Naval Oceanographic and Atmospheric Research Laboratory
NOARL Report 32 January 1992

A Survey of Digital Image Segmentation Algorithms

K. B. Shaw
M. C. Lohrenz
Mapping, Charting and Geodesy Division
Ocean Science Directorate

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JAN 1995		**00-00-1992 to 00-00-1992**

4. TITLE AND SUBTITLE	
A Survey of Digital Image Segmentation Algorithms	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Oceanographic and Atmospheric Research Laboratory,Ocean Science Directorate,Stennis Space Center,MS,39529-5004**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**28**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Foreword

As the Navy's leading laboratory for research and development in mapping, charting, and geodesy (MC&G), the Naval Oceanographic and Atmospheric Research Laboratory is actively involved in applying digital MC&G data to the support of naval weapons systems and in conducting research to improve these data.

This report provides details of significant research on digital image segmentation, a valuable technique for improving underwater mine and submarine detection, for better target recognition, and for improving the quality of automated computer vision output used in autonomous digital mapping.

William B Moseley

W. B. Moseley
Technical Director

L. R. Elliott

L. R. Elliott, Commander, USN
Commanding Officer

Executive Summary

Computer vision is a rapidly expanding field that depends on the capability to automatically segment and, thus, to classify and interpret images. In this report, the primary computer vision subarea—segmentation—is investigated. Many of the latest publications on the subject of segmentation are detailed in a survey format. Special attention is given to a few specialized techniques for segmenting digital images.

Powerful segmentation techniques are available; however, each technique is ad hoc. The creation of hybrid techniques seems to be a promising future research area with respect to current Navy digital mapping applications.

Acknowledgments

The authors thank the AV-8B, V-22, and F/A-18 night attack programs at the Naval Air Systems Command for supporting this document. This work was funded under Aircraft Procurement, Navy H1CC, subhead AV-8B Harrier, program elements 940101 (64262N) and 980101 (APN). Dr. Budd B. Adams was the program manager.

Special thanks are also due Tulane University's Electrical Engineering Department for contributions to this report and, in particular, Drs. Herb Barad and Andrew Martinez for their ongoing support. We would also like to thank everyone associated with NOARL's long-term training program, which enabled the first author to pursue this and other interests while at Tulane University.

Contents

A Survey of Digital Image Segmentation Algorithms

1.0 Introduction

Computer vision is a rapidly expanding area that is dependent on the capability to automatically segment, classify, and interpret images. Segmentation is central to the successful extraction of image features and their subsequent classification. Image segmentation techniques can be grouped into six categories: amplitude thresholding, component labeling, boundary-based segmentation, region-based segmentation, template matching, and texture segmentation.

During segmentation, an image is preprocessed, which can involve restoration, enhancement, or simply representation of the data. Certain features are extracted to segment the image into its key components. The segmented image is routed to a classifier or an image-understanding system. The image classification process maps different regions or segments, into one of several objects. Each object is identified by a label. The image-understanding system then determines the relationships between different objects in a scene to provide a complete scene description.

Powerful segmentation techniques are currently available; however, each technique is ad hoc. The creation of hybrid techniques seems to be a future research area that is promising with respect to current Navy digital mapping applications. For example, improved digital map classification techniques could be developed for automated feature extraction of the digitally scanned map data used in various Navy aircraft and for future shipboard electronic chart systems.

This report discusses the six image segmentation algorithms by describing the technique and comparing different algorithms. The latest publications that describe each technique are given in a survey-type format. The Summary and Conclusions section examines the potential applications if several of the techniques are integrated for developing segmentation methods that will specifically address naval applications.

2.0 Amplitude Thresholding

Amplitude thresholding, or window slicing, is useful whenever an object is sufficiently characterized by the amplitude features. Thresholding techniques are also useful in segmenting such binary images as printed documents, line drawings, and multispectral and x-ray images. A commonly used approach to thresholding follows:

- Examine the histogram of the image to identify peaks and valleys. If the image is multimodal, then the valleys can be used for selecting thresholds.
- Perform thresholding so that a predetermined fraction of the total number of samples is below the threshold.
- Adaptively threshold by examining local (neighborhood) histograms.
- Selectively threshold by examining histograms of only those points that satisfy a chosen criterion. For example, in low-contrast images, the histogram of pixels whose Laplacian magnitude is above a predefined value will exhibit clearer bimodal features than that of the original image.
- Determine the threshold to minimize the probability of error or some other quantity, for instance, Bayes' risk,[1] if a probabilistic model of the different segmentation classes is known.

Multilevel thresholding is generally less reliable than its single-threshold counterpart because establishing multiple thresholds that effectively isolate regions of interest, especially when the number of corresponding histogram modes is large, is difficult. Problems of this nature, if handled by thresholding, are best addressed by a single, variable threshold.

Mathematically, thresholding can be viewed as an operation that involves tests against a function T of the form

$$T = T\ [x, y, p(x, y), f(x, y)]\ , \tag{1}$$

where $f(x, y)$ is the gray level of point (x, y), and $p(x,y)$ denotes some local property of this point: for example, the average gray level of a neighborhood centered at (x, y). It follows that a thresholded image $g(x, y)$ is created by defining

$$g(x, y) = \begin{bmatrix} 1 \text{ if } f(x, y) > T \\ \\ 0 \text{ if } f(x, y) <= T \end{bmatrix} . \quad (2)$$

Therefore, in examining $g(x, y)$, pixels that are labeled 1 (or any other convenient intensity level) correspond to objects, and pixels that are labeled 0 correspond to the background. When T depends only on $f(x, y)$, the threshold is called global.

A simple approach that is often useful for segmenting an image consists of dividing the gray scale into bands and using thresholds to determine regions or to obtain boundary points.[2] Smoothing is also a key component related to thresholding. The gray-level subpopulations that correspond to different types of regions in a picture will often overlap. Under these circumstances, segmenting the picture into regions by thresholding becomes difficult: wherever the threshold is placed, the overlapping subpopulations cannot be cleanly separated. This problem can usually be alleviated by smoothing the picture before thresholding it. For example, the picture could be locally averaged by replacing the gray level at each point with an average of the neighboring pixels' gray levels. Within a given type of region, averaging dampens local gray-level fluctuations and, hence, reduces the gray-level variability while preserving the mean gray level. However, averaging also blurs the borders of the regions; thresholding will still extract the regions more or less correctly, although it will smooth out irregularities in their borders.[3]

3.0 Component Labeling Segmentation

Component labeling is a simple and effective method of segmenting binary images by examining the connectivity of pixels with their neighbors and then labeling the connected sets. Two practical algorithms, pixel labeling and run-length connectivity analysis, are discussed in the following sections.

3.1 Pixel Labeling

Suppose a binary image is raster-scanned from right to left and from top to bottom. The current pixel, say x, is labeled as belonging to either an object (with the pixel value set to 1) or a hole (0) by examining its connectivity to its neighbors. If two or more qualified objects are present, then those objects are declared to be equivalent and are merged. A new object label is assigned when a transition from a set of 0s to an isolated 1 is detected. Once the pixel is labeled, the features of that object are updated. At the end of the scan, such features as the centroid, area, and perimeter are saved for each region of connected 1s.

3.2 Run-Length Connectivity Analysis

An alternate method of segmenting binary images is to analyze the connectivity of run lengths from successive scan lines. Consider black and white runs denoted A, B, C, etc. A segmentation table is created and run A of the first scan line is entered into the first column. The object of run A is named A´. The first run of the next scan line, B, has the same color as A and overlaps A. Hence, B also belongs to object A´ and is placed underneath A in the first column. Run C has a different color, so it is placed in a new column for an object labeled B´. Run D has the same color as A and overlaps A. Since both B and D overlap A, divergence is said to have occurred, and a new column of object A´ is created, in which D is placed. A divergence flag, ID1, is set in this column to indicate that object B´ has caused this divergence. Another flag, ID2 of B´ (column 2), may be set to A´ to indicate that object B has caused divergence in overlap with another run, U, which sets the convergence flags IC1 to C´ in column 4 and IC2 to B´ in column 6. Similarly, W sets the convergence flag IC2 to A´ in column 2, and column 5 is labeled as belonging to object A´.

In this manner, all the objects with different closed boundaries are segmented in a single pass. The segmentation table lists the data relevant to each object. The convergence and divergence flags also give the hierarchy structure of the object. Since B´ causes divergence as well as convergence in A´, and since C´ has a similar relationship with B´, the objects A´, B´, and C´ are assigned levels 1, 2, and 3, respectively.[1]

4.0 Boundary-Based Segmentation and the Hough Transform

Boundary extraction techniques segment objects on the basis of their profiles. Therefore, such techniques as contour following, connectivity, edge linking, graph searching, curve fitting, Hough transform, and others are applicable to image segmentation. Difficulties with boundary-based methods occur when objects are touching or overlapping, or if a break occurs in the boundary due to noise or artifacts in the image.[1]

It has long been recognized that the Hough Transform (HT) represents near-exclusive technique for shape and motion analysis in images that contain noisy, missing, and extraneous data. But its adoption has been slow because of computational complexity and storage problems, as well as the lack of a detailed understanding of its properties. However, in recent years much progress has been made in these areas. An efficient implementation of the HT and results on analytic and empirical performance of various methods are discussed in this section.

The HT was first introduced to detect complex patterns of points in binary image data by determining specific values of parameters that characterize these patterns. Spatially extended patterns are transformed to produce spatially compact features in a space of possible parameter values. The HT converts a difficult global detection problem in image space into a more easily solved, local peak-detection problem in a parameter space.

When the HT is calculated on a digital computer, the continuous parameter space is usually considered to be composed of the union of a number of finite-sized regions. In standard implementations the space is partitioned into suitably sized, multidimensional rectangles. Each rectangle is associated with an element of a multidimensional array called an accumulator array. The elements of the accumulator array act as counters and are incremented when a hypersurface from the back-projection of an image point passes through the region of parameter space associated with the element. When several image points back-project to the same parameter combinations, i.e., their hypersurfaces either intersect or pass close to one another, then the corresponding array element accumulates a large value.

