1982

Noise reduction in images using Kalman filtering.

Sunil. Patil

University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation
https://scholar.uwindsor.ca/etd/2861

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000 ext. 3208.
NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS RECUE
NOISE REDUCTION IN IMAGES

USING KALMAN FILTERING

by

SUNIL PATIL

A Thesis
Submitted to the Faculty of Graduate Studies
Through the Department of
Electrical Engineering in Partial Fulfillment
of the Requirements for the Degree of
Master of Applied Science
University of Windsor
Windsor, Ont., Canada
1982
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>v</td>
</tr>
</tbody>
</table>

## CHAPTER I  
### INTRODUCTION
1.1 Image Enhancement ........................................ 1
1.2 Kalman Filtering and Image Processing ...................... 1
1.3 Kalman Filtering in Two-Dimensions ....................... 2
1.4 Thesis Organization .................................. 3

## CHAPTER II  
### IMAGE ENHANCEMENT
2.1 Introduction .......................................... 6
2.2 Histogram ............................................. 6
   2.2.1 Methods of Producing Transformed-Histograms .......... 7
   2.2.2 Transformation Function ............................ 7
2.3 Histogram Modification .................................. 8
   2.3.1 Histogram Equalization ............................ 9
      2.3.1.1 Continuous Data ................................... 9
      2.3.1.2 Discrete Data ................................... 11
   2.3.2 Adaptive Histogram Modification ..................... 12
2.4 Simple Filtering Methods ................................ 12
   2.4.1 Averaging .......................................... 14
   2.4.2 Image Sharpening ................................. 16
2.5 Results ................................................ 18
CHAPTER III ONE-DIMENSIONAL KALMAN FILTERING FOR NOISE REDUCTION IN IMAGES

3.1 Introduction ........................................ 27
3.2 Discrete Estimation .................................. 28
3.3 One-Dimensional (1-D) Kalman Filter ......... 30
   3.3.1 Formulation of 1-D Kalman Filter ....... 30
   3.3.2 Block Diagram of 1-D Kalman Filter .... 36
3.4 Edge-Preserving Kalman Filter .................... 37
3.5 Filter Performance Index (FPI) .................. 39
3.6 Results ............................................ 40

CHAPTER IV TWO-DIMENSIONAL KALMAN FILTERING

4.1 Introduction ........................................ 44
4.2 Markov Image Models ................................. 45
   4.2.1 Space-Invariant Gaussian Model ......... 45
   4.2.2 Space-Variant Gaussian Model ............ 47
4.3 Two-Dimensional (2-D) Kalman Filtering ....... 49
   4.3.1 Scalar Two-Dimensional Kalman Filter ... 49
   4.3.2 Disadvantages of Scalar 2-D Kalman Filter 52
4.4 Reduced Update Kalman Filter (RUKF) .......... 52
   4.4.1 Formulation of Reduced Update Kalman Filter 53
   4.4.2 Scalar Form of 2-D RUKF ................. 54
   4.4.3 Order of Computation ....................... 56
4.5 Parameters Estimation ............................... 57
   4.5.1 Identification ................................ 57
   4.5.2 Spatial Domain Models ....................... 59
   4.5.3 Parameters for 'Piston-Head', 'Disk-Pack', 'Face' . 60
4.6 Reduced Update Kalman Pre-Processing (RUKP) .. 61
4.7 Results ............................................ 61
CHAPTER V RESEARCH IN KALMAN FILTERING

5.1 Introduction ........................................... 68
5.2 Future Research ...................................... 68
  5.2.1 Image Restoration ............................... 68
  5.2.2 Boundary Conditions ......................... 69
  5.2.3 Adaptive 2-D Kalman Filtering .......... 70
5.3 Summary ............................................. 70
  5.3.1 Image Enhancement ............................ 71
  5.3.2 One-Dimensional Kalman Filtering .... 71
  5.3.3 Two-Dimensional Kalman Filtering .... 71
5.4 Conclusion ........................................... 72

APPENDICES

APPENDIX A FOUR QUADRANTS IN 2-D FILTERING .......... 75
APPENDIX B IDENTIFICATION OF DYNAMICAL MODEL .... 77
APPENDIX C DEGRADATION MODEL ......................... 81
APPENDIX D EDGE TEMPLATES .......................... 83

REFERENCES .............................................. 85
LISTING .................................................. 87
VITA AUCTORIS ........................................ 128
ABSTRACT

In this work an investigation of Kalman filtering for the reduction of noise in images is discussed. Algorithms are based on two assumptions: (1) The image data is uncorrelated with the noise data; (2) The autocorrelation function of the image data is separable. 'Separable' in this case means that the autocorrelation function of the image data is the product of vertical and horizontal autocorrelations.

One-dimensional (1-D) Kalman filtering algorithm is implemented for the reduction of noise in (NxN) size images. Noise reduction is achieved but at the cost of blurring the overall image. Edge-preservation is obtained by modifying the original one-dimensional set of filtering equations.

Direct extension of one-dimensional Kalman filtering equations in two-dimensions (2-D) results in a computationally inefficient and expensive 2-D processor. A reduced updating procedure is suggested forming a very efficient 2-D recursive Kalman filtering algorithm to solve two main problems: computations and storage.

Dynamical models are defined using linear predictive technique, and the support of the model is nonsymmetric half-plane (NSHP).

Image enhancement methods such as Histogram modification, Features Sharpening, etc. are combined with the 2-D Kalman processor to improve the subjective quality of the image, and form the 2-D pre-processing scheme.
ACKNOWLEDGEMENTS

The author wishes to express a feeling of heartiest thankfulness to Dr. M.A. Sid Ahmed for inspiring guidance, supervision and deep interest during this work.

I gratefully acknowledge Dr. M. Shridhar's significant contribution to this research through his intelligent suggestions.

The valuable advice and comments of Dr. G.A. Jullien and Dr. W.C. Miller, are gratefully acknowledged. In addition, the help of many graduate students and Mr. J. Novasad is sincerely appreciated.

To my parents, I extend my sincerest thanks. Without their help and love, though far away, this work would not have started.
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>FIG. #</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Gray-Level Transformation Function</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>Specification of Histograms</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>Averaging</td>
<td></td>
</tr>
<tr>
<td>(a) 5-Point</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>(b) 9-Point</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>Gradient Computation</td>
<td></td>
</tr>
<tr>
<td>(a) 4-Point</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>(b) 9-Point</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>Original Image (Piston-Head)</td>
<td>20</td>
</tr>
<tr>
<td>2.5(a)</td>
<td>Original Image (Face)</td>
<td>20</td>
</tr>
<tr>
<td>2.6</td>
<td>(i) Histogram of the Original Image</td>
<td>21</td>
</tr>
<tr>
<td>(ii) Transformation Function</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>Specified Histogram</td>
<td>22</td>
</tr>
<tr>
<td>2.8</td>
<td>Original Image Transformed to the Specified Histogram</td>
<td>23</td>
</tr>
<tr>
<td>2.9</td>
<td>Noisy Image</td>
<td>24</td>
</tr>
<tr>
<td>2.10</td>
<td>Averaged Image</td>
<td>24</td>
</tr>
<tr>
<td>2.11</td>
<td>Gradients of an Original Image</td>
<td>25</td>
</tr>
<tr>
<td>2.12</td>
<td>Sharpened Image</td>
<td>26</td>
</tr>
<tr>
<td>3.1</td>
<td>Message and Noise Model</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>B.D. of 1-D Kalman Filter</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>(a) Original Image (Piston-Head)</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>(b) Noisy Image</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>(c) Estimated Image</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>(a) Original Image (Face)</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>(b) Noisy Image</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>(c) Estimated Image</td>
<td>42</td>
</tr>
<tr>
<td>3.5</td>
<td>Edge-Preserved Estimated Image</td>
<td>43</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Markov Field ................................ 46</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>Global State Region of NSHP Model .......... 46</td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td>Space-Variant Filtering with On-Line Model 48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Estimation</td>
<td></td>
</tr>
<tr>
<td>4.4</td>
<td>Assignment of Points to Local and Global States 55</td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>Assignment for Approximate Reduced Update Kalman 58</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Filter (RUKF)</td>
<td></td>
</tr>
<tr>
<td>4.6</td>
<td>Reduced Update Kalman Pre-Processing (RUKP) 62</td>
<td></td>
</tr>
<tr>
<td>4.7</td>
<td>(a) Original Image (Piston-Head) 64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(b) Noisy Image ................................ 64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c) Estimated Image                        64</td>
<td></td>
</tr>
<tr>
<td>4.8</td>
<td>(a) Original Image (Disk-Pack) 65</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(b) Noisy Image ................................ 65</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c) Estimated Image                        65</td>
<td></td>
</tr>
<tr>
<td>4.9</td>
<td>(a) Original Image 66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(b) Modified Histogram Image 66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c) Estimated Image                        67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(d) Edges of an Estimated Image 67</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>Quarter Plane Regions 76</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>Non-Symmetric Half Plane 76</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>Degradation Process 81</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>Edge Templates 83</td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>Edges of Disk-Pack 84</td>
<td></td>
</tr>
</tbody>
</table>
LIST OF ABBREVIATIONS

