour. Preliminary results indicate that the zero-crossing segments associated with the North Wall, South Wall, and high contrast eddy boundaries have the highest overall edge strength values (outside of cloudy areas), as would be expected. In a semi-automated sense, these features could be extracted from the image based on a single threshold.

An important consideration in using zero-crossing contours to delineate sea surface features is that the Marr-Hildreth operator can result in what could be called "over-segmentation". Whereas most edge detection schemes produce a multitude of individual edge locations which then must be grouped into a lesser number of continuous edge segments, the Marr-Hildreth operator can provide continuous (and relatively long) edge segments from the outset. At times, however, it may be desirable to divide the contours into segments of more moderate length in order to extract the most significant segments. An example of this can be seen in figure 4b. At the bottom center of this image, the zero-crossing contour delineating the South Wall starts to meander within the Sargasso Sea water. Subjectively, one would like to terminate the segment at some point where the gradient begins to weaken or the edge becomes less sharp (i.e., less narrow). A possible solution could be to break up zero-crossing contours based on positions where the spatial stability index has a value of one. However, due to the present strictness in defining the spatial stability index (see section III.B.), one could lose the continuity provided by the Marr-Hildreth operator in favor of fragmenting the edge description.

Instead of a semi-automated approach based on relative edge strength, it may be possible to automatically extract the North Wall, South Wall, and significant eddies based on their expected gradient magnitude. Hildreth (1983) showed that the slope of the zero-crossing from two scales can be used to determine the absolute width and contrast of the gradient across the zero-crossing. Thus, the gradient could be expressed as

$$\frac{\Delta \text{ digital count}}{\Delta \text{ distance [pixels]}}$$

If a calibrated or MCSST image is used, $\Delta \text{ digital count}$ could be transformed to $\Delta \text{ temperature}$. Furthermore, $\Delta \text{ distance [pixels]}$ easily could be transformed to $\Delta \text{ distance [km]}$ since pixel resolution for an AVHRR image is a function of sample number. Therefore, the gradient could be transformed to

$$\frac{\Delta \text{ temperature}}{\Delta \text{ distance [km]}}$$

If a certain gradient range is expected for the North Wall or any other feature, absolute thresholds expressed in $^{\circ}\text{C/km}$ could be imposed. Since the North Wall often appears as an extremely sharp gradient, an absolute threshold on $\Delta \text{ distance [km]}$ could prove to be useful.

For each image discussed in section IV, the scales at which the eddies were satisfactorily delineated were noted. The scales common to each image were σ equal to five, six, and seven pixels. These scales also result in good delineation of the North and South Walls of the Gulf Stream. Although these scales may be optimal in terms of detecting significant features, due to the smoothing performed by the operator the zero-crossing locations may be displaced somewhat from positions chosen subjectively or those output by a small first derivative operator. This displacement could be rectified by tracing the zero-crossings through scale-space to finer scales (see section II.B). Comparisons with other edge detectors (e.g., Cayula, 1988 and Holyer and Peckinpaugh, 1989) and subjective analyses would have to be performed to see if the displacement of the zero-crossings warrant this.

VI. CONCLUSION

The major conclusions from this study are as follows:

1. The Marr-Hildreth operator provides easily segmentable edge images along with relative gradient magnitude and narrowness information without recourse to thresholds and without the need to optimize any parameters (see section II.D). In terms of extracting significant features, thresholds come into play after the segmentation is complete.
2. A set of scales (σ equal to five, six, and seven pixels) was subjectively determined which seem appropriate for the delineation of fronts and eddies in AVHRR HRPT IR imagery.
3. Via extraction of closed-curve zero-crossing contours, isolated eddies are readily detectable whereas eddies with trailing filaments and partial attachment to the main flow present difficulties.
4. Potential exists for objectively specifying the most "eddy-like" closed curves based on easily derivable descriptive parameters such as size, edge strength, and circularity.
5. A need for cloud-masking at some stage of the feature detection process is apparent.
6. With some further work (see section V), it may be possible to implement the Marr-Hildreth operator as a general edge detector which could extract significant thermal features based on absolute criteria such as gradients expressed in $^{\circ}\text{C}/\text{km}$.
7. Due to the desirable properties of the Marr-Hildreth operator (outlined in the first conclusion above), and, in particular, due to the minimal use of data-specific thresholds and parameters, the scheme described in this study should be easily adaptable to regions other than the Gulf Stream (as evidenced by the marginal ice zone image) and to other types of data (e.g., CZCS imagery).

VII. REFERENCES

- Atkinson, L. (1990). OSIRRUS: Oceanic Symbolic Image, Representation, Recognition and Understanding Software. Proceedings of the Automated Interpretation of Oceanographic Satellite Data Workshop.
- Babaud, J., A.P. Witkin, M. Baudin, and R.O. Duda (1986). Uniqueness of the Gaussian kernel for scale-space filtering. IEEE Trans. Pattern Anal. Machine Intel. 8(1):26-33.
- Berzins, V. (1984). Accuracy of Laplacian edge detectors. Computer Vision, Graphics, and Image Processing 27:195-210.
- Bischof, W.F. and T. Caelli (1988). Parsing scale-space and spatial stability analysis. Computer Vision, Graphics, and Image Processing 42:192-205.
- Cayula, J.-F. (1988). Edge detection algorithm for SST images. Master of Science thesis, Electrical Engineering Dept., University of Rhode Island.
- Hildreth, E. (1983). The detection of intensity changes by computer and biological vision systems. Computer Vision, Graphics, and Image Processing 22:1-27.
- Holyer, R.J. and S.H. Peckinpaugh (1989). Edge detection applied to satellite imagery of the oceans. IEEE Trans. Geoscience and Rem. Sens. 27(1):46-56.
- Lunscher, W.H.H.J. and M.P. Beddoes (1986). Optimal edge detector design I: parameter selection and noise effects. IEEE Trans. Pattern Anal. Machine Intel. 8(2):164-177.
- Lunscher, W.H.H.J. and M.P. Beddoes (1986). Optimal edge detector design II: coefficient quantization. IEEE Trans. Pattern Anal. Machine Intel. 8(2):178-187.
- Lybanon, M. and R.J. Holyer (1990). Historical overview of NOARL automated imagery interpretation studies. Proceedings of the Automated Interpretation of Oceanographic Satellite Data Workshop.
- Marr, D. and E. Hildreth (1980). Theory of edge detection. Proc. R. Soc. Lond. B 207:187-217.

Prager, J.M. (1980). Extracting and labeling boundary segments in natural scenes. IEEE Trans. Pattern Anal. Machine Intel. 2(1):16-27.

Skagseth, O., K. Kloster, K. Barthel, and O. Johannessen (1988). NOAA AVHRR observations during the Marginal Ice Zone Experiment, between Spitzbergen and Greenland, June 7 to July 18, 1984. Naval Environmental Prediction Research Facility, Monterey, CA. Contractor Report CR 88-10.

Torre, V. and T.A. Poggio (1986). On edge detection. IEEE Trans. Pattern Anal. Machine Intel. 8(2):147-163.

Wiejak, J.S., H. Buxton, and B.F. Buxton (1985). Convolution with separable masks for early image processing. Computer Vision, Graphics, and Image Processing 32:279-290.

APPROACHES TO CLOUD DETECTION IN AVHRR IMAGES

James Kowalski, Peter Cornillon,
Jean-François Cayula, James Gallagher,
Daniel Holloway, and Olga Slawinski
Computer Science Department
and
Graduate School of Oceanography
University of Rhode Island

I. INTRODUCTION

Advanced Very High Resolution Radiometer (AVHRR) images, produced by satellite borne passive infrared sensors, are an important and major source of data for oceanographic research. Sea surface temperature (SST) fields generated from AVHRR data are used extensively to locate oceanographic features and to study their dynamic behavior. However, before AVHRR images can be used, they require significant processing to correct for the effects of atmospheric water vapor and to identify and demarcate fog and cloud areas in the image. While errors due to many conditions can be corrected using well-known processing techniques, accurately identifying cloud and fog areas has been more difficult. A variety of algorithmic techniques have been developed to perform this task; however, current techniques are limited by the circumstances in which they can be applied and by the percentage of cloud-contaminated pixels that they can correctly identify. Hence, AVHRR images typically require additional processing by human analysts to mitigate deficiencies in current automated cloud detection techniques. However, the abundance of AVHRR data available makes it advantageous to develop improved methods of cloud detection.

At the Remote Sensing Laboratory of the Graduate School Oceanography, University of Rhode Island, we have been working on developing improved automated cloud detection techniques, approaching the problem from several different perspectives. This paper describes the bases of three different systems which we have developed or are developing and reports on preliminary results comparing these approaches with one another and against the work of a human analyst.

II. CLOUD DETECTION USING IMAGE GRADIENT

The most successful method which we have available at present was developed as part of a more extensive algorithm to detect oceanographic fronts in AVHRR images. Only the cloud detection portions of the procedure will be described here. The reader is referred to [1] for fuller details of the front detection procedure.

In outline the procedure involves the following steps:

- 1) threshold the image to remove obvious clouds;
- 2) compute the x- and y-components and the magnitude of the gradient of the image and median filter the gradient magnitude;

- 3) threshold and label areas of high gradient magnitude;
- 4) analyze high gradient areas for gradient coherence and use the following rules to identify cloud regions
 low coherence => cloud,
 high coherence => not cloud;
- 5) analyze high gradient areas not classified by the above step for elongation and use the following rules to identify cloud regions:
 low elongation => cloud,
 high elongation => not cloud;
- 6) merge small unclassified areas completely surrounded by cloud into the cloud areas.

The first step in the procedure is a simple thresholding step. As is well known, clouds are often significantly colder than water or land masses; hence, the very coldest pixels in an AVHRR image usually represent readings from cloud contaminated areas. By choosing some conservative threshold, these areas can be easily identified. In the present system, the threshold is 2° C, i.e., every pixel with a value corresponding to a temperature of 2° C or less is classified as cloud.

In the second step, the x- and y-component of the gradient together with the gradient magnitude are computed and stored for every pixel in the original image. Letting $t_{i,j}$ represent the value of a pixel at (i,j) in the image coordinate system, the x- and y- components of the gradient are defined by:

$$\nabla_x(i,j) = t_{i+1,j} - t_{i-1,j} \quad \text{and} \quad \nabla_y(i,j) = t_{i,j+1} - t_{i,j-1}.$$

Gradient magnitude is defined as expected:

$$\nabla(i,j) = ||\vec{\nabla}(i,j)||$$

After these are computed, the gradient magnitude image is median filtered using a 7 x 7 pixel window.

The gradient of an AVHRR image is, of course, a measure of how fast the temperature values are changing at a pixel. Low gradient is associated with areas spatially uniform in temperature, such as the interior surface areas of water masses. High gradient is associated with spatially rapid temperature change such as the boundary between water masses of different temperature or cloudy regions. Hence, in the third step of the procedure, thresholding is applied to the gradient magnitude image (at an empirically determined threshold of 16). Areas of high gradient magnitude likely belong to two classes: clouds or edges (fronts).

To differentiate cloud regions from edge regions, several different tests are applied to each simply connected region of pixels whose gradient is above the threshold. This requires a preliminary computation of a labeled image in which each pixel in a region of simply connected, high gradient pixels is assigned a common id number unique to that region. Thus,

after this computation, all pixels in a region will share a common id number and regions can be identified on the basis of the id numbers.

The first regional test, step four of the procedure, is based on the following rationale. The gradient vectors associated with pixels in an edge region tend to point in approximately the same direction. On the other hand, the gradient vectors associated with pixels in a cloud region tend to be randomly oriented. Some measure of coherence of gradient vector direction could thus serve to differentiate the two types of regions. Such a measure can be provided by taking the magnitude of the sum of the gradient vectors in a region and dividing it by the sum of the magnitudes of the gradient vectors in that region. In regions of high gradient coherence, this ratio tends to unity, while in regions of low gradient coherence, the ratio tends to zero because of the random orientation of vectors. More formally, if O is a simply connected region of pixels, each above the gradient magnitude threshold (16), then the gradient coherence of the region is defined as:

$$\text{coherence } \nabla = \nabla_r / \nabla_m,$$

$$\text{where } \nabla_r = \sqrt{\left(\sum_{(i,j) \in O} \nabla_x(i,j) \right)^2 + \left(\sum_{(i,j) \in O} \nabla_y(i,j) \right)^2},$$

$$\text{and } \nabla_m = \sum_{(i,j) \in O} \nabla(i,j).$$

Experimentation with several images led to the following decision rules:

$$\begin{aligned} \text{coherence } \nabla < 0.3 &\implies \text{cloud region,} \\ \text{coherence } \nabla > 0.7 &\implies \text{non-cloud region.} \end{aligned}$$

In the cases of labelled objects left undecided by the analysis of the gradient coherence parameter, an analysis of the elongation of the object is carried out to try to classify it. The rationale here is that objects of moderate gradient magnitude which are elongated are more likely to be associated with fronts than more "circular" objects which are more likely associated with clouds.

To determine an object's elongation, the spatial covariance matrix M is calculated:

$$M = \frac{1}{N} \sum_{(i,j) \in O} \begin{pmatrix} i - i_c \\ j - j_c \end{pmatrix} \cdot \begin{pmatrix} i - i_c & j - j_c \end{pmatrix},$$

where N is the number of pixels in O and (i_c, j_c) is the center of object O , given by

$$\begin{pmatrix} i_c \\ j_c \end{pmatrix} = \frac{1}{N} \sum_{(i,j) \in O} \begin{pmatrix} i \\ j \end{pmatrix}.$$

M is symmetric. Hence, it has two real eigenvalues λ_1 and λ_2 which can be ordered so that $\lambda_1 \geq \lambda_2$. If λ_1 is much larger than λ_2 , this indicates that the object, O , has an elongated profile (in the direction of the eigenvector associated with λ_1). On the other hand, if the two eigenvalues

are close in value, this indicates that the object is compact or bulky rather than elongated. Accordingly, the following decision rules (determined empirically) are used to further classify labeled objects whose gradient coherence parameter lies between 0.3 and 0.7:

$\lambda_1/\lambda_2 \leq 6 \implies$ cloud region,

$\lambda_1/\lambda_2 > 6 \implies$ non-cloud region.

The final step involves merging relatively small, unclassified areas completely surrounded by areas identified as cloud into the surrounding area. In the present algorithm, the threshold for "relatively small" is set at 600 pixels or less.

III. CLOUD DETECTION USING NEURAL NETWORKS

In developing cloud detection methods, one is normally faced with the task of identifying and characterizing the image features that can be used as a basis for classification. This can be quite difficult. One of the attractive features about neural networks in this regard is their ability to learn to classify patterns without being provided an explicit characterization of the properties which serve as the basis for the classification. Hence, one of our efforts has addressed the cloud identification problem through the use of neural networks.

Briefly, a neural net can be viewed as an network of interconnected processing elements or nodes. Each node receives weighted inputs from one or more sources and produces an output as a function of its inputs. Output values are passed on to other nodes or are used directly as one of the components of the final output of the entire network. By suitably configuring the network and adjusting the weights on the interconnections between the nodes, it is possible to have a network act as a classifier, producing an appropriate output for each class of inputs. For certain classes of networks, algorithmic procedures -- learning rules -- are known which can compute the necessary weight adjustments based on the difference between the actual output and the desired output. Training a network involves iteratively presenting it with a series of input patterns, each with its target output, and adjusting the interconnection weights according to the learning rule until the network can correctly classify all, or some predetermined proportion, of the input patterns. After training the network can be presented new patterns and the output produced can be used to classify the pattern.

Neural nets can vary in their architecture, i.e., the number of layers and nodes in the network and the topography of their interconnections. They can also vary with respect to the learning rule that is used to adjust connection weights. In our work so far, we have investigated primarily three layer, totally connected, feedforward networks as the basic network architecture. In such a network, the nodes are arranged into input and output layers, with a middle, or so-called hidden layer, interposed between. Each node in a given layer is connected to each node in the adjacent layer (totally connected) and to only those nodes (feedforward). Fixing these parameters, we have experimented by varying the total number nodes in each layer. In each case, however, backpropagation with a momentum term was used for the learning rule. [2]

In each of the different network architectures investigated, the intended output of the network was a code number in the range 0 - 4 indicating a classification of the test region -- an 8 x 8 window of pixels from an image -- into one of five categories: 100% cloudy; 75% cloudy; 50% cloudy; 25% cloudy; and 0% cloudy. This categorization was chosen rather than a binary cloudy/not-cloudy scheme because of its usefulness in other applications. When used just for cloud identification, any window classified by the network into one of the four cloud cover categories was marked as a cloud region, thus producing a binary classification.

Each network was trained on a training set 500 patterns – 100 from each of the five categories. These had been obtained by having a human analyst select and classify "by hand" different 8 x 8 pixel regions in a number of AVHRR images. An 8 x 8 pixel region was chosen as the basis for the test patterns because this seemed to be an approximate lower-bound on the size of a region for which a human analyst can make accurate estimates of cloud cover.

