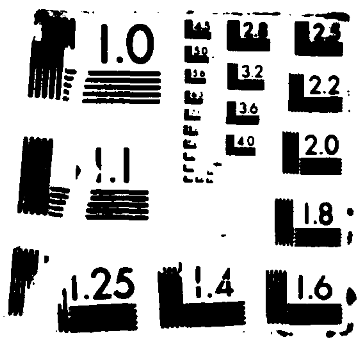
AD-A181 609 COMPARISON OF MODEL-BASED SEGMENTATION ALGORITHMS FOR COLOR IMAGES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA T KUPELI MAR 87 1/1

UNCLASSIFIED F/G 12/9 NL

END
DATE
FILMED


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPARISON OF MODEL-BASED

SEGMENTATION ALGORITHMS FOR COLOR IMAGES

by

Timur Kupeli

March 1987

Thesis Advisor: Charles W. Therrien

Approved for public release; distribution is unlimited

AD A181 609

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 62	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

COMPARISON OF MODEL-BASED SEGMENTATION ALGORITHMS FOR COLOR IMAGES

12 PERSONAL AUTHOR(S)

Kupeli, Timur

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1987 March	71

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Maximum Likelihood, Maximum A Posteriori, Markov Random Field, Karhunen-Loeve Transformation

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

The objective of this thesis is to develop segmentation methods for multichannel and single channel images, and compare these methods. The segmentation algorithms are based on a linear model for the image textures and on inverse filtering to estimate the image textures and their regions. Two specific methods are compared 1) A multichannel filtering algorithm that simultaneously models the three separate signals representing the intensity of red, green, and blue as a function of spatial position and 2) A single channel model applied to a combined image resulting from performing a Karhunen-Loeve transformation on the three signal components. Results of the multichannel image segmentation and the Karhunen-Loeve transformed one-channel image segmentation are presented and compared.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. C.W. Therrien	22b TELEPHONE (Include Area Code) 408-646-3347	22c OFFICE SYMBOL 62Ti

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Approved for public release; distribution is unlimited.

Comparison of Model-Based Segmentation Algorithms For Color Images

by

Timur Kupeli
Lieutenant JG, Turkish Navy
B.S., Turkish Naval Academy, 1980.

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1987

Author: ______________________
Timur Kupeli

Approved by: ______________________
Charles W. Therrien, Thesis Advisor

____
Chin-Hwa Lee, Second Reader

____
Harriett Rigas, Chairman, Department of Electrical and Computer Engineering

____
Gordon E. Schacher,
Dean of Science and Engineering

ABSTRACT

The objective of this thesis is to develop segmentation methods for multichannel and single channel images, and compare these methods. The segmentation algorithms are based on a linear model for the image textures and on inverse filtering to estimate the image textures and their regions. Two specific methods are compered 1) A multichannel filtering algorithm that simultaneously models the three separate signals representing the intensity of red, green, and blue as a function of spatial position and 2) A single channel model applied to a combined image resulting from performing a Karhunen-Loève transformation on the three signal components. Results of the multichannel image segmentation and the Karhunen-Loève transformed one-channel image segmentation are presented and compared. Keywords: maximum likelihood; Markov random fields; computer programs; Theses.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Segmentation techniques are among the most important considerations in the development of the automated image processing systems. Two related algorithms using 2-D *linear prediction models* and the *Karhunen-Loève transformation* for multichannel and color image segmentation are developed and compared in this thesis.

The purpose of segmentation is to partition an image into a set of simpler homogeneous regions. The regions may consist of different gray level, different textures, colors, etc. In some cases an " image " may consist of several spectral components. For example, a color image consists of three separate signals representing the intensity of red, green, and blue as a function of spatial position. If we represent each of these signals by functions F_r (n,m), F_b (n,m), and F_g (n,m), the image is represented by a vector quantity

$$\underline{F}(n,m) = \begin{bmatrix} F_r(n,m) \\ F_b(n,m) \\ F_g(n,m) \end{bmatrix} \quad (1.1)$$

where n and m represent spatial coordinates. We call such an image, consisting of more than one two dimensional (2-D) signal, a *multichannel* image.

In this thesis, a method based upon *linear prediction* is evaluated experimentally. This method has been developed [Ref. 1] for monochrome images and extended to color images [Ref. 2]. That method uses *maximum likelihood* (*ML*) and *maximum a posteriori* (*MAP*) estimation to segment multichannel images into regions of similar textures. The linear prediction is a filtering of the multichannel image to estimate the gray level at a particular spatial coordinate from the gray levels at neighboring positions. It is implemented as a 2-D linear filtering operation. The algorithm uses a previously-determined set of parameters corresponding to the mean of the data in each channel, the covariance matrix of the prediction error, and the weighting coefficients of the estimation filter for each specific texture type.

The method discussed above is compared to a variant of this method based on the *Karhunen-Loève* (*K-L*) *transformation. The* K-L transformation allows the several components of an image to be combined into a single image that retains most of the energy in the original image. Hunt and Kubler [Ref. 3] found that for image restoration, Karhunen-Loève transformation followed by single channel image processing worked nearly as well as multichannel image processing. It was desired to see if the Karhunen-Loève transformation would be equally effective for segmentation. In this part of the work, the K-L transformation has been developed to reduce the 3-channel color problem to a 1-channel problem and the segmentation was performed for the one-channel image. Karhunen-Loève transformation is based upon the statistical characteristics of an image. The advantage of this approach is computational savings; only about one ninth the number of computations is required for this method.

The remainder of this thesis is organized as follows. Chapter II discusses the model and the algorithms used to perform the multichannel image segmentation employing techniques of linear prediction [Ref. 4]. A general class of linear filtering models for texture is first presented. An algorithm is then developed to estimate the filter parameters from a multichannel image. Then, the multichannel image segmentation algorithm is described.

Chapter III presents the models and the algorithms to perform the Karhunen-Loève transformation and one-channel image segmentation. First, the algorithms for determining the eigenvectors and the eigenvalues of the correlation matrix are developed. Then, the transformation using a 3-channel image is presented. Finally, the one-channel image segmentation algorithm is discussed. The examples demonstrating the application of the segmentation methods for color images are presented and compared in Chapter IV. Chapter V has the conclusions about the multichannel image segmentation and one-channel image segmentation.

In Appendix A, the *Relaxation Method* is described briefly as an alternative to the *maximum a posteriori* region estimation for monochrome images. Results are compared with the MAP method. In each of Appendices B and C, the description and use of the computer programs for the multichannel image segmentation and the one-channel image segmentation are presented. The computer program for multichannel image segmentation is contained in Appendix D. The Karhunen-Loève transformation and one-channel image segmentation computer programs are contained in Appendix E. These programs are written in FORTRAN, compiled using Version 4.2 under the VAX / VMS Version 4.1 operation system.

II. IMAGE SEGMENTATION USING A MULTICHANNEL FILTERING MODEL

In this chapter, a multichannel image segmentation algorithm based upon a 2-D linear filtering model is presented. The multichannel images used in this work are color images with three channels representing the red, green, and blue components. The linear filtering model is used to develop approximate expressions for the multivariate probability density functions in terms of the filter error residuals for the entire set of points representing an image. The density expressions are used in the formulation and solution of the multichannel image segmentation problem. It is assumed that the multichannel images contain multiple regions of homogeneous texture. This is found to be the case in dealing with aerial photographs of natural terrain.

The problem of multichannel image segmentation is addressed as an estimation problem for two regions of texture. *Maximum likelihood* (*ML*) and *maximum a posteriori* (*MAP*) region estimates using a *Markov random field* to model region transitions are developed.

This chapter consists of two sections. The first section describes the linear filtering model and develops an expression for the image probability density function in terms of filter error residuals. The last section deals with the algorithm that is developed for the texture estimation and the multichannel image segmentation. The results of texture estimation and image segmentation are presented in Chapter IV.

A. MODEL DEVELOPMENT

In this section, a 2-D multichannel *autoregressive-moving average* (*ARMA*) model [Ref. 5] is first discussed. Then, we concentrate on the multichannel *autoregressive* (*AR*) model with *Gaussian white noise* inputs [Ref. 6]. The development parallels that in [Ref. 1]. A multichannel image is represented by a vector signal $\underline{f}^h$ (n,m) where (n,m) are spatial coordinates and the superscript h is an index representing the texture type. A 2-D multichannel image is shown in Figure 2.1 . A texture of type h is then modeled in general by a multichannel ARMA process defined by

$$\underline{F}^h(n,m) = -\sum_{\substack{i=0\\(i,j)\neq(0,0)}}^{P-1}\sum_{j=0}^{Q-1} A^h_{ij}\,\underline{F}^h(n-i,m-j) + \sum_{\beta} B^h_{ij}\,\underline{W}^h(n,m) \tag{2.1}$$

$$\underline{F}^h_m(n,m) = \underline{F}^h(n,m) + \underline{G}^h \tag{2.2}$$

for $h = 0,1$, $n = 1,\ldots,N$, $m = 1,\ldots,M$, where A_{ij}^h and B_{ij}^h are set of filter weighting coefficient matrices of size K by K. $\underline{W}^h(n,m)$ are a set of independent identically distributed zero-mean random variables, $\underline{G}$ is a constant representing the mean value of the image, and β is finite-extent mask covering the filtered points. $\underline{F}^h(n,m)$, $\underline{W}^h(n,m)$, and $\underline{G}^h$ are vectors of size K, *the number of channels in the image*. The matrices A^h_{ij} are the key parameters in the linear model. For a first quadrant filter with a $P \times Q$ region of support, there are PQ A^h_{ij} matrices with $A^h_{00} = \mathbf{I}$, an identity matrix.

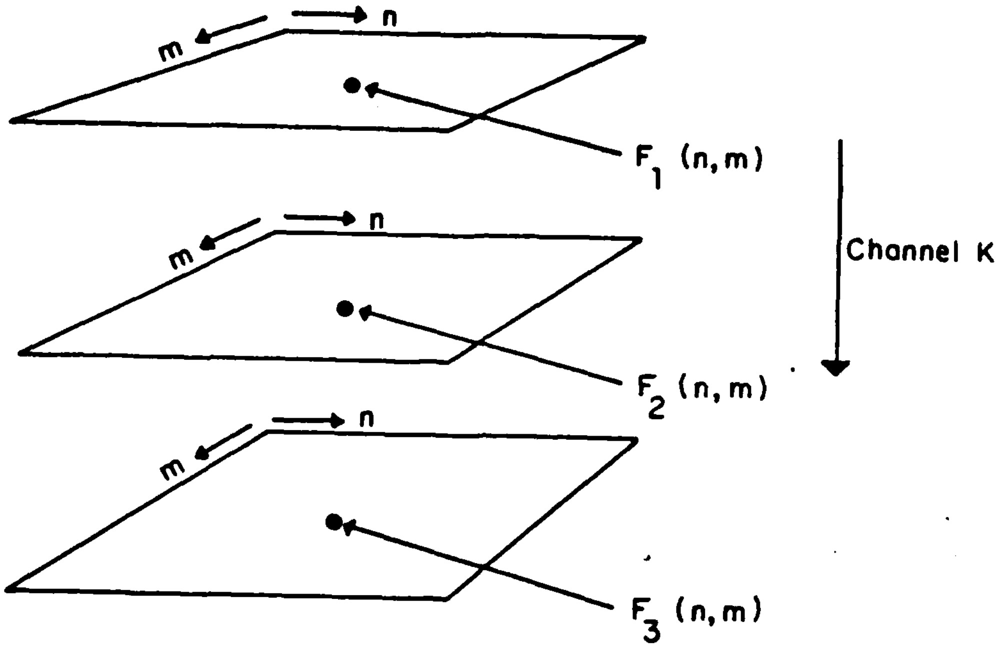
For the *auto-regressive* (*AR*) or *all-pole* model we have $B^h_{ij} = \delta\mathbf{I}_{ij}$ and so that Equation 2.1 reduces to

$$\underline{F}^h(n,m) = -\sum_{\substack{i=0\\(i,j)\neq(0,0)}}^{P-1}\sum_{j=0}^{Q-1} A^h_{ij}\,\underline{F}^h(n-i,m-i) + \underline{W}(n,m) \tag{2.3}$$

If the vectors **f**, **w**, and **g** represent an ordered set of the corresponding image points, then Equation 2.2 and Equation 2.3 are written in a matrix formulation as

$$A(\mathbf{f} - \mathbf{g}) = \mathbf{W} - A_0\,\mathbf{f}_0 \tag{2.4}$$

where A and A_0 are matrices whose nonzero elements are derived from the terms A_{ij} in Equation 2.1 and $\mathbf{f}_0$ represents a set of boundary conditions with support outside of the regions. Since the terms $\underline{W}(n,m)$ are independent with probability density function (PDF) p_w, One can solve Equation 2.4 for **W** and express the multivariate probability density function for the image conditioned on the boundary values as



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

Figure 2.1 2-D Multichannel Image Model.

$$p_{f \mid f_0}(f \mid f_0) = \frac{1}{|A^{-1}|} p_w(A(f-g) + A_0 f_0) \tag{2.5}$$

$$= \prod_{(n,m) \in R} p_w(\mathcal{E}(n,m))$$

where the notation E(n,m) is used to represent the ordered components of the vector $A(f-g) + A_0 f_0$. If the boundary conditions, f_0, are temporarily ignored, then

$$\mathcal{E}(n,m) = A(f - g) \tag{2.6}$$

and the terms $\mathcal{E}(n,m)$ in Equation 2.5 are computed from

$$\mathcal{E}^h(n,m) = \sum_{\substack{i=0 \\ (i,j) \neq (0,0)}}^{P-1} \sum_{j=0}^{Q-1} A^h_{ij} \mathcal{F}^h(n-i,m-j) \tag{2.7}$$

The filter of Equation 2.7 which computes $\mathcal{E}^h(n,m)$ from $\mathcal{F}^h(n,m)$ is referred to as the *'prediction error filter'*. One can think of Equation 2.7 as producing an estimate or *prediction* $\hat{\mathcal{F}}(n,m)$ of the image at point (n,m) and then forming an error $\mathcal{E}(n,m)$ as a difference $\mathcal{F}(n,m) - \hat{\mathcal{F}}(n,m)$. This process is known as 2-D linear prediction and is fundamental to performing multichannel image segmentation.

B. ALGORITHM DEVELOPMENT

In the multichannel segmentation problem, it is assumed that the image consists of multiple connected regions of known texture types, but that the region boundaries and the number of regions are not known. The segmentation of the image is treated as a supervised learning problem, since the regions are considered to consist of known texture types. In this section, the multichannel image segmentation algorithm for textured images is discussed.