\( G_{OR} \)  
Original Gray-Level

\( G_{NE} \)  
New Gray-Level

\( T \)  
Transformation Function

\( P_{D}(G_{OR}) \)  
Prob. Density of \( k \) Gray-Level

\( P_{D}(G_{OR}) \)  
Prob. Density of Original Gray-Level

\( P_{N}(G_{NE}) \)  
Prob. Density of New Gray-Level

\( N_{k} \)  
No. of \( k \)th Gray-Level

\( P_{d}(G_{d}) \)  
Desired Prob. Density

\( h, j, p \)  
Break points

\( \theta_{L} \)  
Angle with Vertical (left)

\( \theta_{R} \)  
Angle with Vertical (right)

\( S_{F}(m,n) \)  
Filtered Pixel

\( r(m,n) \)  
Observation Vector

\( W(m,n) \)  
White Noise
\(\psi(m,n)\) Measurement Error

\(\sigma_s^2\) Variance of Signal

\(\rho\) Horizontal Corr. Coeff.

\(S(m)\) Estimation Error

\(P(m,n)\) Error-Covariance Matrix

\(F(m,n)\) Kalman Gain Matrix

\(\eta\) Filter Performance Index (FPI)

\(G^+\) Set of Future Points in Region

\(G\) Set of Past Points in Region

\(\delta G\) Set of Present Points in Region

\(C\) Model Coeffs.

\(R^{++}\) NSHP Model

\(a\) After

\(b\) Before

\(J(m,n)\) Cost Function

\(F_{ss}\) Steady-State Gain
NPIX: Total No. of Pixels in an Image

TR: Threshold Intensity Level

\( O_{kl} \): Distorting System Parameters

X: Vector of N points

\( r_e \): Observation Vector Plus Edges

S_1: Local State Vector

S_2: Vector of Remaining Points from Global State Vector

NSHP: Nonsymmetric Half-Plane

RUKF: Reduced Update Kalman Filter

RUKP: Reduced Update Kalman Pre-Processing
CHAPTER 1

INTRODUCTION

Digital image processing algorithms with the present state of the art in computers have proved to be a very useful tool in various applications. Two principal application areas are: improvement of pictorial information for human interpretation, and processing of scene data for autonomous machine perception. Image enhancement, coding, filtering, transmission, segmentation are some of the areas in which currently, research is being conducted.

1.1 IMAGE ENHANCEMENT

Image enhancement and restoration procedures are used to process degraded images depicting unrecoverable objects or experimental results too expensive to duplicate. Enhancement techniques are developed and classified into two broad categories: frequency-domain methods and spatial-domain methods. Modification of the Fourier transform of an image is the basis for the first approach, and direct manipulation of the pixels in an image is the basis for the second approach.

1.2 KALMAN FILTERING AND IMAGE PROCESSING

Algorithm based on the concept that though the image is in the form of two-dimensional data, the functioning of the flying-spot scanner which basically scans the picture on a line by line basis converts the 2-D nature of the processing into 1-D.

Later on, Nehi and Habibi [3] suggested a detection-directed recursive algorithm using 1-D Kalman filtering. Biemond and Gerbrands [4] presented new edge-preserving Kalman filtering equations, as it was found that the noise from the image is certainly reduced using Kalman filtering, but at the cost of blurring the image. The work of Mendel [5], who made a detailed study to see the practicability of Kalman filter, should not go unnoticed. Computational time, and storage requirements have always been the areas of concern for the implementation of Kalman filtering algorithms for applications in image processing. Simplification of the 2-D problem into 1-D was performed with the assumption that the noise in images is 'white' in nature and either the horizontal or the vertical correlations of image data can be considered.

1.3 KALMAN FILTERING IN TWO-DIMENSIONS

Success of one-dimensional Kalman filtering in image processing created a new interest to devise an efficient two-dimensional Kalman filtering algorithm. J. Woods [6] formulated the image model equations based on Markov image models. Murphy and Silverman [7], Strintzis [8], Powell and Silverman [9], Woods and Radewan [6] suggested different versions of 2-D Kalman filtering with special
applications to image processing. Jain and Jain [10], Sriniath [11], Woods and Ingle [12] also presented Auto-Regressive Moving Average and improved versions of Kalman filtering equations.

The two forms of 2-D Kalman filters most studied are vector filters and scalar filters. The immediate problem faced in implementing a 2-D filter is the definition of 2-D state vector. In 2-D, the storage and computations pose major problems. Approximations of the vector and scalar filters are implemented without any significant loss of performance in the basic algorithm. The approximated scalar 2-D Kalman filter, called a reduced update Kalman filter, is based on the philosophy that correlation of the pixel, which is currently being processed, is very strong in a certain region and continues to weaken beyond that. This paved the way for an efficient 2-D Kalman filter algorithm with a saving of order in computation and storage.

1.4 THESIS ORGANIZATION

Research work in this thesis is divided in four main parts. In the first part we will discuss various image enhancement techniques. Histogram Equalization, Direct Histogram specification, averaging, and image sharpening using gradient computations are some of the techniques which are implemented with the intention of forming a complete pre-processing scheme combined with an efficient 2-D recursive noise reduction algorithm.
In the second part we will discuss one-dimensional Kalman filtering algorithm. 'Message-noise' model is developed which represents the image data formation in practice. It has been observed in estimated image that noise reduction is achieved, but at the cost of blurring the image. Subjective quality of an estimated image could be improved by using Edge-preservation scheme as suggested in the later sections of this part. Filter performance index (FPI), which represents the improvement in Signal to Noise Ratio (SNR) is also defined.

In the third part of this thesis stress is placed on the extension of 1-D Kalman filter to 2-D. Image data is defined as a 2-D Markovian field which enables one to devise a 2-D recursive processor. Space-variant and space-invariant Markovian image models are also discussed. All the models are defined over a nonsymmetric half-plane (NSHP) region. Updating the estimates over a small region around the point being currently processed instead of the complete image field, forms the basis for the new 2-D Reduced Update Kalman Filter (RUKF). It has been observed that this concept forms a computationally efficient and less expensive 2-D recursive Kalman processor. Definitions of 'Global state vector', 'Local state vector' are presented and used in forming the algorithm.

A dynamical model is designed based on the concept of linear prediction. Parameters of the model represent the coefficients to be used in the definition of state vector for image data. The subjective quality of an estimated image is improved by combining image
enhancement methods with 2-D RUKF which in a block forms an efficient 2-D pre-processing scheme.

The boundary conditions issue has been explained in connection with 2-D recursive estimations, which in itself has strong potential for future advanced research. An interesting discussion about the problems in 2-D Kalman estimation, which could present some insight for further research, precedes the summary and conclusions of the thesis.
CHAPTER II

IMAGE ENHANCEMENT

2.1 INTRODUCTION

In the field of image processing one could encounter various types of images with different backgrounds, textures, etc. One of the several critical features used in image analysis or pattern classification is thresholding in which one-dimensional feature space is used, the feature being the gray-level for the picture. Histogram of the gray levels for the image plays a role in deciding the selection of an optimum threshold. Also the modification of the histogram to the desired specifications could enhance the image and expose the unknown details useful in various applications.

In this chapter the following techniques are discussed:

1. Histogram equalization
2. Direct histogram modification
3. 5-point and 9-point averaging scheme
4. Gradient computations.

These methods are applied to 128 x 128 images.

2.2 HISTOGRAM

Many types of images contain dark objects on a light background or vice-versa. Histograms provide insight about the global description of an image. Histogram formation or histogram modification of an image
to a desired gray-level distribution could enhance the image. The selection of a suitable distribution changes from case to case. Invariably histogram is used for the optimum gray-level threshold selection used in pattern classification in which a one-dimensional feature space is used.