Although the intended function of each of the networks was to classify an 8 x 8 input window, the actual input pattern contained more elements than just the values of the 64 pixels in the test window. In fact in every case, the input patterns given to the networks consisted of the 64 pixel values in the test window together with the mean and standard deviation of the pixel values in various context windows overlapping or bordering the test window. For example, one network configuration received as inputs the 64 pixel values from the test window, the mean and standard deviation of the pixels in the test window, the mean and standard deviation of the eight 8x8-pixel windows surrounding the test window, the mean and standard deviation of the 24 x 24 pixel window centered on the test window, and the mean and standard deviation values of the eight 24 x 24 pixel windows surrounding the 24 x 24 pixel window just mentioned. Thus this network received 100 different values in the input pattern. The rationale for using mean and standard deviation values of these *context* windows was based on the observation that human analysts seem to use some sort of context information in determining cloudy areas.

Although it would have been possible, indeed preferable, to construct a network to receive the 72² individual pixel values used in the computation of the means and standard deviations mentioned above, in our case hardware and time limitations dictated a more restricted number of inputs, hence the device of collapsing a group of possible inputs into a mean and standard deviation value.

IV. KNOWLEDGE-BASED APPROACH TO CLOUD DETECTION

As was suggested in the Introduction, review by human analysts is frequently required to complete and correct declouding carried out by currently available algorithms. In fact human analysis supplementing current algorithms is still the most accurate, albeit subjective, method generally available for cloud detection in AVHRR imagery. Hence a third project in automated cloud detection that we have pursued has been motivated by a desire to try to incorporate information provided by human analysts into a knowledge-based program or expert system approach to the problem.

The breadth of information that human analysts use in cloud detection is quite diverse, involving knowledge about oceanography, meteorology, geography, seasonal variation, time scales of events, the imaging equipment, and so forth. To give some examples, sometimes an analyst will classify an area not identified by algorithmic means as cloud because "it has the wrong shape or wrong orientation to be an oceanographic event." In other cases an area might be identified as cloud "because it closely parallels a region already identified as cloud." Even more obviously, a region may be identified as cloud because it extends simultaneously over both land and water areas. In any case, however, human analysts make judgements about *regions* (contiguous groups of pixels perceived as units) in the image as opposed to individual pixels. Hence, one of the tasks in developing a knowledge-based approach is finding a suitable basis or bases for segmenting the image into regions about which classificational judgements can be made.

Development of the knowledge-based system was begun with the intent of incorporating information which would enable the system to make some decisions on the basis of relational information, for example, that smaller regions near larger cloud regions are likely cloud. In outline the system proceeds as follows:

- 1) segment the image into labeled regions;
- 2) determine size, shape, average temperature, and location of the centroid of each region;
- 3) classify the obvious regions by thresholding on average temperature;
- 4) determine region adjacencies between and among regions;
- 5) until no further classifications can be made, iterate using information obtained in steps 2), 3), 4) and 5) as well as experimentally determined thresholds, and infer that unidentified regions "sufficiently close" in average temperature and adjacent to cloud regions are cloud;
- 6) when no further classifications can be made, classify all remaining unidentified regions as cloud.

Step 1 is analogous to the labelling step described in the discussion of the gradient method. However, for this approach segmentation is based on temperature rather than gradient magnitude. More specifically, contiguous pixels are labelled as part of the same region if their pixel values fall into the same temperature interval as defined by a standard display palette. Other segmentation methods deserve study, but this was chosen initially for its simplicity and because it is based on the segmentation that human analysts perceive on the display screen.

In step 2 various information about each labelled region in the image is recorded. Of particular importance for the initial implementation is the average temperature and the location of the center of the region. The average temperature provides the basis for step 3, a simple thresholding step, in which "obvious" clouds are classified by temperature. The threshold, however, can be varied according to the geographic location of the region. In the current system, a distinction is made only between areas north of the Gulf Stream and areas south of it.

Step 4 yields a region adjacency graph, a data structure which makes it possible to identify all regions adjacent to or neighboring any given region. It is used as a basis for classificational inferences based on nearness to regions already identified as cloud.

Step 5 is an iterative procedure which implements an "islands of certainty" inferencing paradigm. That is, using "tuneable" thresholds, neighboring regions "sufficiently close" in distance and average temperature to regions already classified as cloud are likewise classified as cloud.

As is evident, the present system only begins to make use of the knowledge of human analysts. We are in the process of expanding it to enable it to make additional knowledge-based inferences.

V. PRELIMINARY EVALUATION AND COMPARISONS

For evaluational purposes, each of the systems described above was used to decloud a set of thirty-five 512 x 512 AVHRR images of the western North Atlantic which previously had been declouded by a human analyst. Pixel-by-pixel comparisons between the results produced by each system and the results produced by the human analyst as well as comparisons between each system were tabulated. The results are given in Tables 1 and 2. Table 1 shows the probability that the two methods being compared agree in classification (cloud or water) for each pixel. As can be seen, the gradient method yields the highest probability of agreement, 0.90, with the human analyst.

P(agree(row,col))	K-B System	Neural Network	Gradient Method
Neural Network	.74	X	X
Gradient Method	.76	.87	X
Human Analyst	.81	.84	.90

Table 1: Comparison of all four of the declouding techniques using probability of agreement for a single pixel ($N=35 \times (512 \times 512)$).

Table 2 provides a measure of misclassification produced by the systems by giving the conditional probabilities $P(W_{\text{program}} | C_{\text{analyst}})$ and $P(C_{\text{program}} | W_{\text{analyst}})$. $P(W_{\text{program}} | C_{\text{analyst}})$ is the probability that the method will classify a pixel as water given that the human analyst has classified it as cloud. $P(C_{\text{program}} | W_{\text{analyst}})$ is the probability that the program will classify a pixel as cloud given that the human analyst has classified it as water. Pixels classified as cloud are routinely excluded from further use for most oceanographic purposes; hence, $P(W_{\text{program}} | C_{\text{analyst}})$ is the more significant measure since it indicates the number of erroneous values that could be included in data which is ultimately used. On the other hand, $P(C_{\text{program}} | W_{\text{analyst}})$ gives an indication of how much good data a particular method marks for exclusion.

	K-B System	Neural Network	Gradient Method
$P(W_{\text{program}} C_{\text{analyst}})$	.03	.20	.12
$P(C_{\text{program}} W_{\text{analyst}})$	.57	.16	.10

Table 2: Comparison of the three computer programs with the human analyst using $P(W_{\text{program}} | C_{\text{analyst}})$ and $P(C_{\text{program}} | W_{\text{analyst}})$ ($N=35 \times (512 \times 512)$).

In any event, these results represent preliminary work and should be interpreted with some caution. Development is still continuing on the neural net and knowledge-based approaches. Furthermore, the images which were used as the evaluational standard, i.e., those declouded by the human analyst, were not prepared specifically for this purpose, so comparisons based on them taken as "ground truth" cannot be totally accurate.

VI. FUTURE WORK

Of the three systems described above, only that based on the gradient is considered sufficiently accurate for use in some applications, particularly the edge detection procedure for which it was originally developed. However, even though the network and knowledge based approaches do not equal the gradient method in performance, they have especially great potential for further improvements. In the case of the network approach, only a few of the plausible architectures have been investigated, and in the case of the knowledge-based approach only a small fraction of the the information known to be used by human analysts has been incorporated into the system. Thus, work is continuing on these two approaches as well as systems employing combinations of the three methods.

One other problem that we need to address is the development of a standard comparison set of declouded AVHRR images. Evaluation of cloud detection algorithms requires an accurate standard of comparison. Sufficient *in situ* data is impossible to obtain because of the vast synoptic coverage provided by AVHRR images. As mentioned, we have used a set of images declouded with the help of a human analyst, but the images were prepared for a different purpose and were known to contain classification errors. They were, nonetheless, the best available. What is required is an extensive set, carefully prepared specifically with this purpose in mind, using all available information, e.g., AVHRR visible channel data. Such a data set would be valuable for all working in automated cloud detection in AVHRR imagery.

VII. REFERENCES

- [1] Cayula, Jean-François , and Peter Cornillon (1990). Edge Detection Applied to Sea Surface Temperature (SST) Fields, *Digital Image Processing and Visual Communications Technologies in the Earth and Atmospheric Sciences*, Paul Janota, editor, SPIE 1301, 13-24.
- [2] Pao, Yoh-Han (1989). *Adaptive Pattern Recognition and Neural Networks.*, Addison-Wesley, New York.

MARKOV NETWORKS FOR KNOWLEDGE REPRESENTATION INCORPORATING PROBABILITIES

Michael G. Thomason and Jean R. S. Blair

Department of Computer Science
The University of Tennessee
Knoxville, TN 37996

I. INTRODUCTION

A rule-based system to aid the oceanographic image analyst has provided encouraging results[5]. As an important part of its knowledge, that system uses estimates of expected values of a large number of random variables relevant to Warm Core Ring (WCR), Cold Core Ring (CCR), and Gulf Stream evolution[7]. This paper suggests (i) that statistical estimates of complete probability distributions obtained from satellite data be used, and (ii) that new developments in knowledge representation by Markov networks[3,4,8] offer a powerful, graph-theoretic approach to manipulating that complex knowledge effectively to assist oceanographers or guide lower-level image-processing modules.

II. ESTIMATES OF DISCRETE PROBABILITY DISTRIBUTIONS

Statistical studies of oceanographic phenomena are one approach to obtaining the probability distributions on which Markov networks are based. For simplicity, we illustrate with the following nine random variables concerning WCRs:

Lb: the location of birth of a WCR in degrees north latitude and degrees west longitude. i.e., the position of the center of a WCR at its formation by the Gulf Stream.

Sb: the semi-major axis of a WCR at birth in km.

Lc: the location of coalescence of a WCR as a pair (latitude, longitude) in degrees.

Sc: the semi-major axis of a WCR at coalescence in km.

Tc: the time of coalescence of a WCR as days after birth.

SL: the status of a WCR as short-lived or long-lived where

$$SL = \begin{cases} \text{short} & \text{if } Tc < 140 \\ \text{long} & \text{otherwise.} \end{cases}$$

Te: the age of a WCR in days, i.e., the elapsed time since birth of a WCR up to *Tc*.

Le: the location of a WCR at time *Te* as a pair (latitude, longitude) in degrees.

Ve : the velocity of a WCR at time Te as a pair (westward magnitude, northward magnitude) in cm/sec.

The histograms in Figure 1 are from observations of 87 WCRs[2] and provide the estimates of marginal and conditional distributions for the random variables. The "lifetime" histogram defines $p(Tc | SL, Te)$. For example, if $Tc = 100$ days, then $p(Tc | SL = \text{short}, Te = 100)$ for $100 < Tc < 140$ and $p(Tc | SL = \text{long}, Te = 100)$ for $140 < Tc < 400$ are obtained by relative frequencies of the event counts.

The second histogram of locations of WCR birth and coalescence yields $p(Lb)$, $p(SL | Lb, Te)$, and $p(Lc | SL, Lb)$. The third histogram of WCR semi-major axis yields $p(Sb | SL)$ and $p(Sc | SL, Sb)$. We also assume $p(Ve | Lb)$ and $p(Le | Lb, Te)$ to be obtained from ring statistics. These distributions would typically be stored as tables.

The idea now is that the maximum information available in this system resides in various representations of the joint probability $p(Lb, \dots, Ve)$, from which all marginals or conditionals of interest can be computed.

III. MARKOV NETWORKS

We propose the use of Markov network representations of oceanographic knowledge which incorporates probabilities as an inherent component. See[3,4] for extensive discussion of one method, based fundamentally on graph theory, which we only illustrate here.

Figure 2(a) is the network for the nine WCR random variables in which, e.g., the arcs entering node Sc show that $p(Sc | SL, Sb)$ is defined. Figure 2(b) is the triangulated, undirected graph obtained from 2(a), and Figure 2(c) is a tree of the cliques (the maximally connected subsets of nodes) in 2(b). The arcs in the tree are labelled with the "clique separators", i.e., with the random variables in the clique intersections through which information is passed upward and downward through the tree.

Tree-based manipulation yields $p(C)$ for each clique C , e.g., $p(Le, Ve)$ for $C_5 = \{Le, Ve\}$ in Figure 2. Any marginals then desired are easily computable.

IV. ABSORPTION OF EVIDENCE

An "answer" or "inference" provided by a Markov network is a probability distribution conditioned on any constraints imposed on subsets of the random variables. For instance, suppose that a WCR birth at specific location $\bar{Lb}$ with specific size $\bar{Sb}$ is actually observed in satellite data or is hypothesized at elapsed time $Te = 0$. Suppose further that information about Le is desired for elapsed time $\bar{Te}$, say $\bar{Te} = 14$ days, as an estimate of future location after two weeks. An inference in this Markov network model takes the form of the other random variables' probabilities conditioned on the explicit values $\bar{Lb}, \bar{Sb}, \bar{Te}$.

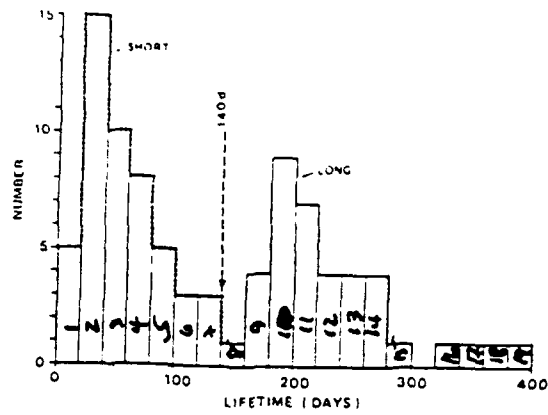
Figure 3(a) shows the undirected graph as modified by the "evidence" $\bar{Lb}, \bar{Sb}, \bar{Te}$. Figure 3(b) is the new clique forest of two disjoint trees created by absorption of this evidence. Tree-processing takes place as with Figure 2: the same algorithms are used. The output is the set of conditional probabilities we require, from which the complete distribution $p(Le | \bar{Lb}, \bar{Sb}, \bar{Te})$ is obtained as the maximum amount of information available about the future location Le projected to $\bar{Te} = 14$ days after birth.

The full distribution may be needed, e.g., to guide a search for edges. If less than complete information is desired, the system could compute and display the conditional expectation $E(Le \mid \bar{Lb}, \bar{Sb}, \bar{Te})$ and the conditional variance $\text{var}(Le \mid \bar{Lb}, \bar{Sb}, \bar{Te})$.

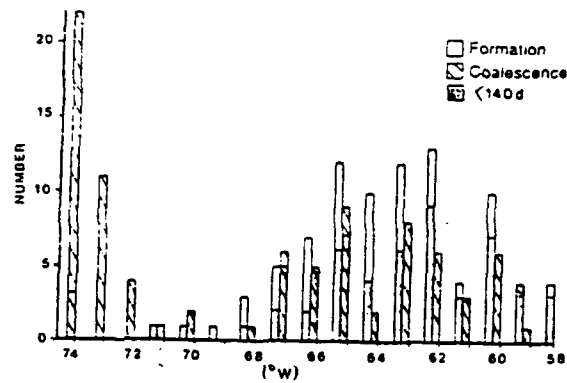
We note finally that recent research has developed methods to update frequency counts systematically in a Markov model as new observations of random variables are obtained; furthermore, this concept of continuous learning via updated statistics also provides for introducing new nodes and arcs in a Markov model to account for events unexpected but reliably observed in real data[8]. For example, these methods can deal with an unexpected event such as the frequency count of zero in the first histogram for 300 to 320 days.

V. REFERENCES

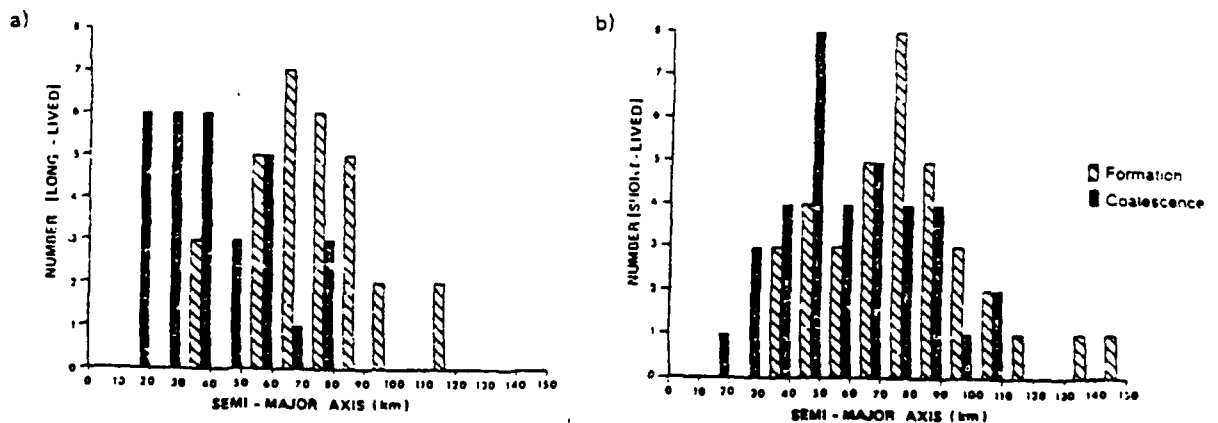
1. C. Berge, *Graphs and Hypergraphs*, North-Holland, Amsterdam, 1973.
2. O. B. Brown, P. C. Cornillon, S. R. Emmerson, and H. M. Carle, "Gulf Stream Warm Rings: a Statistical Study of Their Behavior," *Deep-Sea Research*, vol. 33, pp. 1459-1473, 1986.
3. R. E. England, "Clique Graph Models for Independent Computations," Ph. D. Dissertation, Department of Computer Science, University of Tennessee, Knoxville, December 1989.
4. S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society B*, vol. 50, no. 2, pp. 157-224, 1988.
5. M. Lybanon and R. S. Romalewski, "An Expert System to Aid the Oceanographic Image Analyst," *SPIE 1293 Appls. of A. I. VIII*, pp. 918-928, 1990.
6. J. Pearl, "Fusion, propagation and structuring in belief networks," *Artificial Intelligence*, vol. 29, pp. 241-288, 1986.
7. M. G. Thomason, "Knowledge-Based Analysis of Satellite Oceanographic Images," *Int'l. Jour. Intell. Sys.*, vol. 4, pp. 143-154, 1989.
8. J. A. Whittaker, personal communication.