An overview of the method is as follows. Given a multichannel image of each texture, filter parameters are estimated by computing the covariance matrix from a set of data and solving the Normal equations corresponding to the model of Equation 2.3 . In this case the *'correlation method'* of linear prediction is used to compute the covariance matrix. The filter parameters are derived from a statistical analysis of the textured images, because the image model discussed in the previous section is based upon statistical properties.

Once the filter parameters are known the filters are used to perform the segmentation. The filter weighting coefficients are used to calculate the prediction errors E^h (n,m) of two textures (n,m). Then, a maximum likelihood (ML) region estimate is developed using the prediction errors and the covariance matrices for image. The ML estimate is used as a basis to determine an approximate *maximum a posteriori* (MAP) region estimate. The MAP region estimation utilizes an underlying Markov structure for the region statistics to produce an accurate segmentation.

1. Filter Parameter Estimation Method

The prediction error filter is a *finite-extent impulse response* (*FIR*) or *nonrecursive* filter with selectable mask size and quarterplane region of support. It is always stable. The inverse of the filter used in Equation 2.3 is an all-pole filter (the AR filter). The filter parameter estimation problem requires calculating $\boldsymbol{\mu}^h$, Σ_w^h, and A^h by statistical analysis of data in an estimation window containing the desired texture, where the quantity $\boldsymbol{\mu}^h$ is a mean vector of the average gray level of the image in each of the channels, and Σ^h_w is a covariance matrix of the multichannel white noise

$$\Sigma^h_w = E\left[\, \mathbf{w}^h(n,m)\, (\mathbf{w}^h(n,m))^T \,\right] \tag{2.8}$$

the term Σ^h_w is also refered to as the prediction error covariance matrix, since in a linear prediction problem it represents the covariance of the quantities $\boldsymbol{\varepsilon}$ (n,m) defined in Equation 2.7 . Since Σ^h_w is not in general a diagonal matrix, we see that the 'white noise' is uncorrelated within each channel but correlated between channels.

The covariance of the multichannel white noise (the prediction error covariance) and the filter coefficients will be obtained by estimating the correlation function of the image and solving a set of Normal equations as discussed below. The correlation function itself is estimated from data in a window containing a sample of the desired texture. Two estimation windows are depicted in Figure 2.2 for two different textures in the multichannel image. The reader should realize that the estimation windows do not have to come from the image to be segmented; they can be selected from any image containing the same type of texture.

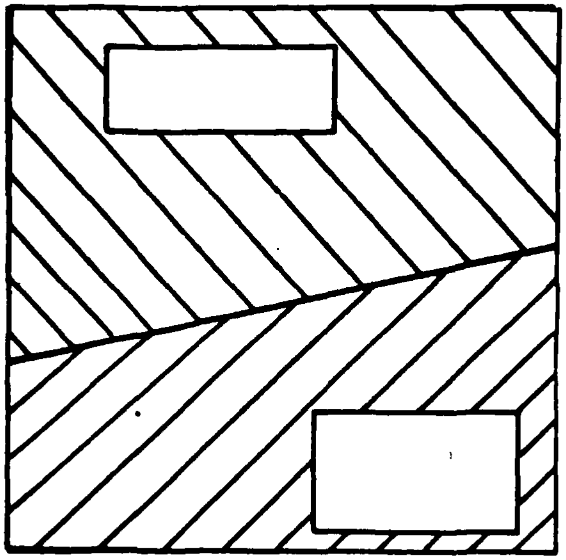


Figure 2.2 Typical estimation windows for two textures.

a. Mean Vector Estimation

In order to model the multichannel image by Equations 2.2 and 2.3 the mean vector of the multichannel image has to be estimated. Knowing $\mathbf{M}$ and selecting the stationary estimation windows of two desired textures of $\mathbf{F}_m$, the zero mean 2-D multichannel image, $\mathbf{F}$, appearing in Equation 2.3 can be obtained by subtracting the mean vectors from the multichannel image. Thus, Equation 2.2 becomes

$$\mathbf{F}^h(n,m) = \mathbf{F}^h_m(n,m) - \mathbf{M}^h \qquad (2.9)$$

where $\underline{M}^h$ corresponds to $\underline{G}^h$ in Equation 2.2, and the term $\underline{M}^h$ in Equation 2.9 are computed from

$$\underline{M}^h = \begin{bmatrix} M^h{}_1 \\ M^h{}_2 \\ M^h{}_3 \end{bmatrix} \quad (2.10)$$

where

$$\underline{M}^h{}_k = \frac{1}{N' M'} \sum_{n=X_1}^{X_2} \sum_{m=Y_1}^{Y_2} \underline{F}^h{}_k\,(n,m) \qquad (2.11)$$

$\underline{M}^h{}_k$ is the mean estimate for the k^{th} channel of the h^{th} texture image. The limits X_1, X_2, Y_1, Y_2 represents the edges of the window which is of size N' by M'. Therefore $N' = X_2 - X_1 + 1$, $M' = Y_2 - Y_1 + 1$, and $0 \leq X_1 < X_2 \leq N$, $0 \leq Y_1 < Y_2 \leq M$,and h represents the two textures of the multichannel image. All variables used in Equation 2.11 are depicted in Figure 2.3 .

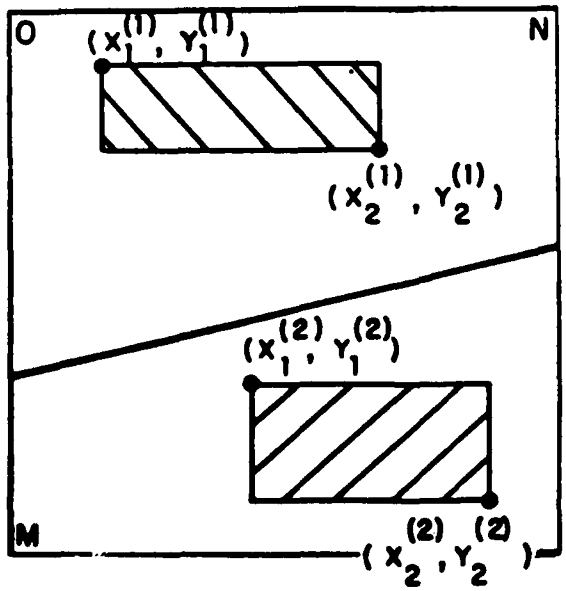


Figure 2.3 Selection of the Size of the Windows for Mean Vector Estimation.

b. Correlation Function Estimation

The correlation function of the zero mean, 2-D multichannel signal has to be calculated in order to estimate the multichannel white noise covariance or prediction error covariance, $\Sigma^h{}_w$, and the filter weighting coefficients, $A^h{}_{ij}$. The theoretical 2-D matrix correlation function for lag " i,j " is given by

$$R^h(i,j) = (R^h(-i,-j))^T \qquad (2.12)$$
$$= E[\mathcal{E}^h(n,m) \cdot (\mathcal{E}^h(n-i,m-j))^T]$$

and can be estimated from the multichannel signal by

$$R^h(i,j) = \frac{1}{N' M'} \sum_{n=n_1}^{n_2} \sum_{m=m_1}^{m_2} \mathcal{E}^h(n,m) (\mathcal{E}^h(n-i,m-j))^T \qquad (2.13)$$

where $R^h(i,j)$ is a matrix of size K by K, and n_1, n_2, m_1, m_2 are defined by

$0 \le n_1 = \max(X_1, X_1 + i) < n_2 = \min(X_2, X_2 + i) \le N$, and

$0 \le m_1 = \max(Y_1, Y_1 + j) < m_2 = \min(Y_2, Y_2 + j) \le M$.

This matrix correlation function is used to form a larger block Toeplitz covariance matrix which is used to estimate the filter parameters. This is discussed next.

c. Filter Coefficients and Prediction Error Covariance

The prediction error filter weighting coefficients and the prediction error covariance must satisfy a set of linear equations known as the Normal Equations when the multichannel image is represented by the model in Equation 2.3 .

Normal equations corresponding to Equation 2.3 can be written as

$$[R] \cdot [A] = [S] \qquad (2.15)$$

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

For a first quadrant filter with a P $\times$ Q region of support, The Normal equations that define Equation 2.15 have the specific form

$$\begin{bmatrix} R(0) & R(-1) & \cdots & R(-P+1) \\ R(1) & R(0) & \cdots & R(-P+2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R(P-1) & R(P-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} A^0 \\ A^1 \\ \cdot \\ \cdot \\ A^{P-1} \end{bmatrix} = \begin{bmatrix} S^0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \qquad (2.16)$$

where

$$R(i) = \begin{bmatrix} R(i,0) & R(i,-1) & \cdots & R(i,-Q+1) \\ R(i,1) & R(i,0) & \cdots & R(i,-Q+2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R(i,Q-1) & R(i,Q-2) & \cdots & R(i,0) \end{bmatrix} \qquad (2.17)$$

$$= R^T(-i)$$

and where $R(i,j)$ is the matrix correlation function described in Equations 2.12 and 2:13 The quantities A^i and S^0 are defined by

$$A^i = \begin{bmatrix} A_{i,0} \\ A_{i,1} \\ \cdot \\ \cdot \\ A_{i,Q-1} \end{bmatrix} \qquad (2.18)$$

and

$$S = \begin{bmatrix} \Sigma_w \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \qquad (2.20)$$

with $A_{00} = I$ and where A_{ij} and the partitions of S^0 are matrices of order K, the number of channels of image.

2. **Multichannel Segmentation Method**

An overview of the segmentation is as follows. By using a Gaussian probability density for the white noise in Equation 2.5 , one can develop explicit estimates for the density functions in terms of the prediction errors $\mathbf{E}$ (n,m) . From this one can form the conditions for ML and MAP estimation of the regions in the image. The theory leading to the estimates is explained below. A brief intuitive explanation of the process is given here.

As mentioned earlier the prediction error filters in Equation 2.3 can be considered as predicting the intensity of a pixel in each channel from data in the region adjacent to the vector of pixels. The prediction errors $\mathbf{E}^h$ (n,m) are the outputs of the filters. In the segmentation algorithm the prediction error is normalized in an expression involving the corresponding prediction error covariance. These normalized errors are compared in an appropriate formula to obtain the ML region estimate. When an area of texture is processed by a filter that is not matched to the texture, the normalized prediction error can be expected to be high. When the same area is processed by the filter that is matched to the texture, the prediction error can be expected to be low.

The *'maximum likelihood'* region estimate, ML(n,m), of texture class is achieved based on the prediction error covariance and the error estimates of the two desired textures for each pixel in the multichannel image. The ML(n,m) region estimate assigns pixels to texture types without regard to the assignments of the adjacent pixels. Then, the *'maximum a posteriori'* region estimate, MAP(n,m), of texture class is achieved for a pixel and a desired number of adjacent pixels of two textures of the ML region estimation result. The MAP region estimation uses the Markov model that refers to the above description. The form of the Markov model and ML and MAP region estimates are presented in detail below.

a. Maximum Likelihood Region Estimation

It is supposed that a multichannel image has many regions, but that each region contains only one or another of two texture types. Given these regions, one can write the Equation 2.5 as

$$p(\mathbf{E} \mid R_i) = \prod_{(n,m)} p_{w_{hi}}(\mathbf{E}(n,m)) \qquad i = 1, \ldots, q \tag{2.21}$$

where the $p_{w_{hi}}$ is the probability density function for the white noise source of type h_i within region R_i ,and q is the number of regions. When the white noise term is Gaussian with density function (mean 0 and covariance Σ_w)

$$p_{\mathbf{W}}(\mathbf{W}) = \frac{1}{(2\pi)^{NM/2} |\Sigma_w|^{1/2}} \exp\left(- \frac{1}{2} \mathbf{W}^T \Sigma_w^{-1} \mathbf{W} \right) \tag{2.22}$$

then, taking minus twice the log of Equation 2.21 and applying Equation 2.22 , we obtain for an N by M pixel multichannel image

$$\begin{aligned} - 2 \ln p(\mathbf{E} | R_1 , R_2 ,\dots\dots\dots,R_q) &= \sum_{(n,m) \in R_1} ([\mathbf{E}^1_h (n,m)]^T [\Sigma^1_h]^{-1} [\mathbf{E}^1_h (n,m)] + \ln|\Sigma^1_h| + .. \\ &+ \sum_{(n,m) \in R_q} ([\mathbf{E}^q_h (n,m)]^T [\Sigma^q_h]^{-1} [\mathbf{E}^q_h (n,m)] + \ln|\Sigma^q_h |) - NM \ln 2\pi \\ &= \sum_{i=1}^{q} \sum_{(n,m) \in R_i} ([\mathbf{E}^i_h (n,m)]^T [\Sigma^i_h]^{-1} [\mathbf{E}^i_h (n,m)] + \ln|\Sigma^i_h |) - NM \ln 2\pi \end{aligned} \tag{2.23}$$

For maximum likelihood estimation, the number of regions q and the regions themselves are considered to be deterministic parameters of the density function. An ML estimate for these parameters is obtained by choosing values that maximize Equation 2.21 or, minimize Equation 2.23 . Since $NM\ln 2\pi$ is constant value, the Equation 2.23 is minimized if every point (n,m) in the multichannel image to a region R_i of type h_i such that the term in brackets is minimum. Thus, one can write a ML region estimation for two textures as

$$ML_1 (n,m) \underset{1}{\overset{0}{\gtrless}} ML_0 (n,m) \tag{2.24}$$

where

$$ML_h(n,m) = [\mathbf{E}^h(n,m)]^T [\Sigma^h{}_w]^{-1} [\mathbf{E}^h(n,m)] + \ln|\Sigma^h{}_w| \qquad (2.25)$$

for h = 0 , 1, where the number above or below the inequality indicates the region class to which the pixel (n,m) is assigned. When the class is 1, the ML(n,m) is assigned 1 for a white pixel. Otherwise ML(n,m) is assigned 0 for a black pixel. Since the ML region estimate assigns pixels to black and white without regard to the assignments of adjacent pixels, this algorithm produces a number of false assignments and a somewhat '*spotty*' result.

b. Maximum A Posteriori Region Estimation

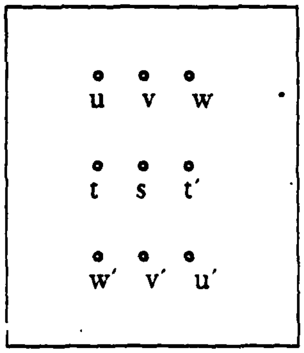
The 'maximum a posteriori' (MAP) region estimation utilizes the Markov model to describe the occurance of regions in the image. The combination of the linear filtering model with the Markov model results in an algorithm to achieve a MAP region estimation. For MAP estimation the regions are considered to be random quantities, and we maximize the probability for a given set of regions conditioned on our observation of the multichannel image. From Bayes rule, the *a posteriori* probability can be written as

$$\Pr[R_1, R_2, \ldots\ldots, R_q \mid \mathbf{E}] = \frac{p(\mathbf{E} \mid R_1, R_2, \ldots\ldots, R_q) \cdot \Pr[R_1, R_2, \ldots\ldots, R_q]}{p(\mathbf{E})} \qquad (2.26)$$

