NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ALGORITHM FOR SEGMENTATION OF MULTICHANNEL IMAGES

by

James F. Janecek

December 1985

Thesis Advisor: Charles W. Therrien

Approved for public release; distribution is unlimited.

T226349

JRITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION
CLASSIFIED

1b. RESTRICTIVE MARKINGS

SECURITY CLASSIFICATION AUTHORITY

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited.

DECLASSIFICATION/DOWNGRADING SCHEDULE

ERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

NAME OF PERFORMING ORGANIZATION
val Postgraduate School

6b OFFICE SYMBOL (If applicable)
62

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

ADDRESS (City, State, and ZIP Code)
nterey, California 93943-5100

7b. ADDRESS (City, State, and ZIP Code)
Monterey, California 93943-5100

NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

TITLE (Include Security Classification)
GORITHM FOR SEGMENTATION OF MULTICHANNEL IMAGES

PERSONAL AUTHOR(S)
necek, James Frank

TYPE OF REPORT
ster's Thesis

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT (Year, Month, Day)
1985 December

15. PAGE COUNT
43

SUPPLEMENTARY NOTATION

COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Multichannel Images, Segmentation, Maximum Likelihood, Maximum A Posteriori

ABSTRACT (Continue on reverse if necessary and identify by block number)

Segmentation is an important step in the computer based analysis of
ages. This thesis addresses the segmentation of images of multiple
annels of data. Such images are referred to as multichannel images.
amples of these are color images, where the channels represent color
mponents, and remote sensing data, where the channels may represent
gnals from various visible and infrared frequency bands. This
esis describes and demonstrates how segmentation of multichannel

DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

NAME OF RESPONSIBLE INDIVIDUAL
of C.W. Therrien

22b. TELEPHONE (Include Area Code)
408-646-2160

22c. OFFICE SYMBOL
62Ti

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

images into homogeneous regions can be accomplished using linear predictive filtering. Results are given for some synthetically generated color images of two textures.

Approved for public release, distribution is unlimited

Algorithm for Segmentation of Multichannel Images

by

James Frank Janecek
Captain, United States Marine Corps
B.S., Marquette University, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1985

Thesis
F2[illegible]
c.1

ABSTRACT

Segmentation is an important step in the computer based analysis of images. This thesis addresses the segmentation of images of multiple channels of data. Such images are referred to as multichannel images. Examples of these are color images, where the channels represent color components, and remote sensing data, where the channels may represent signals from various visible and infrared frequency bands. This thesis describes and demonstrates how segmentation of multichannel images into homogeneous regions can be accomplished using linear predictive filtering. Results are given for some synthetically generated color images of two textures.

DISCLAIMER

Some terms used in this thesis are registered trademarks of commercial products. Rather than attempt to cite each occurrence of a trademark, all trademarks appearing in this thesis are listed below following the name of the firm holding the trademark:

1. COMTAL, Altadena, California

   Vision One/20

2. Digital Equipment Corporation, Maynard, Massachusetts

   VAX/VMS

TABLE OF CONTENTS

I. INTRODUCTION

The purpose of this thesis is to develop a model and associated algorithm for the texture-based segmentation of an image consisting of several related channels of data. Such an image is referred to as a multichannel image. The work is an extension of a model for image segmentation by Therrien [1]. That model used both maximum likelihood and maximum *a posteriori* estimation to segment images into regions of similar textures. When applied to aerial photographs, the textures may represent different types of natural terrain. This extension to multichannel images can be used in the computer-based analysis of color images or LANDSAT imagery[1] received from a multispectral scanner.

Before considering the model, a more precise definition of the terms to be used in this paper is in order. Segmentation is the separation of an image, into homogeneous regions of different textures. The basic image space considered is a channel. The output of a channel is a two-dimensional (2-D) signal representing pixel intensity as a function of spatial position. When two or more images are considered simultaneously, the process is called multichannel. For example, a color image may be represented by a multichannel image consisting of red, green, and blue component images. Hereafter, in this paper, the term *image* is used synonymously with *multichannel image.*

The research of this thesis shows that, regardless of the number of channels being considered, image segmentation can be achieved by the process of 2-D multichannel linear prediction. Linear prediction is a filtering of the image to forecast the intensity at a particular spatial position from the intensity at neighboring data points using a previously determined set of filter parameters. A set of filter parameters corresponds to a specific texture type and consists of the

[1] Land Satellite (LANDSAT) imagery consists of a frame of four registered digital images, two from the visible region and two from the infrared region.

mean of the data in each channel, the prediction filter weighting coefficients, and the covariance matrix of the prediction error.

The effort to extend the model contained in [1] to multichannel image segmentation consisted of three phases. The remaining chapters deal with these phases of the work. In each of Chapters II and III, the theory and relevant algorithms are presented first. This is followed by a description of the implementing computer programs and their use.

Chapter II discusses the generation of test images used in this research. The test images were generated using known sets of filter parameters. The results of segmentation of the test images are then discussed. Chapter III presents the model and algorithms required to perform multichannel image segmentation utilizing techniques of linear prediction [2, 3]. First, the algorithm for determining the filter parameters from a real multichannel image is developed. Next, the algorithm for multichannel image segmentation is developed and the results of segmentation of the images are discussed. Chapter IV presents conclusions about the applicability, advantages, and limitations of the multichannel image segmentation work presented in this thesis.

The programs developed for this thesis were written in FORTRAN, compiled using Version 4.2 under the VAX/VMS Version 4.1 operating system. The algorithms were developed using spatial domain equations. However, implementation using frequency domain methods seems both possible and desirable.

II. MULTICHANNEL IMAGE MODEL

In this chapter, the multichannel image model used to synthesize test images with known statistical characteristics is presented. This is followed by a discussion of the synthesis algorithm and its implementation. The synthetic images used in this thesis are color images with three channels representing the red, green, and blue components. The test image to be segmented is created from two synthetic single-texture images; this simulates a real image with two regions of distinct texture. Each single-texture synthetic image has known filter parameters which are used to generate the image and determine its statistical characteristics. The results of segmenting the test image using known filter parameters are then presented. However, derivation of the segmentation algorithm is deferred to Chapter III.

A. MODEL DEVELOPMENT

In this section, procedures are developed to generate the test image. Let $\mathbf{F}_{2T}$ be the desired two-texture synthetic image which must accurately reflect real images to be processed and displayed. The display system used in conjunction with this work is the COMTAL Vision One/20. It requires one byte (8 bits) to represent the intensity of a pixel. The dark to light range of pixel values is thus 0 to 255. In order to control the statistical properties of the test image and to ensure that the pixel values are within the display range, the images are generated initially as zero-mean 2-D signals with a wide range of values and then shifted and scaled to integer values within the intensity range of the display. Therefore, the procedure used to develop the test model consists of two steps. First, a procedure to generate a pair of single-texture synthetic images, called $\mathbf{F}_D^{(0)}$ and $\mathbf{F}_D^{(1)}$, is presented. This is followed by the process required to form the composite test image.

1. Vector-valued Signal Model

The model used to generate each synthetic image is that of a 2-D multichannel autoregressive process [4]. The equation for this process is

$$\mathbf{F}^{(t)}(n,m) = -\sum_{\substack{i=0 \\ (i,j)\neq(0,0)}}^{P-1} \sum_{j=0}^{Q-1} \mathbf{A}_{ij}^{(t)} \cdot \mathbf{F}^{(t)}(n-i,m-j) + \mathbf{W}^{(t)}(n,m) \tag{2.1}$$

where $\mathbf{F}^{(t)}(n,m)$ is a 2-D signal at spatial coordinates (n,m) representing a zero-mean image of texture "t", $\mathbf{W}^{(t)}$ is a white noise driving term, and the $\mathbf{A}_{ij}^{(t)}$ are a set of filter weighting coefficients. Both $\mathbf{F}^{(t)}$ and $\mathbf{W}^{(t)}$ are vectors of size K, the number of channels in the image, and the $\mathbf{A}_{ij}^{(t)}$ are matrices of size $K \times K$. For a filter whose region of support is $P \times Q$ pixels, there are PQ $\mathbf{A}_{ij}^{(t)}$ matrices with $\mathbf{A}_{00}^{(t)} = \mathbf{I}$, an identity matrix. Although it is not explicitly represented in Equation (2.1), the $\mathbf{A}_{00}^{(t)}$ matrix is used in the linear predictive filtering which is central to the segmentation algorithm. Figure 2-1 illustrates the vector-valued image model for $K = 3$.