Histogram of WCR Lifetime.

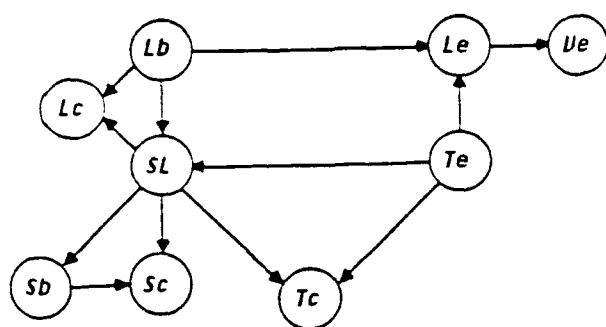


Histograms of WCR Birth and Coalescence.

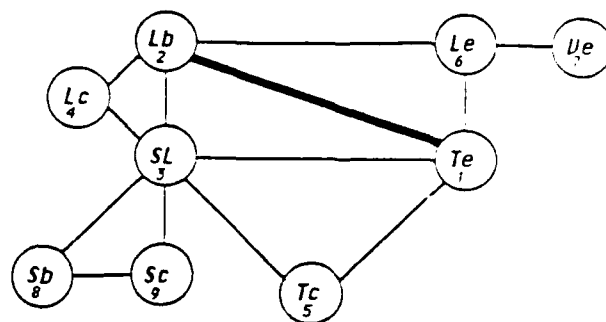


Histograms of WCR Semi-Major Axis at Birth and Coalescence.

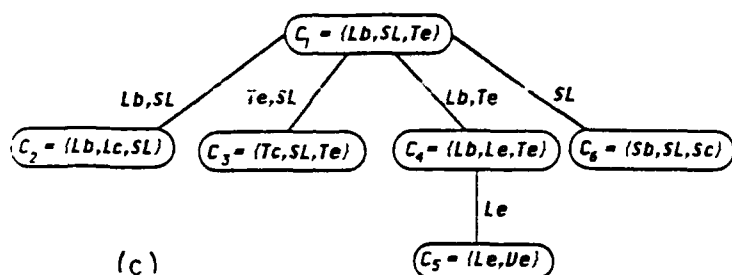
FIGURE 1. WCF. HISTOGRAMS



(a)

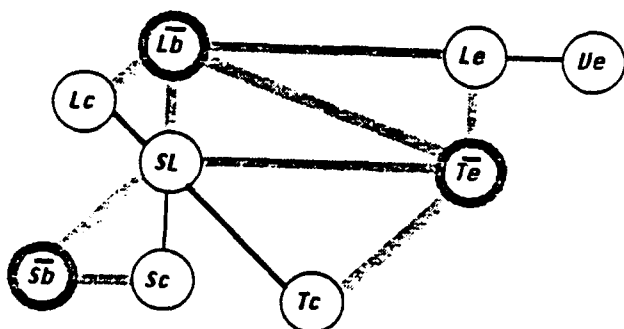


(b)

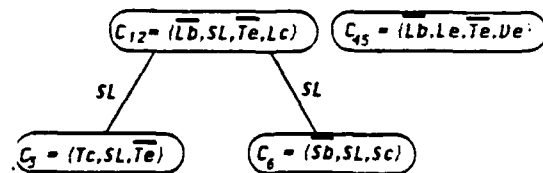


(c)

FIGURE 2. WCR MARKOV NETWORK, GRAPH, AND CLIQUE TREE



(a)



(b)

FIGURE 3. ABSORPTION OF "EVIDENCE"

Edge/Region Correlation in Image Segmentation and Analysis

Vivien J. Cambridge
Applied Math and Physics Section
Engineering and Science Department
Sverdrup Technology Inc, SSC Group

INTRODUCTION

A crucial first step in the computer aided recognition of features in a digital image is the segmentation of the image into subsets which roughly represent objects or phenomena in the depicted scene. The process of of segementation has traditionally been executed in one of two fundamentally different ways: edge based segmentation and region based segmentation. In edge based segmentation the image is scanned for intensity gradients which represent borders between image features, while in region based segmentation features are detected by intensity uniformity between contiguous pixels. The locally oriented edge and the globally oriented region detection techniques provide information that reflects different but important aspects of the image.

The present method for automated understanding of oceanographic IR imagery uses both region and edge detection techniques in a mutually augmentory way. The automated interpretation system uses a cluster shade edge detector capable of distinguishing edges from noise and an iterative split/merge region detector to perform image segmentation.

DISCUSSION

Region detection results and relaxation labeled edges are integrated through a spatial correlation technique. The purpose of spatial correlation of the edges and regions is to identify clusters of edges which are likely to be the detected fragments of one greater edge which would have enclosed a depicted feature in the the image. The premise of the correlation method is that the pixel coherence information inherent in detected regions can be used in the grouping of edges. The following hypothesis is postulated: If association between an edge and a region is defined as the edge and region satisfying some proximity constraint, then edges which delineate the same feature in the depicted scene are likely to be associated with a common region in the image. The correlation method identifies for every edge in the image a group of edges which are associated with at least one region that the edge itself is associated with. It expresses this common association by stating that the edge 'sees' the edges in the related group. An edge correlation table is then constructed which tabulates the group of edges 'seen' by every edge in the image, and the method continues by finding clusters of edges in the correlation table where each edge in the cluster sees at least a specified percentage of the other edges in the

cluster. The spatial association information introduced in the clustering operation can be used to refine the labeling of the edges. It can also be used to increase the significance of member edges of prominent clusters by linking them or sharpening them. Edge grouping following edge region correlation is formally implemented as follows:

Let $T(E_i)$ be the set of edge groups E_i comprised in the correlation table, where every edge group E_i corresponds to a member edge i and comprises the list of edges which this member edge correlates with. An example of such a set T is graphically depicted in the form of a 'correlation map' (Figure 1). The following recursive procedure produces clusters of edges (C_m) where each edge in a cluster correlates with all other edges in that cluster.

$$C_m = (((E_i \wedge E_{M(i,1)}) \wedge E_{M(i,p)}) \wedge E_{M(i,q)}) \wedge \dots) \wedge E_{M(i,k)} \quad (1)$$

where: $M(i,j)$ is member edge j in group E_i
 k is the number of member edges in group E_i
 edge p follows edge 1 in group $(E_i \wedge E_{M(i,1)})$
 edge q follows edge p in group $(E_i \wedge E_{M(i,1)}) \wedge E_{M(i,p)}$

Upon determination of C_m , edge group E_i is reduced as follows:

$$E_i = E_i \cap C_m$$

and the operation of equation 1 is repeated to find C_{m+1} until $E_i = \text{null}$.

This procedure is repeated for all edges in the set T to produce the complete set of correlated edge clusters. Figure 2 shows an algorithm which consistently groups edges into groups where each member edge associates with all member edges.

RESULTS

Figure 3b depicts labeled edges which correspond to two warm core eddies in an oceanographic IR image (Figure 3a). The figure shows improvement in labeling results when edge/region integration is applied (right top and right bottom) over results obtained with a contextual knowledge labeling system using only edge information (left top and left bottom). As this figure indicates, the method improves the interpretation of the detected edges, making it more concise and accurate.

ALGORITHM FOR REDUCING CORRELATION MAP TO EDGE GROUPS

```

FOR ALL EDGE
  FOR ALL MEMBER(LIST_EDGE)
    IF FLAG( MEMBER(LIST_EDGE) ) = FALSE
      LIST_NEW = LIST_EDGE
      WHILE NEXTMEMBER(LIST_NEW) <> END OF LIST_NEW
        LIST_NEW = (LIST_NEW) AND(LIST_NEXTMEMBER(LIST_NEW))
        FLAG( NEXTMEMBER(LIST_NEW) ) = TRUE
        INCREMENT ( NEXTMEMBER(LIST_NEW) )
      END WHILE
      GROUP (NXT) = LIST_NEW
      INCREMENT NXT
    END IF
  END FOR
END FOR

```

Figure 2 - Edge Grouping Algorithm



THE PHOTOGRAPH WAS TAKEN BY A MEMBER OF THE POLICE FORCE WHO WAS ON DUTY AT THE TIME.



Figure 3b -

Finding Ocean Structure Using Mathematical Morphology

Suzanne M. Lea
Department of Mathematics
University of North Carolina at Greensboro
Greensboro, NC 27412

I. THE OPENING AND CLOSING ALGORITHM

The algorithm is based on the image transformations of opening and closing developed by Serra (1982) in connection with texture analysis of digital images. Useful reviews and discussions may be found in Haralick et al. (1987) and Wilson (1989). The transformations can best be understood by considering first their operation on binary images (images where a pixel has intensity 1 if it is in an object and 0 if in the background) and later extending the operations to grey-scale images (images where a pixel may have more than two possible intensity values).

A. The Transformations

The class of transformations considered by Serra results from computing intensity values in a transformed image based on comparing a structure element to the original image. A structure element is a geometric pattern of pixels, usually much smaller than the image. One pixel in the pattern is defined to be its center; the center pixel need not be at the geometric center of the pattern. The properties of the pattern are the properties being sought in the image. Pixels in the structure element may be of three types: those which must be in an object, those which must be in the background, and "don't care" pixels, which may be in objects, in the background, or even outside the image.

A structure element may be represented as a set B_p , centered at the pixel p . It is compared to the image $U = X \cup X^c$, where X is the set of pixels in objects in the image and X^c is the set of pixels in the background, by translating its center across the image from pixel to pixel, one row at a time. At each center position, the intensity value of the corresponding pixel in the transformed image is computed by a rule which depends on the structure element and the transformation chosen.

Pixels in the structure element B_p required to be in the background will be denoted "0"; those required to be in an object will be denoted "1"; and don't-care pixels will be denoted "?". The center pixel p of the structure element will be underscored. For example, the structure element $B_p = \{0\underline{1}\}$ can be used to find all pixels at the left edges of objects; that is, in the transformed image, an intensity value greater than 0 (i.e., 1 in binary images) is assigned only to pixels at the left edges of objects in the original image, and all other pixels are assigned intensity values of 0. (See Fig. 1a.)

As another example, the structure element $B_p = \{0\underline{11}10\}$ can be used to find the center pixels of all objects with (horizontal) length exactly 3 pixels. In the transformed image, an intensity value greater than 0 is assigned only to pixels which in the original image were in the centers of objects with horizontal length exactly 3 pixels; all other pixels, including those at the

edges of objects with horizontal length exactly 3 pixels and those anywhere in objects with horizontal lengths larger or smaller than 3 pixels, are assigned intensity values of 0 (Fig. 1b).

Structure Element	0 <u>1</u>	0 1 <u>1</u> 1 0	1 <u>1</u> 1
Original Image	0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0	0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0	1 1 1 1 1 0 0 1 0 0 0 1 1 1 0 0 0 1 1 0 0 0 1 1 1
Transformed Image	0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0

Fig. 1. (a) (b) (c)

Fig. 1 shows the effects of horizontal structure elements on binary images. In (a), the structure element finds pixels at the left edges of objects. In (b), the structure element finds center pixels in objects exactly 3 pixels long (horizontally). In (c), the structure element finds center pixels in objects 3 or more pixels long.

As a further example, the structure element $B_p = \{?111?\}$ (which may also be written as $B_p = \{111\}$) can be used to find non-edge pixels for objects with horizontal length 3 or greater. In the transformed image, an intensity value greater than 0 is assigned only to pixels which in the original image were in the interiors of objects with horizontal length 3 pixels or more. All other pixels, including those at the edges of objects with horizontal length 3 pixels or more, and those anywhere in objects with horizontal length less than 3 pixels, are assigned intensity values of 0. (See Fig. 1c.)

To find non-edge pixels in objects of size 3 x 3 pixels or greater, the structure element

$$B_p = \begin{matrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{matrix}$$

may be used. This structure element is one that is appropriate in grey-scale images for finding stars and other objects which have uniform intensities or increase in intensity toward their centers. Note that the use of don't-care pixels surrounding this structure element on all sides causes it to be effectively size-independent; objects 3 x 3 or greater in size are found.

B. Defining the Transformations for Binary Images

A two-dimensional image or structure element may be thought of as a three-dimensional intensity surface. For binary images, the allowed intensity values are 0 and 1. (In binary images, the intensity value of a pixel defines whether it is in an object or in the background.)

Comparing the two-dimensional structure element B_p above to a binary image amounts to sliding a 3×3 pixel box of intensity height 1 under the intensity surface of the image. There are two cases of interest: first, the entire box fits inside the intensity surface of the image (the entire box is inside an object); and second, at least one pixel of the 3×3 box fits inside the image intensity surface (at least one pixel of the box is inside an object).

The first case produces an image transformation called erosion when the eroded image is defined as the locus of box centers such that the entire box fits inside the image intensity surface. Fig. 2a shows the effect of erosion on a binary image. In the set representation, an erosion produces a transformed image $U_E = X_E \cup X_E^c$, where $X_E = \{p | B_p \subseteq X\}$ and $X_E^c = \{p | B_p \not\subseteq X\}$. (Recall that p is the center pixel of B_p , not an arbitrary pixel in B_p .) The sense of the definition of X_E^c is that when the entire box does not fit under the image intensity surface (the entire box is not inside an object), the pixel in the transformed image corresponding (in location) to the box center becomes a background pixel. Consequently, edge pixels of an object are always transformed to background pixels in an eroded image. The notation $U \ominus B_p$ may be used for U_E : $U_E = U \ominus B_p$.

The second case produces an image transformation called dilation when the dilated image is defined as the locus of box centers such that at least one pixel of the box fits inside the image intensity surface. Fig. 2b shows the effect of dilation on a binary image. In set notation, a dilation produces a transformed image $U_D = X_D \cup X_D^c$, where $X_D = \{p | B_p \cap X \neq \emptyset\}$ and $X_D^c = \{p | B_p \cap X = \emptyset\}$. The sense of the definition of X_D is that when at least one pixel is in both the box and an object (at least one pixel of the box fits under the intensity surface), the pixel in the transformed image corresponding (in location) to the box center becomes an object pixel. Consequently, pixels which were on the edges of the background are always transformed to pixels in objects in the dilated image. The notation $U \oplus B_p$ may be used for U_D : $U_D = U \oplus B_p$.

The complementary behavior of edge pixels as a result of the operations of erosion and dilation indicates that the two operations are related. In fact, dilation of an object corresponds to erosion of the background: $X \oplus B_p = (X^c \ominus B_p^c)^c$. (The definition as stated here is strictly true only for symmetrical structure elements; see Serra 1982.)

The image transformation of opening is defined as an erosion of the original image followed by a dilation of the eroded image: $U_O = (U \ominus B_p) \oplus B_p$. Fig. 2c shows an opened binary image. Opening an image smoothes object contours and removes small projections, thin connections between objects, and isolated objects smaller than the size of the structure element (in either size dimension).

The image transformation of closing is defined as a dilation of the original image followed by an erosion of the dilated image: $U_C = (U \oplus B_p) \ominus B_p$. Fig. 2d shows a closed binary image. Closing an image fills in small or thin gaps within or between objects.

For binary images, opening (or closing) a previously opened (or closed) image causes no change to the image. Consequently, a binary image which has

been both opened and closed cannot be transformed further by either operation. Fig. 2e shows an opened and closed binary image.