Since the denominator of Equation 2.26 does not depend on the regions, *maximum a posteriori* (MAP) estimate for the regions can be obtained if the R_i are chosen to maximize the numerator

$$p(\mathbf{E} \mid R_1, R_2, \ldots\ldots, R_q) \cdot \Pr[R_1, R_2, \ldots\ldots, R_q] \qquad (2.27)$$

One can define the " state " s(n,m) of point (n,m) as the region type to which that point has been assigned. In our development, the number of region types is assumed to be 2. Since the set of all possible state assignments for points in the image is one-to-one with the set of all possible divisions of the multichannel image into regions, the region estimation problem can be viewed as one of estimating the states of the points. It is assumed that the state of a point is stochastically dependent on some adjacent set of states $S_{n,m}$ in a symmetric support region, as shown in Figure 2.4 . Since S represents a chosen set of state assignments for all points in the multichannel image, Pr [S] denotes the joint probability that the points in the image take on a chosen set of state assignments. The support set defines a neighborhood structure i.e. all elements in the support set are neighbors of each other. It can be shown that if the set of states S is a Markov random field, then the probability of S can be factored as a product of terms of functions depending only on the "cliques" of the support set $S_{n,m}$. The cliques are defined as groups of points such that each set of points are neighbors of each other according to the support set. For a Markov random field, the probability of S can be written as



$S_{n,m}$

Figure 2.4 State support region of the point s for MAP estimation.

$$\Pr[\, S \,] = \prod_{(n,m)} \Pr[\, s(n,m) \mid S_{n,m} \,] \tag{2.28}$$

where the terms in the product are additive functions defined on the cliques of $S_{n,m}$ and the product is over all cliques in S. One simple acceptable form for the terms in the product is

$$\Pr[\, s(n,m) \mid S_{n,m} \,] = \frac{1}{D} \exp[\, s(n,m) \{\alpha + \beta_1 (t+t') + \beta_2 (v+v') + \gamma_1 (u+u') + \gamma_2 (w+w')\}] \tag{2.29}$$

where $t = s(n-1,m)$, $t' = s(n+1,m)$, $S_{n,m}$ is the set of states as snown in Figure 2.4, and D is a normalizing constant. One particular selection of the parameters, namely $\alpha = -4$, $\beta_1 = \beta_2 = \gamma_1 = \gamma_2 = 1$, leads to a particularly simple algorithm. In this case, we have

$$\Pr[\,1 \mid S_{n,m}\,] = \frac{1}{D} \exp\Big(\sum_{(i,j) \in S_{n,m}} (s(i,j) - 1/2) \Big) \tag{2.30}$$

and

$$\Pr[\,0 \mid S_{n,m}\,] = \frac{1}{D} \tag{2.31}$$

The second term in the numerator of Equation 2.26 can be replaced by Pr[S] . Thus maximizing Equation 2.26 is equivalent to minimizing

$$-\,2\ln p(\, \mathcal{E} \mid R_1, \ldots, R_q\,) - 2\ln \Pr[\,S\,] \tag{2.32}$$

One can define the MAP region estimate by combining Equations 2.22 , 2.26 , 2.28 , and 2.32 as

$$\sum_{(n,m)} [\,\mathcal{E}^1(n,m)\,]^T [\,\Sigma^1_w\,]^{-1} [\mathcal{E}^1(n,m)] + \ln|\,\Sigma^1_w\,| - 2\ln \Pr[\,1 \mid S_{n,m}\,] \underset{1}{\overset{0}{\lessgtr}}$$

$$\sum_{(n,m)} [\mathcal{E}^0(n,m)]^T [\Sigma^0_w]^{-1} [\mathcal{E}^0(n,m)] + \ln|\,\Sigma^0_w\,| - 2\ln \Pr[\,0 \mid S_{n,m}\,] \tag{2.33}$$

For $\Pr[\,h \mid S_{n,m}\,]$ in the form of Equation 2.30 and Equation 2.31 computing the terms $-\,2\ln \Pr[h|S_{n,m}]$ is equivalent to counting the number of pixels in $S_{n,m}$ that have value 'h' and dividing by the total number of pixels in $S_{n,m}$, and multiplying by an appropriate normalizing factor, KS.[1] A larger state support region $S_{n,m}$ is depicted in Figure 2.5 as an example. The side of the $S_{n,m}$ must be an *odd* number . Although it does not necessarily lead to the true MAP estimate we find it convenient in practice to maximize Equation 2.33 term by term. That is, we require to use Equation 2.33 without sum. Equation 2.33 can be solved by iteration using the maximum likelihood

[1]The normalizing factor results because the quantities α , β_1 , β_2 , γ_1 ,and γ_2 in Equation 2.29 can be scaled arbitrarily and still result in a legitimate probability function.

state estimates obtained from Equation 2.24 as an initial set of states for MAP region estimate.

0	1	0	1	1
1	0	1	1	0
1	1	.S	1	0
1	0	1	1	1
0	0	1	0	0

Figure 2.5 A Set of States 1's, 0's adjacent to S for MAP Region Estimate.

The computational requirements of the MAP region estimation are reduced by storing the differences,

$$MLD\,(n,m) = ML_1\,(n,m) - ML_0\,(n,m) \tag{2.34}$$

incurred during the calculation of the ML region estimate. Substituting Equation 2.34 in Equation 2.33 gives

$$MLD(n,m) - 2 \ln \Pr [\, 1 \mid S_{n,m} \,] + 2 \ln \Pr [\, 0 \mid S_{n,m} \,] \overset{1}{\underset{0}{\lesseqgtr}} 0 \tag{2.35}$$

At each iteration terms $\Pr [\, h \mid S_{n,m} \,]$ are evaluated based on the values of the states at the previous iteration. For our particular method of selecting $\Pr [\, h \mid S_{n,m} \,]$, Equation 2.35 can be expressed as

$$MLD(n,m) - KS \left| \frac{2\,(\text{number of state 1 pixels}) - (\text{number of pixels in } S_{n,m}) + 1}{\text{number of pixels in } S_{n,m}} \right| \overset{1}{\underset{0}{\lesseqgtr}} 0 \tag{2.36}$$

There are two important points in Equation 2.36 in order to perform the maximum a posteriori image segmentation accurately. The value of the convergence factor, KS, must be assigned properly. 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. In addition, the size of $S_{n,m}$ must be large enough. Otherwise, false assignments produced by the initial ML segmentation may not be removed.

III. KARHUNEN-LOÈVE TRANSFORMATION AND ONE-CHANNEL IMAGE SEGMENTATION

In this chapter, the models and relevant algorithms are presented to perform the Karhunen-Loève (K-L) transformation [Ref. 3] and one-channel image segmentation utilizing the techniques of linear prediction [Ref. 1]. Since linear prediction techniques are presented in detail for multichannel image segmentation in the previous chapter, this discussion concentrates on the K-L transformation. The K-L model and algorithm are first developed to reduce the three-channel color problem to a one-channel problem. Then, a one-channel segmentation procedure is presented that is based on the same model previously discussed. The results of the K-L transformed one-channel image segmentation are presented and compared with the multichannel image segmentation in Chapter IV.

A. MODEL DEVELOPMENT

The K-L transformation developed in this section is based on the statistical properties of an image. This transformation provides an energy compaction between channels of a color image. That is, most of the color image energy is compacted into one channel, and the transformed image channels are uncorrelated. If the multichannel image and transformed multichannel image are expressed in vector form, The K-L transformation is given by [Ref. 7]

$$\underline{\mathbf{Q}}(n,m) = [\mathbf{A}]\ \underline{\mathbf{F}}(n,m) \tag{3.1}$$

where $\underline{\mathbf{F}}(n,m)$ is the original multichannel image, $\underline{\mathbf{Q}}(n,m)$ is the transformed multichannel image, and **A** is the K-L transformation matrix, whose rows are eigenvectors of the between-channel correlation matrix R defined by

$$R = E[\underline{\mathbf{F}}(n,m)\underline{\mathbf{F}}^{\mathbf{T}}(n,m)]. \tag{3.2}$$

The between-channel correlation matrix of the transformed image is

$$\begin{aligned}\Lambda &= E[\underline{\mathbf{Q}}(n,m)\underline{\mathbf{Q}}^{\mathbf{T}}(n,m)] \\ &= [\mathbf{A}][\mathbf{R}][\mathbf{A}]^{-1}\end{aligned} \tag{3.3}$$

where the matrix Λ is a diagonal matrix of the form

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \qquad (3.4)$$

and the λ_k represent eigenvalues of the between-channel correlation matrix. The eigenvalues are ordered such that

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0 \qquad (3.5)$$

The importance of this property is that each eigenvalue λ_k is equal to the variance of the k th channel of the transformed multichannel image whose channels are uncorrelated. Then, it is a well known property of multivariate statistics that the total variability of the color image has the form [Ref. 3]

$$\lambda_T = \sum_{k=1}^{3} \lambda_k \, . \qquad (3.6)$$

which relates the total variability to the decorrelated component variations, λ_k. One can observe that often the λ_k values have a wide range of magnitudes, and the first component, λ_1 , will be sufficient to approximate λ_T with only a small percentage of error. This becomes the key idea for the use of the K-L transformation. Table 1 shows the energy distribution between the transformed color image channels of two test images.

Indeed, in a 3 channel image one can often find that the first channel of the K-L transformed image is sufficient to account for 99 percent of all the variability. That is, it is typical to find that

$$\lambda_1 = 0.99 \, \lambda_T \qquad (3.7)$$

The Karhunen-Loève transformation provides the best energy compaction [Ref. 8] and the advantage of this transformation is computational savings. We will see later that

TABLE 1
ENERGY DISTRIBUTION BETWEEN TRANSFORMED IMAGE CHANNELS

	Percentage of Energy in Channels		
	Q_1	Q_2	Q_3
Image 1	99.13	0.84	0.03
Image 2	99.15	0.78	0.07

one-channel image segmentation achieved by processing only the first channel of the K-L transformed color image will be very close to the result obtained on the original image with a multichannel algorithm, but will be obtained at only one ninth of the computational cost.

B. ALGORITHM DEVELOPMENT

In this section, the algorithm to perform K-L transformation of color images is presented. Since the K-L transformation is based upon the between-channel correlation matrix, **R** , of the color image , the between-channel correlation matrix is first determined. Then the eigenvectors of the between-channel correlation matrix are computed. Finally, the K-L transformation is formulated using the transpose of the eigenvector matrix.

1. Correlation Function Estimation

In order to determine the K-L transformation of Equation 3.1 , the between-channel correlation matrix of the color image must first be estimated. Our estimate for the between-channel correlation matrix is given by

$$R = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \mathbf{F}(n,m)(\mathbf{F}(n,m))^T \qquad (3.8)$$

where the image size is N by N, and the between-channel correlation matrix size is K by K.

2. Karhunen-Loeve Transformation Matrix

Since the rows of the K-L transformation matrix are the eigenvectors, E, of the between-channel correlation matrix, the eigenvectors must be calculated from the between-channel correlation matrix of the color image. The K-L transformation matrix is then obtained using the transpose of the eigenvector matrix. That is

$$[\ A\ ] = \begin{bmatrix} \leftarrow e^T_1 \rightarrow \\ \leftarrow e^T_2 \rightarrow \\ \leftarrow e^T_3 \rightarrow \end{bmatrix} \qquad (3.9)$$

3. Karhunen-Loeve Transformation

Since the K-L transformation matrix satisfies Equation 3.3 , then the K-L transformation is represented by Equation 3.1 with A is given by Equation 3.9, where e^T_i , i = 1 , 2 , 3 are the eigenvectors of R. More explicitly, the components of $\underline{Q}$ are determined from the components of $\underline{F}$ at any pixel (n , m) as

$$\begin{bmatrix} Q_1(n,m) \\ Q_2(n,m) \\ Q_3(n,m) \end{bmatrix} = \begin{bmatrix} \longleftarrow e^T_1 \longrightarrow \\ \longleftarrow e^T_2 \longrightarrow \\ \longleftarrow e^T_3 \longrightarrow \end{bmatrix} \begin{bmatrix} F_1(n,m) \\ F_2(n,m) \\ F_3(n,m) \end{bmatrix} \qquad (3.10)$$

C. ONE-CHANNEL IMAGE SEGMENTATION

The one-channel segmentation procedures presented in this section are based on the same model that was described in detail for the multichannel image in Chapter II. A single channel image is used instead of a three channel image. That is, K is always equal to 1 in the Equations of Chapter II. As a result the vector and matrix quantities become scalars and the equations are simplified.

In order to model the single image by Equations 2.2 and 2.3, the mean of the image is estimated using the Equation 2.11 . Then the correlation function is estimated in the same manner as in Section II-B.b where R(i,j) is a scalar instead of a matrix. This procedure is followed by estimation of the filter coefficients and the prediction error covariance. The equations in Section II-B.c are then used to determine the filter coefficients, A_{ij} , and the the prediction error covariance, Σ_w , now a scalar value. Finally, the one-channel segmentation algorithm is applied. The method is the same as that described in detail for the color images in Section II-B.2. However, instead of Equation 2.24 and 2.33 the following simplified relations are used for the maximum likelihood and the maximum a posteriori region estimates. A maximum likelihood region estimate for a single channel image of the pixel is given by [Ref. 1]

$$\frac{(E^1(n,m))^2}{\Sigma^1_w} + \ln(\Sigma^1_w) \underset{1}{\overset{0}{\gtrless}} \frac{(E^0(n,m))^2}{\Sigma^0_w} + \ln(\Sigma^0_w) \tag{3.11}$$

and the maximum a posteriori region estimate is given by

$$\frac{(E^1(n,m))^2}{\Sigma^1_w} + \ln(\Sigma^1_w) - 2\ln\Pr[1 \mid S_{n,m}] \underset{1}{\overset{0}{\gtrless}} \frac{(E^0(n,m))^2}{\Sigma^0_w} + \ln(\Sigma^0_w) - 2\ln\Pr[0 \mid S_{n,m}] \tag{3.12}$$

where E^1 and E^0 are the result of applying the linear predictive filters to the first channel of the K-L test image.

IV. RESULTS AND COMPARISON OF THE METHODS

In this chapter, the results of the multichannel image segmentation, the Karhunen-Loêve transformed one-channel image segmentation, and the K-L transformed 3-channel image segmentation are presented and compared.