a. White Noise Driving Term

Equation (2.1) shows that $\mathbf{F}^{(t)}(n,m)$ is recursively computed from $\mathbf{W}^{(t)}(n,m)$. Thus, in order to generate an image, one must first have a multichannel array of white noise. This "white noise" is uncorrelated within each channel but correlated between channels. That is, the zero lag covariance of the multichannel noise,

$$\Sigma_W^{(t)} = E[\, \mathbf{W}^{(t)}(n,m) \cdot (\mathbf{W}^{(t)}(n,m))^T \,] \tag{2.2}$$

is a matrix which in general is not diagonal. Multichannel data with this property is hereunder referred to as "multichannel white noise". Data with any desired between channel covariance, Σ_W, can be generated as follows. Since Σ_W is positive-definite and symmetric, it will have positive real eigenvalues and

$$\mathbf{F}^{(t)}(n,m) = \begin{bmatrix} F_1^{(t)}(n,m) \\ F_2^{(t)}(n,m) \\ F_3^{(t)}(n,m) \end{bmatrix}$$

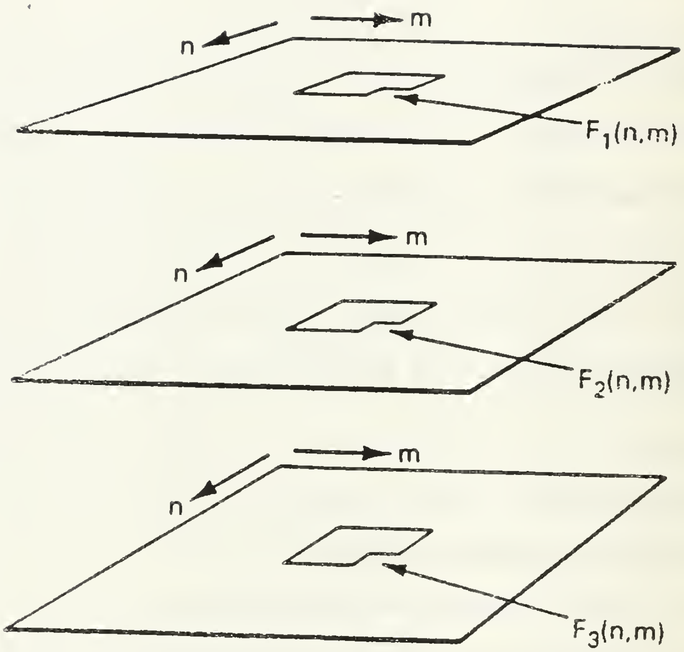


Figure 2-1.

distinct orthonormal eigenvectors. Let $\mathbf{\Phi}$ be the matrix whose columns are the eigenvectors of $\mathbf{\Sigma}_W$ and $\mathbf{\Lambda}^{1/2}$ be the diagonal matrix whose principal elements are the positive square roots of the eigenvalues. Then, if $\mathbf{U}(n,m)$ is a set of data with zero mean and unit variance that is uncorrelated both within and between channels, then the transformation,

$$\mathbf{W}^{(t)}(n,m) = \mathbf{\Phi}\cdot\mathbf{\Lambda}^{1/2}\cdot\mathbf{U}(n,m) \tag{2.3}$$

yields data with the desired covariance. This is true because the covariance of **U** is

$$\Sigma_U = \mathbf{I} \tag{2.4}$$

Thus the covariance of the new data [5] is

$$\mathbf{\Phi}\cdot\mathbf{\Lambda}^{1/2}\cdot\Sigma_U\cdot\mathbf{\Lambda}^{1/2}\cdot\mathbf{\Phi}^T = \mathbf{\Phi}\cdot\mathbf{\Lambda}\cdot\mathbf{\Phi}^T = \Sigma_W \tag{2.5}$$

b. Image Generation Filter Requirements

The synthetic, zero-mean, single textured, 2-D signal representing the image, $\mathbf{F}^{(t)}$, described by Equation (2.1) can be formed once a suitable set of $\mathbf{A}_{ij}^{(t)}$ are provided. The filter used to generate the 2-D signal has a system function

$$\mathbf{H}(z_1,z_2) = \frac{1}{\mathbf{A}(z_1,z_2)} \tag{2.6}$$

where

$$\mathbf{A}(z_1,z_2) = \sum_{i=0}^{P-1}\sum_{j=0}^{Q-1}\mathbf{A}_{ij}\cdot z_1^{-i}\cdot z_2^{-j}$$

is a polynomial derived from Equation (2.1). Since this is an all-pole filter, care must be taken in the selection of weighting coefficients, $\mathbf{A}_{ij}^{(t)}$, to ensure that the resultant filter will be stable [6].

c. Synthetic Single-texture Image Pair

By applying Equation (2.1) recursively with two known sets of $\Sigma_W^{(t)}$ and $\mathbf{A}_{ij}^{(t)}$ parameters, one can generate two zero-mean, 2-D signals, $\mathbf{F}^{(0)}$ and $\mathbf{F}^{(1)}$, that will represent two single texture synthetic images. However, due to the requirements stated earlier, these signals are not suitable for display as images. To allow the overall intensity of the image to be controlled for display, the signal must first be assigned a mean value which is large enough to guarantee that the signal is positive in each channel. That is,

$$\mathbf{F}_M^{(t)}(n,m) = \mathbf{F}^{(t)}(n,m)+\mathbf{M}^{(t)} \qquad \text{for } t = 0,1 \tag{2.7}$$

where $\mathbf{F}_M^{(t)}(n,m)$ is the mean adjusted image and $\mathbf{M}^{(t)}$ is a mean vector of the average gray level or intensity of the image in each of the channels. Since it must be chosen large enough to guarantee positive values throughout $\mathbf{F}_M^{(t)}(n,m)$, the negative of the minimum value of each channel in each 2-D signal is used for the mean of that channel in that image. Then, in order to guarantee that the image has values that are within and fully cover the display range, the signal must be linearly scaled. Since the same scale factor must be used for both textures in the test image, the scaling process must be based on the minimum and maximum values of both textures. The scale factor that satisfies these conditions is given by

$$SF = \frac{255.0}{max\,[max\,(\mathbf{F}^{(0)}), max\,(\mathbf{F}^{(1)})] - min\,[min\,(\mathbf{F}^{(0)}), min\,(\mathbf{F}^{(1)})]} \tag{2.8}$$

Each single-texture synthetic image is then scaled by

$$\mathbf{F}_D^{(t)}(n,m) = SF \times \mathbf{F}_M^{(t)}(n,m) \qquad \text{for } t = 0,1 \tag{2.9}$$

where $\mathbf{F}_D^{(t)}(n,m)$ is a displayable, single-texture, synthetic image. Once $\mathbf{F}_D^{(0)}$

and $\mathbf{F}_D^{(1)}$ have been created, they form a pair that is dependent on the scale factor used. Two typical synthetic images are depicted in Figure 2-2 (a) and (b).

2. Two-Textured Displayable Test Image

If the boundary of a spatial area can be defined as $G(n,m) = 0$ as shown in Figure 2-2(c), then an image with two regions of texture can be created by the process

$$\mathbf{F}_{2T}(n,m) = \begin{cases} \mathbf{F}_D^{(0)}(n,m) \; ; \; G(n,m) \leqslant 0 \\ \mathbf{F}_D^{(1)}(n,m) \; ; \; G(n,m) > 0 \end{cases} \tag{2.10}$$