The HT can be viewed as an evidence-gathering procedure. Each image point "votes" for all parameter combinations that could have produced it, if it were part of the sought-after shape. The votes are counted in the accumulator array, and the final totals indicate the relative likelihood of shapes described by parameters within the corresponding parameter cell.

The HT is closely related to template-matching techniques described later in this report. One obvious difference is that template-matching is carried out entirely in the image domain. For this reason, the HT was included here instead of in the template-matching section. Unlike template-matching techniques, the HT always assumes a match between a given basic template point and a selected image point, and then calculates the transformation parameters that connect them. Thus, although the HT and template-matching calculate the same quantity, the HT is more efficient because it does not generate unessential data.

HT methods offer many desirable features. First, each image point is treated independently; therefore, the method can be implemented using more than one processing unit: i.e., parallel processing of all points is possible. This makes the HT well suited to real-time applications and to be a possible module for shape detection in biological systems. Second, the HTs independent combination of evidence means that it can recognize partial or slightly deformed shapes.

Occlusion is a severe problem for most other shape detection techniques, but the HT degrades gracefully because, to first-order approximation, the size of a parameter peak is directly proportional to the number of matching boundary and template points. The size and spatial localization of the peak provides a measure of similarity in shape and mode. Third, the HT method is robust when random data are introduced by poor image segmentation. Random image points are unlikely to contribute coherently to a single bin of the accumulator and thus produce only a low-level background of counts in the array.

A more serious problem than random data is data from the boundaries of shapes other than those being searched for. These boundaries can produce structured backgrounds, and some care must be taken to either eliminate or identify such situations. Finally, the HT can simultaneously accumulate evidence for several examples of a particular shape class that occurs in the same image. In general, each instance of the shape simply produces a distinct peak or cluster in the accumulator array.

The principal disadvantage of the standard implementation of the HT is its large storage and computational requirements. The determination of q parameters, each resolved into z intervals, requires an accumulator of z^q elements, which can be prohibitively large if either z or q is large. The major computational cost of the algorithm is the calculation of parameter-cell-parameter surface intersections. In the simplest case, the parameter surface spans $(q-1)$ of the q parameter dimensions, so the number of calculations is massive and increases exponentially with the dimensionality of the problem. The efficiency of the HT can be increased by devising methods that use small-sized accumulators or that use extra data to restrict the range of parameters to be addressed.

Early analytical work on the properties and performance of the HT concentrated on the effect of statistical measurement error on the position and the localization of parameter peaks. Shapiro[4,5,6] researched the variance of parameter estimates as a function of measurement error for transforms in which each image or feature point produced a single vote in parameter space. Sklansky[7] suggested a geometric construction for straight-line detection that could be used to investigate the precision of curves derived from estimated parameters. This graphical technique was extended by Shapiro and Iannino[8] to address the case of noisy image measurements, and was used to derive results relating quantization errors to the accuracy of parameter estimation. These guides proved useful in determining accumulator quantization.

Maitre's[9] work is the most recent concerning the effects of random image noise on the density of counts in parameter space.

The HT has proven valuable for solving many machine vision problems, since straight lines and simple polygons occur in most natural and man-made scenes. For example, a remotely sensed image of an inhabited area will contain an abundance of linear features (e.g., roads and railroads) and simple polygonal features (e.g., buildings, parks, and farm fields). Even complex objects can often be identified by their distinctive combination of these basic features.

One of the main characteristics of the HT is that it consists of a series of fairly simple calculations carried out independently on every feature in an image. The following text discusses recent developments in the implementation of the HT with real-time hardware, as well as efforts to capture the HT's inherent parallelism on specialized parallel architectures. Most of these efforts consider only the implementation of the (p, Θ) line-finding HT.

Hanahara et al.[10] implemented a (p, Θ) line-finding HT that pipelines the p intersection calculation and the accumulator increments. They implemented their system in standard TTL (transistor-transistor logic) medium- and small-scale integration circuits using a Motorola MC68000 as the main processor. The process, which includes edge detection, HT calculation, accumulation, and peak detection, was found to take 0.79 seconds for 1024 feature points. Baringer[11] proposed an architecture called PPPE (parallel pipeline projection engine), which uses the ideas of the Radon transform as a projection operation to produce a real-time hardware implementation of the HT. A set of VLSI (very large scale integration) chips is currently being designed, and achievement of real-time implementation of the Radon transform using only one or two ICs is expected.

The Radon transform of a function is defined as its line integral along a line inclined at an angle Θ from the y-axis and at a distance from the origin. Basically, the Radon transform maps the spatial domain to the distance/angle domain, sometimes referred to as the s/Θ space.

Several authors have investigated the implementation of the HT on currently available SIMD (single instruction multiple data) architectures, a type of digital computer. These architectures usually consist of square arrays of simple processing elements (PEs) connected so that each can communicate with its four or eight neighbors. All processors concurrently execute the same instructions on different items of data.

Li[12] considered two schemes for running his fast HT (FHT) on SIMD architecture. In the first scheme, each PE is assigned an image feature, and the coordinates of a parameter cell are broadcast simultaneously to every PE by a central controller. Each PE decides whether the hypersurface generated by its image feature intersects the cell; if so, the PE sends a vote back to the controller. The votes from each PE can be summed by the central controller and stored for later analysis. In the second scheme, each PE is assigned a volume of parameter space and the image features are broadcast. The choice of method depends on the number of available PEs, the number of image features, and the number of parameter cells. For the standard HT the number of parameter cells increases exponentially with dimensionality of the problem; therefore, the first alternative is likely to be the most feasible.

Little et al.[13] describe a possible implementation of the HT on an architecture called the connection machine. This architecture is similar to the SIMD, but in addition to PEs communicating with near neighbors, a hardware router implements rapid communication between any pair of processors. The architecture is based on a 12-dimensional hypercube such that every processor can be reached from any other processor by traversing, at the most, 12 edges of the cube. Their paper concentrates on aspects of programming and addressing but gives no data on the efficiency gained by using this parallel implementation.

Guerra and Hambrusch[14] presented two efficient algorithms for HT line-finding on an $n \times n$ mesh using the massively parallel processor (MPP). Their first method, the block algorithm, involves partitioning the mesh into submeshes, performing projections in these submeshes, and then combining partial results. Their second method is similar in that it projects by tracing lines through the image in a pipeline fashion. Although this tracing algorithm is asymptotically optimal in terms of complexity, Guerra and Hambrusch expect the block algorithm to outperform it in actual implementation.

The HT has attracted attention from researchers interested in human vision, since the HT is a prime example of the ideas of the connectionist school of artificial intelligence.[15] The unifying principle of this approach to intelligence is that low- and medium-level vision tasks are done by massively parallel, cooperative computations on large networks of simple neuron-like units. Low-level, pixel-based properties, such as edge or gray-level estimates, can be represented by nodes in a separate parameter network. Each node records a measure of confidence for the occurrence of its feature or parameter value; direct connections between the two networks define ways in which nodes can influence these confidence values. Different connection patterns can be used to impose different image-space to parameter-space mappings; i.e., connections can be established so that if many low-level units that lie on a straight line have high confidence, then the higher level unit describing the parameters of this line will acquire a large confidence value. The major characteristic of this implementation is the tremendous number of feature and parameter units needed and the very large number of connections required between them.

Blanford's[16] adaptation of the dynamically quantized pyramid method can be naturally mapped to a parallel pyramid machine. However, one possible problem with this algorithm is that it requires some multiplication and division operations, which are inefficient on the simplest bit serial processors.

Fischler and Firschein[17] showed the HT to be an algorithm that can be implemented on a blackboard or database architecture. They invoke a maxim, which they call parallel guessing, that says that it is often computationally beneficial to try to guess a solution rather than to exhaustively compute a solution. They suggest computing the HT incrementally and then terminating computation when a sufficiently significant parameter peak has been identified.[18]

An approach using statistical signal detection theory is effective for curve detection in digital images corrupted by random noise.[19] This approach is a refinement of the HT and results in improved performance, both in deciding the presence or absence of a curve in the image and in determining the location of an existing curve in the image. Location estimation performance is measured by deriving equations for both the HT and the signal detection theory for the probability of correctly estimating the location of a curve in noise. The performance of these two approaches are compared for various signal-to-noise ratios (SNR) and found to be significantly different for some SNR values.

5.0 Region-Based Segmentation

Region-based segmentation techniques are primarily used to identify various regions with similar features in one image. Region-based approaches are generally less sensitive to noise than the boundary-based methods. However, they can be considerably more complex to implement.[1]

Many region-based segmentation techniques are presented in this section, including region-growing and merging, relaxation labeling, symmetric nearest neighbor, hierarchical segmentation, and shadow boundary segmentation. Several well-known image processing techniques are described in the context of region-based segmentation, such as clustering, pattern recognition, edge-detection, noise reduction, and three-dimensional object recognition. An interesting application of region-based segmentation is discussed last: segmentation of handwritten numerical strings.

5.1 A Note on Color

The analysis of color images has received relatively little attention in computer vision research, even though color plays an important role in human vision and provides useful information for many image analysis applications. One simple, powerful method for region-based segmentation of color images uses edge-preserving filters.[20] The method uses a new measure of color edge information based on histograms of absolute color differences. This measure can be used for smoothing, segmentation, and edge detection. Methods for multi-edge-preserving smoothing and region-based segmentation were also developed.[20] A global histogram of absolute color (or gray scale) differences provides a good measure of edge information in an image, because the likelihood that an absolute color difference occurs in the interior of a region decreases monotonically with increasing magnitude of the difference.