2.2.1 Methods of Producing Transformed Histograms

Several methods have been proposed that produce a transformed gray-level histogram based on various objectives: threshold selection sensing the valley or the peak, mapping the original distribution to the desired distribution by selecting the proper transformation function enhancing a particular region of an image, etc. Rosenfeld et al. [13] have suggested various schemes using edge-value operators, e.g., Laplacian, Roberts, 2-by-2, 4-by-4, 8-by-8 neighbourhoods. We have adopted the Direct Histogram specification technique, which has the potential for on-line image processing applications. The transformed image is obtained by specifying a desired histogram distribution and through histogram modification the original image is mapped to one with the desired histogram.

2.2.2 Transformation Function

Transformation function is a simple mapping function for mapping the original gray-level intensities, \( G_{OR} \), in an image to new gray-level intensities, \( G_{NE} \).

Hence:

\[ G_{NE} = T(G_{OR}) \]  \hspace{1cm} (2.1)

This form of transformation function is widely used in contrast
stretching, histogram analysis, segmentation etc. It is recommended to use the normalized image for the transformation, i.e., all the pixel values are normalized to lie in the range,
\[ 0 \leq G_{OR} \leq 1 \] (2.2)
The transformation function defined in Eq. (2.1) must satisfy the conditions:
(i) \( T(G_{OR}) \) is a single value and monotonically increasing in interval \( 0 \leq G_{OR} \leq 1 \)
(ii) \( 0 \leq T(G_{OR}) \leq 1 \) for \( 0 \leq G_{OR} \leq 1 \).
Condition (i) preserves the order from black to white in gray scale, while (ii) spans only between the allowed range of gray levels.
The inverse function will be denoted by,
\[ G_{OR} = T^{-1}(G_{NE}) \quad 0 \leq G_{NE} \leq 1 \] (2.3)
The transformation function satisfying Eq. (2.1) and (2.2) is shown in Fig. 2.1.

2.3 **HISTOGRAM MODIFICATION**
The size of the images used in this work is 128 x 128. Gray-levels are between 0-255, 0 represents black and 255 represents white. Gray-level histogram could be primarily a mixture of two unimodal histograms corresponding to the object and background populations. If the means of these populations are sufficiently far apart, and their standard deviations are sufficiently small, the image histogram will be bi-modal.
The gray-levels in a normalized image are in the interval [0,1]. If the probability density function for the gray-levels in the original image is $P_O(G_{OR})$ and the transformed image is $P_N(G_{NE})$ and if $P_O(G_{OR})$ and $T(G_{OR})$ are known $P_N(G_{NE})$ can be computed as,

$$P_N(G_{NE}) = \begin{bmatrix} P_O(G_{OR}) \frac{dG_{OR}}{dG_{NE}} \end{bmatrix}$$

$$G_{OR} = T^{-1}(G_{NE})$$  \hspace{1cm} (2.4)

The following discussion deals with controlling the transformation function by manipulating the probability density functions to obtain the modified histograms.

2.3.1 Histogram Equalization

2.3.1.1 Continuous Data

Let us consider the transformation function $T(G_{OR})$ as,

$$G_{NE} = T(G_{OR}) = \int_0^{G_{OR}} P_O(0) dD$$  \hspace{1cm} (2.5)

$$0 \leq G_{OR} \leq 1$$

Where 'D' is the dummy variable of integration. The transformation function as assumed in Eq. (2.5) satisfies the necessary conditions discussed earlier.

From Eq. (2.5) we can obtain:

$$\frac{dG_{NE}}{dG_{OR}} = P_O(G_{OR})$$  \hspace{1cm} (2.6)
Fig. 2.1 A Gray-Level Transformation Function
Substituting Eq. (2.6) in Eq. (2.4) we get,

\[ P_N(G_{NE}) = \left[ \begin{array}{cc} \frac{dG_{NE}}{dG_{OR}} & \frac{dG_{OR}}{dG_{NE}} \\ \frac{dG_{OR}}{dG_{NE}} & \frac{dG_{NE}}{dG_{OR}} \end{array} \right] \]

\[ G_{OR} = T^{-1}(G_{NE}) \]

\[ 0 \leq G_{NE} \leq 1 \]

(2.7)

which is a uniform density in the interval of definition of the transformed variable \( G_{NE} \).

2.3.1.2 Discrete Data

In the case of discrete data, or digitized image, we can write the probability density function of gray-level \( k \) in discrete form as,

\[ P_{0}(G_{ Ok}) = \frac{N_k}{NPIX} \]

\[ 0 \leq G_{ Ok} \leq 1 \]

\[ k = 0,1,2,...,L-1 \]

(2.8)

where \( L \) is the number of gray-levels, \( N_k \) is the number of times the \( k \)th level appears in an image and \( NPIX \) is the total number of pixels in an image. The discrete form of the transformation function will be,

\[ G_{NK} = T(G_{Ok}) = \sum_{j=0}^{k} \frac{N_j}{NPIX} \]

\[ = \sum_{j=0}^{k} P_{0}(G_j) \]

\[ 0 \leq G_{Ok} \leq 1 \]

\[ k = 0,1,2,...,L-1 \]

(2.9)

The transformation function in this case also satisfies the necessary conditions.
2.3.2 Adaptive Histogram Modification

On-line image processing applications require flexibility in order to change the gray-level distribution in an image to a desired distribution. If the desired density function is $P_d(G_d)$ and an image has been processed through Eq. (2.9), then the new transformation function for the desired density function will be,

$$G_d = T(G_d) = \int_0^G P_d(w)dw$$  \hspace{1cm} (2.10)

This procedure can be summarized as:

(i) Form histogram equalized image

(ii) Specify the desired density function $P_d(G_d)$ and obtain $T(G_d)$

(iii) Apply inverse transform to the image formed after step (i).

The desired density function could be obtained by specifying the four parameters $m, h, \theta_L, \theta_R$ as shown in Fig. 2.2. Controlling these parameters, various shapes for the desired density functions could be obtained, hence an analogy to 'JOY-STICK' [14].

2.4 SIMPLE FILTERING METHODS

Digital filtering of images is an extensively studied topic and various algorithms have been developed to design different versions of Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. We reviewed some of the simple forms of low-pass filtering (Neighbourhood Averaging) and high-pass filtering (Gradient Computations) in this work. Degraded images by additive white gaussian noise were used to test the performance of these algorithms,
Fig. 2.2 Specification of Histograms
and the results are presented at the end of this chapter.

2.4.1 Averaging

This is a straightforward spatial domain technique in which the smoothed image is obtained by the relation,

\[ S_F(x, y) = \frac{1}{N^2} \sum_{(m,n) \in \Omega} S(m,n) \quad (2.11) \]

where, \( x, y: 1, 2, \ldots, N-1 \) for \( N \times N \) image

\( \Omega \): set of co-ordinates of points in the neighbourhood of and excluding the point \((x, y)\)

\( N^2 \): total number of points in set \( \Omega \).

5-point and 9-point averaging methods as shown in Fig. 2.3 were implemented.

It was confirmed during experimentation that the averaging technique, though it reduces the noise, blurs the vital features of an image. The blurring could be reduced by using thresholding procedure as,

\[
S_F(x, y) = \begin{cases} 
\frac{1}{N^2} \sum_{(m,n) \in \Omega} S(m,n) \\
\text{if } \left| S(x, y) - \frac{1}{N^2} \sum_{(m,n)} S(m,n) \right| > TR \\
S(x, y) \quad \text{otherwise}
\end{cases} \quad (2.12)
\]

where TR is a specified non-negative threshold.
Fig. 2.3  Averaging
(a)  5-point Averaging
(b)  9-point Averaging
2.4.2 Image Sharpening

This method is basically used for sharpening the features of an image, and also in the problems related to feature extractions, segmentation, pattern classification. We have seen by averaging, that is, integrating over the region, the averaged image has blurred features, hence differentiation has the opposite effect. In image processing, differentiation is carried out by computing the gradients at the required point as follows:

\[
\text{GRAD} [S(x, y)] = \begin{bmatrix}
\delta S/\delta x \\
\delta S/\delta y
\end{bmatrix}
\]

(2.13)

The magnitude of the gradient is given by,

\[
M[\text{GRAD}] = [ (\delta S/\delta x)^2 + (\delta S/\delta y)^2 ]^{1/2}
\]