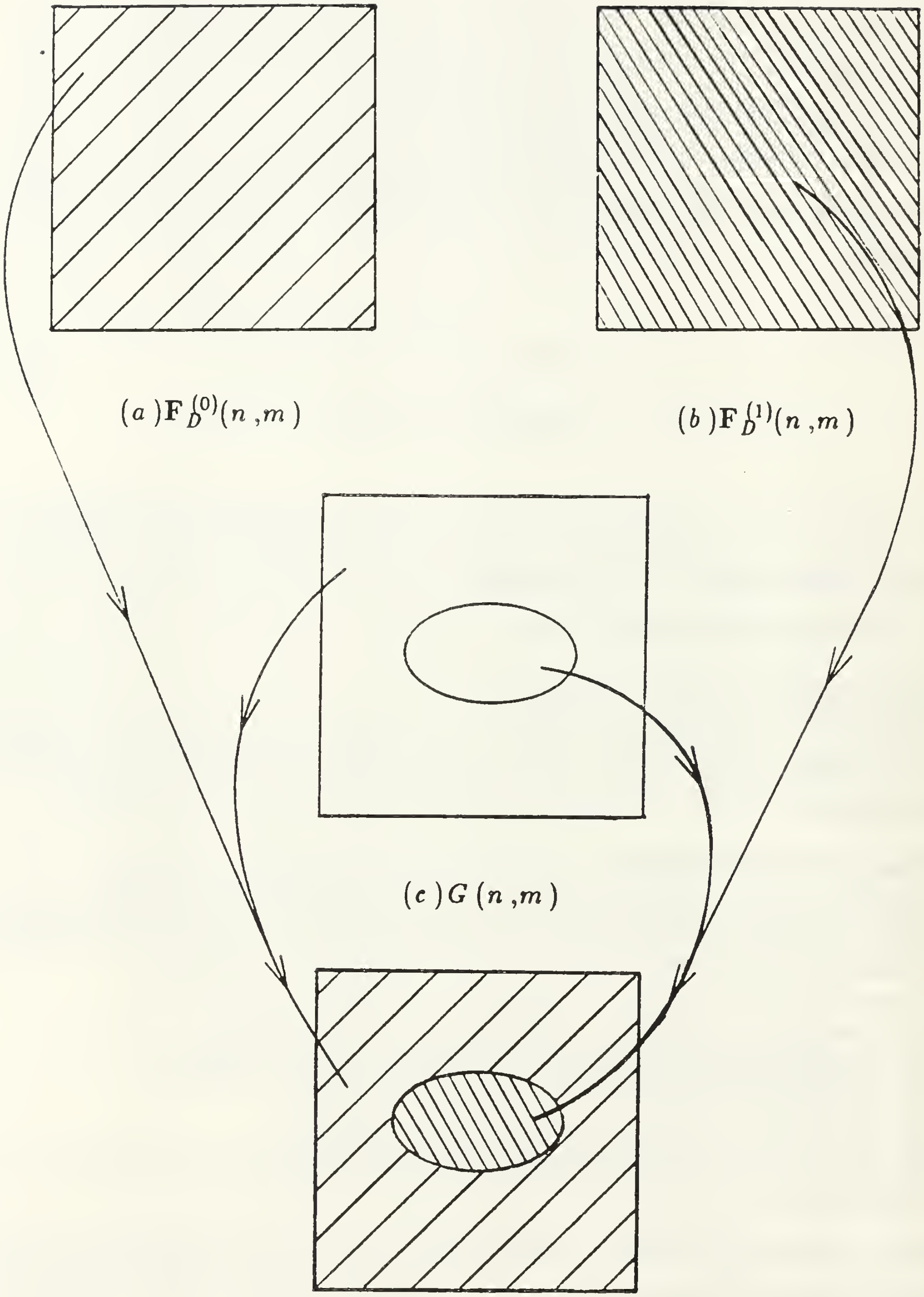
where $\mathbf{F}_{2T}(n,m)$ is the desired two-texture test image. This operation is depicted in Figure 2-2(d). Two possibilities for $G(n,m)$ that were used in this research are an ellipse and a rectangle.

B. IMAGE GENERATION PROGRAM

The program BB2IMAGE generates a dependent pair of synthetic images defined by Equation (2.9). This program first requires the user to input the:

- integer number of rows, N, in each channel (maximum of 128)
- integer number of columns, M, in each channel (maximum of 128)
- integer number of channels, K, in the multichannel image (maximum of 3)
- integer number of rows, P, in each filter (maximum of 10)
- integer number of columns, Q, in each filter (maximum of 10)
- file specification of the first white noise covariance matrix
- file specification of the second white noise covariance matrix
- file specification of the first set of filter coefficients
- file specification of the second set of filter coefficients
- file specification of the first output textured image
- file specification of the second output textured image

In order to ensure dimensional compatibility between the input values and the files that contain the filter parameters, the program performs some compatibility checking. The files should contain the following information in the order stated:



$(a)\mathbf{F}_D^{(0)}(n,m)$

$(b)\mathbf{F}_D^{(1)}(n,m)$

$(c)G(n,m)$

$(d)\mathbf{F}_{2T}(n,m)$

Figure 2-2 Test Image Generation Process

- Covariance matrix files:

  -An integer representing the order of covariance matrix (maximum of 3)
  -The covariance matrix in an order corresponding to the statement:
  READ(*,F15.10) ((Σ_W (i,j),j=1,K),i=1,K)

- Filter coefficient files:

  -The integer number of channels, K, in the image (maximum of 3)
  -The integer number of rows, P, in each filter (maximum of 10)
  -The integer number of columns, Q, in each filter (maximum of 10)
  -The coefficients in an order corresponding to the statement
  READ(*,F15.10) ((((A(i,j,p,q),q=1,Q),p=1,P),j=1,K),i=1,K)

If a failure occurs during the compatibility checks, the program will exit with an appropriate error message.

Upon successful completion of the compatibility checks, the program reads the covariance matrices and filter weighting coefficients from the files. The eigenvalues and eigenvectors of Equations (2.3) and (2.5) are found using standard numerical analysis subroutines. Two uncorrelated white noise images, $\mathbf{U}^{(0)}(n,m)$ and $\mathbf{U}^{(1)}(n,m)$, are formed by the subroutine FORMWL. The transformation of this data to multichannel white noise that is correlated between channels, as expressed in Equation (2.3), is performed by subroutine MXMULT. Figure A-1 in Appendix A is an example of two multichannel white noise images, $\mathbf{W}^{(0)}(n,m)$ and $\mathbf{W}^{(1)}(n,m)$, obtained using the covariance matrices

$$\Sigma_W^{(0)} = \begin{bmatrix} 100.0 & 0.0 & 0.0 \\ 0.0 & 150.0 & -50.0 \\ 0.0 & -50.0 & 150.0 \end{bmatrix} \qquad \Sigma_W^{(1)} = \begin{bmatrix} 150.0 & 100.0 & 50.0 \\ 100.0 & 150.0 & 100.0 \\ 50.0 & 100.0 & 150.0 \end{bmatrix}$$

and scaled as discussed in Section II.A.1.c. Each of these multichannel white noise images along with its chosen set of filter weighting coefficients is then passed to the subroutine FILTER, which generates the zero-mean, single-texture image, $\mathbf{F}^{(t)}(n,m)$, according to Equation (2.1). In addition, FILTER returns the minimum and maximum values of the image. Once both $\mathbf{F}^{(0)}(n,m)$ and

$\mathbf{F}^{(1)}(n,m)$ have been formed, the scale factor, SF, is calculated. The addition of the mean and scaling of each of the images required by Equations (2.8) and (2.9) as well as the conversion of its values to byte format for display is accomplished by subroutine BIMAGE. Figure A-2 is an example of the single-texture synthetic images, $\mathbf{F}_D^{(0)}(n,m)$ and $\mathbf{F}_D^{(1)}(n,m)$, created using the vector-valued image model. The images shown are the result of recursive solution of Equation (2.1) using the multichannel white noise images of Figure A-1 and the sets of filter weighting coefficients listed below, partitioned in the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{01} \\ \mathbf{A}_{10} & \mathbf{A}_{00} \end{bmatrix}$$

$$\mathbf{A}^{(0)} = \left[\begin{array}{ccc|ccc} 0.24 & 0 & 0 & 0.24 & 0 & 0 \\ 0 & 0.24 & 0 & 0 & 0.24 & 0 \\ 0 & 0 & 0.24 & 0 & 0 & 0.24 \\ \hline -0.38 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & -0.38 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & -0.38 & 0 & 0 & 1.0 \end{array}\right]$$

$$\mathbf{A}^{(1)} = \left[\begin{array}{ccc|ccc} -0.24 & 0 & 0 & -0.38 & 0 & 0 \\ 0 & -0.24 & 0 & 0 & -0.38 & 0 \\ 0 & 0 & -0.24 & 0 & 0 & -0.38 \\ \hline -0.38 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & -0.38 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & -0.38 & 0 & 0 & 1.0 \end{array}\right]$$

Each particular synthetic image shown has identical filter weighting coefficients for each channel and no cross terms. This represents a special case when the correlation within channels is separable from the correlation between channels [4]. Nevertheless, an image so generated will be correlated between channels because of the correlations present in the multichannel white noise images, $\mathbf{W}^{(t)}(n,m)$.

The test image, $\mathbf{F}_{2T}(n,m)$, described by Equation (2.10) is created using the program TWOTXTUR. The program allows the user to choose an area bounded by $G(n,m)$ from either an ellipse or rectangle, the location of the center of the area, and its size. The program then evaluates $G(n,m)$ for each pixel location in the test image and determines which texture to use at the current spatial position based on Equation (2.10). The ordered list of required user inputs for the program TWOTXTUR is:

- Choice of rectangle or ellipse for the area bounded by G(n,m)
- The integers for the X-Y center of area
- The integers for the X-WIDTH and Y-HEIGHT of area
- The input file specification of outside texture for channel 1
- The input file specification of inside texture for channel 1
- The output file specification for $\mathbf{F}_{2T}$ for channel 1
- The input file specification of outside texture for channel 2
- The input file specification of inside texture for channel 2
- The output file specification for $\mathbf{F}_{2T}$ for channel 2
- The input file specification of outside texture for channel 3
- The input file specification of inside texture for channel 3
- The output file specification for $\mathbf{F}_{2T}$ for channel 3

The output files contain the three channels of the desired two-textured test image. Figure A-3 is an example of a test image that was generated using this procedure.

The result of segmenting the test image using the known filter parameters is shown in Figure A-4. However, description of the segmentation algorithm is given in the next chapter, as the procedure required is independent of whether the filter parameters were user provided or were the result of the estimation algorithm to be discussed in Chapter III.