5.2 Region-Growing and Merging

One class of region-based techniques involves region-growing. The image is divided into atomic regions of constant gray levels. Similar adjacent regions are merged sequentially until the adjacent regions become sufficiently different. The crux of this procedure is the selection of the merging criterion. Some merging heuristics follow:

- Merge two regions, Ri and Rj, if $w/Pm > \Theta 1$, where $Pm = min\ (Pi, Pj)$; Pi and Pj are the perimeters of Ri and Rj; and w is the number of weak boundary locations (pixels on either side have a magnitude difference less than some threshold y). The parameter $\Theta 1$ controls the size of the region to be merged. For example, $\Theta 1 = 1$ implies that two regions will be merged only if one of the regions almost surrounds the other. Typically, $\Theta 1 = 0.5$.
- Merge Ri and Rj if $w/l > \Theta 2$, where l is the length of the common boundary between the two regions. Typically $\Theta 2 = 0.75$. The two regions are merged if the boundary is sufficiently weak. This step is often applied after the first heuristic has been used to reduce the number of regions.
- Merge Ri and Rj only if there are no strong edge points between them. Note that the run-length connectivity method for binary images can be interpreted as an example of this heuristic.
- Merge Ri and Rj if their similarity distance is less than a threshold. Instead of merging regions, the segmentation problem can be approached by splitting a given region. For example, the image could be split by a quad-tree approach and then similar regions could be merged.

5.3 Hierarchical Segmentation

Gambotto's[21] hierarchical segmentation algorithm can process all regions (in a region-growing manner) in a parallel and recursive fashion. This algorithm simultaneously computes the statistical properties of homogeneous regions, as well as a gradient estimate over the boundaries of the regions to detect the contours.

The algorithm was applied to a synthetic image that contained four regions; each region was obtained by adding a pseudorandom Gaussian noise to a constant value; the variance of the Gaussian noise was equal to 16, and its dynamic was 190. The algorithm produced excellent results, considering the noise in the image. The results of Gambatto's report are given in two phases: the noisy image is first passed through an initial segmentation and then through a final segmentation phase.

5.3.1 Segmenting Contour Line Images

The Gorman and Weill[22] segmentation algorithm groups contour lines into regions. Their method is based in part on a parallel-adjacency criterion, which is defined in their paper. The algorithm was applied to several contour line images, and the resultant regions were given. The apparent key to this algorithm is the way in which it combines image properties to recognize line regions. The main steps in segmenting contour line images with this algorithm follow:

- Split lines at all junctions (bifurcations and line crossings).
- Perform piecewise straight-line fitting so that each line is comprised of straight-line segments.
- Construct an adjacency list of the straight-line segments. This segment adjacency list (SAL) contains, for each segment, all other segments that meet the criteria for proximity, approximate parallelism, and nonzero overlap with respect to that segment.
- Merge the segments of the SAL into groups on the basis of pairwise similarity of line segments due to the parallel-adjacency criterion. The result is the segment group list (SGL).
- Consider each line in its entirety (made up of the straight line segments) and group the lines, again in pairwise fashion, based on line adjacency and similar composition of line segments from the SGL. The result is the line region list that contains line composition of each contour line region.

The performance of this algorithm is dependent on six parameters: maximum distance tolerance between adjacent segments, minimum overlap tolerance between adjacent segments, maximum angular tolerance between adjacent segments, minimum number of segments per group and lines per region, and average line spacing. The experimental results show that the regions determined by the algorithm, when applied to both synthetic and real images, are consistent.

5.3.2 Double Hierarchy of Fusion

Gagalowicz and Monga[23] report a method of region-growing that defines a double hierarchy of fusion of adjacent region pairs. The first level of the hierarchy is defined by the increasing values of a merge criterion and the second one by the order of the fusion criteria used successively.

This double hierarchy allows the algorithm to create sequentially more and more global segmentations. They showed that it was easy to introduce semantic fusion criteria due to the richness of the information available at each iteration. The algorithm did not derive its strength from the classical criteria used, but from the merge strategy, the order, and the succession of the various criteria.

5.4 Global-Local Edge Coincidence

Hall[24] notes that segmentation may be either edge-based or region-based. The edge-based method works well but is sensitive to local noise. A so-called global method is more stable, since it considers the overall characteristics of the image; however, false regions may often be detected. For perfect region segmentation, the global region boundaries should coincide with the local edges. The global-local edge coincidence (GLEC) segmentation method detects the coincidence of the region boundaries (or global edges) and local edges. Since the global edges are obtained from the global characteristics of the image (for instance, the histogram of the intensity of the image), the local noise edges will not be detected in the global edge map; however, perfect region segmentation is barely obtained by using only the intensity information. Basically, GLEC is a merge-oriented region segmentation method. Two regions are merged if the common boundary between these regions does not match the local edges.

5.5 Implementing Data Structures in Pixel-Based Segmentation

Region-based segmentation methods require the use of other data structures in addition to the original pixel array. For merging, we can represent the regions (at a given stage of the merging process) as nodes on a graph, with pairs of nodes joined by arcs if the corresponding regions are adjacent. The statistics associated with each region can be stored at its node; if it is possible to compute the statistics for a (tentatively) merged pair of regions directly from those for the individual regions, the merging process can be done directly on the graph, without reaccessing the original image. For splitting, the quadrants, subquadrants, etc., can be represented at a given stage of the process as nodes of a quadtree, where the root is the whole image and the quadtree nodes correspond to its quadrants. Here, it is necessary to refer to the original image to compute the statistics of the subregions each time a region is split.

The fact that region-based segmentation methods may require access to the image data in an arbitrary order is a potential disadvantage when the image must be accessed from peripheral storage. Thus, such methods are best applied to small images. However, region-based methods do have a potential advantage: in principle, they can be designed to incorporate information about the types of regions (sizes, shapes, colors, textures, etc.) that are expected to occur in images of a given class; thus, merging or splitting can be inhibited if either would violate restrictions on the expected types of regions. As a classic example, a region-based approach can be used to "grow" or "track" global edges (or curves) in an image, starting from pixels that have high edge magnitudes and accepting new pixels (i.e., merging them with the edge fragments already constructed) if they continue along these edges.

Another advantage of these so-called pixel-based segmentation methods over normal region-based segmentation is that pixel-based schemes can be greatly accelerated if parallel hardware is available. This rapidity is accomplished by dividing the image into parts and assigning a separate processor to segment each part; the processors can share global information about segmentation criteria, if desired, and they may also share neighbor information along the common borders of the parts. In principle, parallelism could also be used in region-based schemes by assigning processors to regions or to sets of regions, but this method would require an extremely flexible interprocessor communication scheme to allow processors that contain information about adjacent regions to communicate. In pixel-based schemes, however, the image can be divided into square blocks, for instance, so that the processor responsible for a given block needs only to communicate with a limited number of processors that are responsible for neighboring blocks.[25]

5.6 Clustering

Clustering refers to a class of algorithms used extensively for image segmentation. Clustering assembles unlabeled data by sets, or clusters, of data points with strong internal similarity. Data point values represent characteristic features of interest such as grayscale, color brightness, contrast, etc. During the cluster operation, the clusters are assigned labels that are mapped back into the image, so that the original pixel values are replaced. These labels can be thought of as "class membership" indicators. Similarity is most commonly measured by a distance function in feature space. It is generally desirable to make this function independent of any relevant image transformations being performed (e.g., rotation, translation, or scaling). A criterion function is also used to measure the clustering quality of any given partition of the image function values.

The basic clustering operation examines each pixel individually and assigns it to the cluster that best represents the value of its characteristic vector. This assignment is done according to the selected measure of similarity between the data point and the criterion function that measures clustering quality. The process is repeated, if necessary, until some condition is satisfied by the current grouping of data points. For example, if similarity between pixels is measured in terms of the distance between the value of initial cluster centers, then the cluster centers are assigned the initial values $M1 = M - S$ and $M2 = M + S$, where M is the mean feature vector as measured over the entire image and S is the standard deviation. Clustering, then, would be achieved in the following steps:

- Assign feature vectors to closest cluster centers.
- Compute new cluster centers.
- Compare new and old cluster centers: if they are close enough, then terminate the algorithm; if not, then iterate the procedure from the second step.

The following issues must be considered during clustering:

- The choice of a similarity measure.
- The choice of a criterion function.
- Determination of the appropriate number of clusters.
- Establishing properties of solutions.

5.6.1 Relaxation Labeling

Relaxation labeling provides an improvement to the traditional clustering technique. Instead of mapping a single cluster label back to each image point, the *probability* that an image point belongs to each of the clusters is mapped back to the image. A relaxation process is applied where similar labels will support each other, whereas different labels will compete over neighborhoods. The probabilities are iteratively updated until convergence is reached.

Relaxation was first introduced by Rosenfeld and Kak,[3] who define it as an iterative approach to segmentation. The approach makes fuzzy or probabilistic classification "decisions" at every point in parallel and at each iteration. It then adjusts these decisions at successive iterations based on the decisions made at the preceding iteration for neighboring points. The technique is called relaxation because it resembles a class of iterative numerical methods. The approach is order-independent and can be greatly accelerated by parallel processing. Since each iteration is parallel, only a few iterations are usually necessary. Relaxation is more powerful than one-shot parallel methods, since its initial classifications are refined at each iteration, based on the local context. This approach makes tentative classifications at each stage and repeatedly reconsiders them, unlike other methods that usually make decisions only once at each point (except in cases where sequential methods allow backtracking).

Relaxation labeling estimates the relative likelihoods of nodes in a graph and then reduces the labeling ambiguities in an image. The problem can be formulated by defining the following: a set of nodes, a set of labels for each node, an initial assignment of probabilities for the labels of each node, a set of arcs between nodes to indicate neighboring relations, a constraint relation between node labels, and an updating rule to refine the probabilistic assignment of labels.

Some problems associated with relaxation labeling include the choice of an appropriate updating formula for the probabilities, difficulty in asserting convergence, and difficulty in establishing properties of the solution.[26]

Kittler and Illingworth[27] reviewed various relaxation labeling algorithms that detailed the need to incorporate contextual information into the interpretation of objects. Their literature review highlighted the following technique. Ullmann (*Trans. IRE(IT)* 8(5):74-81, 1962) exploited constraints imposed by triplets of pattern primitives to substantially reduce the errors that occur with a pattern recognition system after a learning sequence of fixed length. Clowes (*Artificial Intelligence* 2:79-116, 1971) and Huffman (*Machine Intelligence* 6:295-323, 1971) used constraints between straight-line segments to eliminate nonsensical interpretations of an ideal line drawing representing a set of polyhedra.