Structure Element	Original Image	(a) Eroded Image	(b) Dilated Image
1 1 1	0 0 0 0 0 0 0 0	x x x x x x x x	x x x x x x x x
1 1 1	0 0 1 1 0 0 0 0	x 0 0 0 0 0 0 x	x 1 1 1 1 1 1 x
1 1 1	0 0 1 1 1 0 0 1	x 0 0 0 0 0 0 x	x 1 1 1 1 1 1 x
	0 0 1 1 1 1 0 0	x 0 0 1 0 0 0 x	x 1 1 1 1 1 1 x
	0 1 1 0 0 0 0 0	x 0 0 0 0 0 0 x	x 1 1 1 1 1 1 x
	0 0 0 1 1 1 0 0	x 0 0 0 0 0 0 x	x 1 1 1 1 1 1 x
	0 0 0 1 1 1 0 0	x x x x x x x x	x x x x x x x x
(c) Opened Image (eroded, dilated)	(d) Closed Image (dilated, eroded)	(e) Opened and Closed Image	
x x x x x x x x	x x x x x x x x	x x x x x x x x	
x 0 0 0 0 0 0 x	x 0 0 0 0 0 0 x	x 0 0 0 0 0 0 x	
x 0 1 1 1 0 0 x	x 0 1 1 1 1 0 x	x 0 1 1 1 0 0 x	
x 0 1 1 1 0 0 x	x 0 1 1 1 1 0 x	x 0 1 1 1 0 0 x	
x 0 1 1 1 0 0 x	x 0 1 1 1 1 0 x	x 0 1 1 1 0 0 x	
x 0 0 0 0 0 0 x	x 0 1 1 1 1 0 x	x 0 0 0 0 0 0 x	
x 0 0 0 0 0 0 x	x 0 0 0 0 0 0 x	x 0 0 0 0 0 0 x	
x x x x x x x x	x x x x x x x x	x x x x x x x x	

Fig. 2

Fig. 2 shows the effects of the erosion, dilation, opening, and closing operations on a binary image. In each case, the starting point is the original image.

C. Defining the Transformations for Grey-scale Images

To apply the opening and closing transformation to a grey-scale image, an intensity threshold t is used to define an equivalent binary image: $X = \{p | p \geq t\}$ and $X^c = \{p | p < t\}$. To smooth the intensity values locally, the intensity of pixels in the transformed image is adjusted. The rule determined by the structure element and transformation chosen decides whether a pixel in the transformed image is in an object or in the background. If a pixel is in the background of the transformed image, its intensity value is set to 0: $x \in X_T^c$ means x is assigned the intensity value 0. If a pixel is in an object in an eroded image, the corresponding pixel in the original image and its n nearest neighbors are examined, and the minimum of their intensities is assigned to the pixel in the eroded image: $x \in X_E$ assigns x the value $\min \{y | y \in B_x \text{ and } B_x \subseteq X\}$. (The number n of nearest neighbors used is determined by the size of the structure element chosen. For the 3×3 structure element B_p , $n=8$.) If a pixel is in an object in a dilated image, the corresponding pixel in the original image and its n nearest neighbors are examined, and the maximum of their intensities is assigned to the pixel in the dilated image: $x \in X_D$ assigns x the value $\max \{y | y \in B_x \text{ and } B_x \cap X \neq \emptyset\}$.

After one opening followed by one closing at a particular threshold intensity, additional openings and closings at the same threshold do not further alter the image. Therefore, for a grey-scale image, an initial threshold intensity value is chosen, and the original image is opened and

closed, producing a transformed image. The opening and closing of an image is referred to as a step in the process of finding objects.

Then the threshold is incremented (defining a new equivalent binary image), the transformed (output) image from the previous step is used as an input image, and the opening and closing transformation is preformed again. The procedure of incrementing the threshold and repeating the opening and closing step on the output of the previous step is repeated until the output image from a step has pixels identical in intensity to those in corresponding locations in the input image at the start of the step. At this point, a stable division into objects and background has been found, and the procedure is said to have converged.

II. THE IMPLEMENTATION

Our implementation presently uses the 3×3 structure element B_p discussed above (or a larger [odd] B_p specified by the user: e.g., 5×5 , 7×7). The square structure element was chosen both because it is a simple structure with a well-defined center (hence easy to implement) and because its symmetry means it can be decomposed into a structure element $B_p^h = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ which operates on the initial image, followed by a structure element $B_p^v = (B_p^h)^T$ (the transpose of the horizontal structure element) which operates on the image resulting from applying B_p^h . The composition of one-dimensional structure elements has the same effect on the initial image as applying the symmetrical two-dimensional structure element directly, but decreases the number of calculations necessary.

The size of the structure element determines the minimum size of objects found: no object smaller than the structure element in either size dimension can be found by the algorithm. However, objects larger than the size of the structure element are found by the algorithm. The user may also specify a minimum size for keeping objects found; this minimum size must be no smaller than the size of the structure element.

The initial intensity threshold, which defines the first equivalent binary image, is selected by the user; an increment is added to the value chosen. The default is the minimum intensity present in the image. The increment value, also selected by the user, may be of any (positive) integer size. It is reasonable to base the choice on a measure of the noise in the image. The default value is the mean of the sum of the square roots of the intensity data in the image.

Choosing a threshold is equivalent to defining (tentatively) the intensity level of an object boundary. Incrementing the threshold is equivalent to (tentatively) redefining the intensity level of an object boundary. In the iteration process, the threshold is incremented by the increment value chosen. The iteration process converges (stops) when the transformed image resulting from an opening and closing step is the same as the image before the step. Hence, the output image for our implementation defines objects to consist of pixels with intensities at least two increment values larger than other (smoothed) intensity values in nearby portions of the image, and belonging to bright areas at least 3×3 pixels in size.

III. ADAPTATIONS OF THE METHOD FOR INFRARED SATELLITE IMAGES

The implementation described above has been adapted to find ocean eddies near the Gulf Stream in NOAA AVHRR images. The images used are 8 bits per pixel, 512 x 512 pixels, and are composites of multiple raw images.

The implementation will not find eddies directly in the composite images for three reasons: first, the implementation expects to find bright objects against a dark background; second, it expects to find objects which are uniform or increase in intensity toward their centers; and third, it expects to find (or reject) the brightest objects in the image. (In other words, the implementation was originally designed to find stars in astronomical images.) Fortunately, it proves possible to pre-process the NOAA images and use the existing implementation to find the eddies.

The initial pre-processing step is to invert the image, so that regions of high intensity correspond to hot objects (Fig. 3a). Since the opening and closing operations are used to remove both very hot and cold objects, it might seem that this step is unnecessary; in practice, however, the implementation finds eddies more reliably if the images are inverted. This increased reliability is ascribed to the fact that warm-core eddies look more like stars in inverted images.

After image inversion, the opening and closing operations are used to find and remove very hot objects (usually land, the Gulf Stream, and some clouds; Fig. 3b). For best results, the criterion for separating objects from background needs to be relaxed: when fewer than 0.33% of the pixels are changing classification (from background to object or vice versa), the opening and closing iteration is stopped. (On the average, weighted inversely by the noise, in the 13 images tested, 0.2% of the pixels were changing. Consequently, on the average, 99.8% of the pixels were unambiguously classified.) In the output image, pixels in objects found are set to 0 intensity, and pixels in the background have their intensities set to the value in the original image. Ambiguous pixels are assigned to whatever category they happen to occupy at the time the iteration is stopped. The behavior of the algorithm at the edges of objects causes a 1-pixel-wide border to be discarded from this output image. Operation of this stage of the processing can be speeded up by starting the iteration at a larger intensity threshold.

The output image from the previous processing step is used as the input image for the final processing step. In this step, the opening and closing operations are used again, with both a smaller increment and a smaller initial threshold (since the brightest objects have been removed), to separate remaining hot objects from the colder background and remove the background (Fig. 3c). The criterion for separating objects from background is relaxed here as well: when fewer than 0.55% of the pixels are changing, the iteration is stopped. (On the average, weighted inversely by noise, in the 13 images tested, 0.3% of pixels were changing. Consequently, 99.7% of the pixels were classified unambiguously.) In the output image from this step, pixels in objects are assigned the intensity of the original image, and pixels in the background are assigned intensity values of 0. Ambiguous pixels again are assigned to whatever category they occupy at the time the iteration stops. A 1-pixel-wide border is also removed from this output image. Operation of this

stage of the processing can be speeded up by starting the iteration at a larger intensity threshold and by keeping only sizable objects (e.g., 50 x 50).

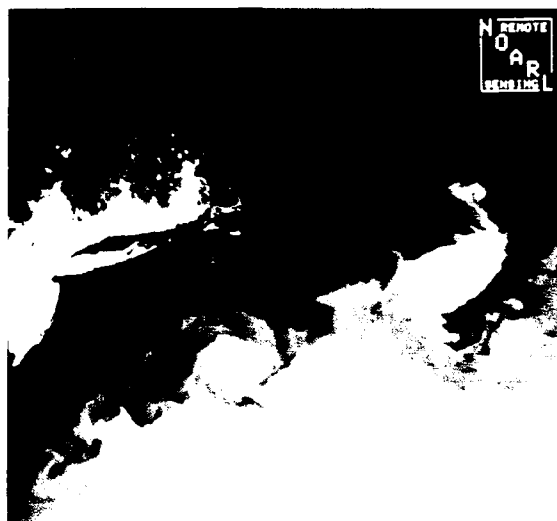
Figs. 3-6 show the results of using the implementation on various images. Fig. 3 is an extraordinarily clean image, whereas Figs. 4-6 are more normal. The algorithm works well on warm core eddies, and less well on cold core eddies. It also finds some unwanted water areas and cloud cover.

Various possible improvements are presently being investigated. Increasing the size of the structure element used would eliminate some of the unwanted objects found in the third processing step. The performance relative to finding cold core eddies could be improved by working with uninverted images, since then cold core eddies are uniform or increase in intensity toward their centers. A check for the presence of cold core eddies could be done by inverting the output of the second processing step before applying the opening and closing steps, and by using the third step to reject bright (cold) objects. Rejection of unwanted water areas and cloud cover might be achievable by using additional processing steps, such as a fourth step which finds skeletons of objects.

IV. REFERENCES

- Haralick, R. M., Sternberg, S. R., and Zhuang, X. (1987). Image analysis using mathematical morphology, IEEE Trans. on Pattern Anal. and Machine Intell., PAMI-9 (no. 4), 532.
- Lea, S. M. and Kellar, L. A. (1989). An algorithm to smooth and find objects in astronomical images. Astron. J. 97, 1238. Erratum (figures reversed): Astron. J. 98, 736.
- Serra, J. (1982). In "Image Analysis and Mathematical Morphology", pp. 34-62, Academic Press, New York.
- Wilson, S. S. (1989). Vector morphology and iconic neural networks, IEEE Trans. on System, Man, and Cybernetics 19 (no. 6), 1636.

(a)



(b)



(c)

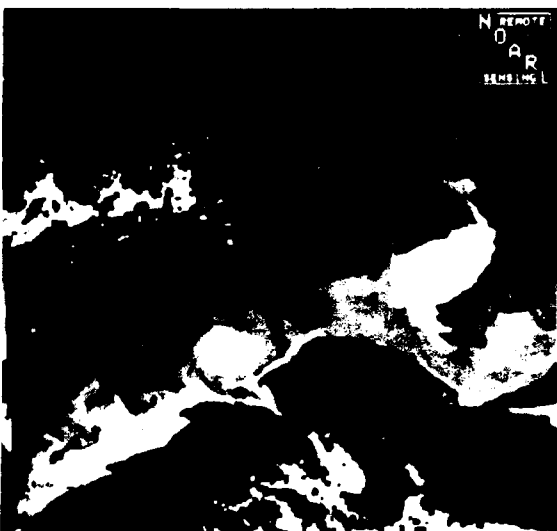


Figure 3. Effect of the procedure on the composite image March 7-8. (a) The original image March 7-8 inverted. (b) March 7-8 with hot objects found and removed by the procedure. (c) Warm objects in image (b) found and kept.

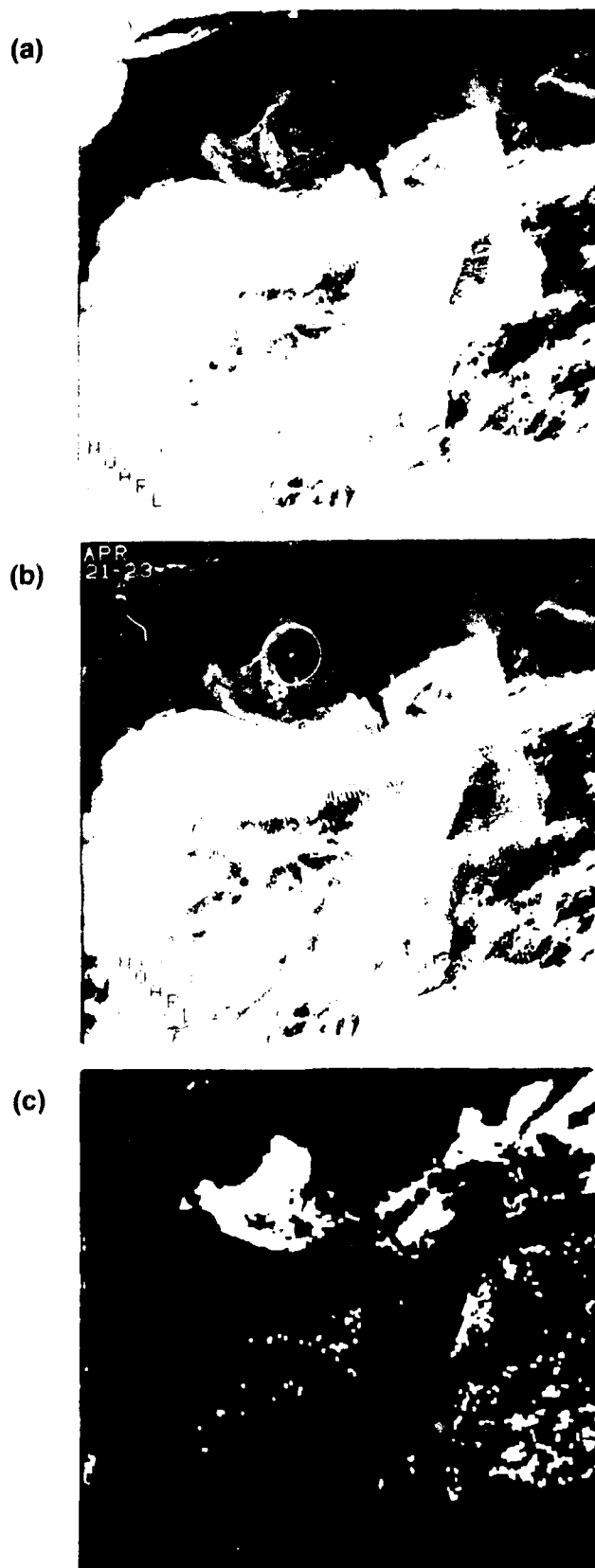


Figure 4 Effect of the procedure on the composite image April 21-23 (a) The original image April 21-23 inverted. (b) April 21-23 with edges marked by a human analyst (red) and another automated procedure (green). Only one eddy was marked in this image (c) April 21-23 with hot objects found and removed and with warm objects found and kept by the procedure

(a)



(b)



(c)

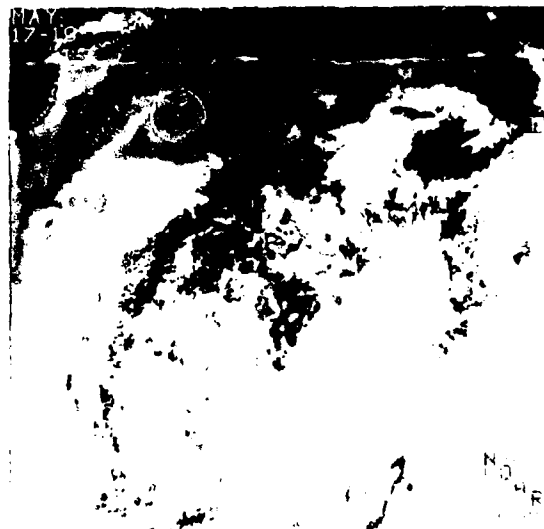


Figure 5. Effect of the procedure on the composite image April 30-May 1. (a) The original image April 30-May 1 inverted. (b) April 30-May 1 with edges marked by a human analyst (red) and another automated procedure (green). Only one eddy was marked in this image. (c) April 30 May 1 with hot objects found and removed and with warm objects found and kept by the procedure.

(a)



(b)



(c)

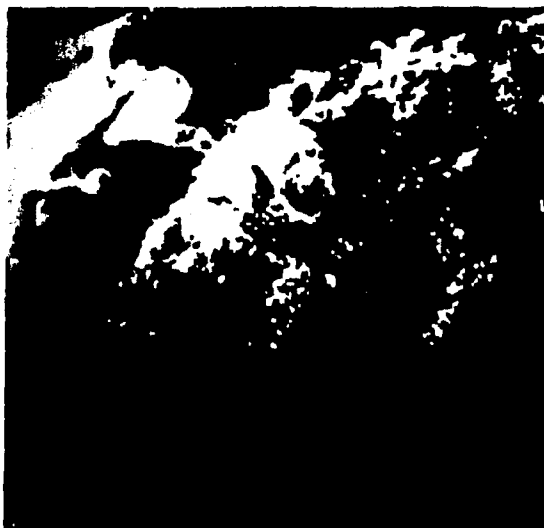


Figure 6. Effect of the procedure on the composite image May 17-19. (a) The original image May 17-19 inverted. (b) May 17-19 with edges marked by a human analyst (red) and another automated procedure (green). Only one eddy was marked in this image. (c) May 17-19 with hot objects found and removed and with warm objects found and kept by the procedure.