III. SEGMENTATION OF TEXTURED IMAGES

In this chapter, the algorithm and implementing programs to perform multichannel image segmentation of textured images are presented. First, the algorithm is developed by presenting the two separate procedures of texture estimation and image segmentation. This is followed by a discussion of two computer programs that implement these procedures. Finally, the results of texture estimation and image segmentation are presented. The segmentation procedures presented in this chapter are based on a model for images described by Equations (2.1) and (2.7). In all cases discussed here, the images have three channels corresponding to the red, green, and blue color components. However, the same algorithm and programs can be used on other types of multichannel images such as LANDSAT, if appropriate filter parameters are first calculated.

A. ALGORITHM DEVELOPMENT

The segmentation of textured images consists of two procedures. First, given an image of each texture, filter parameters that accurately describe that texture are determined. Since the image model presented in the previous chapter is based on statistical properties, the filter parameters must be derived from a statistical analysis of the textured image. Secondly, given filter parameter estimates for two textures, a set of homogeneous regions that "most probably" correspond to the desired textures are found. The latter step is based on 2-D linear predictive filtering.

1. Filter Parameter Estimation Procedures

The problem of filter parameter estimation requires finding $\mathbf{M}^{(t)}$, $\Sigma_W^{(t)}$, and $\mathbf{A}^{(t)}$ by statistical analysis of data in an estimation window containing the desired texture. An estimation window is depicted in Figure 3-1 for one of the textures in the test image. The reader should realize that the estimation window does not have to come from an image to be segmented; it could be found from any image containing the desired texture. The linear prediction filter is the inverse of the filter used in Equation (2.1). Thus it is an FIR filter with quarter-plane region of support and selectable mask size. The prediction filter and the covariance of the multichannel white noise will be found by estimating the correlation function of the signal and solving a set of Normal equations [3,4].

In the follow sub-sections, the image being discussed is a single-texture image. Therefore, the estimation window has the same size as the 2-D extent of the image. Further, the superscript "(t)" is not shown since determination of filter parameters is for a single texture.

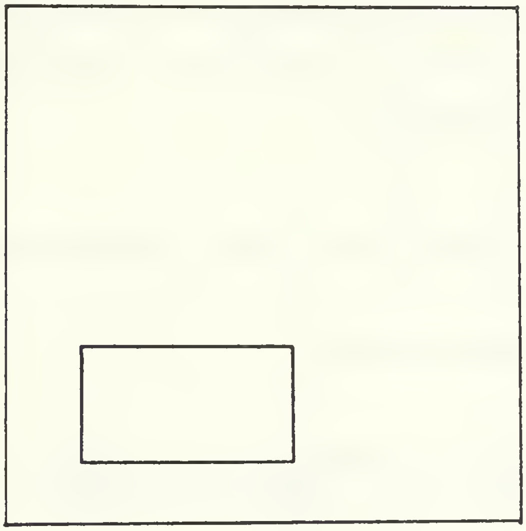


Figure 3-1 Typical Estimation Window for a Single Texture.

a. Mean Vector Estimation

In order to model the image by Equations (2.1) and (2.7), one must first estimate the mean vector of the image. The mean vector estimate is given by

$$\hat{\mathbf{M}} = \begin{bmatrix} \hat{M}_1 \\ \hat{M}_2 \\ \hat{M}_3 \end{bmatrix} \quad where\ \hat{M}_k = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} F_{M_k}(n,m) \qquad (3.1)$$

where $\hat{M}_k$ is the mean estimate for the k^{th} channel of the single texture image,

$\mathbf{F}_M(n,m)$, of size $N \times M$. Knowing $\hat{\mathbf{M}}$ and assuming stationarity of $\mathbf{F}_M$, a zero mean 2-D multichannel signal, $\mathbf{F}$, corresponding to Equation (2.1) can be found by subtracting the mean vector from the image. That is,

$$\mathbf{F}(n,m) = \mathbf{F}_M(n,m) - \hat{\mathbf{M}} \tag{3.2}$$

b. Correlation Function Estimation

In order to estimate the filter weighting coefficients, $\mathbf{A}_{ij}$ and the multichannel white noise covariance, $\hat{\Sigma}_W$, one must first estimate the correlation function of the zero mean, 2-D, multichannel signal. The 2-D correlation function for the "i,j" lag is given by

$$\mathbf{R}(i,j)=\mathbf{R}^T(-i,-j)=E[\mathbf{F}(n,m)\cdot\mathbf{F}^T(n-i,m-j)] \tag{3.3a}$$

and can be estimated from the data by

$$\hat{\mathbf{R}}(i,j) = \frac{1}{NM}\sum_{n=0}^{N-i-1}\sum_{m=0}^{M-j-1}\mathbf{F}(n,m)\cdot\mathbf{F}^T(n-i,m-j) \tag{3.3b}$$

where $\hat{\mathbf{R}}(i,j)$ is a matrix of order K.

c. Filter Coefficients and Prediction Error Covariance

If the multichannel signal is truly represented by the model in Equation (2.1), then the prediction filter coefficients and multichannel white noise covariance must satisfy a set of linear equations known as the Normal equations. In these equations the multichannel white noise covariance is referred to as the prediction error covariance.

Normal equations corresponding to Equation (2.1) can be written as

$$[\,\mathbf{R}\,]\cdot[\,\mathbf{A}\,] = [\,\mathbf{S}\,] \tag{3.4}$$

where $\mathbf{R}$ is is the correlation matrix for the signal, $\mathbf{A}$ is an appropriately ordered matrix of filter coefficients, and $\mathbf{S}$ contains a single non-zero block $\hat{\Sigma}_W$ which is the prediction error covariance. The matrix $\mathbf{R}$ has three levels of partitioning and for any rectangular region of support is block Toeplitz with block Toeplitz blocks.

For a first quadrant filter with a $P \times Q$ region of support, the Normal equations that define Equation (3.4) have the specific form

$$\begin{bmatrix} \mathbf{R}(0) & \mathbf{R}(-1) & \cdots & \mathbf{R}(-P+1) \\ \mathbf{R}(1) & \mathbf{R}(0) & \cdots & \mathbf{R}(-P+2) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \mathbf{R}(P-1) & \mathbf{R}(P-2) & \cdots & \mathbf{R}(0) \end{bmatrix} \begin{bmatrix} \mathbf{A}^{(0)} \\ \mathbf{A}^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{A}^{(P-1)} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^{(0)} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \qquad (3.5)$$

where

$$\mathbf{R}(i) = \mathbf{R}^T(-i) = \begin{bmatrix} \mathbf{R}(i,0) & \mathbf{R}(i,-1) & \cdots & \mathbf{R}(i,-Q+1) \\ \mathbf{R}(i,1) & \mathbf{R}(i,0) & \cdots & \mathbf{R}(i,-Q+2) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \mathbf{R}(i,Q-1) & \mathbf{R}(i,Q-2) & \cdots & \mathbf{R}(i,0) \end{bmatrix} \qquad (3.6)$$

where $\mathbf{R}(i,j)$ is the matrix correlation function described in Equation (3.3) above. The quantities $\mathbf{A}^{(i)}$ and $\mathbf{S}^{(0)}$ both contain two more levels of partitioning and are defined by

$$\mathbf{A}^{(i)} = \begin{bmatrix} \mathbf{A}_{i,0} \\ \mathbf{A}_{i,1} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{A}_{i,Q-1} \end{bmatrix} , \qquad \mathbf{S} = \begin{bmatrix} \Sigma_W \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \tag{3.7,3.8}$$

where $\mathbf{A}_{ij}$ and the partitions of $\mathbf{S}^{(0)}$ are matrices of order K. Further, $\mathbf{A}_{00} = \mathbf{I}$ and Σ_W is the prediction error covariance defined by Equation (2.2). While the covariance in Equation (3.5) has the doubly block Toeplitz structure described above, the innermost blocks are not Toeplitz. The matrix is not block symmetric at any level. Solution of Equation (3.5) gives the coefficients $\mathbf{A}_{ij}$ for $(i,j) \neq (0,0)$ and Σ_W.