The pioneering work in relaxation labeling is normally credited to Waltz (*The Psychology of Computer Vision*, McGraw-Hill, 1957), who considered the problem of line-drawing interpretation studied earlier by Clowes and Huffman. His formulation of the consistent labeling problem allowed only unambiguous interpretation of line segments, achieved by sequentially filtering out inconsistent label pairs of connected segments. This approach was then popularized by Rosenfeld et al. (*IEEE Transactions SMC* 6(6):420-433, 1976), who showed that Waltz's filtering can be carried out in parallel and could therefore be implemented as a network of processors, each associated with one object in the image.

However, the problem considered by Waltz is somewhat unrealistic, since no information as to the identity of each line segment is assumed to be available. In practice, when analyzing real imagery, it is reasonable to assume that some useful information could be extracted from the raw image data. Conversely, the edge representation of a scene is unlikely to look like an ideal line drawing. Rosenfeld et al. argued that the line-drawing interpretation is better formulated in the continuous domain than as a discrete relaxation, even though the latter enforces unambiguous labeling. Fuzzy set and probabilistic frameworks were considered in this respect, but the latter seems to have attracted the most attention to date.

Relaxation has been applied to general cases of improving multi-label classification of multispectral data, especially remotely sensed data and color images. Most of these applications have been approached in the same way. Model clusters are defined in the measurement hyperspace either by automatic clustering or by hand segmentation of ground truth data. The initial label probabilities are calculated as a simple function of distance between the models and the pixel data. The interrelationships among labels are derived empirically from ground truth data by measuring and globally averaging transitional probabilities, correlations, or the mutual information of local pixels. Upon applying these methods, most investigators reported a sharp initial decrease of several percent in classification error rates, followed by a smaller increase as the process converges to a stable solution.

Relaxation labeling has been well used in matching problems, such as two-dimensional (2-D) shape matching or stereo correspondence. Representative feature points, such as corners, are extracted from a template shape and a corresponding real-world image. Initial probabilities can be assigned on the basis of the degree of match between these chosen features, and then these probabilities can be iteratively reinforced on the basis of the occurrence of matches between other features on the template and their respective real-world features.

The stereo correspondence problem is similar. Features are extracted in two images; the features of one image can be regarded as nodes of a graph, and the features in the second image are possible labels for the nodes. The initial probabilities of labels is a simple function of the distance between node and label points in the two images. Node and label assignments are then compatible if similar neighborhood states exist in both images. Although these relaxation schemes might not have great computational advantages relative to other standard matching methods, they are more tolerant of image distortion. In addition, the reinforcing processes are local, so missing matches can be tolerated; therefore, occluded objects can be recognized.[20]

5.7 Symmetric Nearest Neighbor

A powerful symmetric nearest neighbor (SNN) filter can be used for edge-preserving smoothing of gray-scale images. It uses both spatial and nearest-neighbor constraints on image pixels to smooth an image. To compute the gray value for the center pixel in a neighborhood, it selects half the number of pixels in the neighborhood: from each pair of pixels located symmetrically on opposite sides of the center pixel, the one that is closer in gray value to the center pixel is selected. In case of tied pairs, the mean of the pair is used. Then the mean value of those selected is substituted for the original value.

To find SSNs for a multiband image, the following procedure was proposed:

• Compute the multidimensional cumulative histogram of absolute color differences.

• Compute the absolute color differences between the two pixels in the pair and the central pixel, for each symmetric pair of neighbors in a neighborhood.

The pixel with the higher frequency in the cumulative histogram (smaller color difference) is selected. In case of ties, the mean of the symmetric pair is used. The mean of the values of the set of pixels selected is assigned to the center pixel on each band.

The color-SNN filter can be iterated, and it converges without producing artifacts; normally only minute changes occur in the image after two to three iterations. The hardware implementation of the color-SNN using a 3×3 neighborhood should be almost as straightforward as in case of the basic SNN filter.

5.8 Connected Components Algorithm

Color segmentation combines edge-preserving smoothing with a simple connected components (CC) algorithm. Using the CC, adjacent pixels are said to be connected if the likelihood or frequency of the color difference is large (so the magnitude of the difference is small). The algorithms make use of the combined information in a two-band image.

First, the image is smoothed by the color-SNN filter. Normally, about three iterations of 3×3 filtering are needed to sharpen edges and smooth homogeneous areas. To make edges even sharper and to avoid mismerging of regions at some critical points, bands are edge-enhanced with a gray-scale filter known as a MINRANGE filter. The MINRANGE replaces the center value of a 3×3 neighborhood with the mean of the 4-pixel corner subgroup that has the smallest range. Because sharpening is applied to almost completely smoothed bands, no artifacts are generated.

After the color image is smoothed, it is segmented by a two-pass CC algorithm. Here, adjacent pixels are said to be connected if the likelihood or frequency of their absolute color differences are sufficiently large. The only parameter is a threshold, which is expressed as a percentile of frequencies supplied by the user.

The two-band histogram of absolute color differences is then computed. It is first converted to a two-band histogram of cumulative frequencies and then to percentiles of their distribution.

The two passes of the CC algorithm are the same as those of the standard algorithm for binary images. Row by row, pixels are assigned labels by comparing each pixel with the four adjacent pixels above or to the left, which have already been labeled as the image is scanned from top to bottom and from left to right. Then, in the second pass, the pixels with component-equivalent labels are relabeled uniquely.

Therefore, the CC algorithm provides a new measure of edge information for color images based on cumulative histograms of absolute color differences. CC-based methods may be used for edge-preserving smoothing and segmentation. Such methods are relatively simple: they do not have to process many parameters, and they give good results for many different types of images even when using the same set of parameter values for these different images.

5.9 Munsell Color Coordinate System

Tominaga[28] presents a method for segmenting a color image into meaningful regions using three different perceptual attributes of color: hue, lightness, and saturation. This segmentation technique is based on a recursive thresholding method using three histograms, one to depict the range of each attribute. The Munsell color coordinate system (hue, value and chroma) is the color space used to best represent human color perception. This color specification method predicts the color perception of a measured image. A practical segmentation procedure is then presented. A set of subregions with uniform color is extracted from the recursive thresholding on the peak of the histogram set. This operation is repeated to generate a sequence of uniform color regions in the image.

Tominaga describes the segmentation procedure in six key steps:

1. Histograms are computed for each attribute of hue, value, and chroma, using either the entire image as one region or using specified regions within the image. The histograms are smoothed by a moving average to eliminate small peaks.

2. The most significant peak is found in the set of histograms. Peak selection is based on a shape analysis performed on each peak in the histograms. First, some clear peaks are isolated as candidates. Next, the following criterion function (f) is calculated for each candidate peak:

$$f = Sp/Ta\ (100/FWHM)\,, \qquad (3)$$

where Sp represents a peak area between two valleys, V1 and V2 (the lower and upper bounds, respectively), and $FWHM$ is the full width of the peak at half-maximum. Ta denotes the overall area of the histogram, that is, the total number of pixels in the specified image region.

3. Thresholding of a color image is executed using two threshold values derived from the lower bound V1 and the upper bound V2 for the most significant peak in the set of three histograms. This thresholding operation partitions an image region into two sets of subregions. One set consists of subregions corresponding to the color attributes within the threshold limits; the other is a set of

subregions with the remaining attribute values. Only the former set is extracted.

4. The thresholding process is repeated for the extracted subregions. The area of subregions decreases with each succeeding threshold. This process leads to the detection of the most significant cluster. If all the histograms become monomodal, then the cluster detection is finished. One-step segmentation is thus completed, and a suitable label is assigned to the latest extracted subregions.

5. The image labeled by the above segmentation is smoothed on the basis of pixel connectedness. This refinement is intended to smooth out noisy boundaries and eliminate small regions and short lines. The 8-connection property is used in this smoothing algorithm. The operation uses multilevel, rather than binary, smoothing.

6. Steps 1 through 5 are repeated for the remaining regions. The segmentation process is terminated when one area is sufficiently small in comparison to the original image size, or when no histogram has sufficient peaks. The remaining unlabeled pixels are regarded as noisy fluctuations and are merged into neighboring labeled regions of similar colors. The mean values of the color specifications are computed, and a color difference formula is used to choose the nearest color region.

This method has been developed for segmenting a color image into regions with perceptually uniform colors by means of the three Munsell color attributes. The color specification process was first presented to predict color perception of measured images. Experimental results presented by Tominaga demonstrate the feasibility of this method.

5.10 Pattern Recognition Using the Commission Internationale de L' Eclairage Color System

A new computational pattern recognition technique is being used by Celenk and Smith[29] to segment color images of natural scenes. This technique is an unsupervised operation that detects image clusters using one-dimensional (1-D) histograms of the color or feature coordinates in the selected uniform color system or constructed feature space. The detected clusters are then extracted by projecting and separating the classes two at a time. This method tends to integrate the statistical data analysis concept of pattern recognition theory with the fundamental premises of human color perception. It does not operate blindly: an image segment is accepted only after all of its spectral neighbors in the uniform color space are considered. Although this algorithm was developed as an unsupervised operation, it can be implemented in a supervised mode by introducing external knowledge or training prototypes to the system.

The authors' recursive procedure tends to integrate the fundamental methods of human color perception with the statistical data analysis concept of mathematical pattern recognition. It detects image clusters efficiently and determines their boundaries correctly in the CIE uniform color system (L^*, a^*, b^*), which is selected as the feature space. With each iteration, the algorithm detects the most prominent cluster or mode and all of its spectral neighbors in the uniform color space. To make the detection process computationally efficient, the procedure first approximates the underlying clusters with circular-cylindrical volume elements. This approximation provides the best estimates of the 3-D color distributions for these clusters in accordance with the color perception mechanism of the human eye. The boundaries of each volume element are composed of two constant luminance planes, two constant circular chroma cylinders (or loci), and two constant angular hue planes (or loci). Each is derived from the sequentially constructed 1-D zero-, first-, and second-order parametric histograms of the cylindrical coordinates (i.e., L^* = lightness, H^* = hue, C^* = chroma) of the uniform color space. To extract the image region that correctly corresponds to the most prominent cluster (i.e., without significantly deforming its spatial configuration in the image domain), the color vectors lying within the range of the 3-D color distributions of this mode and one of its spectral neighbors are projected onto a line so that the projected color points are well separated and clustered for 1-D thresholding. This projecting and thresholding process is repeated until the selected mode is isolated from all of its neighbors in the uniform color space. The orientation of the line used for every projection is determined according to the Fisher criterion postulated as the measure of effectiveness of a linear discriminate function.