TIME COHERENCY OF GULF STREAM NORMAL MODE COEFFICIENTS

Ronald J. Holyer

Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, Mississippi

I. ABSTRACT

The Naval Oceanographic and Atmospheric Research Laboratory (NOARL) rule-based expert system for mesoscale dynamics in the Gulf Stream region contains rules for eddy motion that show some skill relative to persistence. However, rules for Gulf Stream motion presently consist of a simple downstream phase velocity which probably is not realistic. This work is an attempt to determine if some technique (short of a geophysical fluid dynamical model) can be utilized to furnish improved Gulf Stream motions in the expert system. Several investigators (Carter, 1985; Molinelli and Flanigan, 1987) have investigated the possibility of representing the Gulf Stream shape with a weighted sum of Complex Empirical Orthogonal Functions (CEOFs). This study is an extension of that work where time coherency of the Gulf Stream CEOF coefficients has been examined. The data set considered is 18 months of mesoscale analysis products produced at NOARL for the period of January 1986 through June 1987. The methodology is explained and preliminary results are presented. Conclusive results are not presently available because of the limited size to the data set. Therefore a full paper has not been included in these proceedings. We intend to repeat the analysis with a larger data set.

GENETIC ALGORITHMS AND CURVE-FITTING WITH APPLICATIONS TO AN ALTIMETER DERIVED REFERENCE SURFACE ERROR

by
Kenneth Messa
Department of Mathematical Sciences
Loyola University
New Orleans, LA 70118

I. INTRODUCTION

In calculating the sea-surface height (SSH) residual profile, the presence of mean dynamic topography in the reference surface leads to difficulty in interpretation. When the "geoid" is subtracted, the mean dynamic topography is also subtracted. In the area of the Gulf Stream, this error is manifested as an apparent "counterflow" north of the Gulf Stream's north wall. One approach to removing the mean topography from the residual profiles is to model SSH as instantaneous oceanography less mean oceanography. Thus,

$$\text{SSH} = \text{"true" oceanography} - \text{mean oceanography} + \text{other geophysical errors} \quad (1)$$

This model can then be fit to altimetric data of SSH. Removal of the contamination can be accomplished by adding the mean oceanography term to the SSH profile. The equation representing this SSH profile model is given in (2):

$$\text{SSH}_{\text{fit}} = A \cdot \tanh(B \cdot (X - D - E)) - F \cdot \tanh(C \cdot (X - E)) + G \quad (2)$$

where X is the along-track coordinate. The first term is the true oceanography and the second is the mean oceanography. The final term, G , represents the overall bias. The first coefficients in each term (A and F) represent the amplitude of the hyperbolic tangent curves. The coefficients B and C represent the steepness of the curves and the remaining coefficients (D and E) determine the position of the curves.

The procedure to correct this problem consists of fitting this equation to altimeter data and removing the unwanted mean oceanography. There are quite a few techniques to fit curves to data. Linear least squares/regression is one of the more commonly used methods. There are of course non-linear techniques as well. These traditional mathematics curve-fitting techniques have been tried, but have produced limited success. We introduce an alternate scheme that does not seem to be subject to curve complexity, the number of data points nor the amount of "noise" present in the signal.

II. GENETIC ALGORITHMS

Genetic algorithms are a relatively new optimization technique. They apply the "generate and test" technique to large groups of organisms. In genetic algorithms, the generate and test procedure is repeated iteratively in parallel to successive groups of candidate solutions, called *organisms*. Unlike traditional techniques, genetic algorithms use random search and selection rather than deterministic methods. Many organisms are examined as a group. The genetic algorithm uses one group of organisms to form another group. Each successive group is called a *generation*. Thus we have an initial generation, $G(0)$, and for each generation $G(t)$, the genetic algorithm generates a new one, $G(t + 1)$. An algorithm to implement genetic algorithms is given by:

```
generate initial population,  $G(0)$ ;  
evaluate  $G(0)$ ;  
 $t := 1$ ;  
repeat  
    generate  $G(t)$  using  $G(t-1)$ ;  
    evaluate  $G(t)$ ;  
     $t := t+1$ ;  
until solution is found.
```

Like all generate and test methods, the genetic algorithm requires the two main steps of generation and evaluation. In order to evaluate a generation, a *fitness* function is needed. In nature, a species responds in some way to environmental pressure. The genetic algorithm analog to this pressure is the fitness function. The fitness function is built from domain specific information and returns the relative merit or fitness of the organism. For further information about genetic algorithms, see Goldberg [1989].

III. REPRESENTATION

The organisms in a genetic algorithm represent solutions to the problem. In our case, solutions are real values assigned to each coefficient in the curve model. There is also a measurement of the goodness of fit with respect to the data D . Thus if $f=f(a_1, a_2, \dots, a_n; x)$ is the curve that we would like to fit, we are searching for real number values for the coefficients $a_1, a_2, \dots, a_n$. Thus vectors $\langle r_1, r_2, \dots, r_n \rangle$ where a_i is replaced by real numbers r_i , $i = 1, 2, \dots, n$ are candidate solutions. Therefore, organisms for the genetic algorithm used in curve fitting are vectors of real numbers, $\langle r_1, r_2, \dots, r_n \rangle$.

This view of the representation is useful at the higher level of curve-fitting problem. However, the genetic algorithm works at a lower level -- the level of bits. In order to successfully use the genetic algorithm, we need to consider a representation of the real numbers r_i at the bit level. Given upper and lower bounds for each r_i , u_i and l_i respectively, we can look at r_i as an unsigned binary integer with m bits and calculate its value with respect to l_i and u_i .

Given a binary integer b where b is in $[0, 2^k - 1]$, we can derive its corresponding real value using the formula

$$r = b/2^m * (u - l) + l \quad (3)$$

where u and l are the upper and lower bounds respectively. Combining these two levels, we construct an organism as $\theta = \langle b_{11} b_{12} b_{13} \dots b_{1m}, b_{21} b_{22} \dots b_{2m}, \dots, b_{n1} b_{n2} \dots b_{nm} \rangle$ where each binary integer $b_{i1} b_{i2} \dots b_{im}$ corresponds to a real number r_i which lies in the interval $[l_i, u_i]$. The correspondence is given in (3).

Computing the value of the fitness function of an organism θ requires the two steps -- first, converting each binary integer $b_{i1} b_{i2} \dots b_{im}$ into its corresponding real value r_i ; then, second, evaluating the curve $f = f(r_1, r_2, \dots, r_n; x)$ at the data points of $\mathcal{D}$.

IV. EVALUATION

The fitness function is a measurement of how well the organism fits the data. It is the entity that focuses the genetic algorithm toward the solution. The fitness used is modeled after least squares/regression. Given an organism θ , it is converted into a vector of real numbers, $r = \langle r_1, r_2, \dots, r_n \rangle$ using equation (3). The fitness is then computed as the sum of the squares of the differences between the $f(r; x_i)$ and y_i (that is, the residuals). Thus,

$$\text{fitness}(\theta) = \sum (f(r; x_i) - y_i)^2 \quad (4)$$

where the summation is taken over all data points (x_i, y_i) of $\mathcal{D}$. With this fitness function, a value of 0 is considered a perfect fit and an organism is highly fit if its fitness value is low.

V. CONVERGENCE

The genetic algorithm is designed to improve the relative merit of the organisms over time. While the average fitness of one generation may be lower than the preceding one or the best solution from one generation may not be as good as the best from a previous one, in general, fitnesses improve as generations unfold. Figure 1 is an example of this point. In earlier generations, there is a great deal of variability among the organisms in a single generation. There is a wide range of fitness values in these earlier generations. As happens occasionally, a few organisms are generated whose fitnesses are exceedingly poor. This reduces the average fitness of the overall generation. The number of poor solutions generated is in proportion to the fitness of the generation as a whole. Thus in the early generations, a larger number of poor solutions are formed. However, as the generations improve and larger numbers of the organisms have good fitnesses, the occasional poor performer does not affect the population as much. The affect of these less fit organisms in later generations is minimal.

of data points to be evaluated, mean that the genetic algorithm requires a long time to execute. After several test cases, it became clear that we needed to be less concerned about premature convergence than the time required to converge. Hence in later cases, we have elected to use the best of the current generation which produced faster convergence. See Figure 2.

VI. RESULTS

We have employed genetic algorithms as described above to fit the curve given in equation (2) to over 100 altimetric data points obtained from GEOSAT. The table below shows results of three of the genetic algorithm runs.

coefficient	best of generation		
	run # 1	run # 2	run # 3
A	.186291	.186294	.186295
B	.757695	.757692	.757697
C	.174908	.174905	.174901
D	24.8010	24.8008	24.8009
E	-58.8000	-58.7998	-58.8000
F	.162838	.162842	.162843
G	-.025869	-.025869	-.025869
error	.558747	.558747	.558747

Table 1. Coefficients for SSH model.

As discussed above, we were required to provide the genetic algorithm with upper and lower bounds for each coefficient. From the data and previous experience, we knew the values of the steepness and amplitude coefficients were small. We chose as the starting intervals for these values the range -5 to 5. Likewise, we knew the error term, G, was small, so we estimated it to be in that range as well. In fact, we knew G to be near 0, but chose the wider range to be more cautious.

The range for the position coefficients D and E was harder to determine, so we tried a range of width 100. After obtaining the results listed in Table 1, we narrowed the intervals to obtain the improved results as listed in Table 2. Figure 2 reflects convergence using these narrower intervals.

coefficient	best of generation		
	run # 1	run # 2	run # 3
A	.186290	.186290	.186290
B	.757685	.757680	.757682
C	.174907	.174907	.174907
D	24.8009	24.8009	24.8009
E	-58.8000	-58.8000	-58.8000
F	.162838	.162838	.162838
G	-.025870	-.025870	-.025870
error	.558747	.558747	.558747

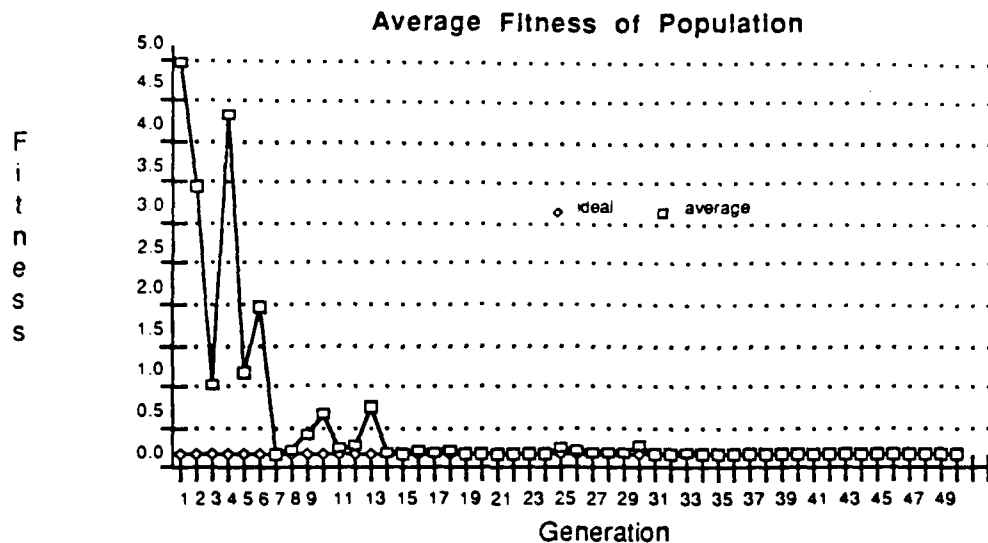


Figure 1. Convergence in Genetic Algorithms

As the generations improve, the average fitness stabilizes. As a result, most of the organisms are nearly identical. This stabilization is called *convergence*, and the genetic algorithm is said to converge to the organism that appears most often. Of course, at convergence, nearly all organisms are identical. This commonality is the solution to the problem.

We examine two approaches to genetic algorithm convergence. The first of these methods employs the best organisms from the current generation to determine convergence, while the second uses the average fitness value of the generation. By using the best of the generation, one can determine the optimal solution sooner than by using the average. Even if a generation has an organism with optimal fitness, the average of that generation may not be very good. It could take many iterations before the average approaches the optimal value and correspondingly, before most of the organisms are like the best one. Thus one gets faster results by using the best of the generation.

While faster convergence supports the best of the generation method it also has drawbacks. A good (but not optimal) organism may appear in an early generation. Through succeeding generations, the average may get better, but the best organism may remain the same. One could be lead to believe that the genetic algorithm has converged. By using the average instead of the best of the current generation, one is more conservative in determining convergence. There is less chance of stopping the genetic algorithm and declaring convergence at a good, but suboptimal solution. Figure 2 illustrates this point. One could erroneously stop the genetic algorithm after observing the same value for generations 45 through 65 of run # 2.

For the purpose of the curve fitting problem, we initially used the average of the current generation, as depicted in Figure 1. The complexity of the fitness function and the large number

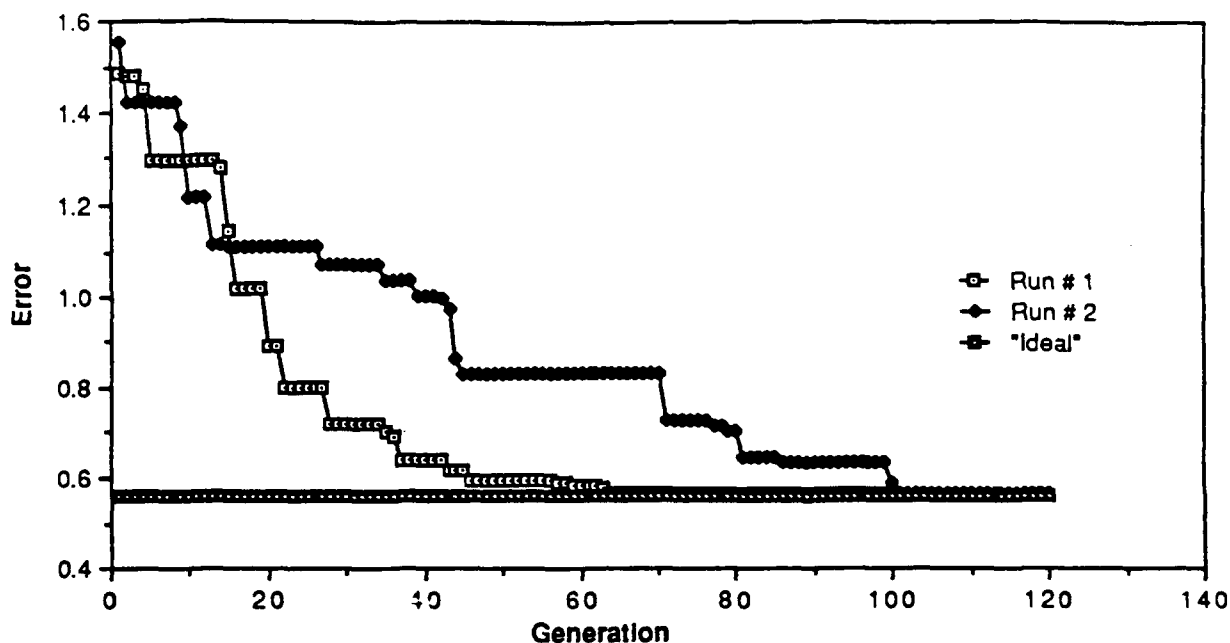


Figure 2. Convergence using "best of generation"

VII. CONCLUSIONS

We have demonstrated that genetic algorithms form a basis for another method of curve fitting. Once the genetic algorithm test bed is built, only minor modifications in the parameters and strategies are needed to achieve a fair degree of accuracy. When some knowledge of the coefficients is available, the genetic algorithm can quickly and accurately determine the optimal fit. When there is little knowledge, some experimentation with the genetic algorithm may be necessary in order to achieve a high degree of accuracy. In any case, genetic algorithms have been used successfully to fit curves to data. The chief application of this technique reveals a method for removing the contamination caused by the mean topography from the residual profiles of altimetric data. This technique can be applied to obtain a more accurate SSH profile. Figure 3 shows the result of removing the mean dynamic topography from the SSH residual profile. The correction term has been shifted downward for easier viewing.