The digitized image size used in this work is 128 by 128 pixels with gray levels represented on a scale of 0 to 255 (8 bits). A digitized color photograph of a rural area containing trees (the green region) and fields (the yellow region) are shown in Figure 4.1 . A quarter-plane filter for each texture class (2 × 2 pixels) was designed and applied to the color image to achieve the multichannel image segmentation. A state support region of 7 by 7 pixels was used for MAP region estimation. The results of the maximum likelihood (ML) and the maximum a posteriori (MAP) segmentations of Figure 4.1 image are shown in Figures 4.2 and 4.3 respectively. The segmentation results show the field regions as black and tree regions coded as white. The ML result (Fig 4.2) is spotty, but the true tree and field regions are distinguishable. The MAP segmentation result (Fig 4.3) of Figure 4.1 image is quite clear. The MAP segmentation was not able to remove a few improperly classified points in the left side of the field region. Since there is an inherent ambiguity in the estimation of the region boundaries due to the finite size of the masks, the edges are expectedly somewhat rough. Figure 4.4 shows another color image of a rural area containing trees and fields. Figures 4.5 and 4.6 show the results of the ML and the MAP segmentation of the Figure 4.4 image. The ML estimation was achieved using the filters designed for the image of Figure 4.1 . The result of the ML segmentation (Fig 4.5) is again quite spotty, i.e. Although two regions are discernible. The MAP algorithms segmented the regions quite accurately as shown in Figure 4.6 . The MAP estimates presented above converged after 10 iterations and KS was assigned to 100.

In the results presented above the ML procedure produced a poor result with a lot of false regions. This is due to the lack of prior information about region connectivity. On the other hand, MAP estimation using the Markov model to represent region transitions produced results that was quite accurate.

Figure 4.7 shows the first channel of the Karhunen-Loêve (K-L) transformed image of the Figure 4.1 . The image size is 128 by 128 pixels with the scaled gray levels within the intensity range of the display. The result of the one-channel ML segmentation of Figure 4.7 is depicted in Figure 4.8 . The one-channel ML segmentation result is quite spotty, but the two regions are perceptually discernible. On the other hand, the MAP segmentation result (Fig 4.9) of Figure 4.7 is quite smooth, although there are a few incorrectly classified points in both regions.

The first channel of the Karhunen-Loève transformed image of Figure 4.4 is shown in Figure 4.10 , and Figures 4.11 and 4.12 show the results of the one-channel ML and MAP segmentations of Figure 4.10 respectively. The one-channel ML segmentation result (Fig 4.11) has a lot of spots, but both segmented regions are distinguishable. The maximum a posteriori segmentation (Fig 4.12) of Figure 4.10 is quite smooth, but again there are a few incorrectly classified sets of pixels in both regions.

The results of multichannel image segmentation and the Karhunen-Loève transformed one-channel image segmentations were presented consecutively. These results show that the ML estimation of the Karhunen-Loève transformed single channel image is much more spotty than the ML estimation of the multichannel image. But, the MAP estimation results of the K-L transformed single channel images are as clear as the MAP segmentation results of the multichannel images.

In summary the results presented in this chapter show that the best segmentation result is provided by the multichannel image segmentation method. However the segmentation results of the Karhunen-Loève transformed single channel images are very close to multichannel image segmentation results, i.e. Because most of the color image energy (99 percent) is compacted into the first channel.

Figure 4.1 Two-Texture Color Image.

Figure 4.2 ML Region Estimation of Figure 4.1 Image.

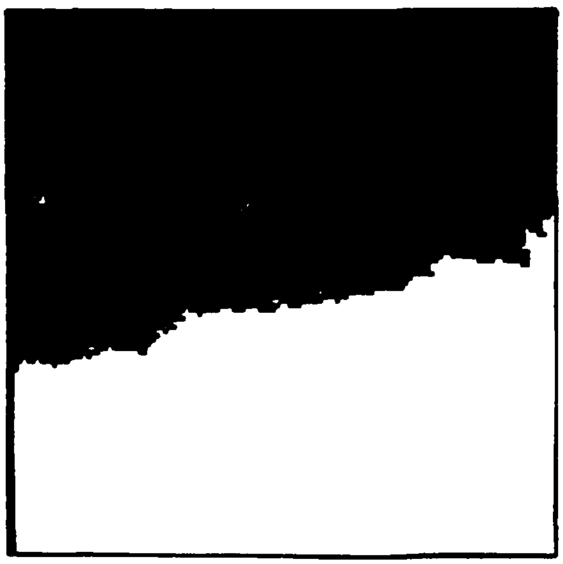


Figure 4.3 MAP Region Estimation of Figure 4.1 Image.

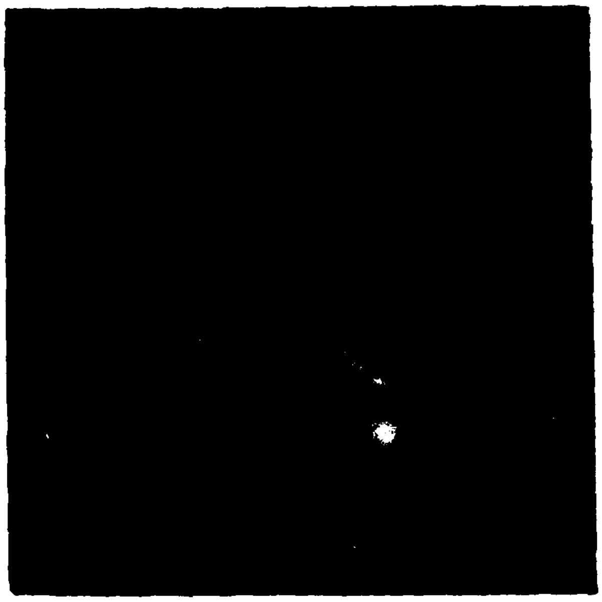


Figure 4.4 Color Image Containing Two-Texture.

Figure 4.5 ML Region Estimation of Figure 4.4 Image.

Figure 4.6 MAP Region Estimation of Figure 4.4 Image.

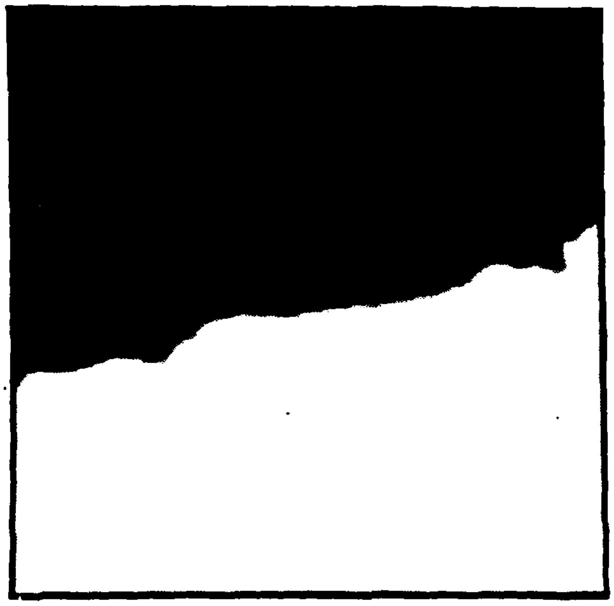


Figure 4.7 First Channel of K-L Transformation of Figure 4.1 Image.

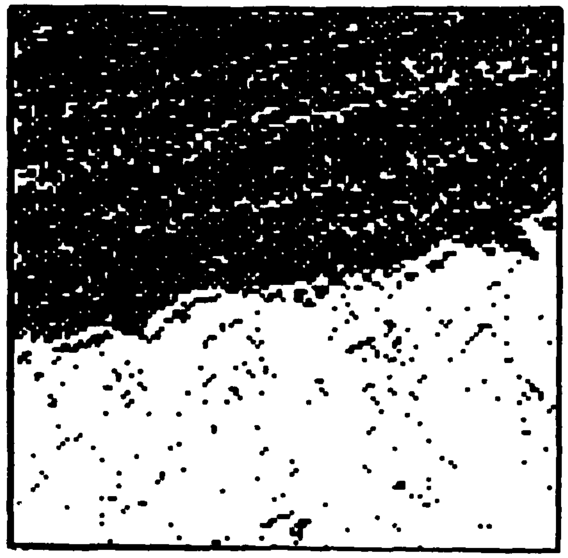


Figure 4.8 ML Region Estimation of Figure 4.7.

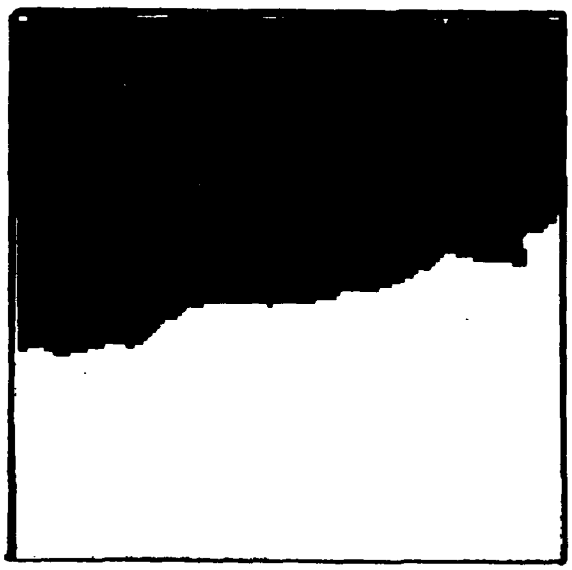


Figure 4.9 MAP Region Estimation of Figure 4.7.

Figure 4.10 K-L Transformed Single Channel Image of Figure 4.4.

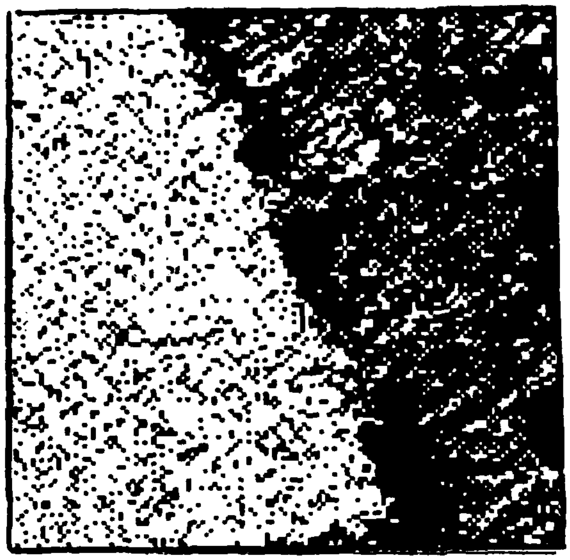


Figure 4.11 ML Segmentation of Figure 4.10.

Figure 4.12 MAP Segmentation of Figure 4.10.

V. CONCLUSIONS

The segmentation of terrain images is an important part of image analysis methods for military and civilian applications. The work in this thesis utilized a 2-D stochastic linear filtering model and compared algorithms for multichannel image segmentation of color images. Two levels of structure were used for multichannel segmentation development. The fundamental structure based on the linear filtering concepts represents the texture in local regions of terrain. Superimposed on this structure is a Markov random field that describes transitions from one region type to another. The segmentation was considered as a region estimation problem and maximum likelihood and maximum a posteriori region estimatation methods were developed. The ML region estimation produced a spotty result, but the MAP region estimation produced quite accurate results for the multichannel and single channel image.

The other piece of work developed in this thesis was the Karhunen-Loêve transformation model that based on the statistical characteristics of color image.The one-channel image segmentation was then applied to the first channel of the Karhunen-Loêve transformed color image to see the effectiveness of the K-L transformation for segmentation.

We observed that multichannel image segmentation results were quite accurate. Similarly the results of the K-L transformed one-channel image segmentation were very smooth. In summary the results of both segmentation methods were very close to each other, and the K-L transformation is very effective for segmentation.

APPENDIX A
RELAXATION METHOD

The Relaxation Method algorithm utilizes a set of iterative numerical techniques to compute a posterior probability of the pixel (n,m) from a prior probability of the same pixel and a set of prior probabilities of the adjacent pixels. The prior probabilities are estimated from a 2-D image data using the linear filtering model (see Equation 2.25).

The Relaxation formula is defined [Ref. 9] by

$$P_{ij}^{\,k+1}(n,m) = \underset{s}{\mathrm{Avg}} \left(\frac{\lambda_{ij}^{s} P_{ij}^{k}(n,m)}{\sum_{t=1}^{T} \lambda_{it}^{s} P_{it}^{k}(n,m)} \right) \qquad \text{(A.1)}$$

where k is the number of the iteration, λ_{ij}^{s} is the relaxation factor, $P_{ij}^{k}(n,m)$ are a set of prior probabilities, T is the number of textures, and S is the number of pixels. The updated estimates $P_{ij}^{k+1}(n,m)$ are obtained by averaging all of the terms in parentheses. The relaxation factor, λ_{ij}^{s} , in Equation A.1 is given by

$$\lambda_{ij}^{s} = \sum_{t=1}^{T} c(i,j \mid s,t)\, P_{st}^{k}(n,m) \qquad \text{(A.2)}$$

where c(i,j|s,t) is a nonnegative compatibility function whose value is small if the neighboring pixel is black when the estimated pixel is white, otherwise its value will be large. λ_{ij}^{s} is used to update the probability $P_{ij}^{k}(n,m)$. Note that λ_{ij}^{s} is large if the compatibilities c(i,j|s,t) are large and the probabilities P_{st}^{k} are high, otherwise λ_{ij}^{s} will be small.

The results of the Relaxation Method segmentation are shown in Figures A.1 and A.2 . The Figure A.1 presents the segmentation of Figure 4.7 with the compatibility c(i,x|s,x) is equal to 0.1, where x can be 0 or 1. The result is very spotty but both regions are discernable. The Figure A.2 gives the segmentation with the compatibility c(i,x|s,x) is equal to 0.9. This result is better than the previous result, but there are still several spots especially in the field region. Both results are obtained after

10 iterations. Figure 4.9 shows the result of MAP segmentation of Figure 4.7 . The MAP segmentation result is much more accurate than the Relaxation method segmentation.

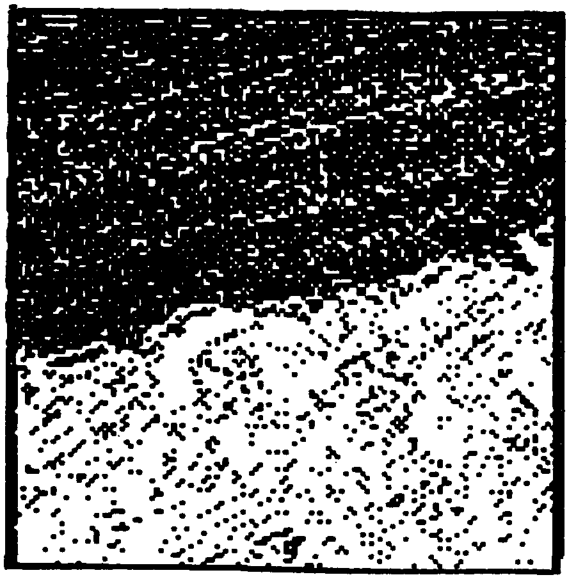


Figure A.1 Relaxation Method Segmentation with c = 0.1.