Celenk's and Smith's technique minimizes the error rate associated with region isolation in the feature space. The computational cost involved in region (segment) extraction and cluster detection is significantly reduced by using only 1-D image histograms.

The 1976 CIE (L^*, a^*, b^*) uniform color system used in Celenk's and Smith's procedure was chosen as the feature space to best approximate human vision perception. This system implicitly satisfies the condition that numerical differences in the feature space should be directly proportional to perceptual differences in the human visual system.

5.11 Motion-Based Segmentation

Bouthemy and Rivero[30] present a new approach to the motion-based segmentation problem. The designed formalism includes 2-D motion models and relies on explicit partial motion information through a stochastic approach, which allows the computation of a likelihood ratio test embedded in a split-and-merge procedure. Therefore, regions are structured according to motion homogeneity criteria considered in a hierarchical way: segmentation starts with as simple a motion model as possible, and after a complete iteration cycle, a more elaborate motion model can be taken into account (e.g., a linear one).

The authors developed a new region-based segmentation method that relies on motion information and follows a stochastic approach. The criterion for a spatio-temporal homogeneity decision is the computation of a proper likelihood ratio test based on some motion model. They present a constant motion model and a linear model in a hierarchical manner within a split-and-merge procedure.

5.12 Edge Detection and Noise Reduction

5.12.1 Statistical Theories of Edge Detection

Edge detection is critical to segmentation and, thus, to computer vision, since edges are essential to the segmenting of regions. Results of Huang's and Tseng's[31] research on edge detection are noted here. They use a statistical theory of hypothesis testing to apply filtering and edge detection simultaneously to a noisy image. A simple decision rule is derived, and the application of this result to more complicated situations is discussed in detail. The decision rule can decide whether there is an edge, a line, a point, a corner edge, or just a smooth region in a given small neighborhood. Computational by-products of the decision rule are the mean and the variance of the neighborhood, which can be used for split and merge analysis. Utilization of the mean can essentially filter the neighborhood pixels. Huang's and Tseng's new technique was implemented to run on a VAX-11/780, and their experimental results indicate a high feasibility for the method. The generalization to more generic cases is also discussed: Haralick's sloped-facet model seems to be the most suitable case. The decision rules used by Huang and Tseng are computationally intensive, but besides simply detecting edges they also yield line and point detection, as well as good estimates of the region's mean and variance for further analysis.

5.12.2 Markov Random Fields

Goutsias and Mendel[32] also address the issue of noisy images. They use a doubly stochastic image model and assume that the image is the sum of the realizations of two independent random fields: the uncorrupted image and the noise field, both of which consist of independent, identically distributed, Gaussian random variables. Their image segmentation technique represents the image with a semi-Markov random field that has been corrupted by additive white noise. Markov random fields (MRFs) are 2-D, noncausal, Markovian stochastic processes.[33] Goutsias and Mendel develop an adaptive Bayesian parameter estimation/image detection algorithm to estimate the unknown image and its underlying parameters in an optimal manner. Their proposed algorithm is demonstrated during the smoothing and segmenting of two 4-gray-level real images. The semi-MRF is defined in terms of two, not necessarily independent, random fields: an MRF that describes the statistical behavior of the boundary pixels (pixels that are located at the boundary between adjacent regions of the image), and a random field that describes the statistical behavior of the regional pixels (pixels that are located inside a region) of the image.

Another use of MRFs is documented by Murray and Buxton,[34] who present results of computer experiments with an algorithm to perform scene decomposition and motion segmentation from visual motion or optic flow. The maximum a posteriori (MAP) criterion is used to formulate the best segmentation or interpretation of the scene, where the scene is assumed to be made up of some fixed number of moving planar surface patches. Their Bayesian approach has two prerequisites: specification of prior expectations for the optic flow field, which Murray and Buxton model as spatial and temporal MRFs, and a way of measuring how well the segmentation predicts the measured field. The MRFs incorporate the physical constraints that objects and their images are probably spatially continuous, and that their images are likely to move quite smoothly across the image plane. To compute the flow predicted by the segmentation, a method for reconstructing the motion and orientation of planar surface facets is used. Then the search for the globally optimal segmentation is performed using simulated annealing. Their most important result was the formulation of scene segmentation from visual motion as an optimization problem that is weakly constrained, or guided, by prior physical expectations.

Two serious problems were noted in Murray's and Buxton's experiment. First, the current implementation of the segmentation process requires the specification of the number of objects likely to be found. The more serious problem is that the algorithm is computationally inefficient.

5.12.3 Hierarchical Edge Detection

McLean and Jernigan[35] discuss the design of efficient edge-detection operators. The need for such operators is reviewed and a set of design and performance criteria is developed. These criteria are then used to evaluate existing edge-detection techniques, as well as to suggest some new approaches to this important problem.

The following requirements for an edge detector are considered:

- capable of working in a purely local context,
- efficient when applied in any order (it cannot derive efficiency by exploiting redundancies when applied in a particular fashion),
- insensitive to the orientation or to the magnitude of the edge,
- work well in the presence of noise,
- relatively insensitive to threshold specifications.

McLean and Jernigan describe a method of hierarchical edge detection (HED), which is based on nonlinear edge detectors. HED is a two-step process consisting of a coarse-oriented gradient measurement followed by the application of a particular orientation of one of the efficient 1-D edge detectors. The gradient preprocessing step serves as an initial filter, so that only those pixels which exceed the gradient threshold become candidates for the more expensive, oriented, edge-detection scheme. In this manner, the HED scheme becomes very efficient: the time required is related to the amount of edge and noise activity that exists within the image.

The burden of edge detection, then, falls heavily on the gradient preprocessing scheme used, since it must not remove true edges and must block as many nonedged pixels as possible. Also, it must be directionally sensitive, so that the correctly oriented edge detector will be applied. Simple logic tests were used to determine edge orientation after the two gradients were thresholded.

McLean's and Jernigan's HED-based method is shown to perform well. An encouraging result of their work is that effective edge detection can be performed while maintaining a highly structured, highly localized approach to this aspect of image processing. The HED scheme is well suited for inclusion in systems that encompass multiple levels of processing. A future research task, and follow on to this effort, could investigate the possibility of adapting the edge extraction process to follow contours, thus eliminating wasted processing.

5.13 Splitting and Merging

5.13.1 Attributed String Matching and Merging

Tsai and Yu[36] give a new structural approach to shape recognition that utilizes string matching with merging. They first present disadvantages of conventional symbolic string matching, which uses changes, deletions, and insertions. Attributed strings are suggested for matching, where each attributed string is an ordered sequence of shape boundary primitives that represents a basic boundary structural unit (a line segment) with two types of numerical attributes (a length and a direction). A new type of primitive edit operation, called a merge, is then introduced. The merge can be used to combine and then match any number of consecutive boundary primitives in one shape with those in another. The resultant attributed string matching with merging approach is then shown to be useful for recognizing distorted shapes. Experimental results are also given to prove the feasibility of this approach.

There were occasional erroneous classifications for the following reasons:

- The number of boundary primitives for each shape was limited to 10 to increase processing speed.
- The image resolution was low (128×128) for the relatively high shape complexity.
- The three images selected for this test (pliers) were similar in shape.
- The selection of thresholds and constants used in the algorithms was not optimal.
- Poor computational accuracy resulted from the exclusive use of integer calculations (to avoid the relatively low-speed real-number computations on the available microcomputer).

The authors also test attributed string matching with merging on images with no occlusion. They suggest that some extensions to this approach are possible, e.g., the improvement of operation cost functions, the inclusion of primitive splitting into the matching algorithm, more intelligent solutions to the shape orientation problem, applications of the proposed approach to recognizing occluded shapes, etc.

5.13.2 Segmenting Range Imagery into Planar Regions

A fast technique for segmenting range imagery into planar regions is discussed by Taylor et al.[37] Range images provide direct measurements of the 3-D surface coordinates of a scene. The technique rapidly divides range imagery surfaces into regions that satisfy a common homogeneity criterion. Key features that enhance the algorithm's speed include the development of appropriate region descriptors and the use of fast region comparison techniques for segmentation decisions. Their split-and-merge algorithm bases its homogeneity criterion on a three-parameter, planar surface description, in which the three parameters are two angles (to describe the orientation of the normal to the local best fit plane) and the original range value. Speed is achieved because both the region splitting and the rejection of merge possibilities can often be based on simple comparisons of these two orientation parameters.

A fast, but more complex, region-to-region range continuity test is also developed for use when the orientation homogeneity tests are inconclusive. The importance of merge ordering is discussed: one particularly effective ordering technique, which is based on dynamic criteria relaxation, is demonstrated within their paper. Sample segmentations of simple and complex range data images are also shown, and the effects of noise and preprocessing are examined.

The authors state that splitting and merging are done conservatively in their algorithm and produce oversegmented images. Extra region boundaries are detected, but no major boundaries are broken; thus, the oversegmentation is due to fragmentation of true regions. Additional merge phases with relaxed homogeneity criteria are used to reduce this fragmentation. No general procedure was offered for selecting these values, but it was noted that the values are not very sensitive for planar data, as long as the general relaxation trends described by the authors are followed.

Overall, the authors' algorithm can rapidly segment an object in a range image into a surface composed of planar regions. The results from this algorithm show that it could be an initial step to a more complex merging technique.

5.13.3 Segmenting Aerial Photographs

A method of segmenting aerial photographs, described by Laprade,[38] approximates the image intensity surface with planar facets. The approximation is accomplished by using a split-and-merge approach that is somewhat different than those previously mentioned. A combination of an F-test and a mean predicate is used to test the uniformity of regions. When two regions are merged into a new region, nine variables are needed to compute the least-squares plane (the components of the 3×3 matrix in the normal equations) for the new region. These variables can be computed by adding the corresponding variables for the individual regions. This process leads to an efficient algorithm.