(2.14)

Eq. (2.14) could be written using absolute values of gradients in \(x\) and \(y\) directions for the discrete data. Roberts suggested 4-point gradient computation as shown in Fig. 2.4a.

\[
M[\text{GRAD}] = | S(x, y) - S(x+1, y+1) | + | S(x + 1, y) - S(x, y+1) |
\]

(2.15)

Sobel suggested a 9-point method as shown in Fig. 2.4b. The gradients in \(x\) and \(y\) direction will be,
Fig. 2.4 Gradient Computation

(a) 4-point Gradient (Robert's Cross-Difference)
(b) 9-point Gradient (Sobel's Operator)
\begin{align}
\text{GRAD}_x &= (S_7 + 2S_8 + S_9) - (S_1 + 2S_2 + S_3) \\
\text{GRAD}_y &= (S_3 + 2S_6 + S_9) - (S_1 + 2S_4 + S_7)
\end{align}

(2.16) \hspace{1cm} (2.17)

The gradients at point \( S_5 \) using absolute values of \( \text{GRAD}_x \) and \( \text{GRAD}_y \) is given by:

\[
M[\text{GRAD}] = |\text{GRAD}_x| + |\text{GRAD}_y|
\]

(2.18)

Gradient computations could be combined with the thresholding procedure to modify the intensity levels in an image of background, of edges or for segmentation.

2.5 RESULTS

In this work, the images used have gray-levels between 0-255. The image is degraded by adding Gaussian noise to it. The package 'NOISY' (Appendix C) generates the noisy image of the desired signal to noise ratio (SNR). Noisy image of SNR = 4.0 dB was obtained. Throughout this investigation noisy images of 'PISTON-HEAD', 'DISK-PACK', 'FACE' were used and SNR = 4.0 dB was specified.

Simple low-pass filtering (averaging) for the noise reduction is implemented. Both 5-point and 9-point techniques as explained earlier were tried to reduce the noise from the noisy image of 'PISTON-HEAD'. We confirmed the results discussed in the literature that the averaging technique does reduce the noise in an image, but causes blurring of the features, hence loss of structural information.
in an image. Fig. 2.5, 2.9, and 2.10 show an original image, noisy image (SNR = 4.0 dB) and an averaged image. Improvement in SNR is found to be 6.0 dB.

Gradient computations are done for various purposes, e.g., image sharpening, edges detection, segmentation, etc. There are different methods suggested in the literature for this computation. We used Robert's cross-difference (4-point) and Sobel's (5-point) gradient computation schemes. Gradients of the 'PISTON-HEAD' are as shown in Fig. 2.11 and an image with sharpened features is as shown in Fig. 2.12.

Histogram modification is also one of the image enhancement methods. We concentrated specially on the Direct Histogram modification scheme, which has the potential for the on-line processing. The gray-level distribution in an original image, Fig. 2.5a, which is extremely poor in quality is shown in Fig. 2.6. The desired histogram is specified for $m = 0.8$, $h = 1.0$, $\theta_L = 45^\circ$, $\theta_R = 5^\circ$, as shown in Fig. 2.7. We can generate any of the desired histograms by varying $m$, $h$, $\theta_L$ and $\theta_R$. The original image transformed to the specified histogram is as shown in Fig. 2.8.

Gradient computation schemes will be used for the extraction of edges from the noisy image. These edge weights are added to the observed sequence, to reduce the blurring caused by I-D Kalman estimation. Histogram modification scheme will be combined with 2-D Kalman Filter as we observed that the transformed image has a significant improvement on the subjective quality of the original image, but the overall image is noisy. Image enhancement techniques discussed in this chapter will be part of the building blocks of Kalman pre-processing scheme discussed later.
Fig. 2.5 Original Image (Piston-Head)

Fig. 2.5(a) Original Image (FACE)
Fig. 2.6 (ii) Transformation Function
Fig. 2.7 Specified histogram
Fig. 2.8 Original image transformed to the specified program.
Fig. 2.9 Noisy image

Fig. 2.10 Averaged Image
Fig. 2.11 Gradients of Original Image

(i) PISTON-HEAD

(ii) FACE
Fig. 2.12 Sharponco Image

PISTON-HEAD
CHAPTER III

ONE-DIMENSIONAL KALMAN

FILTERING FOR NOISE REDUCTION IN IMAGES

3.1 INTRODUCTION

A well known problem in digital image enhancement is the reduction of noise in image. The noise may originate from the imaging system, the transmission medium, the recording process or any combination of these and is usually assumed to be additive white signal, independent random noise with known statistics. In certain applications, the 'a priori' knowledge of the image itself is only of a statistical nature. In other words, the image does not contain characteristic features which would then have to be utilized in the enhancement processes.

The image enhancement, subject to the above restrictions, describes the classical problem of statistical estimation, and filtering, where one basically attempts to filter out the noise from an observation. The most efficient statistical recursive estimation technique of Kalman has been extended for image enhancement. However, it is necessary that the observation be a function of one independent variable, such as time, in contrast to an image which is defined on a plane.
In the following sections, the basic theory of the Kalman filter is discussed, and a suitable form for image enhancement is derived. Experiments show that the noise is effectively reduced, but the edges in the image are blurred by the Kalman filtering operation and the image contrast is reduced as well. Several researchers [2,3,15] have attempted to use additional information about the edges in noisy image during the filtering process. Edge information from a noisy image has been obtained and used during the filtering process, and the results show considerable noise reduction and significant improvement in the subjective quality of the image.

3.2 DISCRETE ESTIMATION

The discrete random process is defined by,

\[ \begin{align*}
X(m+1) &= AX(m) + BW(m) \quad (3.1) \\
Y(m) &= CX(m) + DV(m) \quad (3.2) \\
E[W(m)] &= 0, \quad E[V(m)] = 0 \quad (3.3) \\
E[W(m1)W^T(m2)] &= K_0 m_1 m_2^T \\
E[V(m1)V(m2)] &= L_0 m_1 m_2^T \quad (3.4)
\end{align*} \]

where A, B, C, D, K, L are in the form of matrices and are the function of m. Y(m) is defined as an observation vector. Eq. (3.1) and (3.2) represent the Message and noise model as shown in Fig. 3.1.

Estimation of N-vector X(m+1) which is a linear function of observations Y(1), Y(2), ..., Y(N) is based on the concept that the estimation error is orthogonal to the observations,
Fig. 3.1 Message and Noise Model
\[
E[X(m+1) - \hat{X}(m+1)] Y^t(1) = 0 \quad i=0,1,2,...,N
\]

The set of Kalman filter equations are derived with the help of Eq. (3.5) as discussed in the following sections.

### 3.3 ONE-DIMENSIONAL (1-D) KALMAN FILTER

The formulation of Kalman filtering equations with the application to image processing could be obtained, but we make a few assumptions such as:

1. The noise data and image data are uncorrelated
2. The autocorrelation function of an image data is separable
3. Noise is 'white' in nature.

We also assume that the ensemble autocorrelation function of the scanned lines is exponentially decaying.

\[
R(m) = \sigma_s^2 \rho^{-|m|} \quad (3.6)
\]

where,

- \( \sigma_s^2 \): Image variance
- \( \rho \): Horizontal correlation coefficient
- \( m \): Increment or shift in position

#### 3.3.1 Formulation of 1-D Kalman Filter

We will use the assumptions, discussed earlier to formulate 1-D Kalman filter. Horizontal correlation only is considered, in line with the assumption that autocorrelation is separable. Image
model [4] in 1-D scalar difference equation can be written as,

\[ S(m+1) = \rho S(m) + (\sqrt{1 - \rho^2}) W(m) \]  
\[ r(m) = S(m) + V(m) \]

where,

- \( S(m) \): Original data sequence
- \( W(m) \): White noise sequence
- \( r(m) \): Observed sequence
- \( V(m) \): Observed noise sequence

\[ E[W(m)] = 0, \quad E[V(m)] = 0 \]  
\[ E[S(m)] = \sigma_s^2 \]  
\[ E[V(m)] = \sigma_n^2 \]

\( \sigma_s^2 \): Variance of image data
\( \sigma_n^2 \): Variance of noise

It is desired to find the estimate of an \( N \)-vector \( S(m+1) \) denoted by \( \hat{S}(m+1) \), which is a linear function of observations \( r(1), r(2), \ldots, r(N) \). Hence we can write \( \hat{S}(m+1) \) as,

\[ \hat{S}(m+1) = \sum_{k=1}^{M} \alpha(k) r(k) \]  

where each \( \alpha(k) \) is an \( N \times N \) matrix and each \( r(k) \) is an \( N \)-vector. We will make use of the "orthogonality principle", which states that the linear estimate \( \hat{S}(m+1) \) which minimizes the estimation error \( [S(m+1) - \hat{S}(m+1)] \) is uncorrelated with every one of the observations.
\[ \mathbb{E}[S(m+1) - \hat{S}(m+1)] r'(1) = 0, \quad i=1,2,\ldots,N-1 \quad (3.13) \]

Let us represent the linear estimator as the following:

\[ \hat{S}(m+1) = Q_1 \hat{S}(m) + Q_2 r(m) \quad (3.14) \]

where \( Q_1 \) and \( Q_2 \) are matrices, functions of \( m \), determined by the statistics associated with the process defined in Eq. (3.7)-(3.11).