2. Segmentation Procedure

Image segmentation is achieved by determining the "most probable" texture class of each pixel in the image, given the filter parameters of two textures. The filter parameters are used to implement a set of filters which are the inverse of the filters in Equation (2.1), presumed to have generated the image. These inverse filters can be thought of as predicting the intensity of a pixel in each channel from data in the region adjacent to this vector of pixels. The outputs of the filters are the prediction errors $\mathbf{E}^{(t)}(n,m)$. The basic idea of the algorithm is then as follows. When an area of texture is processed by the filter matched to that texture, the output of the filter, the prediction error, will be low. When that area is processed by a filter that is not matched to the texture, the output will be high. The segmentation algorithm uses this idea in the following way. First the entire image to be segmented is processed by two linear predictive filters, one matched to each of the texture types desired. Thus each filter produces a prediction error estimate for each pixel in the image. Next, based on the error estimates of the two given textures, a "most-likely" choice, $\mathbf{ML}(n,m)$, of texture class is made for each pixel in the image. Finally, a "maximum a posteriori" selection, $\mathbf{MP}(n,m)$, of texture class is made based on the prediction errors for a pixel and the texture classification of a small number of surrounding pixels. These steps are described in greater detail below.

a. Error Estimation

The first step in making an error estimate on a textured image is to subtract the mean for each texture class. This is performed in accordance with Equation (3.2) resulting in two signals, $\mathbf{F}^{(0)}$ and $\mathbf{F}^{(1)}$. The prediction error estimates for the two textures are computed from

$$\mathbf{E}^{(t)}(n,m) = \sum_{i=0}^{P-1} \sum_{j=0}^{Q-1} \mathbf{A}_{ij}^{(t)} \mathbf{F}^{(t)}(n-i,m-j) \qquad t = 0,1 \tag{3.9}$$

where $\mathbf{A}_{00}$ is an identity matrix. As noted earlier, the filter in Equation (3.9) is the inverse of the filter in Equation (2.1). Therefore, if the textured image were really generated by Equation (2.1) with the same weighting coefficients, then $\mathbf{E}^{(t)}(n,m)$ would be multichannel white noise.

b. Most-likely Decision Rule

The "most-likely" decision rule used in this work is an extension of the rule used in [1] for single channel image segmentation. If the image intensities in all channels are characterized by a multivariate distribution, then it can be shown [1] that a maximum likelihood estimate for the classes of the pixel is given by

$$[\mathbf{E}^{(0)}(n,m)]^T [\Sigma_W^{(0)}]^{-1} \mathbf{E}^{(0)}(n,m) + \ln |\Sigma_W^{(0)}| \tag{3.10}$$

$$\overset{0}{\underset{1}{\lessgtr}} \; [\mathbf{E}^{(1)}(n,m)]^T [\Sigma_W^{(1)}]^{-1} \mathbf{E}^{(1)}(n,m) + \ln |\Sigma_W^{(1)}|$$

where the numbers above and below the inequality indicates the texture class or "state" to which the current point (n,m) belongs. If the class is 0, then ML(n,m) is assigned 0 for a black pixel. Otherwise ML(n,m) is assigned 1 for a white pixel. As was pointed out in [1], it is expected that ML will give a somewhat spotty result because the estimate is made only at a single pixel without regard for its neighbors.

c. Maximum A Posteriori Decision Rule

The *maximum a posteriori* (MAP) decision rule is based on a Markov model for classes of pixels. The Markov model is described in detail in [1]. For the multichannel image case the MAP procedure begins with Equation (3.10) and incorporates a set of Markov transition probabilities $Pr[\, t \mid S_{n,m}]$. Thus, the MAP segmentation rule for multichannel images is

$$[\mathbf{E}^{(0)}(n,m)]^T[\Sigma_W^{(0)}]^{-1}\mathbf{E}^{(0)}(n,m)+\ln\,|\,\Sigma_W^{(0)}|\,-2\ln Pr\,[0\,|\,S_{n,m}] \tag{3.11}$$

$$\overset{0}{\underset{1}{\lessgtr}}\ [\mathbf{E}^{(1)}(n,m)]^T[\Sigma_W^{(1)}]^{-1}\mathbf{E}^{(1)}(n,m) + \ln|\,\Sigma_W^{(1)}|\,-2\ln Pr\,[1\,|\,S_{n,m}]$$

where $S_{n,m}$ is a small square region centered at the current point (n,m) and $Pr[\, t \mid S_{n,m}]$ is the probability that the pixel at (n,m) is of state "t" given the states of the surrounding pixels in $S_{n,m}$. The side of $S_{n,m}$ should be an odd number. The solution of this equation is performed by iteration using the states of ML(n,m) obtained from Equation (3.10) as the initial conditions for MP(n,m). Since most of the terms in Equation (3.11) are the same as those in Equation (3.10), the computational requirements of the MAP decision rule are significantly reduced by storing the differences,

$$\begin{aligned} MLD(n,m) \;=\; & [\mathbf{E}^{(0)}(n,m)]^T[\Sigma_W^{(0)}]^{-1}\mathbf{E}^{(0)}(n,m) + \ln|\,\Sigma_W^{(0)}| \\ & - [\mathbf{E}^{(1)}(n,m)]^T[\Sigma_W^{(1)}]^{-1}\mathbf{E}^{(1)}(n,m) - \ln|\,\Sigma_W^{(1)}| \end{aligned} \tag{3.12}$$

incurred during the determination of the *most-likely* segmentation. Substituting Equation (3.12) in Equation (3.11) yields

$$MLD(n,m) - 2\ln Pr\,[0 \mid S_{n,m}] + 2\ln Pr\,[1 \mid S_{n,m}] \overset{0}{\underset{1}{\lessgtr}} 0 \qquad (3.13)$$

as the decision rule for assignment of states to MP(n,m). During each iteration, the Markov transition probabilities are held fixed until all new states are determined. Then the states of MP(n,m) are updated and another iteration of the process is performed. The procedure ends when just a few class labels of the pixels change during an iteration. For a certain symmetric form of the transition probabilities it can be shown [1] that the terms $\ln Pr\,[\,t \mid S_{n,m}]$ in Equation (3.13) are proportional to the number of pixels in $S_{n,m}$ with state "t" divided by the total number of surrounding pixels in $S_{n,m}$. Thus Equation (3.13) can be expressed as

$$MLD(n,m) + KS\left[1 - \frac{\#\ of\ \ state\ 1\ pixels}{\#\ of\ \ surrounding\ \ states}\right] \overset{0}{\underset{1}{\lessgtr}} 0 \qquad (3.14)$$

where KS is a convergence factor for the iterative solution. Normally, the number of surrounding states will be the area of the square used for the Markov region minus one, but care must be taken to properly adjust this value at the image boundaries when the Markov region exceeds the 2-D extent of the image.

B. COMPUTER IMPLEMENTATION OF ALGORITHMS

The procedures develop for estimating filter parameters and performing segmentation were implemented as separate computer programs. Since it is desirable to be able to view the image when doing filter parameter estimation, this process should be done interactively. The segmentation process, on the other hand, has no need for user intervention and can be run in a "batch" mode. Thus, the user can find two sets of filter parameter estimates, set up command file to drive the segmentation program, put the job for the segmentation program in background, and then, if desired, go back and find more filter parameter estimates simultaneously. A description of these programs and their requirements is presented in the following sections.

1. Program for Filter Parameter Estimation

The program, TFLTRM0, implements the filter parameter estimation procedures described in Section III.A.1. For this program, the user defines two areas within the 128 × 128 multichannel image for which filter parameters are desired and the program generates and files the two sets of filter parameters. The following sections describe the required inputs, the implementation of the procedures required to estimate the filter parameters, and the outputs of the texture estimation algorithm.

a. User Inputs

In order to generate the two sets of filter parameters, TFLTRM0 requires the user to input certain program parameters in the order listed below.

- The integer number of rows, P, in each filter (maximum of 10)
- The integer number of columns, Q, in each filter (maximum of 10)
- The integer number of rows, N, in each channel (maximum of 128)
- The integer number of columns, M, in each channel (maximum of 128)
- The integer number of channels, K, in multichannel image (maximum of 3)
- The scale factor of the image (greater than 0)
- The file specification of the image channel 1
- The file specification of the image channel 2
- The file specification of the image channel 3
- The integer number of the upper-left corner for the first filter
- The integer number of the lower-right corner for the first filter
- The output file specification for the first set of filter parameters
- The integer number of the upper-left corner for the second filter
- The integer number of the lower-right corner for the second filter
- The output file specification for the second set of filter parameters

For the first six inputs, the program performs error checking after each entry to ensure that the value entered is with the stated maximum. In the event of an improper entry, the program requests that the value be entered again.

b. Calculation of Mean Vector

The calculation of the mean vector estimate is performed by the subroutine MEAN. When passed the array containing the estimation window of the image, the array dimensions, and the estimation window dimensions, this subroutine calculates the mean of each channel each channel within the estimation window using Equation (3.1). It then adjusts the values in each channel by subtracting the mean in accordance with Equation (3.2). The returned values of this subroutine are $\hat{\mathbf{M}}$ and $\mathbf{F}(n,m)$.