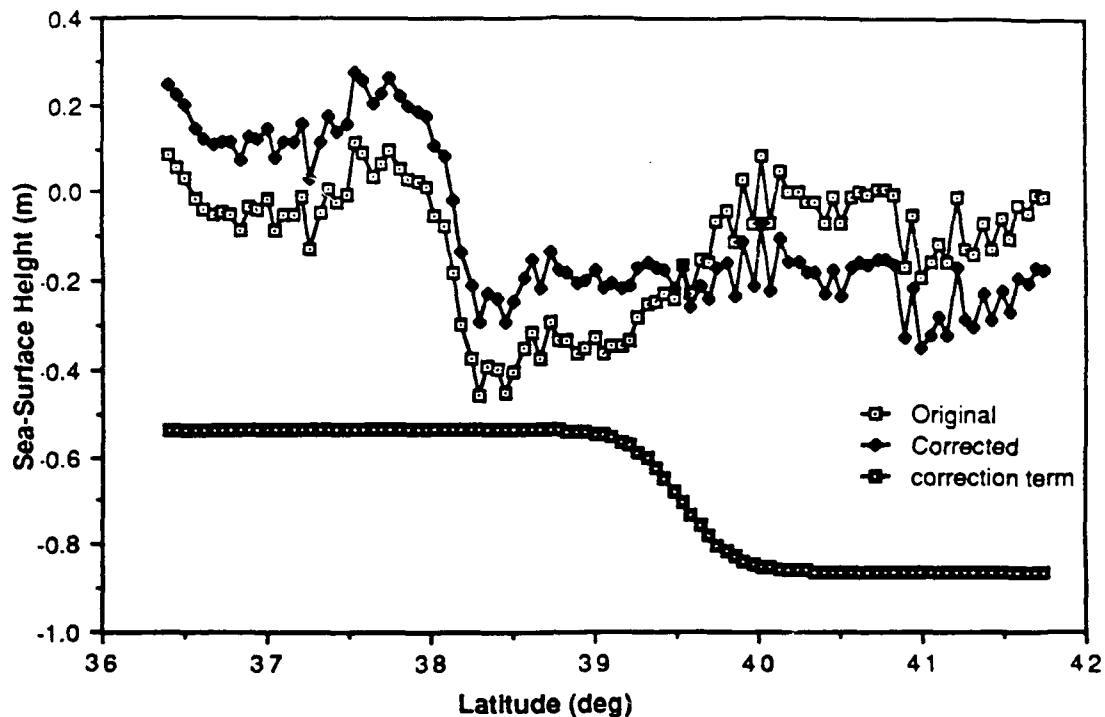


Figure 3. Original Altimeter Data and Adjusted SSH Profile

VIII. REFERENCES

- Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*:14-21.
- Calman, J. (1987). Introduction to sea-surface topography from satellite altimetry. *Johns Hopkins APL Technical Digest*: 206-211.
- Carruana, R. A. & Schaffer, J. D. (1988). Representation and hidden bias: Gray vs. binary coding for genetic algorithms. *Proceedings of the 5th International Conference on Machine Learning*:153-161.
- Daniel, C. & Wood, F. S. (1980). *Fitting Equations to Data*, 2nd Ed. New York: John Wiley & Sons.
- Davis, L. (Ed.) (1987). *Genetic Algorithms and Simulated Annealing*, London: Pitman.

- De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems. (Doctoral dissertation, University of Michigan). Dissertation Abstracts International 36(10), 5104B.
- Eshelman, L. J., Carvana, R. A. & Schaffer, J. D. (1989). Biases in the crossover landscape. *Proceedings of the Third International Conference on Genetic Algorithms*:10 - 20.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.
- Grefenstette, J. (1986). Optimization of control parameters of genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-16(1), Jan/Feb 1986.
- Guest, P. G. (1961). *Numerical Methods of Curve Fitting*. Bristol, Great Britain: Cambridge University Press.
- Holland, J. (1975). *Adaption in Natural and Artificial Systems*, Ann Arbor, Michigan: University of Michigan Press.
- Holyer, R. J. & Peckinpaugh, S. H. (1989). Edge Detection Applied to Satellite Imagery of the Oceans. *IEEE Trans. on Geoscience and Remote Sensing* 27 (1):45-56.
- Lybanon, M., Crout, R., Johnson, C. & Pistek, P. (1990). Operational altimeter-derived oceanographic information: The NORDA GEOSAT ocean applications program, *Journal of Atmospheric and Oceanic Technology*, Vol. 7(3):357-376.
- Lybanon, M., Johnson, D. R. & Romalewski, R. S. (1988). Separation of the mean Gulf Stream topography from an altimeter-derived reference surface, *EOS Transactions, American Geophysical Union*, Vol. 69, no. 44:1281.
- Lybanon, M. & Crout, R. L. (1987). The NORDA GEOSAT ocean applications program, *Johns Hopkins APL Technical Digest*, Vol.8, No. 2.
- Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of an International Conference on Genetic Algorithms and Their Applications*:93 -100.
- Siedlecki, W. & Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, Nov. 1989:335-347.
- Suh, J. Y. & Van Gucht, D. (1987). Incorporating information into genetic search. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*:100-107.
- Whitley, D. & Shaner, D. (1988). Representation issues in genetic algorithms, Technical report # CS-88-102, Department of Computer Science, Colorado State University.

AN AUTOMATED TECHNIQUE FOR LOCATING THE GULF STREAM IN ALTIMETER PROFILES

MIDN Steve L. Cantrell

U.S. Naval Academy
Annapolis, Maryland

Ronald J. Holyer

Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, Mississippi

I. INTRODUCTION

Several satellite-borne radar altimeters have been in space, beginning with Skylab in 1973, followed by GEOS-3 in 1975, and SEASAT in 1978. In 1985 the U.S. Navy launched the GEOSAT altimeter. The purpose of GEOSAT was to measure the oceanic geoid. The GEOSAT primary mission began with its launch on March 12, 1985, into an 800-kilometer-altitude, 108-degree-inclination orbit that generated a ground track grid with 4 kilometer spacing. Sea level observations over this grid continued for 18 months.

On October 1, 1986 the satellite was maneuvered into a 17-day exact repeat orbit optimized for collecting oceanographic data. During this Exact Repeat Mission (ERM) the GEOSAT spacecraft ground track was maintained within the SEASAT orbit specifications in order to allow unclassified distribution of the data to users. The ERM terminated on January 5, 1990 due to instrument failure. More than three years of GEOSAT data from the ERM are now available to researchers through the National Oceanic and Atmospheric Administration (NOAA). An overview of the Navy GEOSAT mission is available in McConathy and Kilgus (1987). The Navy plans to launch a second altimeter called SALT into the SEASAT orbit during 1990.

A satellite altimeter measures the range from the satellite to the sea surface based on travel time of a radar pulse. An independent range measurement (relative to a reference ellipsoid) is derived from satellite tracking. The difference between these two range measurements gives the instantaneous height of the sea surface relative to the reference ellipsoid. Deviations of surface height from the ellipsoid result from several factors, including variability in the earth's gravitation field, tides, currents, storm responses, and other open-ocean processes. The gravity field is, far the largest contributor to deviation of the sea surface with respect to the reference ellipsoid. An ocean at rest would assume a shape corresponding to a surface of equal gravitational potential. Such an equipotential surface is called the marine geoid. The height of the geoid can vary from -104 meters to +64 meters relative to the reference ellipsoid. If the geoid undulation is subtracted from sea surface heights with respect to the reference ellipsoid, then the result is sea surface height residuals which can be attributed to oceanographic phenomena.

Many investigators have utilized altimeter data for analysis of time-varying geostrophic currents. Among these are Cheney (1981, 1982), Bernstein,

et al. (1982), Thompson et al., and Menard (1983) using SEASAT data. Mitchell et al. (1987), Lybanon et al. (1987), and others have performed similar investigations using GEOSAT data.

A problem arises in using altimetry for investigation of ocean features that are stationary in both time and space. The problem is that the geoid is derived from altimeter data, which means that this type of quasi-permanent ocean feature, e.g., a western boundary current, is present as an error in the altimetric geoid. When such a geoid is subtracted from an altimeter profile to produce the oceanographic residual, the feature of interest can be eliminated because of its presence in both terms of the subtraction. If not entirely eliminated, the shape of the feature can be modified, depending on the relative shape and position of the feature as seen in the geoid and as captured instantaneously in the individual altimeter overpass. Figure 1 demonstrates this problem diagrammatically. The problem of oceanographic signals in the altimetric geoid can make the detection of features such as the Gulf Stream difficult, and can cause errors in determining the exact location of features even if their presence is detected. Kelly and Gille (1990) present more complete diagrams showing these effects.

This paper presents a new approach to determining the location of the Gulf Stream in altimeter profiles. The new technique avoids the problem of oceanographic contamination of the geoid by using a polynomial as a equipotential reference surface. Automation of the Gulf Stream location problem is also accomplished by using the linear Hough transform as a detection and location paradigm.

II. THE DATA SET

Thirty-six altimeter profiles are included in this study. Sixteen profiles are from ERM track A2 and twenty are from track A3. The locations of these two tracks are plotted in Figure 2. These data were collected throughout the period November 17, 1986 to October 27, 1987. The profiles of sea surface height with respect to the reference ellipsoid are shown in Figure 3. An altimetric geoid has been formed for each of these tracks by averaging the profiles shown in Figure 3. A typical example of an oceanographic residual formed by subtraction of the altimetric geoid from one of the A2 profiles is shown in Figure 4. An experienced analyst associated with the GEOSAT Ocean Applications Program (GOAP) (Lybanon and Crout, 1987) has identified the gulf Stream and a cold eddy in Figure 4 at the locations labeled GS and CE, respectively. These selected Gulf Stream and eddy locations are not at all apparent to the untrained eye. There would appear to be several other features in Figure 4 that could just as well have been interpreted as ocean features. Figure 4 is a good example of problems resulting when the oceanographic residual is referenced to an altimetric geoid. This same profile will be shown later where the altimetric geoid has been replaced with an alternate geoid. The reader will see how the interpretation of ocean features can change completely when the geoid is modified so as not to include any Gulf Stream corruption.

III. A POLYNOMIAL MARINE GEOID MODEL

There are several approaches that could lead to a modified marine geoid

which would not contain errors as the result of quasi-permanent mesoscale ocean features. For example, the geoid could be low-pass filtered with a frequency cutoff point chosen to eliminate features of spatial extent smaller than 100 kilometers (or some typical width for the Gulf Stream). Another option would be to replace the Gulf Stream portion of the geoid with interpolated values based on geoid shape to the north and south of the stream. Mitchell, et al. (1990) used bathythermograph surveys along several ERM tracks to construct a synthetic geoid that is free from Gulf Stream influence. The approach followed here is to fit a polynomial to the altimetric geoid. The order of the polynomial is selected so as not to fit features with spatial scales of the size associated with the Gulf Stream. Such a polynomial provides a smoothed version of the altimetric geoid which does not contain the mean Gulf Stream signature.

Figure 5 shows the altimetric geoid for tracks A2 and A3 over the latitude range 32 to 43 degrees north. These altimetric geoids are formed by averaging the profiles for these tracks shown in Figure 3. That portion of the altimetric geoids falling in the 34 to 43 degree latitude range have been fit using least-squares techniques (McCalla, 1967) with an n th order polynomial of the following form,

$$\text{height} = a_0 + a_1 \text{ lat} + a_2 \text{ lat}^2 + \dots + a_n \text{ lat}^n,$$

where lat is (latitude - 38.5). Table 1 gives the coefficients for the polynomial geoids shown as dashed lines in Figure 5. A 10th order polynomial captured the general profile shape along track A2 without including the mean Gulf Stream signal. For track A3 the 10th order polynomial was too high because it followed the mean Gulf Stream topography. The order of the polynomial for track A3 was reduced to eight in order to produce a synthetic geoid that did not contain the mean Gulf Stream signal.

At the scale of Figure 5 the altimetric and polynomial geoids are nearly indistinguishable. It is instructive to plot the difference between the altimetric and polynomial geoids on expanded scale as shown in Figure 6. The difference in the two geoids shows the high-frequency content of the altimetric geoid which is not contained in the smoother polynomial. As expected, a Gulf Stream signature appears in Figure 6. The Gulf Stream in the altimetric geoid is seen to be approximately 70 centimeters high and to be spread over approximately 1.5 degrees of latitude. This description of the mean Gulf Stream contribution to the altimetric geoid compares favorably with a mean height value of 80 cm reported by Mitchell et al. (1990).

Small-scale gravity features also appear in the difference traces of Figure 6. These small-scale gravity features are the disadvantage of using the polynomial geoid, since these gravity artifacts could be confused with oceanographic features. However, this factor is not as serious as one might think, because an automated technique is incorporated to detect and locate the Gulf Stream. Since the small-scale gravity artifacts will always have the same magnitude and shape, it is straightforward to program the Gulf Stream detector to ignore these known gravimetric features.

Figure 7 shows the oceanographic residuals with respect to the polynomial geoid for GEOSAT orbit 10981 (the same orbit shown in Figure 4). Also noted in Figure 7 are the cold eddy and Gulf Stream locations determined from Figure 4 by the GOAP analyst. Note that the previously identified cold eddy and Gulf Stream

signatures have disappeared from the profile when the polynomial geoid is used rather than the altimetric geoid which was used in Figure 4. Figure 7 shows a typical Gulf Stream signature located at approximately 39.5 degrees north (dashed line in Figure 7). The oceanographic interpretation of this sea surface height profile has changed completely as a result of a change in the geoid reference surface. The interpretation resulting after subtraction of the polynomial geoid is believed to be correct in this case.

IV. AUTOMATIC DETECTION OF THE GULF STREAM

The polynomial model for the marine geoid insures that the shape of the Gulf Stream in oceanographic sea surface height residuals will be more consistent from orbit to orbit than would be the case with an altimetric geoid. Specifically, the Gulf Stream signature that we would expect to see is a nearly linear feature sloping with increased height toward the south. Cheney and Marsh (1981) give examples of ideal Gulf Stream profiles from SEASAT data. Geostrophic balance relates slope of the sea surface to current velocity, so the range of slopes associated with the Gulf Stream signature is bounded by the range of observed Gulf Stream velocities.

If we assume this linear model for the Gulf Stream signature, the linear Hough transform becomes one possibility for automatically detecting the Gulf Stream in an altimeter profile. The linear Hough transform is a widely used digital image analysis technique for detection of linear features in images (Ballard and Brown, 1982). The Hough transform consists of remapping edge points from image space into parameter space. The idea behind the method is that colinear points in the image space fall into points (or at least small clusters) in the parameter space. In the case of the linear Hough transform, parameter space is two dimensional, with slope being one dimension and intercept the other. In principle, all points in an altimeter profile can be transformed into parameter space and then linear features in the profile can be found by searching for clusters in parameter space. The slope and intercept values in parameter space that mark the location of the cluster, define the corresponding line in the profile. Details of the implementation of the linear Hough feature detector that would allow the reader to reproduce the algorithm are beyond the scope of this short summary. However, the general procedure is as follows:

- (1) Subtract the polynomial geoid from the raw sea surface height data.
- (2) Set up a two dimensional (slope, intercept) accumulator array in memory.
- (3) For each point in the profile of sea surface height with respect to the polynomial geoid, given in (lat, height) coordinates, increment all possible (slope, intercept) combinations in the accumulator which represent lines that pass through the (lat, height) point under consideration.
- (4) Search the accumulator array for peaks at (slope, intercept) combinations that represent reasonable fits to the Gulf Stream slope.
- (5) Once the Gulf Stream peak has been found in the accumulator, the original profile is examined to find the northernmost point in the altimeter profile that causes the selected peak cell within the accumulator to be

incremented.

- (6) Search the profile to the south for seven points from this northernmost point looking for the profile sample with the smallest height value. The latitude of this point with the lowest height value is called the North Wall position.

Further explanation is required to clarify what is meant by "reasonable" fits to the Gulf Stream in step (4) above. Slope of the sea surface is related to surface current velocity via the geostrophic equation. Therefore, a slope value associated with an accumulator peak should translate into a realistic current value if we are to claim that the peak is generated by the Gulf Stream slope. For example, if an accumulator peak has a slope coordinate that translates into a current velocity of 500 cm/s, or that represents a current toward the west, that peak probably is not the peak resulting from the Gulf Stream. Likewise, if the intercept coordinate of a peak places the Gulf Stream on the continental shelf, that also is not a reasonable solution. In the present study the only accumulator peaks that were considered were those that represented eastward flowing currents with velocities ranging from 34 to 156 cm/s and that indicated Gulf Stream positions within 1 deg of latitude of the climatological mean position along tracks A2 and A3.

We found it necessary to bias the judgement of which Hough accumulator peak was the largest in favor of larger slope values for those cases where two peaks were nearly equal in height, but one was at a larger slope coordinate than the other.

V. RESULTS AND DISCUSSION

All thirty-six altimeter profiles were used to evaluate the accuracy of the Hough linear feature detector in automatically locating the North Wall of the Gulf Stream. The dashed line in Figure 7 is an example of the Gulf Stream fit as it was automatically determined by the Hough algorithm. This example from track A2, orbit 10981 is typical of all of the cases analyzed. Following the general steps outlined in Section 4, a best linear fit to the Gulf Stream slope was established, and a latitude value for the intersection of the North Wall with the GEOSAT ground track was determined. The accuracy of these North Wall determinations can be evaluated by comparison with the GOAP mesoscale analysis charts (Crout, 1987). An example of these mesoscale analyses are shown in Figure 8. These charts were produced by human interpretation of IR imagery and altimeter data.