From Eq. (3.5), (3.7) and (3.14) we can write,

\[ \mathbb{E} \left[ \rho S(m) + \sqrt{1 - \rho^2} W(m) - (Q_1 \hat{S}(m) + Q_2 r(m)) \right] x \]

\[ r'(1) = 0 \quad (3.15) \]

Since \( W(m) \) and \( r(m) \) are uncorrelated, it follows that,

\[ \mathbb{E}[W(m) r'(i)] = 0 \]

\[ \mathbb{E}[u(m) r'(i)] = 0 \quad \forall \quad i \leq i < N \quad (3.16) \]

We obtain from Eq. (3.15),

\[ \mathbb{E} [\rho S(m) - Q_1 \hat{S}(m) - Q_2 r(m)] r'(i) = 0 \]

\[ i = 1,2,\ldots,N-1 \quad (3.17) \]

Substituting for \( r(m) \) from Eq. (3.8) into Eq. (3.17),

\[ \mathbb{E} [\rho S(m) - Q_1 \hat{S}(m) - Q_2 S(m) - Q_2 V(m)] r'(1) = 0 \quad (3.18) \]

Eq. (3.17) holds for the optimal estimate at the previous step.

Hence,

\[ \mathbb{E}[S(m) - \hat{S}(m)] r'(i) = 0, \quad i=1,2,\ldots,N-1 \quad (3.19) \]
Using Eq. (3.19) into Eq. (3.18) we obtain,

\[ E[ \rho - Q_1 - Q_2 ] S(m) r'(1) = 0 \]  

(3.20)

Rearranging Eq. (3.20),

\[ [\rho - Q_1 - Q_2] E[S(m) r'(1)] = 0 \]  

(3.21)

This equation is satisfied if the matrix \( Q_1 \) is chosen to be,

\[ Q_1 = \rho - Q_2 \]  

(3.22)

We will define the noise free observation as,

\[ \hat{r}(m) \triangleq \hat{S}(m) \]  

(3.23)

\[ S(m) \triangleq S(m) - \hat{S}(m) \]  

(3.24)

\[ \tilde{r}(m) \triangleq r(m) - \hat{r}(m) = r(m) - S(m) \]  

(3.25)

The vector \( S(m) \) is then the estimation error. Since \( r(m) \) is linear in \( S(m) \) and because \( \hat{S}(m) \) is in definition linear in \( r(i) \), we get:

\[ E[S(m+1) - \hat{S}(m+1)] \hat{r}'(m) = 0 \]  

(3.26)

Subtracting Eq. (3.26) from Eq. (3.13) and using Eq. (3.25),

\[ E[S(m+1) - \hat{S}(m+1)] \tilde{r}'(m) = 0 \]  

(3.27)

Substituting Eq. (3.27) and Eq. (3.14) in Eq. (3.27) and using Eq. (3.25), it follows that,

\[ E[\rho S(m) - \rho S(m) + (Q_1 \hat{S}(m) + Q_2 r(m))] \tilde{r}'(m) = 0 \]  

(3.28)
\[ E[p\ S(m) - Q_1 \ S(m) - Q_2 \ r(m)] [r(m) - \ S(m)]' = 0 \quad (3.29) \]

Substitute for \( r(m) \) from Eq. (3.8) and for \( Q_1 \) from Eq. (3.22)
and using Eq. (3.24),

\[ E[p\ S(m) - (p-Q_2) \ S(m) - Q_2 (S(m) + V(m))] \]

\[ [S(m) + V(m) - \ S(m)]' = 0 \quad (3.30) \]

\[ E[(p-Q_2) \ S(m) - Q_2 V(m)] [\ S(m) + V(m)]' = 0 \quad (3.31) \]

Let us define the estimation error covariance as,

\[ P(m) \triangleq E[\ S(m) \ S'(m)] \quad (3.32) \]

we can simplify Eq. (3.31) as,

\[ E[(p-Q_2) \ S(m) \ S'(m) - Q_2 V(m) \ S'(m) +
(p-Q_2) \ S(m) V'(m) - Q_2 V(m) V'(m)] = 0 \quad (3.33) \]

Rearranging we obtain,

\[ [(p-Q_2) \ E(\ S(m) \ S'(m)) - Q_2 \ E(V(m) \ S'(m)) +
(p-Q_2) \ E(\ S(m) V'(m)) - Q_2 \ E(V(m) V'(m))] = 0 \quad (3.34) \]

After taking the required expectations we get,
\[ (p_{-Q_2}) P(m) - Q_2 \sigma_n^2 = 0 \]  

From Eq. (3.35) we can find out the relation for the matrix \( Q_2 \) as,

\[ Q_2 = \frac{p \ P(m)}{P(m) + \sigma_n^2} \]  

Now we can define \( P(m+1) \) as,

\[ P(m+1) = E[\hat{S}(m+1) \hat{S}^T(m+1)] \]  

\[ = E \left\{ [S(m+1) - \hat{S}(m+1)] [S(m+1) - \hat{S}(m+1)]^T \right\} \]  

\[ = E \left\{ [p S(m) + (\sqrt{1 - p^2}) W(m) - Q_1 S(m) \right. \]  

\[ + Q_2 r(m)] [p S(m) + (\sqrt{1 - p^2}) W(m) \]  

\[ - (Q_1 \hat{S}(m) + Q_2 r(m))]^T \} \]  

After substituting for \( Q_1 \) from Eq. (3.36) and \( r(m) \) from Eq. (3.8), Eq. (3.39) simplifies to,

\[ P(m+1) = (p_{-Q_2}) P(m) (p_{-Q_2})^T + (1-p^2) \sigma_s^2 + Q_2 \sigma_n^2 \]  

Denoting \( Q_2 = F \), called as Kalman gain, in Eq. (3.41) and from Eq. (3.22) and Eq. (3.14) we can write the set of equations for one-dimensional
Kalman filter for an image as,

\[
\hat{S}(m+1) = (\rho - F(m)) \hat{S}(m) + F(m) r(m) \tag{3.42}
\]

\[
F(m) = \frac{\rho P(m)}{P(m) + \sigma_s^2} \tag{3.43}
\]

\[
P(m+1) = (\rho - F(m))^2 P(m) + (1 - \rho^2) \sigma_s^2 + (F(m))^2 \sigma_n^2 \tag{3.44}
\]

The recursive algorithm as stated in Eq. (3.42) - (3.44) requires
the statistical 'a priori' information as \(\hat{S}(0)\) and \(P(0)\) from the image
data. In our case we assume,

\[
\hat{S}(0) = 0 \tag{3.45}
\]

\[
P(0) = \sigma_s^2
\]

### 3.3.2 Block-Diagram of 1-D Kalman Filter

We formulated one-dimensional Kalman filter algorithm in the
preceeding section based on 'orthogonality principle'. Message and
noise model was defined by Eq. (3.7) and (3.8) and Eq. (3.42) - (3.44)
form the algorithm. We know that Kalman estimated data will be,

\[
\hat{S}(m+1) = (\rho - F(m)) \hat{S}(m) + F(m) r(m)
\]

Rearranging we get,

\[
\hat{S}(m+1) = \rho \hat{S}(m) + F(m) (r(m) - \hat{S}(m)) \tag{3.46}
\]
The block diagram representation of the Message model and an optimal estimator is shown in Fig. 3.2.

3.4 EDGE PRESERVING KALMAN FILTER

The estimated image from the observed noisy data using 1-D Kalman filter equations derived in the preceding section show that the noise is effectively reduced, but that the edges in the image are blurred by the filtering operations. These effects decrease the subjective quality of the image.