c. Calculation of the Signal Correlation Matrix

The calculation of the correlation matrix, **R**, requires three tasks. First, a loop is established in the main program to find the required PQ^2 blocks of 2-D correlation functions required by Equation (3.4) and to determine the necessary parameters for the calculation of the current block. The actual calculation of each 2-D correlation function given by Equation (3.3b) was performed in this loop by calling subroutine ACF. ACF uses the covariance method [2] for computing the 2-D correlation function. After finding the correlation matrix **R**, the program calls an available linear equation routine to solve for a "pseudofilter" **B** that can be defined by multiplying Equation (3.4) by the inverse of the multichannel white noise covariance to be found. That is

$$[\,\mathbf{R}\,]\cdot[\,\mathbf{B}\,] = [\,\mathbf{S_I}\,] \qquad (3.15)$$

where $\mathbf{B}_{00} = [\,\hat{\Sigma}_W\,]^{-1}$ and $[\,\mathbf{S_I}\,]$ is all zero except for an identity matrix in its first partition. After extracting $\mathbf{B}_{00}$ from **B**, the linear equation routine is again used to solve for the desired covariance estimate given by

$$\mathbf{B}_{00}\cdot\hat{\Sigma}_W = \mathbf{I} \qquad (3.16)$$

d. Calculation of Filter Weighting Coefficients

The calculation of the filter weighting coefficients is easily performed. Since **B** and $\hat{\Sigma}_W$ are both available, the subroutine MXMULT is called to solve for **A** given by

$$[\,\mathbf{A}\,] = [\,\mathbf{B}\,]\cdot\hat{\Sigma}_W \qquad (3.17)$$

e. Outputs of Filter Parameter Estimation

Each set of filter parameters found is written to the user specified files. In order to provide for dimensional compatibility checking when the filters are used in the segmentation process, each output file contains a header that describes some essential characteristics of the filter. The ordered list of the contents of each output filter file is provided below.

- The integer number of channels
- The integer number of filter rows
- The integer number of filter columns
- The scale factor of the image
- The mean vector of the image in channel number order
- The filter weighting coefficients in an order corresponding to the statement READ(*,F15.10) ((((A(i,j,p,q),q=1,Q),p=1,P),j=1,K),i=1,K)
- The covariance matrix in an order corresponding to the statment READ(*,F15.10) ((Σ(i,j),i=1,K),j=1,K)

2. Image Segmentation Program

The program, MAIN2, implements the image segmentation algorithms of Section III.A.2. The following sections describe the required inputs and the implementation of the procedures required to segment an image into homogeneous regions.

a. User Inputs

The program MAIN2 requires the user to input certain program parameters in the order listed below.

- The integer number of rows, N, in each channel (maximum of 128)
- The integer number of columns, M, in each channel (maximum of 128)
- The integer number of channels, K, in multichannel image (maximum of 3)
- The integer number of rows, P, in each filter (maximum of 10)
- The integer number of columns, Q, in each filter (maximum of 10)
- The integer size of a side of the square Markov region (maximum of 9)
- The integer number for the limit on iterations
- The value to use for the convergence factor, KS
- The file specification of the image channel 1
- The file specification of the image channel 2

- The file specification of the image channel 3
- The file specifications for the two sets of filter parameters
- The output file specification for the segmented image (filetype of ".SGT")

For the first six inputs and the last input, the program performs error checking to ensure that the value entered meets the stated constraint. In the event of an improper entry, the program requests that the value be input again. The filter files to be input must be ordered as shown in Section III.B.1.e for proper assignment of variables. As the file for each filter is input, the header is read and compatibility checks are performed to ensure dimensional compatibility of input values. If a failure should occur during any compatibility check, the program will exit with an appropriate error message.

b. Calculation of Error Estimates for Two Textures

Before performing the linear predictive filtering, the images, $\mathbf{F}^{(0)}$ and $\mathbf{F}^{(1)}$, are formed from the input image. This process is performed by the subroutine CBR4NM. When passed the array containing the image, the array dimensions, the image dimensions, the scale factor of the image, and the two mean vectors, this subroutine returns $\mathbf{F}^{(0)}$ and $\mathbf{F}^{(1)}$ given by Equation (3.2). The linear predictive filtering is performed by subroutine FILTER. When passed $\mathbf{F}^{(t)}$,$\mathbf{A}^{(t)}$, and the proper dimensioning arguments, this subroutine returns the prediction error, $\mathbf{E}^{(t)}$, of Equation (3.9). $\mathbf{E}^{(0)}$ and $\mathbf{E}^{(1)}$ are formed by consecutive calls to FILTER.

c. Calculation of Most-likely Image Segmentation

The calculation of the *most likely* image segmentation, *ML,* is performed by looping over the 2-D extent of the image. At each point *(n,m)* in the image, Equation (3.10) is evaluated and ML(n,m) is assigned the resulting binary value (0 or 1). In addition, the value of MLD(n,m) is retained for use during the *MAP* segmentation.

d. Maximum A Posteriori Image Segmentation

In order to perform the maximum *a posteriori* image segmentation of Section III.A.2.c, a value for the convergence factor, *KS,* must be assigned. If

KS is assigned too small, the segmentation may not remove improperly classified pixels. On the other hand, if *KS* is assigned too large, correctly classified pixels could be changed due to overdriving the MAP estimate. The MAP segmentation is also performed using a loop over the 2-D extent of the image. However, in order to control the iterative process, steps were taken to limit the process in both the number of iterations allowed and the count of changing states during the an iteration. Prior to the calculation of Equation (3.14) at each point (n,m), the program checks to see if the Markov region is within the 2-D extent of the image. If a boundary problem is detected the program provides the required adjustment to the counting of the states and the total number of states being considered. The assignment of the new state is then placed in a temporary array and the process repeats for the next point in the image. At the completion of each iteration loop over the 2-D extent of the image, the contents of the temporary array are transferred into MP(n,m). Then if the controls on total iterations are not exceeded, the temporary array is reset to zero and another iteration is begun.

e. Outputs of Image Segmentation Program

There are two outputs provided by program MAIN2. First, the user specified MAP segmentation file is stored. In addition, the program outputs the *most-likely* segmentation under the same filename used for the MAP segmentation, but changes the filetype to ".ML_". Both files are ready for display on the COMTAL Vision One/20.

C. RESULTS OF IMAGE SEGMENTATION

The programs were executed on the image shown in Figure A-3. Filter parameters were estimated using the estimation windows depicted in Figure 3-2.

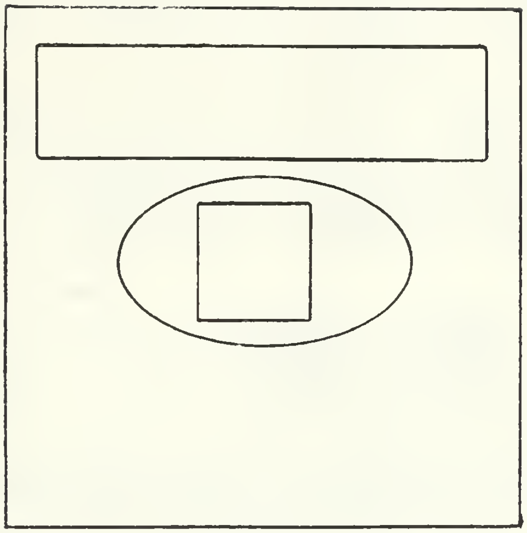


Figure 3-2 Test Estimation Windows

The estimated filter parameters were found to be close approximations to the values used to generate each texture (see Chapter 2). The filter parameter estimates are provided below.

$$\hat{\mathbf{A}}^{(0)} = \left[\begin{array}{ccc|ccc} 1.0000 & 0.0000 & 0.0000 & -0.3662 & 0.0068 & 0.0046 \\ 0.0000 & 1.0000 & 0.0000 & -0.0022 & -0.3370 & -0.0471 \\ 0.0000 & 0.0000 & 1.0000 & -0.0210 & 0.0199 & -0.3710 \\ \hline -0.3666 & -0.0197 & -0.0121 & -0.2379 & 0.0106 & -0.0132 \\ -0.0011 & -0.3727 & -0.0125 & -0.0057 & -0.2662 & 0.0490 \\ -0.0026 & 0.0080 & -0.3753 & 0.0233 & -0.0226 & -0.2345 \end{array}\right]$$

$$\hat{\Sigma}_W^{(0)} = \begin{bmatrix} 94.64 & 2.36 & 2.83 \\ 2.36 & 143.76 & -47.75 \\ 2.83 & -47.75 & 142.39 \end{bmatrix} \qquad \hat{\mathbf{M}}^{(0)} = \begin{bmatrix} 125.63 \\ 123.41 \\ 127.26 \end{bmatrix}$$

$$\hat{\mathbf{A}}^{(1)} = \left[\begin{array}{ccc|ccc} 1.0000 & 0.0000 & 0.0000 & 0.1953 & -0.0134 & -0.0164 \\ 0.0000 & 1.0000 & 0.0000 & 0.0616 & 0.2748 & 0.0505 \\ 0.0000 & 0.0000 & 1.0000 & -0.0083 & -0.0001 & 0.2407 \\ \hline -0.4072 & -0.0181 & -0.0208 & 0.2642 & 0.0030 & 0.0093 \\ -0.0021 & -0.3700 & 0.0444 & -0.0144 & 0.2413 & 0.0027 \\ 0.0238 & 0.0038 & -0.3850 & 0.0147 & -0.0233 & 0.2066 \end{array}\right]$$

$$\hat{\Sigma}_W^{(1)} = \begin{bmatrix} 142.670 & 93.071 & 43.175 \\ 93.071 & 143.343 & 96.255 \\ 43.175 & 96.255 & 147.520 \end{bmatrix} \qquad \hat{\mathbf{M}}^{(1)} = \begin{bmatrix} 67.281 \\ 67.216 \\ 67.686 \end{bmatrix}$$

The image of Figure A-3 was then processed using these estimated filters. The results of both the *most likely* and the *maximum a posteriori* segmentations are presented in Figure A-5.