Figure A.2 Relaxation Method Result with c = 0.9.

APPENDIX B

FILTER PARAMETER ESTIMATION AND MULTICHANNEL SEGMENTATION

The interactive program, **FLTR1**, estimates two sets of filter parameters from a 2-D three-channel image. The corresponding algorithms were presented in Section II.B.1. In order to run this program, the user must input the required program parameters in the order listed below

* The number of filter rows, P. Maximum is 4.
* The number of filter columns, Q. Maximum is 4.
* The number of rows, N, in each image channel. Maximum is 128.
* The number of columns, M, in each image channel. Maximum is 128.
* The number of channels, K, in image.
* The filenames of the image channels.
* The coordinates of the estimation windows of textures.
* Two output filenames for the estimated filter parameters.

The mean vectors of the data in the two estimation windows is first computed by FLTR1. Then, the program subtracts the mean from the image and determines the correlation functions. After calculating the correlation matrix, the program determines the covariance matrix, Σ , using the equation

$$[R][B] = [S_I] \tag{B.1}$$

where B is a dummy matrix that has the same dimension with A matrix, $B_{00} = [\Sigma_w]^{-1}$ and S_I is all zero except for an identity matrix in its first partition. The covariance matrix must satisfy the relation

$$B_{00} \cdot \Sigma_w = I \tag{B.2}$$

Finally, the filter weighting coefficients are estimated using

$$[A] = [B] \cdot [\Sigma] \tag{B.3}$$

MULTICHANNEL IMAGE SEGMENTATION PROGRAM

The interactive program, **SGMT1**, segments a color image with two textures when given the three-channel image, the image dimension, and two sets of filter parameters. Again, the user must input the required program parameters to run the program. These parameters have to be in the order listed below :

* The number of rows, N, in the image channels.
* The number of columns, M, in the image channels.
* The number of filter rows, P.
* The number of filter columns, Q.
* The number of textures, T, in the image.
* The number of channels, K, in the image.
* The filenames of the image channels.
* The filenames of the filter parameters.
* The output filename of the ML segmentation.
* The output filename of the MAP segmentation.

This program estimates the errors of two textures using Equation 2.7, then performs the ML segmentation and the MAP segmentation using Equations 2.25 and 2.33 . The convergence factor KS, and the size of $S_{n,m}$ must be assigned properly by user to perform the MAP segmentation accurately.

APPENDIX C
KARHUNEN-LOÈVE TRANSFORMATION

The interactive program, **KRLV,** implements the Karhunen-Loève transformation algorithms described in Section III.B. This program requires the user to assign the program parameters in the order listed below :

* The number of channels, K, in the image.
* The number of rows and columns, N, in each channel.
* The filenames of the image channels.
* The output filenames of the transformed color channels.

The correlation matrix is first calculated. Then the program calls the IMSL subroutine EIGRS to calculate the eigenvalues and the eigenvectors of the correlation matrix. Finally, the transformed color image is obtained by multiplying the transpose of the eigenvectors matrix by the original color image. The program scales the transformed color image for display on the COMTAL image prossessing system.

ONE-CHANNEL IMAGE SEGMENTATION PROGRAM

The program, **FLTR2,** calculates two sets of filter parameters from a 2-D single channel image. The user must input the required program parameters to run the program. These parameters except the number of channel, K, are given in Appendix B.

The program, **SGMT2,** implements a single channel image segmentation. The required program parameters given in Appendix B have to be assigned to run the program. Equations 3.11 and 3.12 are used to perform the ML region estimation and the MAP region estimation.