Features that differ from the standard split-and-merge algorithm are described in Laprade's paper. One such feature is the use of multiple predicates, specifically the mean predicate and F-test, at certain stages of the algorithm. Regions are allowed to merge with other regions during the region-growing process, as opposed to the usual practice of allowing a growing region to absorb only quads that have not been assigned to another region. Multiple predicates were used because the F-test is not sensitive to the magnitude of differences between regions, only to their uniformity.

In other words, two regions may differ only slightly in their means or slopes, but if the residuals of these two regions from their facet fits are much smaller than these differences, then the F-test will classify these regions as being distinct. In addition to the F-test, two regions were compared by looking at the maximum difference between their facet representations at points where the regions were adjacent.

In general, the results of this technique offer an improvement over the flat facet results. However, there is one problem associated with this technique: the splitting procedure finds very few uniform areas from which to grow regions. Such oversegmentation is necessary to ensure that quads, which include edges of regions, are split. If one region contributes only a small part of a quad's area, then the threshold that controls splitting must be very tight to ensure that splitting occurs. If the output of an edge detector is used to ensure that quads containing edge pixels are split, this key problem may be greatly reduced.

5.13.4 Other Split-and-Merge Approaches

Taylor et al.[39] developed a technique to rapidly divide surfaces in range imagery into regions that satisfy a common homogeneity criterion. The result is a segmentation of the range information into approximately planar surface regions. Key features that enhance the algorithm's speed include the development of appropriate region descriptors and the use of fast region comparison techniques for segmentation decisions. Their algorithm takes a split-and-merge approach, where the homogeneity criteria is based on three planar surface description parameters: two angles (to describe the orientation of the normal to the local best-fit plane) and the original range value. Speed is achieved because both region-splitting and the rejection of merge possibilities can often be based on simple comparisons of only the two orientation parameters. Another fast, but more complex, region-to-region range continuity test is developed for inconclusive orientation tests.

In Taylor et al., the concept of the split-and-merge technique is extended from gray-level imagery to range imagery. The algorithm segments range images into a set of planar surface regions by using efficient planar region comparison and multiple merge phases. Splitting and merging are done conservatively, yielding oversegmented images. Extra region boundaries are detected, but no major boundaries are broken; thus, the oversegmentation is due to the fragmentation of true regions. Additional merge phases with relaxed homogeneity criteria are then used to reduce this fragmentation. No general procedure for selecting these values is offered; however, it is suggested that the values are not very sensitive for data that are actually planar, as long as the general, indicated, relaxation trends are followed as instructed by the authors.

The algorithm is effective in the presence of a large amount of noise; for this situation, image filtering becomes important. Two central filtering techniques are discussed; mean filtering is used first, followed by median filtering. Also, varying window sizes are used when the noisy image data are filtered. Other filtering possibilities are suggested, including filtering in parameter space and a hierarchical, multiple-window, planar estimation scheme.

Merging options available for the split-and-merge algorithm have varying effects on the results:

- Forcing single-pixel regions to merge with their best-match neighbors improves the quality of the segmentation, particularly at the boundaries of true regions.
- Allowing the algorithm to merge regions as they are encountered, rather than in descending order of size, work best.
- Considering the neighbors of a region by angle value closeness for merging slightly degrades the algorithm's performance.
- Performing multiple mergings with successive relaxed homogeneity criteria dramatically improves the segmentation results.

The results from Taylor's algorithm suggest that it is a potential "front-end" stage for a more complex merging technique.

5.14 Object Recognition in 3-D Space

Hoffman and Jain[40] describe the recognition of objects in 3-D space for use in computer vision systems. Range images, which directly measure 3-D surface coordinates of a scene, are well suited for this task. The authors present a procedure to detect connected planar, convex, and concave surfaces of 3-D objects. Their algorithm is implemented in three stages. The first stage segments the range image into "surface patches" by a square-error, criterion-clustering algorithm that uses surface points and associated surface normals. The second stage classifies these patches as planar, convex, or concave; the classification is based on a nonparametric statistical test for trend, curvature values, and eigenvalue analysis. In the final stage, boundaries between adjacent surface patches are classified as crease or noncrease edges, and this information is then used to merge compatible patches and produce reasonable faces of the object(s).

The authors demonstrate that a square-error, criterion-clustering algorithm performs well for segmenting a variety of range images into patches. Information is provided about the geometric structure of objects by producing surface patches that do not cross over natural jump or crease edges. They chose clustering to implement their segmentation phase of the algorithm because clustering was found to perform better than any edge-based technique. Among all possible clustering algorithms, the authors chose one, appropriately named CLUSTER, which was developed by Dubes and Jain.[41]

Hoffman's and Jain's procedure is useful for object representation and recognition based on surface primitives. The authors used the method specifically to find natural object faces in a range image. The procedure segments and classifies range images and merges the resultant surface patches. The clustering algorithm chosen was shown to be effective over a variety of range images for partitioning an image into surface patches that do not cross over crease or jump edges. A classification of these surface patches as planar, convex, or concave is strongly based on a nonparametric statistical test. Although the test is simple in concept, it proves to be a powerful tool for this application. A disadvantage of the nonparametric test, however, is the need for moderate sample sizes, which, in this technique, translates into moderate patch sizes. The decision tree for patch classification is designed to fall back on curvatures and eigenvalues if a patch is too small to make a meaningful decision by the trend test. A crease edge-detection technique is used to guide the reconstruction of the natural object faces, which were oversegmented by the cluster technique. It was noted that the eigenvalue threshold may have to be increased for those images in which only a small portion of the available depth values are used.

5.15 Shadow Boundary Segmentation

Hambrick et al.[42] have documented a new technique to interpret arbitrarily shaped surfaces by segmenting and labeling the shadow boundary. The technique is called the Entry-Exit Method of Shadow Boundary Segmentation, and its distinguishing attributes can be summarized as follows:

- Extracts shape-related information from the shadow cast by arbitrarily shaped objects on known surfaces.
- Works independently of a priori knowledge of the scene, requiring only the shadow boundary and the illumination vector.
- Provides a general structure for shadow boundaries by identify-identifying the basic segments and establishing the relationships among them.
- Defines the minimum set of segment types required to describe and interpret a shadow.
- Delineates the possible identities of isolated boundary segments.
- Automatically recognizes ambiguities caused by occlusions, coincidences, and intermediate errors.

The shadow-handling technique is based on the key principle that each point on a shadow boundary is either an entry or an exit point. That is, a light ray projected across the boundary would either enter or exit the shadow at that point. Segments consisting of entry points are called entry segments; they face toward the light source. Segments of exit points are called exit segments; they face away from the light source. A pair of entry and exit segments whose end points are aligned along light rays compare a shadow-making line and its corresponding shadow line. An exit segment connected to the shadow-making line is an occluding line. An entry segment connected to the shadow line is the shadow of a hidden shadow-making line.

A thresholding technique was developed by Perez[43] for segmenting digital images with bimodal reflectance distributions under nonuniform illumination. The algorithm works in a raster format, thus making it an attractive segmentation tool in situations requiring fast data throughput. The theoretical base for this algorithm is a recursive Taylor expansion of a continuously varying threshold tracking function.

5.16 Segmentation of Handwritten Numerical Strings

Shridhar and Badreldin[44] give a context-directed segmentation algorithm for handwritten numerical strings, in which connected numerical strings are split into their key components. The algorithm is hierarchical in that it tests various hypotheses ranging from the case in which the numerals are completely isolated to that in which the numerals may be connected, touching, or existing in overlapping fields.

Test results for this technique revealed that on 200 numerals, an accuracy of 92% was obtained. The recognition errors were mainly due to the pseudofeatures generated by the connecting tail that was still present after segmentation. The authors felt that these errors could be reduced by modifying the recognition algorithm to account for the pseudofeatures. Errors could also be reduced by providing information to the recognition stage segmenter that would indicate where the numerals were disconnected, thus allowing the algorithm to predict where the features of each numeral might be computed.

Many assumptions were made in this test. The algorithm assumes that the numeral strings were written in a "normal" way, i.e., each numeral in the string had roughly the same height. It was also assumed that the numerals were written on a specified line with orientations limited to 20 degrees from the vertical. Finally, the number of numerals in the string must have been specified prior to processing.

6.0 Template-Matching Segmentation

One direct method of segmenting an image is to match it against templates from a given list. The detected objects can then be segmented out, and the remaining image can be analyzed by other techniques. This method can be used to segment busy images, such as journal pages that contain text and graphics. The text can be segmented by template-matching techniques and graphics can be analyzed by boundary-following algorithms.[1]

In 1980, Tsuji et al.[45] documented a dynamic scene analyzer that separated moving objects (such as animated films) from the background and analyzed their motion patterns in dynamic line images. Since the objects move and rotate in a 3-D world, occlusion often occurs, and the shapes, sizes, and structures of the moving object images change from frame to frame. The background and stationary objects may also move in the images due to movements of the camera while tracking interesting objects. The task of the analyzer is to segment the scene into meaningful constituents and to obtain a structural description of each object that contains properties, spatial relations, and motion patterns.

This flexible template-matching method finds correspondence between regions and their respective segments in a sequence of input frames. Also, the analyzer tracks the moving regions and segments within the dynamic images. A similarity test of segment movements then detects the background movement and classifies the regions into stationary and nonstationary. Each region in the latter group is further labeled as partly occluded, false, or moving by examining both the motion patterns of its segments and the temporal change of its structure. Finally, the analyzer merges the segments of each moving object into groups with similar motion patterns to obtain a meaningful partition that corresponds to its components, such as hands or legs.

The authors conclude by stating that the system is primitive because a simple scene model is used. They list important problems for future related research, such as analysis of the rotations in the 3-D world, analysis of the structural change of the line image, semantic interpretation of moving and stationary regions, and understanding the meaning of the movements.