IV. CONCLUSIONS

The work in this thesis demonstrated that segmentation of multichannel images using linear predictive filtering is feasible. Although the testing was performed on synthetic images, similar results are anticipated for real images of texture such as natural terrain.

The assignment of a value to the convergence factor, *KS*, appears to be one of professional judgement. Testing was conducted on several synthetic images formed from six different types of texture. Testing of these images showed that very large values of *KS* were sometimes required to get complete segmentation in fewer than ten iterations. Experimentation also showed that more moderate values of *KS* are adequate for the initial iterations but higher values are preferred in the final stages. Further work is recommended to determine a method of dynamically varying *KS* .

As part of this thesis, an additional project was undertaken to develop an interactive program for image segmentation. This single program will allow the user to perform all processing required for the image segmentation. In addition to estimating filter parameters and performing image segmentation, this program provides several utilities. These utilities are listed below.

- transferring of a 512×512 image to the COMTAL
- selection of up to 128×128 image subsections
- storage of the image subsections
- labeled overlays for recording of image subsections that were used
- writing of a command file for the segmentation program
- execution of the image segmentation in a background job
- viewing of segmentation results

The first four utilities have been completed and further work is being done to complete the remaining items listed.

Additional research is recommended to improve program execution times. Presently, the execution time for performing each error prediction described by

Equation (3.9) is about three minutes on a VAX 11/780. This time is based on processing a three channel, 128×128 image using a 2×2 filter.

LIST OF REFERENCES

1. Therrien, Charles W., "An Estimation-Theoretic Approach to Terrain Image Segmentation," **Computer Vision, Graphics, and Image Processing 22,** pp. 313 - 326, 1983.

2. Makhoul, John, "Linear Prediction: A Tutorial Review," **Proc. IEEE 63,** pp. 561 - 580, 1975.

3. Therrien, C. W., Quatieri, T. F., Dudgeon, D. E., "Statistical Model-Based Algorithms for Image Processing," **Proceedings of the IEEE,** (accepted for publication).

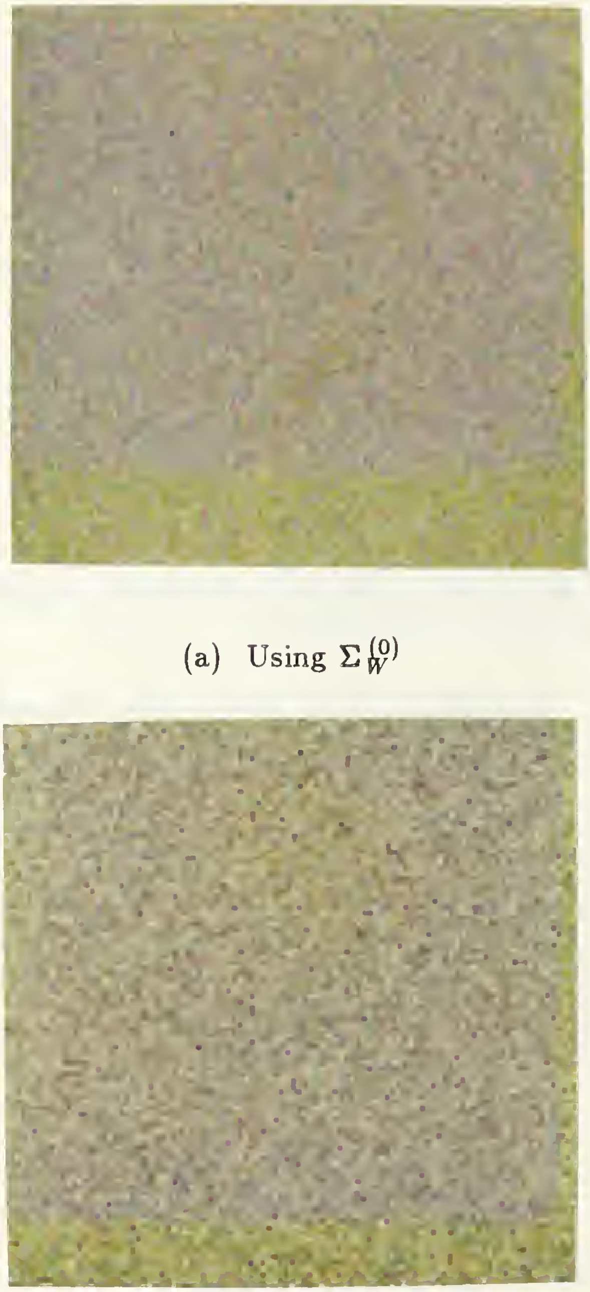
4. Therrien, Charles W., "Linear Prediction for 2-D Vector Random Fields," **Proceedings of the 19th Asilomar Conference on Circuits, Systems, and Computers,** 6-8 November 1985.

5. Helstrom, C.W., **Probability and Stochastic Processes for Engineers,** Macmillan Publishing Company, New York, 1984.

6. Dudgeon, Dan E. and Mersereau, Russell M., **Multidimensional Digital Signal Processing,** Prentice-Hall Inc., Englewood Cliffs, N.J., 1984.

APPENDIX A

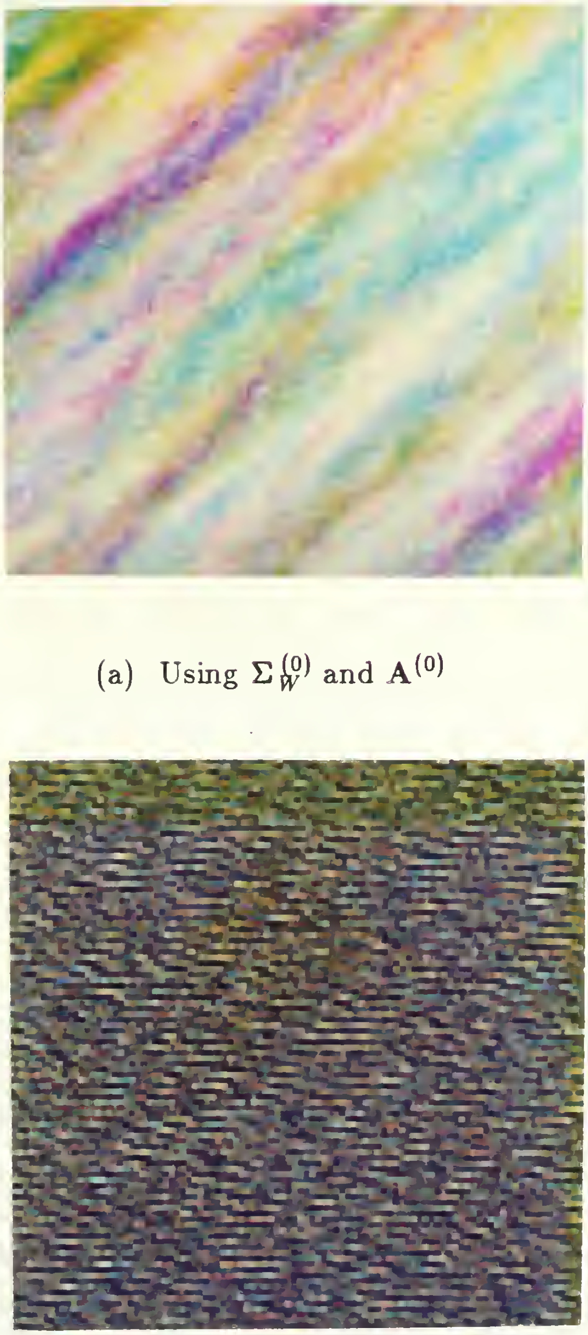
Photographed Results



(a) Using $\boldsymbol{\Sigma}_W^{(0)}$

(b) Using $\boldsymbol{\Sigma}_W^{(1)}$

Figure A-1 Multichannel White Noise Images.



(a) Using $\Sigma_W^{(0)}$ and $\mathbf{A}^{(0)}$

(b) Using $\Sigma_W^{(1)}$ and $\mathbf{A}^{(1)}$

Figure A-2 Single Texture Synthetic Images.