Several researchers have attempted to reduce the distortion caused by filtering by using additional structural information from the image. We used edge information from the noisy image in the original Kalman filter equations. The edge detection is done by using gradient computation techniques as explained earlier and also by using various edge templates as discussed in Appendix (D).

The edge weights $e(m)$ thus obtained are added to the observed data as follows:

$$r_{e}(m) = r(m) + \psi e(m)$$

(3.47)

The factor $\psi$ is determined [4] from the steady state value of the Kalman gain $F$, denoted as $F_{ss}$, obtained after a certain number of iterations, where the absolute difference between two subsequent values of $F$ computed using Eq. (3.40), (3.41) is less than a certain criterion.
Fig. 3.2 One-Dimensional Kalman Filter
\[ \varphi = \frac{a}{1 - a^2} \]  
\[ (3.48) \]

where,

\[ a = \rho - F_{ss} \]  
\[ (3.49) \]

The observed sequence with the addition of edge information as shown in Eq. (3.47) is processed through the Kalman filter equations.

3.5. FILTER PERFORMANCE INDEX (FP1)

To study the effectiveness of the performance of the filter, a measure of performance \( \eta \) is introduced. We define \( \eta \) in decibels (dB) as

\[ n = 10 \log_{10} \left( \frac{\sum_{i=1}^{128} \sum_{j=1}^{128} \left[ r(i,j) - S(i,j) \right]^2}{\sum_{i=1}^{128} \sum_{j=1}^{128} \left[ S(i,j) - \hat{S}(i,j) \right]^2} \right) \text{ dB} \]  
\[ (3.50) \]

Eq. (3.50) in fact represents the ratio of variance of signal to variance of noise. \( r(i,j) \) is the observed data, \( S(i,j) \) the corresponding original data and \( \hat{S}(i,j) \) is the estimated data.
3.6 RESULTS

One-dimensional Kalman filter algorithm is developed for applications in image processing. In this case the global state vector and local state vector are the same. Initial conditions, \( S(0) = 0 \) and \( P(0) = \sigma_s^2 \) are assumed. Noisy images have SNR = 4.0 dB. Only horizontal correlation is considered in the algorithm and the coefficients are assumed constant throughout recursion. When noisy images were processed through the Kalman filter, it was found that the algorithm works very effectively to reduce the noise but at the cost of blurring an image. Fig. 3.3 and 3.4 show for the comparison of original and noisy and estimated images, that improvement in SNR (FPI) was found to be 6.9 dB and 7.0 dB respectively.

Edge-preserving one-dimensional Kalman filtering algorithm as discussed earlier was implemented. Edge-weights were computed using (3x3) operator. The observation vector was modified accordingly. The estimated image with heavy emphasis on edges has retained the structural information and brings about a significant improvement in the subjective quality of an image. FPI was improved to 7.4 dB. Edge-preserving estimated image is as shown in Fig. 3.5.
Fig. 3.3  
(a) Original Image (PISTON-HEAD)  
(b) Noisy Image  
(c) Estimated Image
Fig. 3.4  
(a) Original Image (FACE)  
(b) Noisy Image  
(c) Estimated Image
Fig. 3.5 Edge-Preserved Estimated Image
CHAPTER IV

TWO-DIMENSIONAL KALMAN FILTERING

4.1 INTRODUCTION

In the recent past, there has been much interest in the problem of two-dimensional recursive filtering in particular, with application to images. Strintzis [8], Jain [10] and Murphy and Silverman [7] suggested various image estimation techniques. Strintzis presented an autoregressive-moving average (ARMA) modelling approach to recursive processing. Jain considered both implicit and explicit partial difference equation models for images. The computational methods are based on fast transform algorithms. This method is not suitable for spatially varying image models. Murphy and Silverman considered a vector scanning approach to image restoration.

Efforts to achieve a truly recursive 2-D Kalman filter were of only limited success due to both the difficulty in establishing a suitable 2-D model, and also to the high dimension of the resulting state vector. In this chapter we will discuss Markov image models for both space-invariant and space-variant Gaussian cases, followed by 2-D Kalman filtering algorithms. Part of the chapter is also devoted to the discussion of 2-D model identification and a complete pre-processing scheme based on Kalman filtering. Exploration of the possibility of using this scheme with various feature extraction algorithms has been presented in the latter part of the chapter.
4.2 MARKOV IMAGE MODELS

In this section, the discussion is divided into two parts. The first part covers space-invariant Gaussian models and the second part, space-variant Gaussian models.

4.2.1 Space-Invariant Gaussian Model

It is shown [16] that nonsymmetric half-plane (NSHP) recursive models generate a special class of the Markov random fields. The first-order and separable models, often used in image processing, are seen to be the special cases of the general discrete Markov random field.

**Definition:** A discrete Markov random field:

Let $S$ be a random field on $\mathbb{Z}^2$, the 2-D lattice. Let a band of minimum width $p$, $\partial G$ ("the present") separate $\mathbb{Z}^2$ into two regions $G^+$ ("the future") and $G^-$ ("the past"). Then $S$ is Markov-$p$ if 

$[S/G^+ \text{ given } S/\partial G]$ is independent of $[S/G^-]$ for all $\partial G$, as shown in Fig. 4.1.

It is shown that for the homogeneous, zero-mean, Gaussian case, this definition is equivalent to the following difference equation model,

$$S(m, n) = \sum_{\partial p} C_{kl} S(m-k, n-l) + W(m, n)$$  \hspace{1cm} (4.1)

where

1) $E[S(m, n) W(k, l)] = c^2 \delta_{mk} \delta_{nl}$

2) $\partial p = \{k, l k^2 + l^2 < p^2 \text{ and } (k, l) \neq (0, 0)\}$
Fig. 4.1 Regions Used in Definition of Markov Field

\[ R_{\Theta^+} = SG \cup G^- \]

Fig. 4.2 Global State Region of NSHP Model
(iii) \( W(m,n) \) is a Gaussian, zero-mean, homogeneous random field.

(iv) The \( C_{kl} \) are the interpolator coefficients of the minimum mean square error (MMSE) 

(v) \( \sigma_u^2 \) is the mean square interpolation error.

Using the above properties it was shown that

\[
R_s(m,n) = \sum_{\delta_p} C_{kl} R_s(m-k, n-l) + \delta_{mn} \sigma_u^2
\]  

(4.2)

It is shown that for every 2-D Markov field, there does not correspond a finite-order NSHP model. However, the inverse is true. A finite order NSHP model driven by white noise generates a Markov random field. Thus we arrive at general Markov image models of the form,

\[
S(m,n) = \sum_{R \oplus} C_{kl} S(m-k, n-l) + W(m,n)
\]  

(4.3)

where \( R \oplus = \{m \geq 0, n \geq 0\} \cup \{m < 0, n > 0\} \) as shown in Fig. 4.2

and where \( W(m,n) \) is a white noise. Thus the state region of Fig. 4.1 has expanded into the very large global state region of Fig. 4.2.

This is expected since the state region must separate the past from the future.

4.2.2 Space-variant Gaussian Model

The model would be space-variant by allowing the model coefficients \( C_{kl} \) to be a function of position. The model would thus be denoted \( \{C_{kl}^{mn}\} \). In order to use these models two problems are to be overcome. The first problem is model design. The second problem is the storage
Fig. 4.3 Space-Variant Filtering With
On-Line Model Estimation
requirements for the model coefficients. For example, if the model was of (2x2) order, having 12 coefficients, the amount of storage required for the coefficients would be 12(NxN), assuming that the coefficients changed at each pixel. It is suggested that these problems could be solved by using noisy data for identification as shown in Fig. 4.3. The resulting processor would then be an approximation to the space-variant Gaussian models.

4.3 TWO-DIMENSIONAL (2-D) KALMAN FILTERING

Kalman filtering is well established for the one-dimensional case. The assumption is that the signal can be modelled by a first-order vector dynamical system. The vector is called the state vector and summarizes the effect of the past on the future for the 1-D Markov process. The set of Kalman filter equations have the following simple interpretation. First we project the last estimate forward using the dynamics of the system model. Then we update this estimate using the new observation. The remaining equations compute the error covariances to compute the gain matrix required for updating.