APPENDIX D

COMPUTER PROGRAMS FOR MULTICHANNEL IMAGE SEGMENTATION

```
C **************************************************************
C *                                                            *
C *   PROGRAM FLTR1                                            *
C *                                                            *
C *   PURPOSE    To develop two sets of filter parameters from a 2-D   *
C *              three-channel input image. These parameters are the   *
C *              mean vector, the covariance, and the filter coeffici- *
C *              ents of the image.                            *
C *                                                            *
C *   REQUIRED IMSL ROUTINES                                   *
C *                                                            *
C *         LEQT2F, LUDATN, LUELMN, LUREFN                     *
C *                                                            *
C *   IMPLEMENTED BY LTJG TIMUR KUPELI Nov 1986                *
C *                                                            *
C *                                                            *
C **************************************************************
C
C       ****   VARIABLE DEFINITIONS   ****
C
C       BINPUT = [ Input ] Image data in byte format.
C       FINPUT = [ Input ] image data in real*4 format.
C       P , Q  = Rows, Columns of the linear filter.
C       N , M  = Rows, Columns of the image data.
C       K      = The number of channels in image.
C       T      = The number of filters for processing.
C       IMAGE  = [ Input ] Filename of the image channels.
C       FILTER = [ Output ] Filename of the filter parameters.
C       MEAN   = Array of mean vectcrs of estimation windows.
C       R      = The correlation matrix.
C       IMATRX = Identity matrix.
C       KW     = The covariance matrix
C       A      = The filter coefficients matrix.
C       ISIZE  = The estimation window size.
C       NO, MO = Row, column delay(shift) of correlation.
C
C       This program uses 2 by 2 pixels filter. If user wants to use
C       different size filter, the dimensions of the following
C       variables must be modified.
C               R, SI, A, WAREA
C       For exemple, for 4 by 4 pixels filter, the dimensions must be
C               R(48,48), SI(48,3), A(48,3), WKAREA(2448)
C
C
        INTEGER*4   P,Q,ROW,COL,STARTN(2),STARTM(2),ENDN(2),ENDM(2),
     *              RNROW,RNCOL,RROW,CCOL,X,Y,K,IER,QK,PKQ,L,J,JJ,JJJ,
     *              HNNO,HMMO,HCOL,HROW,LNNO,LNMO,LCOL,LROW,RN,RM,T,
     *              NO,MO,RRN,RRM,ROW1,COL1,MMO,NNO,IDGT,I
C
        REAL*4    SUM,TEMP,FINPUT(128,128,3),MEAN(2,3),WKAREA(180),
     *            R(12,12),SI(12,3),IMATRX(3,3),BOO(3,3),SUM1,SUM2,SUM3,
     *            KW(3,3),A(12,3),ISIZE
C
        CHARACTER*50    IMAGE(3),FILTER(2)
C
        BYTE    BINPUT(128)
C
        T = 2
C
C               GET PROGRAM INPUT PARAMETERS.
```

```
C
         TYPE 10
   10   FORMAT(' ENTER NUMBER OF FILTER ROWS FROM 2-4 DESIRED :',$)
         READ 11,P
   11   FORMAT(I3)
C
         TYPE 12
   12   FORMAT(' ENTER NUMBER OF FILTER COLUMNS FROM 2-4 DESIRED :',$)
         READ 11,Q
C
         TYPE 13
  13    FORMAT(' ENTER NUMBER OF ROWS IN IMAGE ',$)
         READ 11,N
C
         TYPE 14
  14    FORMAT(' ENTER NUMBER OF COLUMNS IN IMAGE ',$)
         READ 11,M
C
         TYPE 15
   15   FORMAT(' ENTER NUMBER OF CHANNELS [ MAX = 3 ] IN IMAGE ;',$)
         READ(*,11) K
C
C              GET THE MULTICHANNEL IMAGE
C
         DO 16 J = 1 , K
C
           WRITE(*,17) J
   17     FORMAT(' ENTER FILENAME OF IMAGE CHANNEL ' I3,$)
           READ(*,18) IMAGE(J)
   18     FORMAT(A50)
C
C       CONVERT THE IMAGE FROM BYTE FORMAT TO REAL*4 FORMAT
C
         OPEN(UNIT=1,FILE=IMAGE(J),STATUS='OLD',ACCESS = 'DIRECT')
C
         DO 180  ROW = 1 , N
                READ (1'ROW) (BINPUT(COL),COL=1,M)
                DO 181  COL = 1 , M
                        TEMP = BINPUT(COL)
                        IF (TEMP.LT.0.0) THEN
                           TEMP = TEMP + 256
                        END IF
                        FINPUT(ROW,COL,J) = TEMP
 181             CONTINUE
 180    CONTINUE
         CLOSE (UNIT=1)
C
 16     CONTINUE
C
C       GET 'T' AREAS FOR WHICH FILTERS ARE DESIRED AND OUTPUT FILENAMES
C       FOR EACH AREA'S FILTER COEFFICIENTS AND COVARIANCE MATRIX.
C
         DO 19   I = 1,T
            WRITE(*,20) I
   20     FORMAT(' ENTER UPPER-LEFT ROW FOR AREA ',I2,':',$)
            READ(*,11) STARTN(I)
C
            WRITE(*,21) I
   21     FORMAT(' ENTER UPPER-LEFT COLUMN FOR AREA',I2,':',$)
            READ(*,11) STARTM(I)
C
            WRITE(*,22) I
   22     FORMAT(' ENTER LOWER-RIGHT ROW FOR AREA',I2,':',$)
            READ(*,11) ENDN(I)
C
            WRITE(*,23) I
   23     FORMAT(' ENTER LOWER-RIGTH COLUMN FOR AREA',I2,':',$)
            READ(*,11) ENDM(I)
C
```

```
          WRITE(*,24) I
   24     FORMAT(' ENTER OUTPUT FILE-SPEC FOR FILTER',I2,':',$)
          READ(*,18) FILTER(I)
C
C      FIND THE MEAN VECTOR OF ESTIMATION WINDOW , T, AREAS OF IMAGE
C      CHANNELS. THE MEAN VECTOR CONSISTS OF THE MEAN FOUND EACH CHANNEL
C
        ISIZE = (ENDN(I) - STARTN(I) + 1)*(ENDM(I) - STARTM(I) + 1)
        SUM1 = 0.0
        SUM2 = 0.0
        SUM3 = 0.0
        DO 25   L = STARTN(I),ENDN(I)
            DO 26   J = STARTM(I),ENDM(I)
                SUM1 = SUM1 + FINPUT(L,J,1)
                SUM2 = SUM2 + FINPUT(L,J,2)
                SUM3 = SUM3 + FINPUT(L,J,3)
   26       CONTINUE
   25   CONTINUE
        MEAN(I,1) = SUM1 / ISIZE
        MEAN(I,2) = SUM2 / ISIZE
        MEAN(I,3) = SUM3 / ISIZE
        WRITE(*,27) (MEAN(I,II),II=1,K)
   27   FORMAT('   MEAN:',3F9.3,$)
C
C             CORRECT THE IMAGE TO BE ZERO MEAN
C
        DO 271  J = 1 , K
        DO 28  L = STARTN(I), ENDN(I)
            DO 29  LL = STARTM(I),ENDM(I)
                FINPUT(L,LL,J) = FINPUT(L,LL,J) - MEAN(I,J)
   29       CONTINUE
   28   CONTINUE
   271  CONTINUE
C
C      DETERMINE THE 2-D CORRELATION FUNCTION OF THE IMAGE CHANNEL.
C      THE CORRELATION MATRIX APPROPRIATE TO THE INPUT INPUT PARAMETERS
C      OF P, K, Q
C
        WRITE(*,291) I
  291   FORMAT('              CORRELATION.MATRIX',I2,$)
        PKQ = P * K * Q
        QK = Q * K
        DO 30  RNROW = 1,P
           X = QK * (RNROW-1)
           DO 31  RNCOL = 1,P
                NO = RNROW - RNCOL
                Y = QK * (RNCOL-1)
                DO 32  RMROW = 1,Q
                    RN = K * (RMROW - 1) + X
                    DO 33  RMCOL = 1,Q
                        MO = RMROW - RMCOL
                        RM = K * (RMCOL - 1 ) + Y
C
                    LNNO = STARTN(I) + NO
                        LMMO = STARTM(I) + MO
                        HNNO = ENDN(I) + NO
                        HMMO = ENDM(I) + MO
C
                        LCOL =MAX0(STARTM(I),LMMO)
                        HCOL =MIN0(ENDM(I),HMMO)
                        LROW =MAX0(STARTN(I),LNNO)
                        HROW =MIN0(ENDN(I),HNNO)
C
        DO 133  ROW1= 1 , 3
           RRN = RN + ROW1
             DO 233  COL1 = 1 , 3
                RRM = RM + COL1
                SUM = 0.0
                DO 333  ROW = LROW , HROW
```

```
                              NNO = ROW - NO
                              IF (NNO .LT. LROW) THEN
                                   RROW = MAXO(ROW,NNO)
                              ELSE IF (NNO .GT. HROW) THEN
                                   RROW = MIN(ROW,NNO)
                              ELSE
                                   RROW = NNO
                              END IF
C
                              DO 433  COL = LCOL , HCOL
                                   MMO = COL - MO
                                   IF (MMO .LT. LCOL) THEN
                                      CCOL = MAXO(MMO,COL)
                                   ELSE IF (MMO .GT. HCOL) THEN
                                      CCOL = MINO(MMO,COL)
                                   ELSE
                                         CCOL = MMO
                                   END IF
C
              SUM = SUM + FINPUT(ROW,COL,ROW1) * FINPUTSAVEW,CCOL,COL1)
C
  433           CONTINUE
  333         CONTINUE
                R(RRN,RRM) = SUM / ISIZE
  233       CONTINUE
  133     CONTINUE
C
   33                CONTINUE
   32             CONTINUE
   31          CONTINUE
   30     CONTINUE
C
C        THE FOLLOWING FORMAT MUST BE MODIFIED TO USE DIFFERENT FILTER SIZE
C        THEN 2 by 2 PIXELS.
C
                    DO 330 L = 1 , PKQ
                    WRITE(*,331) (R(L,J),J=1,PKQ)
                    WRITE(*,332)
 331              FORMAT(' ',12F6.0)
 332              FORMAT(' ')
 330              CONTINUE
C
C        RESET THE IDENTITY MATRIX.
C
                    DO 36 J = 1 , K
                       DO 37 L = 1 , K
                          IF (J.EQ.L) THEN
                             IMATRX(J,L) = 1.0
                          ELSE
                             IMATRX(J,L) = 0.0
                          END IF
 37                 CONTINUE
 36               CONTINUE
C
C    RESET SI(J,L) TO HAVE [ I ] IN FIRST PARTITION AND [ 0 ] IN ALL OTHERS
C
                    DO 38   J = 1 , PKQ
                       DO 39  L = 1 , K
                          IF (J.EQ.L) THEN
                             SI(J,L) = 1.0
                          ELSE
                             SI(J,L) = 0.0
                          END IF
 39                 CONTINUE
 38               CONTINUE
C
C        SOLVE EQUATION [ R ] * [ B ] = [ SI ] . NOTE THAT THE BELOW
C        CALL TO IMSL ROUTINE LEQT2F, [ B ] RETURNS IN [ SI ]
C
```

```
          IDG = 3
C
          CALL LEQT2F (R,K,PKQ,PKQ,SI,IDG,WKAREA,IER)
C
C        SOLVE EQUATION [ B00 ] * [ KW ] = [ I ] FOR COVARIANCE
C        MATRIX [ KW ] AFTER COLLING IMSL ROUTINE LEQT2F, [ KW ]
C        RETURNS  IN [ IMATRX ].
C
          DO 45 J = 1 , K
             DO 46 JJ = 1 , K
                  B00(J,JJ) = SI(J,JJ)
 46          CONTINUE
 45       CONTINUE
C
          CALL  LEQT2F(B00,K,K,K,IMATRX,IDG,WKAREA,IER)
C
C        SOLVE FOR THE FILTER COEFFICIENTS , [ A ] = [ B ] * [ KW ],
C        WHICH IN THE PROGRAM IS [ A ] = [ SI ] * [ IMATRX ]
C
          DO 47 J = 1 , PKQ
             DO 48 JJ = 1 , K
                  TEMP = 0.0
                  DO 49 JJJ = 1 , K
                     TEMP = TEMP + SI(J,JJJ) * IMATRX(JJJ,JJ)
 49               CONTINUE
                  A(J,JJ) = TEMP
 48          CONTINUE
 47       CONTINUE
C
          WRITE(*,491) ((IMATRX(J,JJ),JJ=1,K),J=1,K)
 491      FORMAT(' ',<K>(F6.2,3X))
          WRITE(*,53) I
 53       FORMAT('     FILTER COEFFICIENTS :', I2,$)
          WRITE(*,491) ((A(J,JJ),JJ=1,K),J=1,PKQ)
C
C        WRITE OUT MEAN, COVARIANCE MATRIX, AND FILTER COEFFICIENTS TO THE
C        USER INPUT FILE.
C
          OPEN (UNIT=2,FILE=FILTER(I),STATUS='NEW',CARRIAGECONTROL='LIST',
     *         FORM='FORMATTED')
C
          WRITE(2,495) (MEAN(I,J),J=1,K)
 495      FORMAT(F10.4)
          WRITE(2,495) ((IMATRX(J,JJ),JJ=1,K),J=1,K)
          WRITE(2,495) ((A(J,JJ),JJ=1,K),J=1,PKQ)
C
          CLOSE(UNIT=2)
C
   19     CONTINUE
          WRITE(*,55)
 55       FORMAT(' THE PROGRAM FLTR1 IS OVER',$)
          STOP
          END
```

```
C ******************************************************************
C *                                                                *
C *       PROGRAM SGMT1                                            *
C *                                                                *
C *       PURPOSE To segment a color image with two textures given *
C *               the image, the image dimensions, the filter dimen- *
C *               sions, and the two sets of filter parameters to  *
C *               be used.                                         *
C *                                                                *
C *       REQUIRED IMSL ROUTINES                                   *
C *               LINV3F, LUDATN, LUELMN                           *
C *                                                                *
C *       IMPLEMENTED BY LTJG TIMUR KUPELI  Nov 1986               *
C *                                                                *
C ******************************************************************
C
C
C       ****    VARIABLE DEFINITIONS    ****
C
C       BINPUT   = [ Input ] Image data in the byte format.
C       ML       = [ Output ] The result of ML segmentation in byte format.
C       MAP      = [ Output ] The result of MAP segmentation in byte format.
C       FNAME    = [ Input ] Filename of filter parameters set.
C       IMAGE    = Filename of the image channel.
C       P , Q    = Rows , Columns of filter.
C       N , M    = Rows , Columns of image.
C       K        = Number of channel of image.
C       KW       = [ Input ] The covariance matrix.
C       MEAN     = [ Input ] Mean vector of estimation windows.
C       ERROR    = Prediction error estimation
C       A        = [ Input ] Filter coefficients matrix.
C       TEXTURE  = Zero-mean image data in real*4 format.
C       C01      = Number of removed false points in the first texture.
C       C10      = Number of removed false points in the other texture.
C       IN , IM  = Maximum number of rows and columns in the image.
C       P1 , Q1  = Maximum number of rows and columns in the filter.
C       TMAX     = Maximum number of filters for processing.
C       IK       = Maximum number of channels in image.
C
C
        INTEGER IN,N,IM,P1,P,Q1,Q,TMAX,T,PQ,ROW,COL,I,J,JJ,JJJ,L,LL,KK,
     *          LLL,LLLL,COUNT,LI,HI,LJ,HJ,K,PP,QQ,IK,II,III,C01,C10,M
C
        REAL  KW(1:3,1:3,1:2),MEAN(1:3,1:2),TEMP,SUM1,SUM2,PML11,LN(2),
     *        ERROR(1:128,1:128,1:3,1:2),PML1,PML2,PML(1:128,1:128),
     *        AREA,KS,A(1:3,1:3,1:2,1:2,1:2),D1,D2,KW1(1:3,1:3),KW2(3,3),
     *        EKW1(1,1:3),EKW2(1,1:3),PML22,DD2,TEXTUR(1:128,1:128,3,3)
     *        WKAREA(6),AA(1:128,1:128)
C
        CHARACTER*50    IMAGE(1:3),FNAME(1:2),MLTEST,MAPTEST
C
        BYTE    BINPUT(1:128,1:128,1:3),ML(1:128,1:128),MAP(1:128,1:128)
     *          MLI(1:128,1:128)
C
        IN = 128
        IM = 128
        P1 = 4
        Q1 = 4
        TMAX = 2
        IK = 3
C
C       GET THE INPUT PARAMETERS OF THE PROGRAM
C
 1      WRITE(*,2) IN
 2      FORMAT(' ENTER THE NUMBER OF ROWS IN IMAGE.LIMIT OF',I3,':',$)
        READ(*,3) N
 3      FORMAT(I3)
        IF((N.LT.1) .OR. (N.GT.IN)) GOTO 1
C
```

```
4       WRITE(*,5) IM
5       FORMAT(' ENTER THE NUMBER OF COLUMNS IN IMAGE.LIMIT OF',I3,':',$)
         READ(*,3) M
         IF((M.LT.1) .OR. (M.GT.IM)) GOTO 4
C
6       WRITE(*,7) P1
7       FORMAT(' ENTER THE NUMBER OF ROWS IN FILTER.LIMIT OF',I3,':',$)
         READ(*,3) P
         IF((P.LT.2) .OR. (P.GT.P1)) GOTO 6
C
8       WRITE(*,9) Q1
9       FORMAT(' ENTER THE NUMBER OF COLUMNS IN FILTER.LIMIT OF',I3,':',$)
         READ(*,3) Q
         IF((Q.LT.2) .OR. (Q.GT.Q1)) GOTO 8
C
10      WRITE(*,11) TMAX
11      FORMAT(' ENTER NUMBER OF TEXTURES TO PROCESS.LIMIT OF',I3,':',$)
         READ(*,3) T
         IF((T.LT.2) .OR. (T.GT.TMAX)) GO TO 10
C
12      WRITE(*,13) IK
13      FORMAT(' ENTER THE NUMBER OF IMAGE CHANNELS.LIMIT OF',' ',I3,$)
         READ(*,3) K
         IF((K.LT.1) .OR. (K.GT.IK)) GO TO 12
C
C       GET ALL CHANNELS OF THE IMAGE
C
         DO 19 I = 1 , K
         WRITE(*,20) I
20      FORMAT(' ENTER FILENAME OF THE IMAGE CHANNEL NUMBER',' ',I2,$)
         READ(*,21) IMAGE(I)
21      FORMAT(A50)
C
         OPEN(UNIT=1,FILE=IMAGE(I),STATUS='OLD',ACCESS='DIRECT')
C
         DO 23 ROW = 1 , N
                READ(1'ROW) (BINPUT(ROW,COL,I),COL = 1 , M )
23      CONTINUE
C
         CLOSE( UNIT = 1 )
C
19      CONTINUE
C
C       GET THE FILTER COEFFICIENT,MEAN, AND COVARIANCE MATRICES
C
         DO 25 I=1,T
         WRITE(*,26) I
26      FORMAT('ENTER FILENAME OF FILTER PARAMETERS SET NUMBER',' ',I2,$)
         READ(*,21) FNAME(I)
C
         OPEN(UNIT=2,FILE=FNAME(I),STATUS='OLD',FORM='FORMATTED')
C
         DO 260 J = 1 , K
         READ(2,27) MEAN(J,I)
27      FORMAT(F10.4)
260     CONTINUE
C
         DO 28   J=1,K
                READ(2,27) (KW(J,JJ,I),JJ=1,K)
28      CONTINUE
C
         DO 29 PP=1,P
           DO 30 QQ=1,Q
             DO 31 ROW=1,K
          READ(2,33) (A(ROW,COL,PP,QQ,I),COL=1,K)
33      FORMAT(F10.4)
31      CONTINUE
30      CONTINUE
29      CONTINUE
```

```
          CLOSE(UNIT=2)
 32      FORMAT(3(F10.4,3X))
C
 25      CONTINUE
C
C        GET THE OUTPUT FILENAMES OF THE ML AND MAP SEGMENTATION RESULTS.
C
          READ(*,220) MLTEST
          READ(*,220) MAPTEST
 220     FORMAT(A80)
C
C        CONVERT A 2-D BYTE INPUT IMAGE DATA ARRAY IN THE RANGE OF -128 TO
C        127 THAT REPRESENTS APPROPRIATE INTENSITY LEVELS IN THE RANGE OF
C        0 TO 255 .
C
          DO 35 J = 1 , K
          DO 36 ROW = 1 , N
                DO 37 COL = 1 , M
                      TEMP = BINPUT(ROW,COL,J)
                      IF(TEMP.LT.0.0) THEN
                            TEMP = TEMP+256
                      END IF
                      TEXTUR(ROW,COL,J,1) = TEMP - MEAN(J,1)
                      TEXTUR(ROW,COL,J,2) = TEMP - MEAN(J,2)
 37             CONTINUE
 36      CONTINUE
 35      CONTINUE
C
C        CALCULATION OF ERROR ESTIMATE FOR TWO TEXTURES
C
          DO 40 I = 1 , T
            DO 41  L = 1 , N
              DO 42 LL = 1 , M
               DO 421 KK = 1 , K
                ERROR(L,LL,KK,I) = 0.0
                DO 43 III = 1 , P
                  J = 1 - III
                DO 44 LLL = 1 , Q
                  JJJ = LL - LLL
                DO 45 II = 1 , K
                DO 46 JJ = 1 , K
                IF(J.LE.0) J=1
                IF(JJJ.LE.0) JJJ=1
C
          ERROR(L,LL,KK,I)=ERROR(L,LL,KK,I)+A(II,JJ,III,LLL,I)*TEXTUR(J,JJJ,KK,I)
C
 46             CONTINUE
 45           CONTINUE
 44         CONTINUE
 43        CONTINUE
 421       CONTINUE
 42       CONTINUE
 41      CONTINUE
 40      CONTINUE
C
          DO 47 JJ=1,K
            DO 48  LL=1,K
                KW1(JJ,LL) = KW(JJ,LL,1)
                KW2(JJ,LL) = KW(JJ,LL,2)
 48       CONTINUE
 47      CONTINUE
C
          D1 = 1.0
          CALL  LINV3F(KW1,6,1,K,K,D1,D2,WKAREA,IER)
          DET1 = D1 * 2**D2
          LN(1) = ALOG(DET1)
C
          D1 = 1.0
          CALL  LINV3F(KW2,6,1,K,K,D1,DD2,WKAREA,IER)
```

```
          DET2 = D1 * 2**DD2
          LN(2) = ALOG(DET2)
C
C       CALCULATION OF MAXIMUM-LIKELIHOOD IMAGE SEGMENTATION
C
          OPEN(UNIT=3,FILE=MLTEST,STATUS='NEW',ACCESS='DIRECT',
     *          RECL=(IM/4),MAXREC=IN)
C
          DO 50 ROW =1,N
           DO 51 COL = 1,M
            DO 52 I = 1,K
               EKW1(1,I) = 0.0
               EKW2(1,I) = 0.0
                  DO 53 J=1,K
                   EKW1(1,I)=EKW1(1,I)+ERROR(ROW,COL,J,1)*KW1(J,I)
                   EKW2(1,I)=EKW2(1,I)+ERROR(ROW,COL,J,2)*KW2(J,I)
 53              CONTINUE
 52        CONTINUE
            PML11 = 0.0
            PML22 = 0.0
            DO 54 L=1,K
              PML11=PML11+EKW1(1,L)*ERROR(ROW,COL,L,1)
              PML22=PML22+EKW2(1,L)*ERROR(ROW,COL,L,2)
 54        CONTINUE
            PML1 = 0.0
            PML2 = 0.0
            PML1 = PML11 + LN(1)
            PML2 = PML22 + LN(2)
C
                  ML(ROW,COL)=0
            PML(ROW,COL) = PML2 - PML1
                  IF(PML1 .GT. PML2) THEN
                    ML(ROW,COL)= -1
                  END IF
                  MLI(ROW,COL) = ML(ROW,COL)
 51         CONTINUE
C
             WRITE(3'ROW) (ML(ROW,COL),COL=1,M)
 50       CONTINUE
C
          CLOSE(UNIT=3)
C
C       MAXIMUM A POSTERIORI IMAGE SEGMENTATION
C
          OPEN(UNIT=4,FILE=MAPTEST,STATUS='NEW',ACCESS='DIRECT',
     *          RECL= (IM/4), MAXREC = IN)
C
          KS = 10.0
          C01 = 0
          C10 = 0
          DO 600  II = 1 , 5
          DO 60 I=1,N
             DO 61 J = 1 , M
                SUM1 = 0.0
                SUM2 = 0.0
                LI = I - 3
                HI = I + 3
                LJ = J - 3
                HJ = J + 3
C
                IF(LI.EQ.0) LI = 1
                IF(HI.GT.N) HI = N
                IF(LJ.EQ.0) LJ = 1
                IF(HJ.GT.M) HJ = M
C
                AREA = (HI - LI + 1) * (HJ - LJ + 1)
C
                DO 62 ROW = LI , HI
                  DO 63 COL = LJ , HJ
```

```
63                      CONTINUE        SUM1 = SUM1 - MLI(ROW,COL)
62                  CONTINUE
                          SUM1 = SUM1 + MLI(I,J)
                          MAP(I,J)= 0
C
                     SUM2 = PML(I,J) - ((KS/AREA)*(2*SUM1-area+1))
                     IF(SUM2.LT.0.0) THEN
                        MAP(I,J) = -1
                     END IF
                     MLI(I,J) = MAP(I,J)
C
 61         CONTINUE
 60     CONTINUE
 600    CONTINUE
C
        DO 70 I = 1 , N
          DO 71 J = 1 , M
                 IF(ML(I,J) .EQ. 0) THEN
                   IF(MAP(I,J) .NE. ML(I,J)) C01 = C01 + 1
                 ELSE
                   IF(MAP(I,J) .NE. ML(I,J)) C10=C10+1
                 END IF
71         CONTINUE
                   WRITE(4'I) (MAP(I,J),J=1,M)
70      CONTINUE
         CLOSE(UNIT=4)
                   WRITE(*,64) C01,C10
64      FORMAT(' C01:',I5,5x,'C10:',I5)
         STOP
         END
```