The distribution of errors between the North Wall positions derived automatically and those taken from the GOAP charts was bimodal. Thirty-three of the thirty-six error values fall in the range -0.4 deg to 0.6 deg of latitude. The remaining three errors range from 1.4 to 2.0 deg of latitude. Our contention is that these three cases with exceedingly large errors result from situations where the human analyst and the automated technique identified different features. Orbit 10981 shown in Figures 4 and 7 is one of these outliers. It could be argued that the automated technique is actually correct in this case, and that the human interpreter was fooled as a result of the corruption of the geoid by the mean Gulf Stream signal. Therefore, the three outliers were not considered in evaluating accuracy since it is debatable as to which value

represents "truth".

The remaining thirty-three cases result in a mean difference of 0.09 deg of latitude (automatic techniques biased toward the north) between the automatically determined positions and those taken from the GOAP charts. The rms variation about this mean error is 0.21 deg. This is a fairly small scatter when one considers that the positions can be retrieved from the GOAP charts to only 0.1 deg accuracy. Kelly and Gille (1990) extracted Gulf Stream positions from GEOSAT profiles and compared these with AVHRR imagery. Their standard deviation between altimeter and IR estimates of the Gulf Stream position was 0.55 deg of latitude. (Kelly and Gille did not analyze the same tracks that we included here so direct comparison of accuracy is not possible.) However, our rms error of 0.21 deg does compare favorably with other work.

VII. CONCLUSIONS

The combination of a synthetic polynomial geoid and a Hough linear feature detector has automatically extracted North Wall positions that compare favorably with positions determined by manual interpretation in thirty-three out of thirty-six cases. In three cases where the automated and manual methods were in disagreement, it is not clear which of the methods is in error.

The polynomial geoids developed here for tracks A2 and A3 worked well enough that the Hough linear feature detector could correctly locate the Gulf Stream. This result speaks more for the robustness of the automatic detection procedure than for the suitability of the polynomial approach to generating synthetic geoids. A higher quality synthetic geoid is now available for some of the ERM tracks in the Gulf Stream region (Mitchell et al. 1990). Use of this geoid would be preferable to the polynomial approach. However, the polynomial geoid remains as a useful concept in those cases where a better alternative is not available.

The polynomial geoid is justified in the application presented here, namely, Gulf Stream detection. However, the polynomial approach is not suitable as a generally applicable geoid. For example, it would not be good for eddy detection because it contains gravity features of the same height and space scales as eddies. Likewise, polynomial geoids are not suited for accurate measurement of surface current velocities since they do not represent a true equipotential surface.

VII. REFERENCES

- Ballard, D. H. and C. M. Brown, 1982: Computer Vision, Prentice-Hall, 123-131.
- Bernstein, R. L., G. H. Born, and R. H. Writner, 1982: SEASAT altimeter observations of dynamic topography of the Gulf Stream region. J. Geophys. Res., 87, 3261-3268.
- Cheney, R. E. and J. G. Marsh, 1981: SEASAT altimeter observations of dynamic topography of the Gulf Stream region. J. Geophys. Res., 86, p. 473.
- Cheney, R. E., 1982: Comparison data for SEASAT altimetry in the western North

Atlantic. J. Geophys. Res., 87, 3247-3253.

Crout, R., 1987: "Remotely sensed environmental mesoscale products V.Contr. N66604-86-D-0129, Planning Systems Inc., Slidell, LA.

Kelly, K. A. and S. T. Gille, 1990: Gulf Stream surface transport and statistics at 69° W from GEOSAT altimeter. J. Geophys. Res., 95, 3149-3161.

Lybanon, M. and R. L. Crout, 1987: The NORDA GEOSAT ocean applications program. Johns Hopkins APL Tech. Digest 8, 212-218.

Lybanon, M., R. L. Crout, C. H. Johnson, and P. Pistek, 1990: Operational altimeter-derived oceanographic information: the NORDA GEOSAT ocean applications program. J. Atmos. Oceanic Technology, to be published in 1990.

McCalla, T. R., 1967: Introduction to Numerical Methods and FORTRAIN Programming. John Wiley, 43-250.

McConathy, D. R. and C. C. Kilgus, 1987: The Navy GEOSAT mission: An overview. Johns Hopkins APL Tech. Digest, 8, 170-175.

Menard, Y., 1983: Observations of eddy fields in the Northwest Atlantic and Northwest Pacific by SEASAT altimeter data. J. Geophys. Res., 88, 1853-1866.

Mitchell, J. L., J. M. Dastuge, W. J. Teague, and Z. R. Hallock, 1990: The estimation of geoid profiles in the NW Atlantic from simultaneous satellite altimetry and AXBT sections submitted to J. Geophys. Res.

Thompson, J. D., G. H. Born and G. A. Maul, 1983: Collinear-track altimetry in the Gulf of Mexico from SEASAT: Measurements, models, and surface truth. J. Geophys. Res., 88, 1625-1636.

Table 1

coefficient	A2	A3
a_0	-3.9826×10^1	-3.7449×10^1
a_1	7.6695×10^{-1}	-1.2994×10^{-1}
a_2	1.3531×10^0	1.1147×10^0
a_3	4.1488×10^{-1}	4.6058×10^{-1}
a_4	-1.4874×10^{-1}	-1.1760×10^{-1}
a_5	-6.4752×10^{-2}	-3.7850×10^{-2}
a_6	8.7302×10^{-3}	6.6423×10^{-3}
a_7	3.7933×10^{-3}	9.0036×10^{-4}
a_8	-2.6936×10^{-4}	-1.4506×10^{-4}
a_9	-7.6505×10^{-5}	--
a_{10}	3.7103×10^{-6}	--

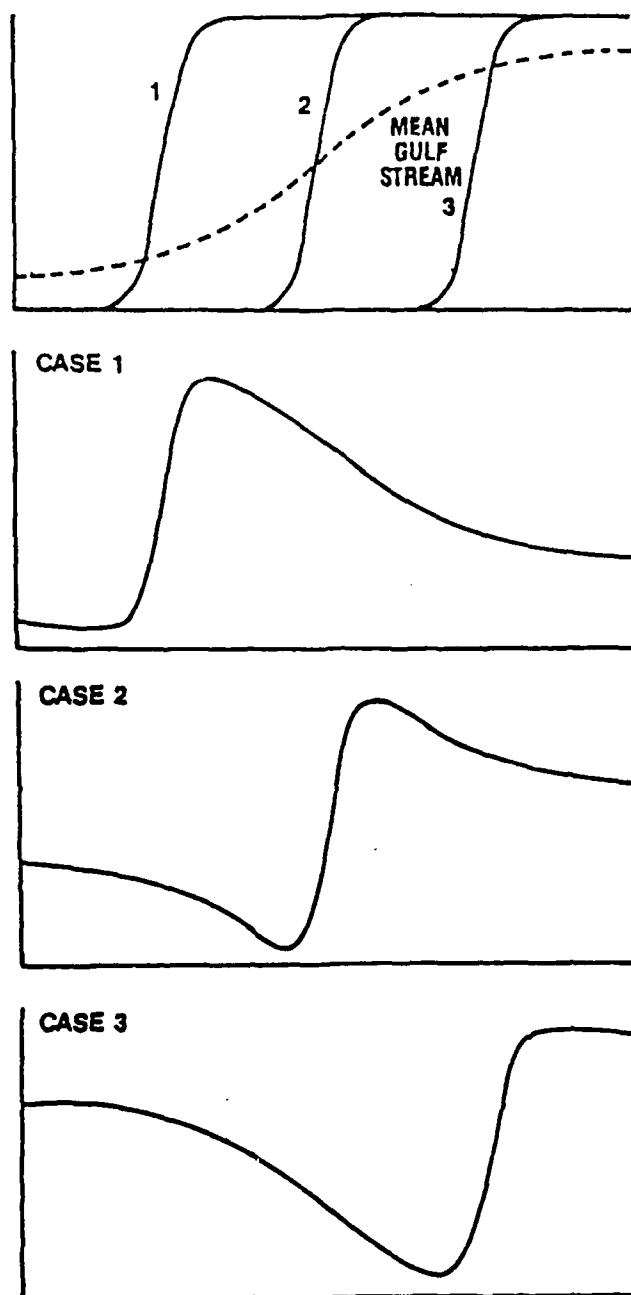


Fig. 1. An idealized illustration of the problem associated with use of an oceanographically contaminated geoid. All plots show sea surface height as the ordinate and distance as the abscissa. (a) Dashed curve is an idealized broad Gulf Stream as it might appear in an altimetric geoid. Solid lines labeled 1, 2, and 3 represent Gulf Stream topography in individual altimeter profiles where the Stream is positioned differently. (b) Plots labeled case 1 through case 3 are the idealized oceanographic residuals created by subtracting the dashed curve from each of the solid curves in (a).

ERM TRACKS A2 AND A3

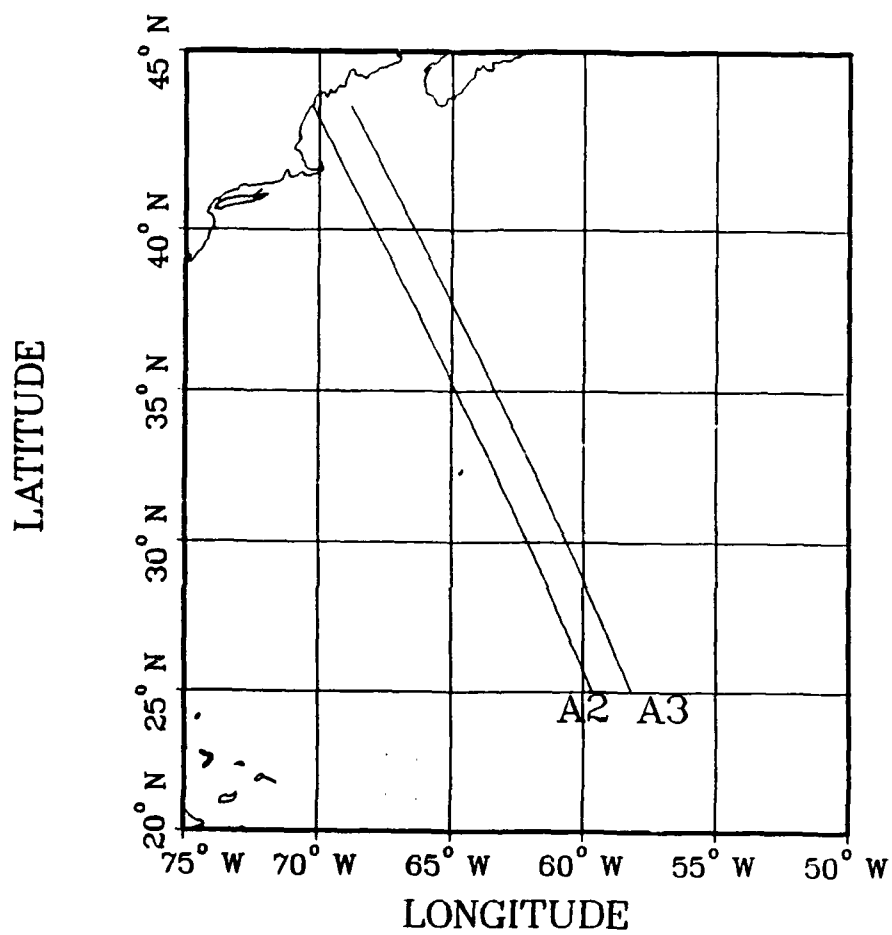


Fig. 2. Geographic location of ERM tracks A2 and A3.

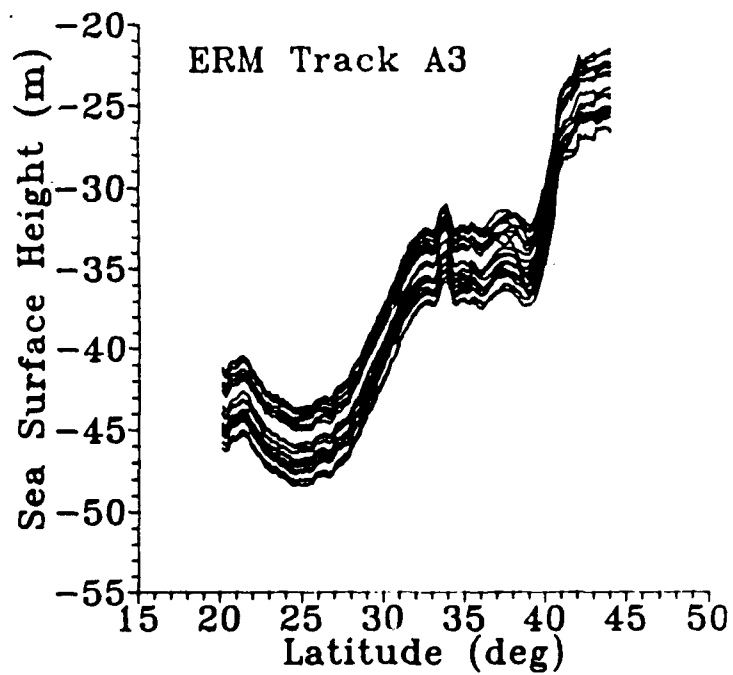
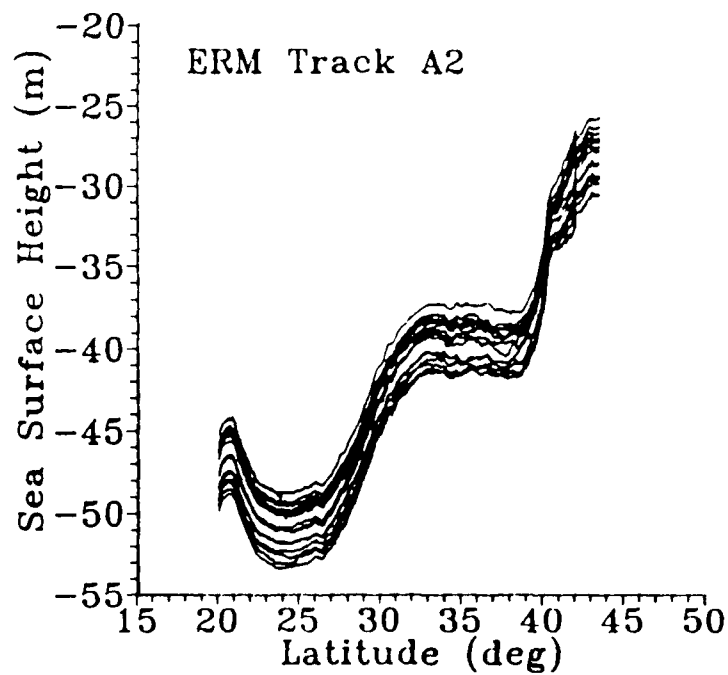


Fig. 3. The altimeter data set for this study consisting of sea surface height residuals with respect to the reference ellipsoid for ERM tracks A2 and A3.

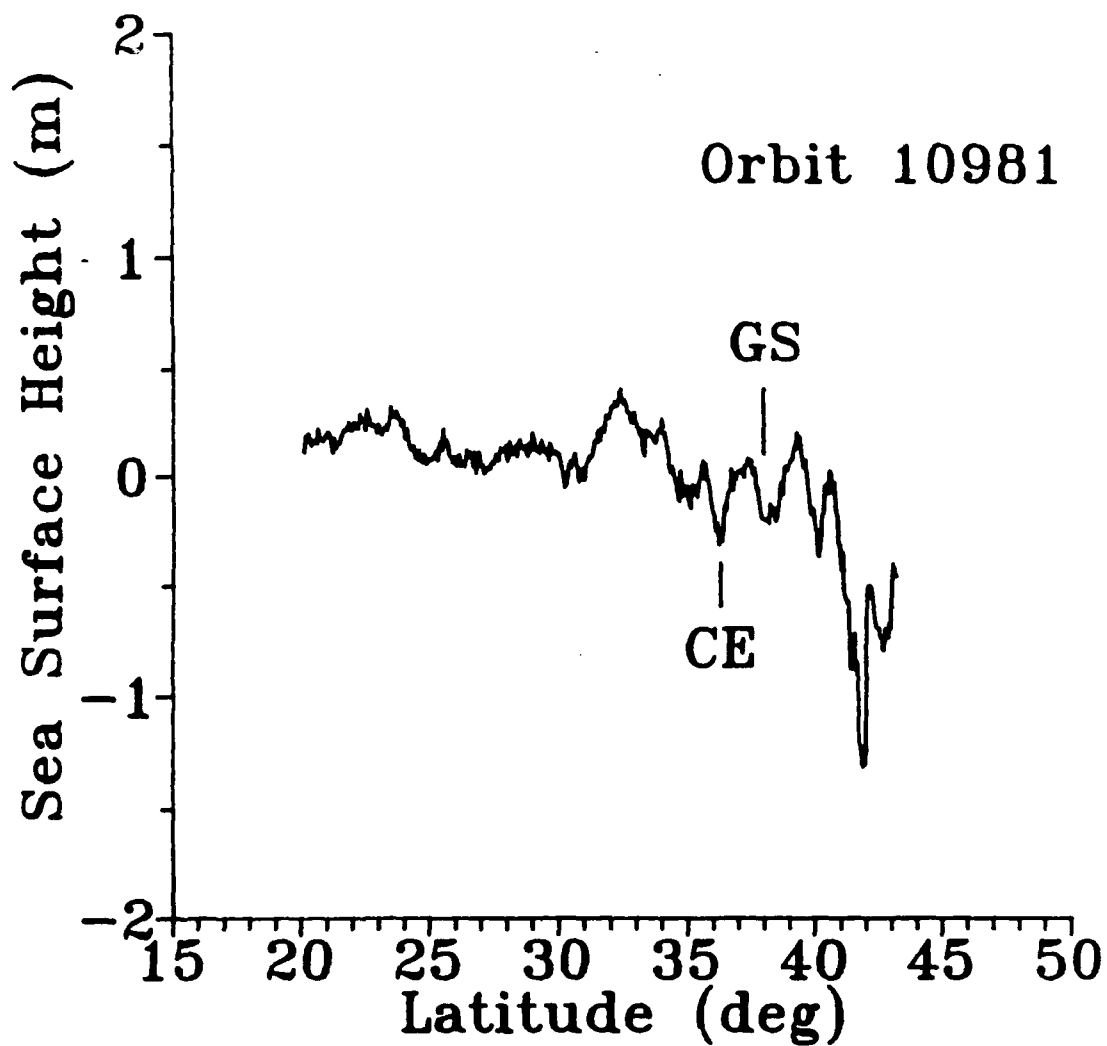


Fig. 4. Sea surface height residuals with respect to an altimetric geoid for GEOSAT orbit 10981 along ground track A2 on April 19, 1987. The labels CE and GS mark the locations of a cold eddy and the Gulf Stream respectively according to an analysis performed in support of the Navy GOAP program (Crout, 1987).