7.0 Texture Segmentation

Texture segmentation becomes important when objects in a scene have a textured background. Since texture often contains a high density of edges, boundary-based techniques may become ineffective unless the texture is filtered out. Clustering and region-based approaches applied to textured features can be used to segment textured regions. In general, texture segmentation and classification is a complicated problem. Use of a priori knowledge about the existence and kinds of textures that may be present in a scene can be beneficial when applied to practical problems.[1]

Raafat and Wong[46] present a new method for image segmentation and region classification based on the texture content of different regions in an image. This technique uses a new measure of texture information to initiate texturally homogeneous core regions. Next, the information measure, together with a new texture distance measure (known as the event set distance) is used to direct the growth of various homogeneous regions. Since the texture information measure reflects both the local and global properties of an image, the segmentation process is highly adaptable to various images. The event set distance is defined over a set of gray-level and gradient vector histograms derived from the texture content within image blocks. The method is data-directed, computationally efficient, and operationally flexible to accommodate various textural properties and distances. Their algorithm for segmenting and classifying textured images is based on the low-level vision approach, where no a priori knowledge is available about the number and types of textures present in the image. This technique is a region-growing method directed by the texture information inherent in various regions of the image (i.e., it is based solely upon the relative visual characteristics of the image). The authors note that from experiments with this technique, the algorithm proved effective and efficient.

7.1 Shift-Match Method

An approach to the segmentation of dynamic scenes that contain textured objects moving against a textured background is presented by Jayaramamurthy and Jain.[47] Their multistage approach first uses a differencing operation to obtain active regions in the frames that contain moving objects. In the next stage, an HT technique is used to determine the motion parameters associated with each active region. Finally, the intensity changes and motion parameters are combined to obtain masks of the moving objects.

The approach to recovering masks of moving objects is referred to as the "shift-match method," which does not require prior segmentation of frames. This technique depends solely on motion to obtain segmentation. It seems to have performed well in a textured environment, even in the presence of occlusion.

7.2 Segmentation Using Temporal Textures

Samy et al.[48] give an image sequence segmentation algorithm for the analysis of dynamic textured scenes. A measure of time-varying textures, based on classical spatial texture measures, and temporal filters to enhance moving regions are also given. Their segmentation algorithm is a straightforward extension of real-time target-tracking algorithms based on adaptive statistical clustering.

Hyde et al.[49] give a means of producing a multiresolution, multipredicate representation of image data. They demonstrated that object segmentation and classification can be represented in the context of image interpretation. In the same computational process, they also showed that multiresolution data can be provided at successively decreasing resolutions to provide region segmentations or, equivalently, edge segmentations as required. This method has also been used for color segmentation and extended to the temporal domain to provide optical flow information.

7.3 Markov Random Fields in Texture Segmentation

Cohen and Cooper[33] suggest simple, parallel, hierarchical and relaxation algorithms to segment noncausal MRFs. Two conceptually new algorithms are presented for segmenting textured images into individual regions. The data from the regions are then modeled as an MRF. The algorithms are designed to operate in realtime when implemented on parallel computer architectures.

A doubly stochastic representation is used in the image modeling process. Cohen and Cooper use a Gaussian MRF to model textures in visible light and infrared images, and an autobinary (or autoternary) MRF to model a priori information about the local geometry of the textured image regions. For image segmentation, the true texture class regions are treated a beforehand either as completely unknown or as a realization of a binary (or ternary) MRF. In the former case, image segmentation is realized as true maximum, likelihood segmentation. In the latter case, the segmentation is realized as true maximum, a posteriori likelihood segmentation.

7.4 Texture Segmentation Using Fractal Geometry

Keller and Chen[50] give a new method for estimating the fractal dimension from image surfaces and show that the method describes and segments generated fractal sets well. Since the fractal dimension alone is not sufficient to characterize natural textures, a new class of texture measures based on the concept of lacunarity is defined and used in conjunction with the fractal dimension to describe and segment natural texture images. They also developed new methods for computing the fractal dimension and lacunarity. Finally, they state that equivalent performance could be obtained by using a supervised segmentation algorithm and perhaps including other texture features.

8.0 Discussion

Several digital image segmentation experiments were recently completed at the Naval Oceangraphic and Atmospheric Research Laboratory using acoustical imagery. These experiments confirmed the hypothesis that combinations of digital image segmentation techniques must be used to adequately differentiate common geoacoustic regions.[51] The findings are briefly summarized:

- Texture-based segmentation can be applied to acoustical imagery to yield additional segmentations as compared to simply using the intensity values alone.
- Texture measures can be applied to an acoustical image to yield texture bands that can be combined to form a multiband image, which then can be used as input to these conventional digital image segmentation techniques.
- Clustering techniques are effective segmenters for acoustical imagery and yield complementary segmentations when combined with the multitexture band image. This type of hybrid segmentation was effective because it combined the texture of spatial statistical information gained from texture-based segmentation, as well as using a region-based clustering segmentation algorithm.

9.0 Summary and Conclusions

Six principal categories of digital image segmentation have been surveyed, with emphasis on the techniques appearing in the technical literature that relate to Navy seafloor segmentation/classification. Many of the papers provided new algorithms for addressing particular segmentation tasks.

Specific conclusions from this study are as follows:

• Hybrid techniques (a combination of two or more of the discussed segmentation techniques) should prove effective on some seafloor segmentation/ classification problems—for example, the combination of region-based segmentation advantages (being adaptive dependent on the region) with the benefits of the texture-based techniques, which normally require a priori knowledge for full utilization.

• Current NOARL research programs can build on the texture-based approaches outlined, as well as the material on region-based segmentation and Hough Transform utilization for improved edge detection.

• Real-time implementation of the region-based techniques will require special emphasis on the merging rules, since this task is computationally intensive.

• Amplitude thresholding alone will not provide the discrimination necessary for geoacoustic province selection given sidescan sonar imagery, since more information than the histogram alone must be examined.

• The most useful boundary-based segmentation technique for current seafloor acoustic imagery exploitation is the Hough Transform (the formulation which handles vertical lines also).

• Three-dimensional object recognition techniques described by Hoffman and Jain[40] should prove useful for 3-D geoacoustic province selection for sidescan sonar imagery.

• Template-matching segmentation techniques could be useful for very simple and well-defined regions but not for the rapidly changing ocean bottom and water column.

• Fractal geometry (e.g., fractal dimension) can be used to augment other segmenting descriptors for seafloor acoustic imagery. As noted in this paper, fractal dimension will not sufficiently characterize natural textures.

10.0 References

1. Jain, Anil K. (1989). *Fundamentals of Digital Image Processing*. Englewood Cliffs (NJ): Prentice Hall.

2. Gonzalez, Rafael C. and Paul Wintz (1987). *Digital Image Processing*. Addison Wesley.

3. Rosenfeld, Azriel and Avinash C. Kak (1982). *Digital Picture Processing*. Orlando (FL): Academic Press.

4. Shapiro, S. D. (1972). Transformations for the computer detection of curves in noisy pictures. *CGIP* 4:328–338.

5. Shapiro, S. D. (1976). Aspects of transform method for curve detection. *Joint Workshop on Pattern Recognition and Artificial Intelligence*, p. 90–97.

6. Shapiro, S. D. (1974). An extension of the transform of curve detection for textured image data. *Third IJCPR*, Coronado, p. 205–207.

7. Sklansky, J. (1978). On the Hough Transform technique for curve detection. *IEEE T-COMP* 27:923–926.

8. Shapiro, S. D. and A. Iannino (1979). Geometric constructions for predicting Hough Transform performance. *IEEE T-PAMI* 1:310–317.

9. Maitre, H. (1986). Contribution to the prediction of performances of the Hough Transform. *IEEE T-PAMI* 8:669–674.

10. Hanahara, K., T. Maruyama, and T. Uchiyama (1988). A real-time processor for the Hough Transform. *IEEE PAMI* 10(1):121–125.

11. Baringer, W. B. (1987). A VLSI implementation of PPPE for real-time image processing in Radon space. *IEEE Workshop on CAPAMI*, Seattle, WA.

12. Li, H. (1985). Fast Hough Transform for Multidimensional Signal Processing. Yorktown Heights, Westchester, New York, IBM Research Report RC 11562.

13. Little, J. J., G. Blelloch, and T. Cass (1987). Parallel algorithms for computer vision on the connection machine. *First IEEE International Conference on Computer Vision*, London, p. 587–591.

14. Guerra, C. and S. Hambrusch (1987). Parallel algorithms for line detection on a mesh. *IEEE Workshop on CAPAMI*, Seattle, WA, p. 99–106.

15. Ballard, D. H. (1984). Parameter Nets. *Artificial Intelligence* 22:235–267.

16. Blanford, R. P. (1987). Dynamically quantized pyramids for Hough Vote Collection. *IEEE Workshop on CAPAMI*, Seattle, WA, p. 145–152.

17. Fischler, M. A. and O. Firschein (1987). Parallel guessing: a strategy for high speed computation, *Pattern Recognition* 20:257–263.

18. Illingworth, J. and J. Kittler (1988). A survey of the Hough Transform. *Computer Vision, Graphics, and Image Processing* 44:87–116.

19. Hunt, D., L. Nolte, and W. Ruedger (1988). Performance of the Hough Transform and its relationship to statistical signal detection theory. *Computer Vision, Graphics, and Image Processing* 43:221–238.

20. Cappellini and Marconi (editors) (1986). Segmentation of color images using edge-preserved filters. In: North-Holland, *Advances in Image Processing*.

21. Gambotto, J. P. (1986). A hierarchical segmentation algorithm. *IEEE Eighth Conference on Pattern Recognition*, Paris, France, Oct. 27–31, p. 951–953.

22. O'Gorman, L. and G. I. Weill (1986). An approach toward segmenting contour line regions. *IEEE Eighth Conference on Pattern Recognition*, Paris, France, Oct. 27–31, p. 254–257.

23. Gagalowicz, A. and O. Monga (1986). A new approach for image segmentation. *IEEE Eighth Conference on Pattern Recognition*, Paris, France, Oct. 27–31, p. 265–267.

24. Hall, E. L. (1979). *Computer Image Processing and Recognition*. Academic Press.

25. Ekstrom, M. P. (1984). *Digital Image Processing Techniques*. Academic Press.

26. Mitiche, A. and J.K. Aggarnal (1985). Image segmentation by conventional and information-integrating techniques: a synopsis. In: Butterworth and Company, *Image and Vision Computing*.