4.3.1 Scalar Two-Dimensional Kalman Filter

When we try to extend 1-D Kalman filter equations to 2-D, we encounter a problem concerning the definition of the state vector. To design the 2-D set of Kalman filter equations we consider the scanning of a 2-D square region consisting of an NxN regularly spaced lattice. Thus at any point in the picture some points will be "past", one point will be "present", and the remaining points will be the...
"future". These words have their conventional meaning with respect to the order in which the points are processed, and correspond to the Markov field regions as shown in Fig. 4.2. The scalar version of 1-D dynamical model could be written as,

\[ S(m) = \sum_{k=1}^{M} C_k S(m-k) + W(m) \]  (4.4)

From the nature of Eq. (4.1) and (4.4) we are in a position to appreciate the difficulty with state in two dimensions. We will center the discussion around the (MxM) order model. For each computation of a new point, Eq. (4.1) uses \( M^2 \) points, so the amount of computation is related to the order of the model. When Eq. (4.1) moves across the image, values of \( S \) will be needed from the previous \( M \) lines. Thus the memory requirements of Eq. (4.1) are not of the order related to the order of the model as in Eq. (4.4). Because of this large memory requirement, the state for Eq. (4.1) must contain \( M \) previous lines and hence has \( MN \) components. It is convenient to call this full state the Global State and to take the part of the state on the support of the model and call it the Local State.

Local State: The minimum part of the global state needed to compute the next output given the present input. This local state has \( M \) number of elements related to the order of the model and hence its size determines the order of computation of the model.

At this point we will discuss the Kalman filtering equations for the model of Eq. (4.1) and the observation equation,

\[ r(m,n) = S(m,n) + V(m,n) \]  (4.5)
where \( W \) and \( V \) are white Gaussian sources. Scanning operation transforms 2-D problem into an equivalent 1-D problem. Thus Eq. (4.1) can be transformed into the following form after the global state vector of \( M(N+1) \) components.

\[
S(m,n) = [ S(m,n), S(m-1,n), \ldots, S(1,n); S(N,n-1), \ldots, S(1,n-1); \ldots, S(N,n-M), \ldots, S(n-M+1,n-M) ]^T
\]

(4.6)

With this transformation we obtain

\[
S(m,n) = C S(m-1,n) + W(m,n)
\]

(4.7)

with the corresponding observational model. The Kalman equations for this model are,

\[
\hat{S}_b(m) = C \hat{S}_a(m-1)
\]

(4.8)

\[
P_b(m) = C P_a(m-1) C^T + \sigma_s^2
\]

(4.9)

\[
F(m) = P_b(m) H^T (H P_b(m) H^T + \sigma_n^2)^{-1}
\]

(4.10)

\[
\hat{S}_a(m) = \hat{S}_b(m) + F(m) [ r(m) - H \hat{S}_b(m) ]
\]

(4.11)

\[
P_a(m) = [ I - F(m) H ] P_b(m)
\]

(4.12)

where

\[
P_i(m) \overset{\Delta}{=} E[(S-S)^T] \cdot (S-S)_i^T, \ i = a, b
\]

(4.13)

The subscripts 'a' and 'b' indicate after and before updating respectively.
### 4.3.2 Disadvantages of Scalar 2-D Kalman Filter

The difficulty with the set of equations in the previous section is the amount of computation and storage requirements associated with them. For example, consider an \((M \times M)\) order system model and an observation region consisting of \(N \times N\) square with \(N \gg M\). The dimension of matrix equations is approximately \(MN\). This means that in general the order of computation is \(M^3 N^3\). For the spatial invariant model this will be \(M^2 N^2\). If \(M = 4\) and \(N = 100\), we require 160,000 operations per output point. The overall total computation for the 10,000 element picture would be \(10^9\). At 1 \(\mu s\)/operation, the computer time for such calculations would be approximately 20 minutes. In addition storage problems are immense.

### 4.4 Reduced Update Kalman Filter

In the previous sections the discussion was devoted to the scalar 2-D Kalman filter in the formulation of which the global state is \(MN\) dimensional where \(M\) is the order of the recursive model and \(N\) is the width of the picture. The main concept of the reduced update filter derives from the fact that the Kalman equations are composed of two steps: a prediction part and an update part. The prediction part as shown in Eq. (4.8) is a computationally straightforward projection of \(M^2\) previous estimates. However, the update part involves calculations involving each of the \(MN\) points in the global state. Since \(N \gg M\), one can reduce the bulk of computation by reducing the update process. Thus we update only those elements
of the global state within a certain distance of the point currently being processed \((m,n)\). Omitting the update of points far away should not make a significant change in the performance of the filter. Hence we update only the local state vector for each observation.

### 4.4.1 Formulation of Reduced Update Kalman Filter (RUKF)

In this section we will discuss the formulation of reduced update Kalman filter. Let the model be \((M \times M)\) order NSHP as discussed before. Let \(r(m,n)\) be the observation over \(N \times N\) rectangular region. We will divide the global state into two vectors: \(S_1(m,n)\) is a local state vector and \(S_2(m,n)\) contains the remaining points. The global state vector can be written as,

\[
S(m,n) \triangleq [S_1^T(m,n), S_2^T(m,n)]^T
\]  

(4.14)

The drive vector for the dynamical model is given by:

\[
W(m,n) \triangleq [W(m,n), 0, \ldots, 0]^T
\]  

(4.15)

The observation equation is given by,

\[
r(m,n) = h^T S(m,n) + V(m,n)
\]  

(4.16)

where

\[
h^T = (1, 0, \ldots, 0)
\]

If we partition \(h^T = (h_1^T, h_2^T)\) and \(h_2 = 0\), we will obtain a new observation equation,

\[
r(m,n) = h_1^T S(m,n) + V(m,n)
\]  

(4.17)

At this point we can write the Kalman filtering as follows:
PREDICTION: \( m \rightarrow m+1 \)

\[
P_b(m,n) = C P_a(m-1,n) C^T + \sigma_s^2
\]  
(4.18)

\[
S_b(m,n) = C S_a(m-1,n)
\]  
(4.19)

UPDATE:

\[
K(m,n) = P_b(m,n) h [h^T P_b(m,n) h + \sigma_n^2]^{-1}
\]  
(4.20)

\[
S_a(m,n) = S_b(m,n) + [r(m,n) - h^T S_b(m,n)]
\]  
(4.21)

\[
P_a(m) = [I - F(m,n)h^T] P_b(m,n)
\]  
(4.22)

In the above set of equations, different terms have the same meaning as explained before. Assignment of points for the local state vector and global state vector is as shown in Fig. 4.4.

4.4.2 Scalar Form of 2-D RUKF

The set of equations from (4.18) to (4.22) can provide considerable computational savings over the standard Kalman filtering equations directly extended from 1-D. Equations (4.18) to (4.22) could be written in the scalar form as follows,

\[
S_b(m,n) = \sum_{R \oplus k,l} C_{k,l} S_a(m-1,n)(m-k,n-l)
\]  
(4.23)

\[
R_b(m,n;k,l) = \sum_{op \oplus m-o,n-p} C_{op} R_a(m-1,n)(m-o,n-p;k,l)
\]  
(4.24)
Assignment of Points to Local $s_1$ and Global $S$ State
\[ R_b^{(m,n)}(m,n;m,n) = \sum_{k,l} C_{kl} R_b^{(m,n)}(m,n;m-k,n-l) + \sigma^2_{n} \]  
(4.25)

\[ F(m,n)(i,j) = R_b^{(m,n)}(m,n;m-i,n-j) / (R_b^{(m,n)}(m,n;m,n) + \sigma^2_{n}) \]
\( (i,j) \in \Theta \)  
(4.26)

\[ S_a^{(m,n)}(i,j) = S_b^{(m,n)}(i,j) + F(m,n)(m-i,n-j) [r(m,n) - S_b^{(m,n)}(m,n)] \]  
(4.27)

\[ R_a^{(m,n)}(i,j,k,l) = R_b^{(m,n)}(i,j,k,l) - F(m,n)(m-i,n-j) \]
\[ R_b^{(m,n)}(m,n;k,l) \]  
(4.28)

Eq. (4.23) represents propagation of the previous estimates through the dynamics of the system. In these scalar equations, the superscript indicates the step in the filtering, while argument represents the position of the data. Eq. (4.18) expresses the error in this predicted estimate which can be written in scalar form as in Eq. (4.24) and (4.25). Eq. (4.20) is written as Eq. (4.26) in scalar form.