APPENDIX E

PROGRAMS FOR K-L TRANSFORMATION, AND ONE-CHANNEL SEGMENTATION

```
C
C     ****************************************************************
C     *                                                              *
C     *  PROGRAM KRLV                                                *
C     *                                                              *
C     *  PURPOSE To implement Karhunen-Loève transformation from a 2-D *
C     *          color image which size is 128 by 128 pixels.        *
C     *                                                              *
C     *  REQUIRED IMSL ROUTINES                                      *
C     *                                                              *
C     *          EIGRS, EQRT2S, EHOBKS, EHOUSS, UERTST, USPKD, UGETIO *
C     *                                                              *
C     *  IMPLEMENTED BY LTJG TIMUR KUPELI  Dec 1986                  *
C     *                                                              *
C     ****************************************************************
C
C
      INTEGER  I,J,L,K,N1,N2,NN,ROW,COL,ITEMP,QC(128,128),
     *         MAXVAL,MINVAL,IDIF,INTVAL,N
C
      REAL R(1:3,1:3),E(1:3,1:3),FINPUT(1:128,1:128,1:3),
     *     D(3),WK(3),TEMP,Q(1:128,1:128,1:3),SLOPE
C
       CHARACTER*50    IMAGE(1:3) , FNAME(1:3)
C
       BYTE    BINPUT(1:128,1:128),QQ(1:128,1:128,1:3)
C
       TYPE 100
 100  FORMAT( 'ENTER THE NUMBER OF ROWS, COLUMNS IN THE IMAGE',$)
       READ 101,N
 101  FORMAT(I3)
C
       TYPE 102
 102  FORMAT(' ENTER THE NUMBER OF CHANNELS IN THE IMAGE',$)
       READ 101,K
C
C     GET THE FILENAMES OF THE RED,GREEN,AND BLUE COMPONENTS
C     OF THE COLOR IMAGE.
C
       DO 1 I = 1 , K
       WRITE(*,2) I
  2   FORMAT(' ENTER THE FILENAME OF THE IMAGE CHANNEL NUMBER ',' ',I2,$)
       READ(*,3) IMAGE(I)
       WRITE(*,3) IMAGE(I)
       WRITE(*,45)
  3   FORMAT(A50)
C
C     CONVERT THE IMAGE BTYE FORMAT TO THE REAL NUMBER FORMAT
C
       OPEN(UNIT=1, FILE=IMAGE(I), STATUS='OLD', ACCESS='DIRECT')
C
       DO 5 ROW = 1 , N
              READ(1'ROW) (BINPUT(ROW,COL),COL=1,N)
              DO 6 COL = 1 , N
                  TEMP = BINPUT(ROW,COL)
                  IF (TEMP.LT.0.0) THEN
                        TEMP = TEMP + 256
                  END IF
                  FINPUT(ROW,COL,I) = TEMP
  6           CONTINUE
  5   CONTINUE
```

```
C
        CLOSE(UNIT = 1)
C
  1    CONTINUE
C
C      CALCULATE THE CORRELATION MATRIX
C
        NN = N * N
        DO 20 I = 1 , K
            DO 21 J = 1 , K
                   R(I,J) = 0
                   DO 22 N1 = 1 , N
                       DO 23 N2 = 1 , N
                          R(I,J) = R(I,J) + FINPUT(N1,N2,I)*FINPUT(N1,N2,J)
 23                CONTINUE
 22             CONTINUE
                   R(I,J) = R(I,J) / NN
 21       CONTINUE
 20    CONTINUE
C
        WRITE(*,45)
        WRITE(*,30)
        WRITE(*,39)
 39    FORMAT(' ----------------------------------')
 30    FORMAT('   ',5X,' THE CORRELATION MATRIX',$)
        WRITE(*,31) ((R(I,J),J=1,K),I=1,K)
 31    FORMAT( <K>(F9.2,4X))
        WRITE(*,39)
C
C      CALCULATE EIGENVALUES AND EIGENVECTORS OF THE
C                CORRELATION MATRIX
C
        JOBN = 11
C
        CALL EIGRS(R,K,JOBN,D,E,K,WK,IER)
C
C      SORT THE EIGENVALUES IN DECREASING ORDER
C
        TEMP = D(1)
        D(1) = D(3)
        D(3) = TEMP
C
        DO 49 I = 1 , K
                   TEMP   = E(I,1)
                   E(I,1) = E(I,3)
                   E(I,3) = TEMP
 49    CONTINUE
C
        WRITE(*,39)
        WRITE(*,45)
 45    FORMAT(' ')
C
        WRITE(*,41)
        WRITE(*,39)
 41    FORMAT(' ',5X,' THE EIGENVALUES ',$)
        DO 46 J=1,K
        WRITE(*,42)  D(J)
 42    FORMAT(5X,F9.2)
 46    CONTINUE
        WRITE(*,39)
        WRITE(*,45)
        WRITE(*,43)
        WRITE(*,39)
 43    FORMAT(' ',5X,' THE EIGENVECTORS', $)
        WRITE(*,44) ((E(I,J),J=1,K),I=1,K)
 44    FORMAT(3(F9.2,4X))
        WRITE(*,39)
C
C      USE THE TRANSPOSE OF THE EIGENVECTORS MATRIX TO IMPLEMENT
```

```
C       KARHUNEN-LOEVE TRANSFORMATION.
C
          DO 50 I = 1 , K
               DO 51 N1 = 1 , N
                    DO 52 N2 = 1 , N
                         Q(N1,N2,I) = 0.0
                         L = 4 - I
                         DO 53 J = 1 , K
                              Q(N1,N2,I) = Q(N1,N2,I) + E(J,I)*FINPUT(N1,N2,J)
 53                 CONTINUE
 52              CONTINUE
 51          CONTINUE
 50      CONTINUE
C
          DO 60 I = 1 , K
          WRITE(*,61) I
 61      FORMAT(' ENTER FILENAME OF THE TRANSFORMED IMAGE ',' ',I2,$)
          READ(*,3) FNAME(I)
          WRITE(*,3) FNAME(I)
          WRITE(*,45)
C
C       CONVERT THE TRANSFORMED IMAGE TO BYTE FORMAT
C
          DO 62  N1 = 1 , N
          DO 63  N2 = 1 , N
                    QC(N1,N2) = JNINT(Q(N1,N2,I))
 63      CONTINUE
 62      CONTINUE
C
C       SCALE THE TRANSFORMED IMAGE TO BE WITHIN DISPLAY RANGE
C
          IF (I .EQ. 1) THEN
             MAXVAL = QC(1,1)
             MINVAL = QC(1,1)
             DO ROW = 1 , N
                DO COL = 1 , N
                    IF (QC(ROW,COL) .GT. MAXVAL) THEN
                         MAXVAL = QC(ROW,COL)
                    ELSE IF (QC(ROW,COL) .LT. MINVAL) THEN
                         MINVAL = QC(ROW,COL)
                    END IF
                ENDDO
             ENDDO
             INTVAL = MAXVAL - MINVAL
             SLOPE = 255 / REAL(INTVAL)
             DO ROW = 1 , N
                DO COL = 1 , N
                    IF (QC(ROW,COL) .EQ. MINVAL ) THEN
                                QC(ROW,COL) = 0
                    ELSE IF (QC(ROW,COL) .EQ. MAXVAL) THEN
                                QC(ROW,COL) = 255
                    ELSE
                                IDIF = QC(ROW,COL) - MINVAL
                                QC(ROW,COL) = INT(SLOPE*IDIF)
                    END IF
                ENDDO
             ENDDO
          ELSE
               MINVAL = QC(1,1)
               DO ROW = 1 , N
                  DO COL = 1 , N
                    IF (QC(ROW,COL) .LT. MINVAL) THEN
                         MINVAL = QC(ROW,COL)
                    END IF
                  ENDDO
               ENDDO
               DO ROW = 1 , N
                  DO COL = 1 , N
                                IDIF = QC(ROW,COL) - MINVAL
```

```
                         QC(ROW,COL) = INT(SLOPE*IDIF)
              ENDDO
           ENDDO
        END IF
C
        DO 64 N1 = 1 , N
        DO 65 N2 = 1 , N
              ITEMP = QC(N1,N2)
              IF (ITEMP .GT.127) THEN
                    ITEMP = ITEMP - 256
              END IF
              QQ(N1,N2,I) = ITEMP
 65     CONTINUE
 64     CONTINUE
C
        OPEN(UNIT=2,FILE=FNAME(I),STATUS='NEW',ACCESS='DIRECT',
     *       RECL= (N/4), MAXREC=N )
C
C
        DO 66  N1 = 1 , N
        WRITE(2'N1) (QQ(N1,N2,I),N2=1,N)
 66     CONTINUE
C
        CLOSE(UNIT= 2)
C
 60     CONTINUE
C
        STOP
        END
```

```
C     ************************************************************************
C     *                                                                      *
C     *    PROGRAM FLTR2                                                     *
C     *                                                                      *
C     *    PURPOSE To develop two sets of filter parameters from a 2-D single*
C     *            channel image which size is 128 by 128 pixels. These para-*
C     *            meters are the mean, the covariance, and the filter       *
C     *            coefficients.                                             *
C     *                                                                      *
C     *    REQUIRED IMSL ROUTINES                                            *
C     *                                                                      *
C     *            LEQT2F, LUDATN, LUELMN, LUREFN                            *
C     *                                                                      *
C     *    IMPLEMENTED BY LTJG TIMUR KUPELI  Sep 1986                        *
C     *                                                                      *
C     ************************************************************************
C
C
      INTEGER*4        P,Q,ROW,COL,STARTN(2),STARTM(2),ENDN(2),ENDM(2),IDGT,
     *                 PQ,RNROW,RNCOL,RROW,CCOL,X,Y,
     *                 HNN0,HMM0,HCOL,HROW,LNN0,LNM0,LCOL,LROW,RN,RM,IER
C
      REAL*4   SUM,TEMP,FINPUT(128,128),MEAN(2),WKAREA(28),
     *         R(4,4),SI(4,1),IMATRX(1,1),B00,
     *         KW(1,1),A(4,1),ISIZE
C
      CHARACTER*50     IMAGE,FILTER(2)
C
      BYTE     BINPUT(128)
C
      T = 2
C
C             GET PROGRAM INPUT PARAMETERS.
C
      TYPE 10
10    FORMAT(' ENTER NUMBER OF FILTER ROWS FROM 2-4 DESIRED :',$)
      READ 11,P
11    FORMAT(I3)
C
      TYPE 12
12    FORMAT(' ENTER NUMBER OF FILTER COLUMNS FROM 2-4 DESIRED :',$)
      READ 11,Q
C
      TYPE 13
13    FORMAT(' ENTER NUMBER OF ROWS IN IMAGE ',$)
      READ 11,N
C
      TYPE 14
14    FORMAT(' ENTER NUMBER OF COLUMNS IN IMAGE ',$)
      READ 11,M
C
C             GET FILENAME OF SINGLE CHANNEL IMAGE
C
      TYPE 15
15    FORMAT(' ENTER FILENAME OF IMAGE ',$)
      READ 16,IMAGE
16    FORMAT(A50)
C
C     CONVERT THE IMAGE BTYE FORMAT TO REAL NUMBER FORMAT
C
      OPEN(UNIT=1,FILE=IMAGE,STATUS='OLD',ACCESS = 'DIRECT')
      DO 17   ROW = 1 , N
              READ (1'ROW) (BINPUT(COL),COL=1,M)
              DO 18   COL = 1 , M
                      TEMP = BINPUT(COL)
                      IF (TEMP.LT.0.0) THEN
                         TEMP = TEMP + 256
                      END IF
                      FINPUT(ROW,COL) = TEMP
```

```
18              CONTINUE
17     CONTINUE
        CLOSE (UNIT=1)
C
C
C      GET 'T' AREAS FOR WHICH FILTERS ARE DESIRED AND OUTPUT FILENAME FOR
C      EACH AREA'S FILTER COEFFICIENTS AND COVARIANCE MATRIX.
C
        DO 19   I = 1,T
           WRITE(*,20) I
20       FORMAT(' ENTER UPPER-LEFT ROW FOR AREA ',I2,':',$)
           READ(*,11) STARTN(I)
C
           WRITE(*,21) I
21       FORMAT(' ENTER UPPER-LEFT COLUMN FOR AREA',I2,':',$)
           READ(*,11) STARTM(I)
C
           WRITE(*,22) I
22       FORMAT(' ENTER LOWER-RIGHT ROW FOR AREA',I2,':',$)
           READ(*,11) ENDN(I)
C
           WRITE(*,23) I
23       FORMAT(' ENTER LOWER-RIGTH COLUMN FOR AREA',I2,':',$)
           READ(*,11) ENDM(I)
C
           WRITE(*,24) I
24       FORMAT(' ENTER OUTPUT FILE-SPEC FOR FILTER',I2,':',$)
           READ(*,16) FILTER(I)
C
C      FIND THE MEAN VECTOR OF ESTIMATION WINDOW AREA OF IMAGE
C
        ISIZE = (ENDN(I) - STARTN(I) + 1)*(ENDM(I) - STARTM(I) + 1)
        SUM = 0.0
        DO 25   L = STARTN(I),ENDN(I)
              DO 26   J = STARTM(I),ENDM(I)
                 SUM = SUM + FINPUT(L,J)
26          CONTINUE
25     CONTINUE
        MEAN (I) = SUM / ISIZE
        WRITE(*,27) I,MEAN(I)
27     FORMAT('  MEAN','(',I2,')',':',F9.2,$)
C
C               CORRECT THE IMAGE TO BE ZERO MEAN
C
        DO 28  L = STARTN(I), ENDN(I)
              DO 29  J = STARTM(I),ENDM(I)
                 FINPUT(L,J) = FINPUT(L,J) - MEAN(I)
29          CONTINUE
28     CONTINUE
C
C      DETERMINE CORRELATION MATRIX
C
        WRITE(*,291) I
291    FORMAT('        CORRELATION.MATRIX',I2,$)
        PQ = P * Q
        DO 30  RNROW = 1,P
           X = Q * (RNROW-1)
           DO 31  RNCOL = 1,P
                 NO = RNROW - RNCOL
                 Y = Q * (RNCOL-1)
                 DO 32  RMROW = 1,Q
                    RN = X + RMROW
                    DO 33  RMCOL = 1,Q
                          MO = RMROW - RMCOL
                          RM = Y + RMCOL
                          LNNO = STARTN(I) + NO
                          LMMO = STARTM(I) + MO
                          HNNO = ENDN(I) + NO
                          HMMO = ENDM(I) + MO
C
```

```
                          LCOL =MAX0(STARTM(I),LMM0)
                          HCOL =MIN0(ENDM(I),HMM0)
                          LROW =MAX0(STARTN(I),LNN0)
                          HROW =MIN0(ENDN(I),HNN0)
C
                          SUM = 0.0
                          DO 34 ROW = LROW,HROW
                                NN0 = ROW - N0
                            DO 35  COL = LCOL,HCOL
                                MM0 = COL - M0
                                SUM = SUM + FINPUT(ROW,COL)*FINPUT(NN0,MM0)
   35                  CONTINUE
   34                CONTINUE
                          R(RN,RM) = SUM / ISIZE
C
   33              CONTINUE
   32           CONTINUE
   31        CONTINUE
   30     CONTINUE
C
C        RESET SI(J,1) TO HAVE 1 IN THE FIRST ROW AND 0 IN ALL OTHERS
C
          DO 41  J = 1,PQ
            IF (J.EQ.1) THEN
                SI(J,1) = 1.0
                IMATRX(J,1) = 1.0
            ELSE
                SI(J,1) = 0.0
            END IF
   41     CONTINUE
C
C        THE FORMATS BELOW MUST BE MODIFIED TO USE DIFFERENT FILTER
C        SIZE THAN 2 by 2 PIXELS.
C
          IDGT = 1
          DO 36 K=1,PQ
          WRITE(*,37) (R(K,J),J=1,PQ)
          WRITE(*,38)
 37      FORMAT(' ',4F9.2,$)
 38      FORMAT('   ',$)
 36      CONTINUE
C
C        SOLVE EQUATION [ R ] * [ B ] = [ SI ]
C
          CALL LEQT2F (R,IDGT,PQ,PQ,SI,3,WKAREA,IER)
C
          WRITE(*,360) (SI(J,1),J=1,PQ)
 360     FORMAT(F10.2)
C
C        SOLVE EQUATION   B00 * KW = I
C
          B00 = SI(1,1)
C
C        GET THE FILTER COEFFICIENTS USING THE EQUATION
C                    [ A ]= [ B ] * KW
C
          KW(1,1) = IMATRX(1,1) / B00
          WRITE(*,*) 'COVARIANCE'
          WRITE(*,360) KW(1,1)
C
          DO 43  ROW = 1,PQ
             A(ROW,1) = SI(ROW,1) * KW(1,1)
   43     CONTINUE
C
C        WRITE OUT MEAN, COVARIANCE, AND FILTER COEFFICIENTS TO THE
C        USER INPUT FILE.
C
          OPEN (UNIT=2,FILE=FILTER(I),STATUS='NEW',CARRIAGECONTROL='LIST',
     *         FORM='FORMATTED')
```

```
C
      WRITE(2,50) KW(1,1)
   50 FORMAT( F10.5)
      WRITE(2,500) MEAN(I)
  500 FORMAT( F10.5)
      DO 51 J = 1,PQ
            WRITE(2,52) A(J,1)
   51 CONTINUE
   52 FORMAT (F10.5)
C
      CLOSE(UNIT=2)
C
      WRITE(*,53) I
   53 FORMAT('  FILTER COEFFICIENTS :',I2,$)
      WRITE(*,54) (A(J,1),J=1,PQ)
   54 FORMAT(F10.5)
   19 CONTINUE
      STOP
      END
```

```
C     ***********************************************************************
C     *                                                                     *
C     *  PROGRAM SGMT2                                                      *
C     *                                                                     *
C     *  PURPOSE To segment a single image with two textures given the      *
C     *          image, the image dimensions, the filter dimensions,        *
C     *          and two sets of filter parameters to be used.              *
C     *                                                                     *
C     *  REQUIRED IMSL ROUTINES                                             *
C     *                                                                     *
C     *          NONE                                                       *
C     *                                                                     *
C     *  IMPLEMENTED BY LTJG TIMUR KUPELI   Sep 1986                        *
C     *                                                                     *
C     ***********************************************************************
C
C
      INTEGER  IN,N,IM,P1,P,Q1,Q,TMAX,T,PQ,ROW,COL,I,J,JJ,JJJ,L,LL,
     *         LLL,LLLL,COUNT,LI,HI,LJ,HJ
C
       REAL     KW(2),MEAN(2),AA(4,1,2),TEMP,SUM1,SUM2,TEXTUR(128,128,2),
     *          LN(2),ERROR(128,128,2),PML1,PML2,PML(128,128),
     *          AREA,KS,A(2,2)
C
       CHARACTER*50    IMAGE,FNAME,MLTEST,MAPTEST
C
       BYTE     BINPUT(128,128),ML(128,128),MAP(128,128),MLI(128,128)
C
       IN = 128
       IM = 128
       P1 = 4
       Q1 = 4
       TMAX = 2
C
C      GET THE INPUT PARAMETERS OF THE PROGRAM
C
 1     WRITE(*,2) IN
 2     FORMAT(' ENTER THE NUMBER OF ROWS IN IMAGE.LIMIT OF',I3,':',$)
        READ(*,3) N
 3     FORMAT(I3)
        IF((N.LT.1) .OR. (N.GT.IN)) GOTO 1
C
 4     WRITE(*,5) IM
 5     FORMAT(' ENTER THE NUMBER OF COLUMNS IN IMAGE.LIMIT OF',I3,':',$)
        READ(*,3) M
        IF((M.LT.1) .OR. (M.GT.IM)) GOTO 4
C
 6     WRITE(*,7) P1
 7     FORMAT(' ENTER THE NUMBER OF ROWS IN FILTER.LIMIT OF',I3,':',$)
        READ(*,3) P
        IF((P.LT.2) .OR. (P.GT.P1)) GOTO 6
C
 8     WRITE(*,9) Q1
 9     FORMAT(' ENTER THE NUMBER OF COLUMNS IN FILTER.LIMIT OF',I3,':',$)
        READ(*,3) Q
        IF((Q.LT.2) .OR. (Q.GT.Q1)) GOTO 8
C
 10    WRITE(*,11) TMAX
 11    FORMAT(' ENTER NUMBER OF TEXTURES TO PROCESS.LIMIT OF',I3,':',$)
        READ(*,3) T
        IF((T.LT.2) .OR. (T.GT.TMAX)) GO TO 10
C
C      GET THE SINGLE-CHANNEL IMAGE
C
        PQ = P * Q
        WRITE(*,20)
 20    FORMAT(' ENTER FILENAME OF IMAGE ',$)
        READ(*,21) IMAGE
 21    FORMAT(A50)
```

```
C
          OPEN(UNIT=1,FILE=IMAGE,STATUS='OLD',ACCESS='DIRECT')
C
          DO 22 ROW = 1 , N
                READ(1'ROW) (BINPUT(ROW,COL),COL = 1 , M )
 22      CONTINUE
C
          CLOSE( UNIT = 1 )
C
C        GET THE FILTER COEFFICIENTS,MEANS, AND COVARIANCES
C
          DO 23  I = 1 , T
                WRITE(*,24) I
 24      FORMAT(' ENTER THE FILENAME OF FILTER PARAMETERS SET NUMBER',I2,':',$)
          READ(*,21) FNAME
C
          OPEN(UNIT=2,FILE=FNAME,STATUS='OLD')
C
          READ(2,240)  KW(I)
 240     FORMAT(F10.5)
          READ(2,240) MEAN(I)
          DO 242 KK= 1,PQ
          READ(2,240) AA(KK,1,I)
C
 242     CONTINUE
C
          CLOSE(UNIT = 2)
C
          WRITE(*,241) KW(I),MEAN(I),AA(1,1,I),AA(2,1,I),AA(3,1,I),AA(4,1,I)
 241     FORMAT(6F10.5)
 23      CONTINUE
          READ(*,220) MLTEST
          READ(*,220) MAPTEST
 220     FORMAT(A80)
C
C        CONVERT A 2-D BYTE INPUT IMAGE DATA IN THE RANGE OF -128 TO 127 THAT
C        REPRESENTS APPROPRIATE INTENSITY LEVELS IN THE RANGE OF 0 TO 255
C
          DO 30 ROW = 1 , N
                DO 31 COL = 1 , M
                      TEMP = BINPUT(ROW,COL)
                      IF(TEMP.LT.0.0) TEMP = TEMP+256
                      TEXTUR(ROW,COL,1) = TEMP - MEAN(1)
                      TEXTUR(ROW,COL,2) = TEMP - MEAN(2)
 31           CONTINUE
 30      CONTINUE
C
C        CALCULATOIN OF ERROR ESTIMATE FOR TWO TEXTURES
C
          DO 40 I = 1 , T
             J = 1
             DO 41 JJ = 1 , P
                DO 42 JJJ = 1 , Q
                   A(JJ,JJJ) = AA(J,1,I)
                   J = J + 1
 42           CONTINUE
 41        CONTINUE
C
             DO 43  L = 1 , N
               DO 44 LL = 1 , M
                  ERROR(L,LL,I) = 0.0
                  DO 45  LLL = 1 , P
                     J = L - LLL
                     DO 46 LLLL = 1 , Q
                        JJ = LL - LLLL
                        IF((J.GT.0) .AND. (JJ.GT.0)) THEN
             ERROR(L,LL,I) =ERROR(L,LL,I)+ A(LLL,LLLL)*TEXTUR(J,JJ,I)
                        END IF
 46               CONTINUE
```

```
45              CONTINUE
44           CONTINUE
43         CONTINUE
C
            LN(I) = ALOG(KW(I))
40      CONTINUE
C
C       CALCULATION OF MAXIMUM LIKELIHOOD IMAGE SEGMENTATION
C
          OPEN(UNIT=3,FILE=MLTEST,STATUS='NEW',ACCESS='DIRECT',
     *           RECL=(IM/4),MAXREC=IN)
C
          L = 1
          LL = 2
          DO 50  I = 1 , N
              DO 51 J = 1 , M
                    PML1 = 0.0
                    PML2 = 0.0
                    PML1 = ((ERROR(I,J,L)**2)/KW(L)) + LN(L)
                    PML2 = ((ERROR(I,J,LL)**2)/KW(LL)) + LN(LL)
                    PML(I,J) = PML2 - PML1
                    IF(PML1 .GT. PML2) THEN
                       ML(I,J)= -1
                    END IF
                    MLI(I,J) = ML(I,J)
51         CONTINUE
C
            WRITE(3'I) (ML(I,J),J=1,M)
50      CONTINUE
C
C
          CLOSE(UNIT=3)
C
C       MAXIMUM A POSTERIORI IMAGE SEGMENTATION
C
          OPEN(UNIT=4,FILE=MAPTEST,STATUS='NEW',ACCESS='DIRECT',
     *           RECL= (IM/4), MAXREC = IN)
C
          KS = 100
          DO 600 II = 1 , 5
          DO 60 I = 1 ,N
              DO 61 J = 1 , M
                  SUM1 = 0.0
                  SUM2 = 0.0
                  LI = I - 3
                  HI = I + 3
                  LJ = J - 3
                  HJ = J + 3
C
                  IF(LI.EQ.0) LI = 1
                  IF(HI.GT.N) HI = N
                  IF(LJ.EQ.0) LJ = 1
                  IF(HJ.GT.M) HJ = M
C
                  AREA = (HI - LI + 1) * (HJ - LJ +1)
C
                  DO 62 ROW = LI , HI
                     DO 63 COL = LJ , HJ
                        IF((ROW.NE.I) .OR. (COL.NE.J)) THEN
                                 SUM1 = SUM1 - MLI(ROW,COL)
                        END IF
63                 CONTINUE
62              CONTINUE
                     MAP(I,J)=0
C
                  SUM2 = PML(I,J) - ((KS/AREA)*(2*SUM1-AREA+1))
                  IF(SUM2.LT.0) THEN
                     MAP(I,J) = -1
                  END IF
```

```
                MLI(I,J) = MAP(I,J)
 61          CONTINUE
 60        CONTINUE
 600       CONTINUE
C
           DO 70 I = 1 , N
                WRITE(4'I) (MAP(I,J),J=1,M)
 70        CONTINUE
           STOP
           END
```