27. Kittler, K. and J. Illingworth (1985). Relaxation labeling algorithms–a review. In: Butterworth and Company, *Image and Vision Computing*.

28. Tominaga, S. (1986). Color image segmentation using three perceptual attributes. *IEEE Conference on Pattern Recognition*, p. 628–630.

29. Celenk, M. and S. Smith (1986). Color segmentation by clustering and parametric-histogramming technique. *IEEE Eighth Conference on Pattern Recognition*, Paris, France, Oct 27–31, p. 883–886.

30. Bouthemy, P. and J. Santillana Rivero (1987). A hierarchical likelihood approach for region segmentation according to motion-based criteria. *IEEE First International Conference on Computer Vision*, p. 463–467.

31. Huang, J. S. and Dong H. Tseng (1988). Statistical theory of edge detection. *Computer Vision, Graphics, and Image Processing*, 43:337–346.

32. Goutsias, J. and J. Mendel (1987). Semi-Markov random field models for image segmentation. *IEEE International Conference on Acoustics, Speech and Signal Processing*.

33. Cohen, F. S. and D. Cooper (1987). Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian random fields. *IEEE PAMI, PAMI*-9(2):195–219, March.

34. Murray, D. and B. Buxton (1987). Scene segmentation from visual motion using global optimization. *IEEE T-PAMI, PAMI*-9(2), March.

35. McLean, G. F. and M.E. Jernigan (1988). *Hierarchical Edge Detection*. Academic Press.

36. Tsai, W.-H. and S.-S. Yu (1985). Attributed string matching with merging for shape recognition. *PAMI, PAMI*-7(4):453–462, July.

37. Taylor, R. W., M. Savini, and A. P. Reeves (1989). *Fast Segmentation of Range Imagery into Planar Regions*. Academic Press, p. 42–60.

38. Laprade, R. (1988). Split-and-merge segmentation of aerial photographs. *Computer Vision, Graphics, and Image Processing* 44:77–86.

39. Taylor, R. W. M. Savini, and A. P. Reeves (1989). Fast segmentation of range imagery into planar regions. *Computer Vision, Graphics, and Image Processing* 45:42–60.

40. Hoffman, R. and A. K. Jain (1987). Segmentation and classification of range images. *IEEE PAMI, PAMI*-9(5):608–620, September.

41. Dubes, R. and A. K. Jain (1976). *Pattern Recognition* 8:247–260.

42. Hambrick, L., M. Loew, and R. Carroll (1987). The entry-exit method of shadow boundary segmentation. *IEEE PAMI, PAMI*-9(5):597–607, September.

43. Perez, A. (1987). An iterative thresholding algorithm for image segmentation. *IEEE PAMI, PAMI*-9(6):742–750, November.

44. Shridhar, M. and A. Badreldin (1987). Context-directed segmentation algorithm for handwritten numeral strings. *Image and Vision Computing* p. 3–9.

45. Tsuji, S., M. Osada, and M. Yachida (1980). Tracking and segmentation of moving objects in dynamic line images. *IEEE PAMI, PAMI*-2(6):516–521, November.

46. Raafat, H. M. and A. K. C. Wong (1988). A texture information-directed region growing algorithm for image segmentation and region classification. *Computer Vision, Graphics, and Image Processing* 43:1–21.

47. Jayaramamurthy, S. N. and R. Jain (1983). An approach to the segmentation of textured dynamic

scene. *Computer Vision, Graphics, and Image Processing* 21:239–261.

48. Samy, R. A., P. H. Lheritier, and C. A. Bozzo (1986). Image sequence segmentation using time-varying textures. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 403–406.

49. Hyde, J., J. A. Fullwood, and D. R. Corrall (1985). An approach to knowledge-driven segmentation. *Image and Vision Computing* 198–205.

50. Keller, J. M. and S. Chen (1989). Texture description and segmentation through fractal geometry. *Computer Vision, Graphics, and Image Processing* 45:150–166.

Distribution List

Applied Physics Laboratory
Johns Hopkins University
Johns Hopkins Road
Laurel MD 20707

Applied Physics Laboratory
University of Washington
1013 NE 40th St.
Seattle WA 98105

Applied Research Laboratory
Pennsylvania State University
P.O. Box 30
State College PA 16801-0030

Applied Research Laboratories
University of Texas at Austin
P.O. Box 8029
Austin TX 78713-8029

Assistant Secretary of the Navy
Research, Development & Acquisition
Navy Department
Washington DC 20350-1000

Chief of Naval Operations
Navy Department
Washington DC 20350-2000
Attn: OP-71
OP-951
OP-987

Chief of Naval Operations
Oceanographer of the Navy
U.S. Naval Observatory
34th & Massachusetts Ave. NW
Washington DC 20392-1800
Attn: OP-096
OP-0961B

David W. Taylor Naval Research Center
Bethesda MD 20084-5000
Attn: Commander

Defense Mapping Agency
Systems Center
12100 Sunset Hill Road #200
Reston, VA 22090-3207
Attn: Code SGWN

Defense Mapping Agency
Systems Center
8613 Lee Hwy.
Mail Stop A-13
Fairfax VA 22031-2138
Attn: Code PRN

Fleet Antisub Warfare Tng Ctr-Atl
Naval Station
Norfolk VA 23511-6495
Attn: Commanding Officer

Fleet Numerical Oceanography Center
Monterey CA 93943-5005
Attn: Commanding Officer

Marine Acoustics, Inc.
2231 Crystal Drive, Suite 410
Arlington, VA 22202

National Ocean Data Center
WSC1 Room103
6001 Executive Blvd.
Rockville, MD 20852
Attn: Director
G. W. Withee

National Ocean Data Center
1825 Connecticut Ave., NW
Universal Bldg. South, Rm. 206
Washington DC 20235
Naval Air Development Center
Warminster PA 18974-5000
Attn: Commander

Naval Air Systems Command HQ
Washington DC 20361-0001
Attn: Commander

Naval Civil Engineering Laboratory
Port Hueneme CA 93043
Attn: Commanding Officer

Naval Coastal Systems Center
Panama City FL 32407-5000
Attn: Commanding Officer

Naval Facilities Engineering
Command HQ
200 Stovall St.
Alexandria VA 22332-2300
Attn: Commander

Naval Oceanographic Office
Stennis Space Center MS 39522-5001
Attn: Commanding Officer
Library
Technical Director, L. Bernard

Naval Oceanography Command
Stennis Space Center MS 39529-5000
Attn: Commander

Naval Oceanographic & Atmospheric
Research Laboratory
Atmospheric Directorate
Monterey CA 93943-5006
Attn: Code 400

Naval Oceanographic & Atmospheric
Research Laboratory
Stennis Space Center MS 39529-5004
Attn: Code 100
Code 105
Code 115
Code 125L (10)
Code 125P
Code 200
Code 300

Naval Ocean Systems Center
San Diego CA 92152-5000
Attn: Commander

Naval Postgraduate School
Monterey CA 93943
Attn: Superintendent

Naval Research Laboratory
Washington DC 20375
Attn: Commanding Officer
Library, M. Bradley (3)
Naval Sea Systems Command HQ
Washington DC 20362-5101
Attn: Commander

Naval Surface Warfare Center Det
Silver Spring
White Oak Laboratory
10901 New Hampshire Ave.
Silver Spring MD 20903-5000
Attn: Officer in Charge
Library

Naval Surface Warfare Center
Dahlgren VA 22448-5000
Attn: Commander

Naval Underwater Systems Center
Newport RI 02841-5047
Attn: Commander

Naval Underwater Systems Center Det
New London Laboratory
New London CT 06320
Attn: Officer in Charge

Office of Naval Research
800 N. Quincy St.
Arlington VA 22217-5000
Attn: Code 10D/10P, Dr. E. Silva
Code 112, Dr. E. Hartwig
Code 12
Code 10

Office of Naval Research
ONR European Office
PSC 802 Box 39
FPO AE 09499-0700
Attn: Commanding Officer

Office of Naval Technology
800 N. Quincy St.
Arlington VA 22217-5000
Attn: Code 20, Dr. P. Selwyn
Code 228, Dr. M. Briscoe
Code 228, CDR. L. Bounds
Code 22, Dr. T. Warfield

Scripps Institution of Oceanography
University of California
291 Rosecrans St.
San Diego CA 92106-3505

Space & Naval Warfare Sys Com
Director of Navy Laboratories
SPAWAR 005
Washington DC 20363-5100
Attn: Commander

Woods Hole Oceanographic Institution
P.O. Box 32
Woods Hole MA 02543
Attn: Director

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. Agency Use Only *(Leave blank).*

2. Report Date. January 1992

3. Report Type and Dates Covered. Final

4. Title and Subtitle.
A Survey of Digital Image Segmentation Algorithms

5. Funding Numbers.

Job Order No.	13512C
Program Element No.	0604262N
Project No.	00101
Task No.	001072
Accession No.	DN258022

6. Author(s).
K. B. Shaw and M. C. Lohrenz

7. Performing Organization Name(s) and Address(es).
Naval Oceanographic and Atmospheric Research Laboratory
Ocean Science Directorate
Stennis Space Center, Mississippi 39529-5004

8. Performing Organization Report Number.
NOARL Report 32

9. Sponsoring/Monitoring Agency 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.

11. Supplementary Notes.

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited. Naval Oceanographic and Atmospheric Research Laboratory, Stennis Space Center, Mississippi 39529-5004

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

Computer vision is a rapidly expanding field that depends on the automated capability to automatically segment and, thus, to classify and interpret images. In this report, the primary computer vision subarea—sementation—is investigated. Many of the latest publications on the subject of segmentation are detailed in a survey format. Special attention is given to a few specialized techniques for segmenting digital images.

Powerful segmentation techniques are available; however, each technique is ad hoc. The creation of hybrid techniques seems to be a promising future research area with respect to current Navy digital mapping applications.

14. Subject Terms.
digital image processing, segmentation, computer vision

15. Number of Pages. 24

16. Price Code.

17. Security Classification of Report.	18. Security Classification of This Page.	19. Security Classification of Abstract.	20. Limitation of Abstract.
Unclassified	Unclassified	Unclassified	None