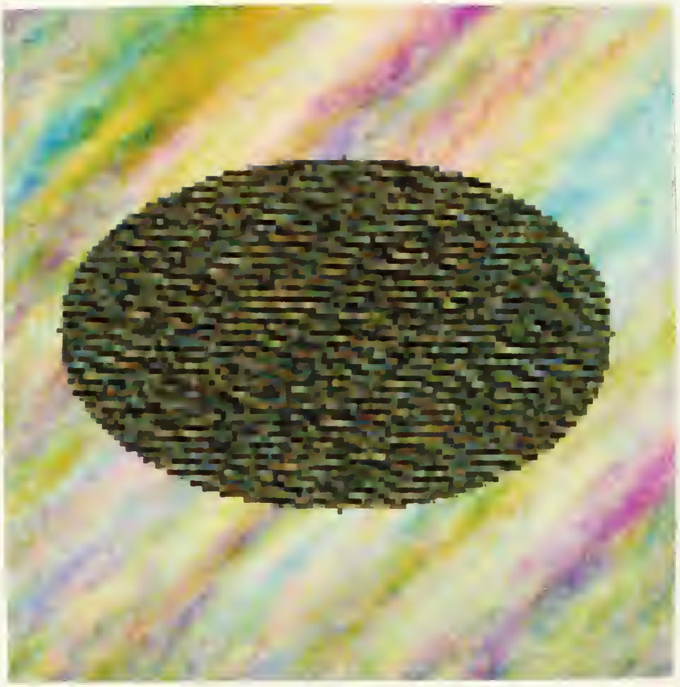
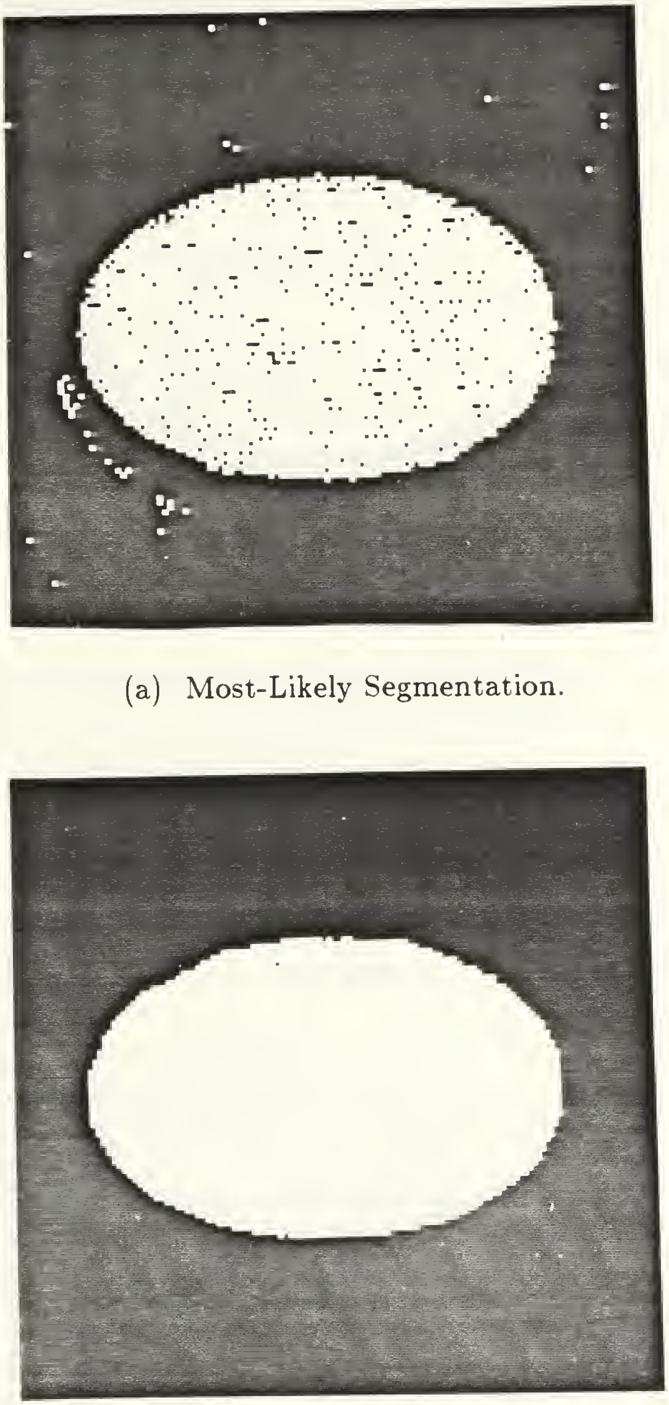


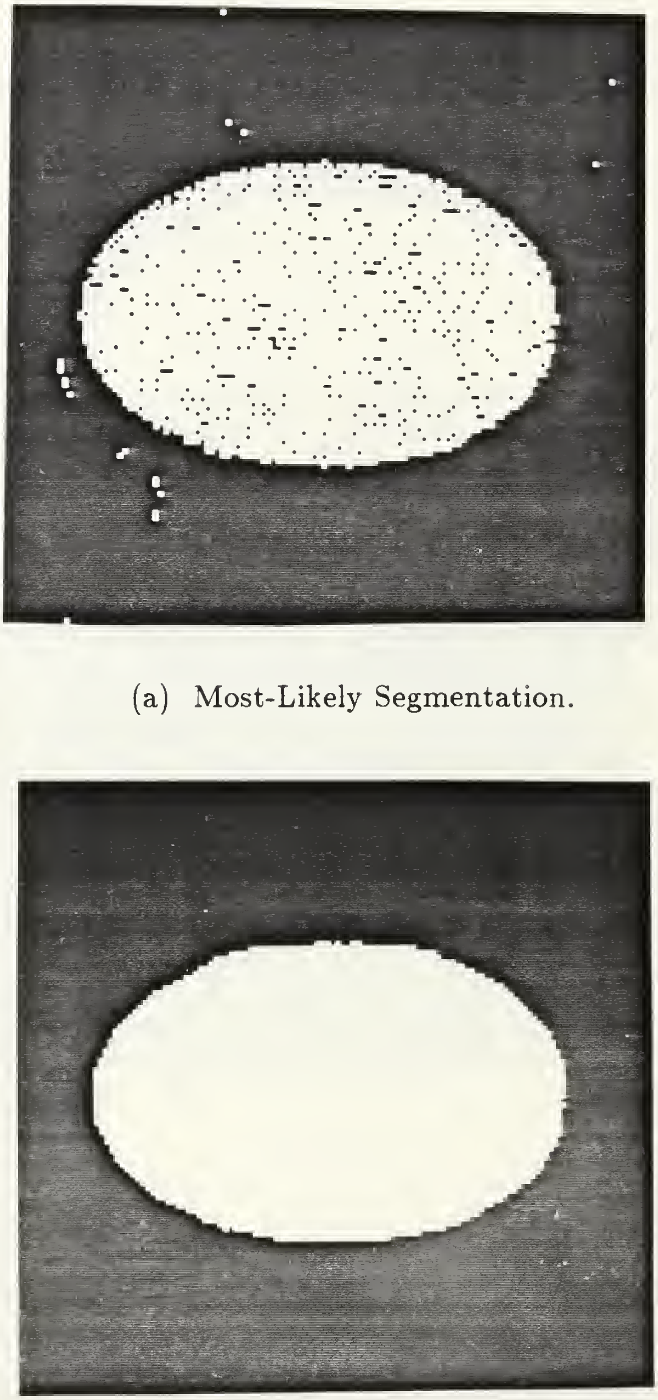
Figure A-3 Synthetic Two Texture Test Image.



(a) Most-Likely Segmentation.

(b) Maximum A Posteriori Segmentation.

Figure A-4 Segmentation of Test Image Using Known Filter Parameters.



(a) Most-Likely Segmentation.

(b) Maximum A Posteriori Segmentation.

Figure A-5 Segmentation of Test Image Using Estimated Filter Parameters.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
3.	Department Chairman, Code 62 Electrical and Computer Engineering Dept. Naval Postgraduate School Monterey, California 93943-5100	1
4.	Professor Charles W. Therrien, Code 62Ti Electrical and Computer Engineering Dept. Naval Postgraduate School Monterey, California 93943-5100	3
5.	Professor Chin-Hwa Lee, Code 62Le Electrical and Computer Engineering Dept. Naval Postgraduate School Monterey, California 93943-5100	3
6.	Capt. James F. Janecek 14272 Silverdale Drive Dale City, Virginia 22193	3
7.	Director, Research Administration, Code 012 Naval Postgraduate School Monterey, California 93943-5100	1

602 197

216653

Thesis
J274 Janecek
c.1 Algorithm for segmentation of multichannel images.

216653

Thesis
J274 Janecek
c.1 Algorithm for segmentation of multichannel images.

AMERICAN BINDERIES

thesJ274
Algorithm for segementation of multichan
3 2768 000 65177 2
DUDLEY KNOX LIBRARY