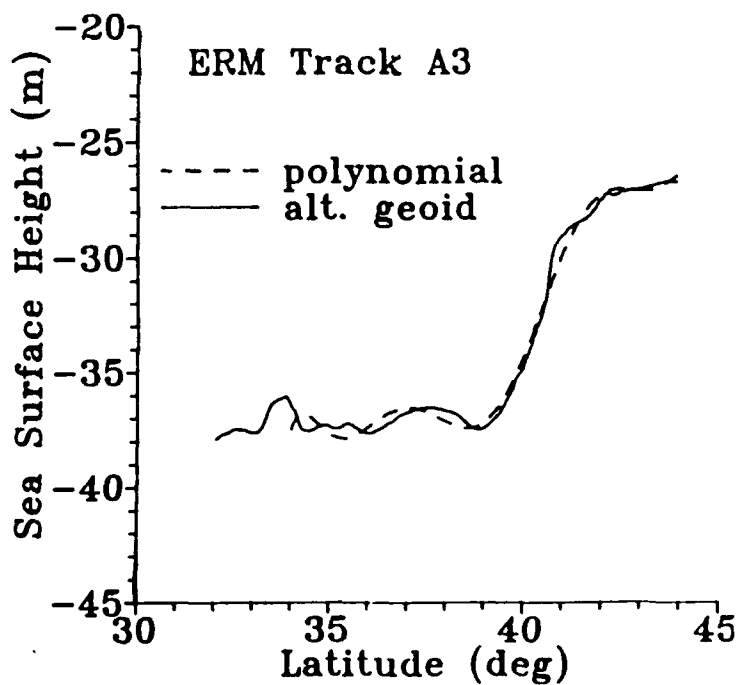
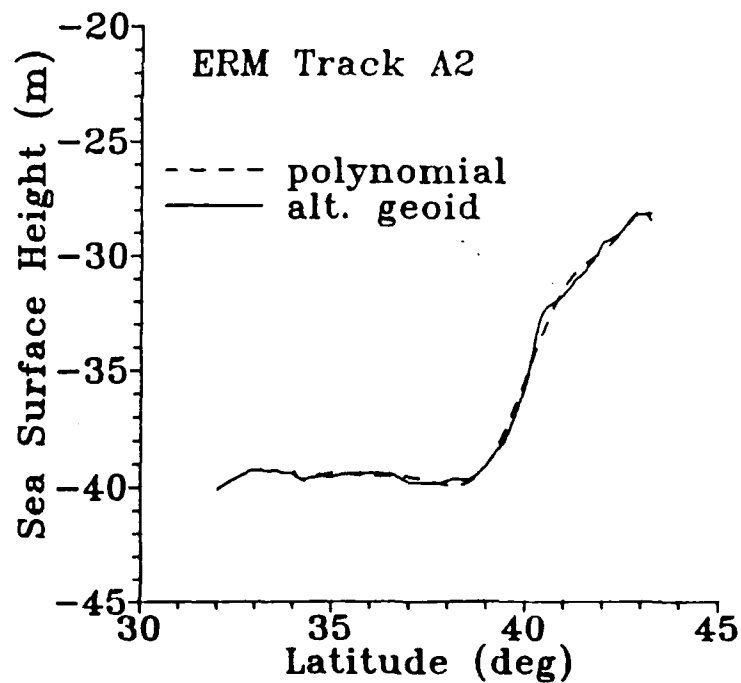


Fig. 5. Altimetric and polynomial geoids for ERM tracks A2 and A3.

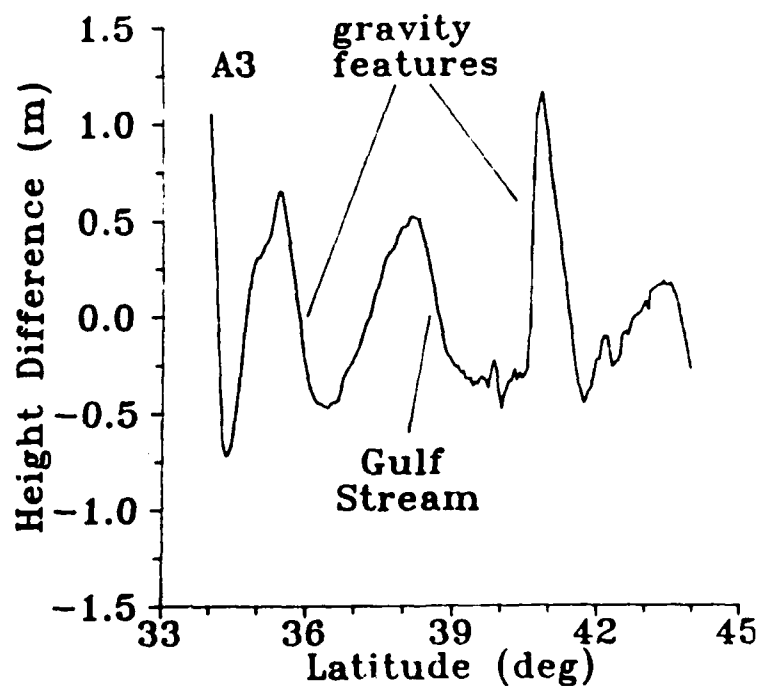
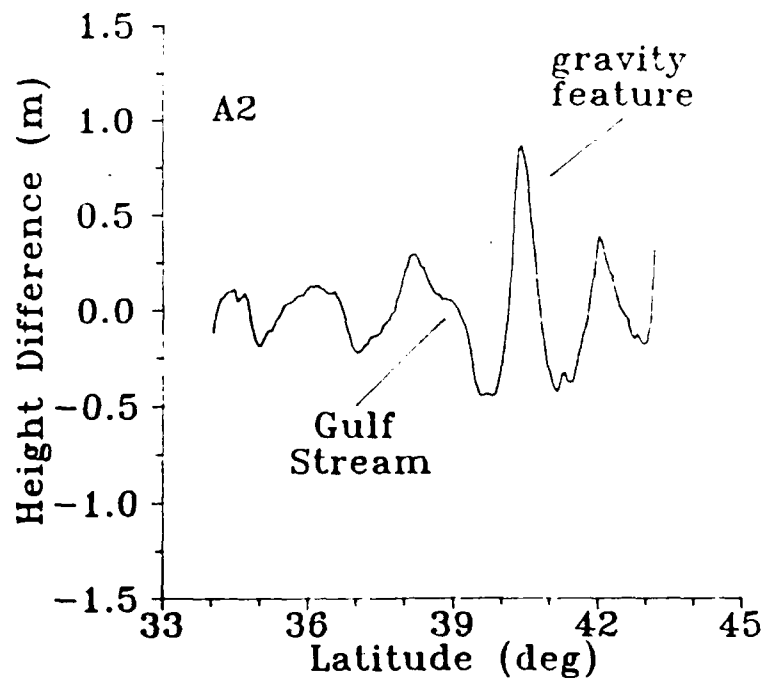


Fig. 6. Difference between altimetric and polynomial geoids for tracks A2 and A3.

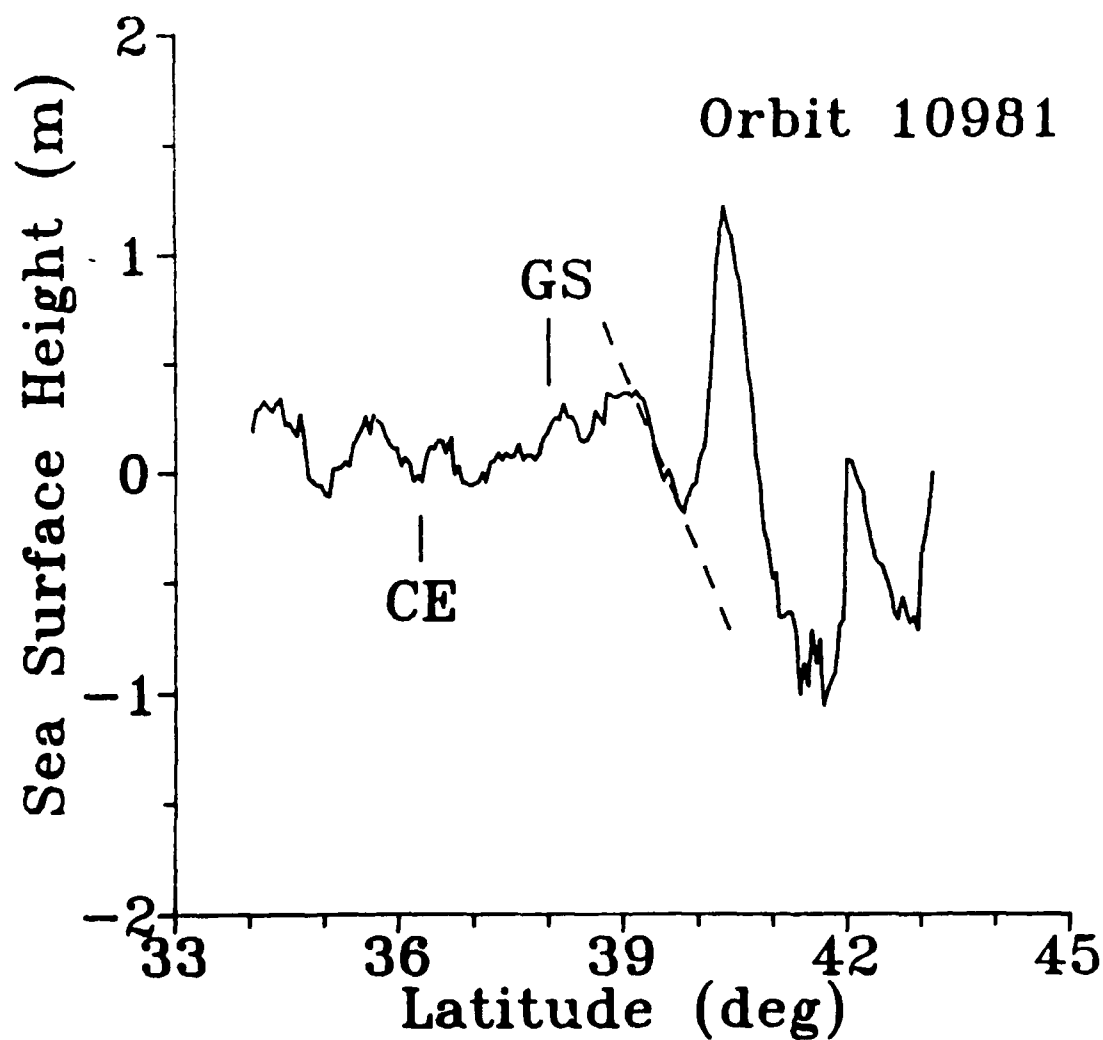


Fig. 7. Sea surface height residuals with respect to the polynomial geoid for the same GEOSAT orbit shown in Figure 4. The polynomial geoid is constructed from the coefficients given in Table 1. GS and CE labels mark positions of the Gulf Stream and cold eddy, respectively, as interpreted from Figure 4. Dashed line represents Gulf Stream location as interpreted from this figure.

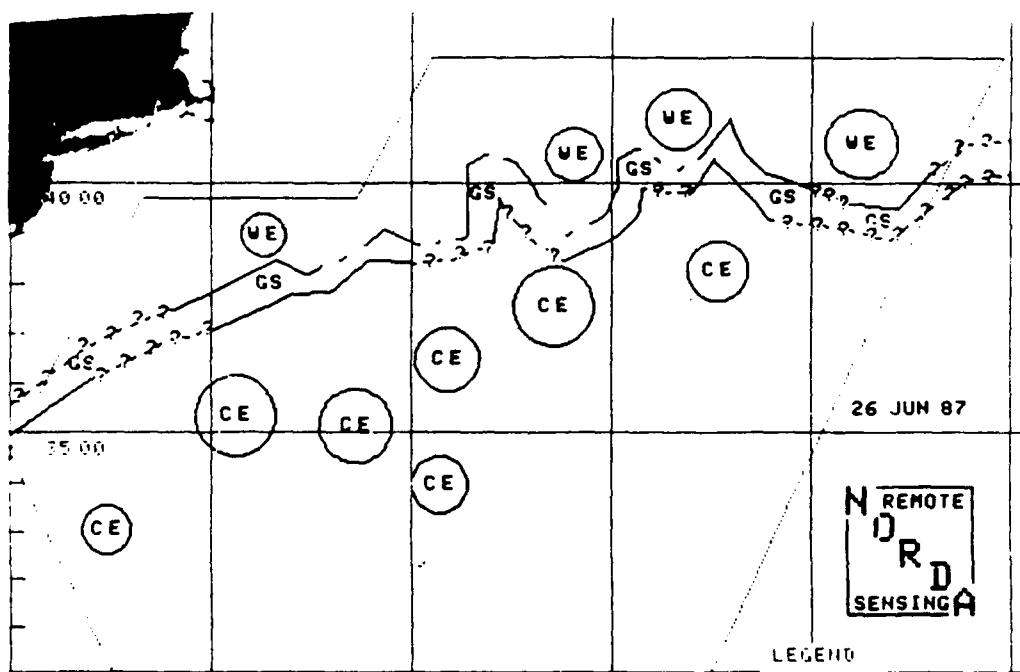


Fig. 8. GOAP mesoscale analysis product (from Crout, 1987)

REPORT DOCUMENTATION PAGE

Form Approved
OBM No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. Agency Use Only (Leave blank).		2. Report Date. February 1991	3. Report Type and Dates Covered. Final
4. Title and Subtitle. Proceedings: Automated Interpretation of Oceanographic Satellite Images Workshop		5. Funding Numbers. Program Element No. 0602435N Project No. 3582 Task No. M0G Accession No. DN256010	
6. Author(s). Matthew Lybanon		8. Performing Organization Report Number. NOARL SP 001:321:91	
7. Performing Organization Name(s) and Address(es). Naval Oceanographic and Atmospheric Research Laboratory Ocean Science Directorate Stennis Space Center, Mississippi 39529-5004		10. Sponsoring/Monitoring Agency Report Number. NOARL SP 001:321:91	
9. Sponsoring/Monitoring Agency Name(s) and Address(es). Naval Oceanographic and Atmospheric Research Laboratory Exploratory Development Program Group Stennis Space Center, Mississippi 39529-5004			
11. Supplementary Notes.			
12a. Distribution/Availability Statement. Approved for public release; distribution is unlimited.		12b. Distribution Code.	
13. Abstract (Maximum 200 words). The papers collected in this volume describe work that is potentially of great benefit to the U.S. Navy. The provision of automated or semiautomated tools to aid the oceanographic satellite data interpreter has the potential to make significant contribution to the exploitation of remotely sensed observations to satisfy Navy requirements. The techniques themselves are of considerable scientific interest, since they indicate a high degree of success in dealing with a number of difficult technical problems. The cooperation between Government, academic, and industrial researchers which the papers represent serves as a model for future partnerships.			
14. Subject Terms. Remote Sensing, Artificial Intelligence, Data Assimilation, Satellite Data		15. Number of Pages. 228	
		16. Price Code.	
17. Security Classification of Report. Unclassified	18. Security Classification of This Page. Unclassified	19. Security Classification of Abstract. Unclassified	20. Limitation of Abstract. SAR