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. Janecek, J.F., *Algorithm for Segmentation of Multichannel Images.* M.S.Thesis, Naval Postgraduate School, Monterey, California, December 1985.

3. Hunt, B.R., Kubler, O., " Karhunen-Loeve Multispectral Image Restoration, Part I:Theory," *IEEE Transaction on Acoustics, Speech, and Signal Processing*, Vol. Assp-32, No.3, pp. 592-600, June 1984.

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

5. NPS Report NPS-62-87-002 " The Analysis of Multichannel Two-Dimensional Random Signals," by Therrien, C.W., 31 October 1986.

6. Therrien, Charles W., Quatieri, T.F., Dudgeon, D.E., " Statistical Model-Based Algorithms for Image Analysis," *Proceedings of the IEEE*, Vol.74, pp. 532-551, 1986.

7. Pratt, W.K., *Digital Image Processing*, NewYork:Wiley, 1978.

8. Pratt, W.K., " Spatial Transform Coding of Color Images," *IEEE Trans. on Communication Technology*, Vol. Com-19, No.6, 1971.

9. Rosenfeld A, Kak A.C., *Digital Picture Processing* Vol.2, New York, 1982.

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 93943-5002	2
3.	Deniz Kuvvetleri Komutanligi Bakanliklar, Ankara / Turkey	5
4.	Deniz Harp Okulu Komutanligi Kutuphanesi Tuzla, Istanbul / Turkey	1
5.	Department Chairman, Code 62 Electrical and Computer Engineering Dept. Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor Charles W. Therrien, Code 62Ti Electrical and Computer Engineering Dept. Naval Postgraduate School Monterey, California 93943-5000	3
7.	Professor Chin-Hwa Lee, Code 62Le Electrical and Computer Engineering Dept. Naval Postgraduate School Monterey, California 93943-5000	3
8.	Ltjg, Timur Kupeli Bati Lojmanlari, Cerbe B-1, Yuzbasilar Golcuk, Kocaeli / Turkey	3

LMED

-8