4.3.3 Order of Computation

We will investigate the order of computation of this reduced update Kalman filter. Eq. (4.23) will be \( M^2 \) for an \( M^th \) order model. Thus Eq. (4.23) will be \( M^2 \) for each \( k,l \). Since there are \( MN \) points in \( \Theta \), we need \( M^3 N \) for Eq. (4.24). Eq. (4.28) has to be computed for each pair of \( i,j \) and \( k,l \). Overall total computation per
point is $M^3N$. This is to be compared with $M^3N^3$ for the general Kalman filter. The overall savings of a factor $N^2$ results from two simplifications. First, the reduced update has reduced the orders of the matrices from $MN \times MN$ to $M^2 \times MN$. Second, for the $(M \times M)^\text{th}$ order filter model, the scalar equations only require the newly computed values for the point $(m,n)$. Fig. 4.5 explains the regions assignment for approximate reduced update Kalman filter.

4.5 PARAMETERS ESTIMATION

There are various algorithms discussed in the literature, which tackle the problem of determining the set of parameters required for the implementation of filtering and prediction. The interest in this area started with the work of Yule [17], who introduced the motion of an autoregressive series in connection with the study of sun spots in 1927. Box and Jenkins [18] and Parzen [19] have provided further insights into the problem.

4.5.1 Identification

A number of approaches to the Identification problem have been proposed. Some of them are as follows:

1. Spectral modelling using 2-D spectral factorization to develop frequency domain models.

2. Linear predictive techniques for developing spatial domain models.

3. Multiple model techniques suitable for the estimation of models in adaptive multiple-model algorithms.
Fig. 4.5 Assignment for Approximate Reduced Update Kalman Filter (RUKF)
4.5.2 Spatial Domain Models

The common practice is to use either of the techniques from:

i) Linear least squares methods

ii) Stochastic approximation methods

iii) Maximum likelihood methods

iv) Linear predictive methods

In the linear prediction approach, one minimizes the square residual,

$$J_{m,n} = \sum_{(i,j) \in \Omega_{m,n}} [S(i,j) - \hat{C}^T S_l(i,j)]^2$$  \hspace{1cm} (4.29)

Here $S_l(i,j)$ is the local state vector at $(i,j)$ and $\Omega_{m,n}$ is local data history at pixel $(m,n)$. These techniques assume that a noise-free sample of signal is available for parameter estimation. This assumption is appropriate in the case of constant models where on-line parameter estimation is not required. However, in the case of non-constant models where the spatial variation of model parameters is to be determined, it is necessary to base estimation on the noisy data.

$$J_{(m,n)} = \sum_{(i,j) \in \Omega_{m,n}} [r(i,j) - \hat{C}^T r_l(i,j)]^2$$  \hspace{1cm} (4.30)

For the scalar-input and scalar-output system, parameters $\hat{C}$ are given as:
\[
\begin{bmatrix}
\hat{C}_1 \\
\hat{C}_2 \\
\vdots \\
\hat{C}_n \\
\end{bmatrix}
\begin{bmatrix}
\hat{D}_1 & \cdots & \hat{D}_n \\
\vdots & \ddots & \vdots \\
\hat{D}_n & \cdots & \hat{D}_{2n-1} \\
\end{bmatrix}
^{-1}
\begin{bmatrix}
\hat{D}_{n+1} \\
\vdots \\
\hat{D}_{2n} \\
\end{bmatrix}
\]

(4.31)

where,

\[
\hat{D}_k = \frac{1}{N} \sum_{i=k}^{N} S_i \cdot S_{i-k}
\]

(4.32)

The parameters were estimated for the different images. They are listed in the following sections.

### 4.5.3 Parameters for 'Piston-Head', 'Disk-Pack', 'Face'

<table>
<thead>
<tr>
<th>S No.</th>
<th>'PISTON-HEAD'</th>
<th>'DISK-PACK'</th>
<th>'FACE'</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0980523</td>
<td>-0.00839745</td>
<td>-0.0529677</td>
</tr>
<tr>
<td>2</td>
<td>-0.198515</td>
<td>-0.189912</td>
<td>-0.0554307</td>
</tr>
<tr>
<td>3</td>
<td>0.355872</td>
<td>0.344637</td>
<td>0.269529</td>
</tr>
<tr>
<td>4</td>
<td>0.396838</td>
<td>0.0187334</td>
<td>0.0351073</td>
</tr>
<tr>
<td>5</td>
<td>-0.0943953</td>
<td>0.00131497</td>
<td>0.00298018</td>
</tr>
<tr>
<td>6</td>
<td>-0.0668639</td>
<td>-0.0614636</td>
<td>-0.00964986</td>
</tr>
<tr>
<td>7</td>
<td>0.412931</td>
<td>-0.0405616</td>
<td>-0.110690</td>
</tr>
<tr>
<td>8</td>
<td>0.3311532</td>
<td>0.4227348</td>
<td>0.422228</td>
</tr>
<tr>
<td>9</td>
<td>-0.208882</td>
<td>0.0216542</td>
<td>0.00433340</td>
</tr>
<tr>
<td>10</td>
<td>0.0791450</td>
<td>0.0351858</td>
<td>0.0336115</td>
</tr>
<tr>
<td>11</td>
<td>-0.0190956</td>
<td>0.0845092</td>
<td>0.0731387</td>
</tr>
<tr>
<td>12</td>
<td>-0.0866957</td>
<td>0.364330</td>
<td>0.346117</td>
</tr>
</tbody>
</table>
4.6 REDUCED UPDATE KALMAN PRE-PROCESSING (RU KP)

We prepared a complete pre-processing scheme combining histogram modification, two-dimensional Kalman processor and some of edge detection, gradient computations and template matching algorithms. The performance of this scheme was found to be very encouraging. The input image was extremely poor in quality, hence the histogram of the gray-levels in the original image was modified to the one specified by an interactive technique. The modified image had a significant improvement in the subjective quality but was noisy. This image was processed through the Kalman processor and substantial reduction of noise was obtained. Edge detection schemes enhance some useful structural information from the Kalman estimated image. This block could form an integral part of the systems used for pattern classification, criminology, medical diagnosis, etc. The block diagram of the scheme is as shown in Fig. 4.6, and the results are shown at each stage.

4.7 RESULTS

Two-dimensional Kalman filter algorithm was developed based on the definition of the Markov image field. To solve the problems of computations and storage associated with the above algorithm, a reduced updating two-dimensional Kalman filter set of equations were
Fig. 4.6 Reduced Update Kalman Pre-Processing (RUKP)
derived.

The dynamical model of (2x2) order was developed based on the linear predictive technique. This forms a set of 12 coefficients to be used in the design of the Kalman filter. The support of this model was on a nonsymmetric half-plane (NSHP).

Updating of the covariance and the estimate was restricted over the region of 10 columns and 20 rows. The definition of the steady state region is problem-oriented. During experimentation it was found that the correlation at the boundaries of the region is substantially decayed. Two-dimensional recursion was performed from top to bottom and left to right. Noisy images as shown in Fig. 4.7b and 4.8b were processed through the filter, and the estimated images were obtained as shown in Fig. 4.7c and 4.8c. Improvement in FPI was found to be 8.2 dB and 8.1 dB respectively. The algorithm was very effective in the reduction of noise.

The two-dimensional pre-processing scheme formed by histogram modifier and Kalman filter is proposed. This effort is done with the intention of extending these algorithms for use in feature extraction, pattern classification etc. Histogram modified image as shown in Fig. 4.9b was processed through the 2-D Kalman filter. Significant improvement in FPI (6.4 dB) and in the quality of an image was obtained. The edges of an estimated image are as shown in Fig. 4.9d.
Figure 4.7 
(a) Original Image (PISTON-HEAD) 
(b) Noisy Image 
(c) Estimated Image
Fig. 4.8  
(a) Original image (DISK-PACK)  
(b) Noisy image  
(c) Estimated image
Fig. 4.9  (a) Original Image
        (b) Modified Histogram Image
Fig. 4.9 (c) Estimated image
(d) Edges of an Estimated Image
CHAPTER V

RESEARCH IN KALMAN FILTERING

5.1 INTRODUCTION

Kalman filtering algorithm has proved to be a very useful tool to solve problems in various areas, e.g., space applications, system identification, decision making, optimal state estimations and now in image processing. Multi-dimensional Kalman filtering algorithms have suffered a severe setback because of excessive computations and storage until 1979, when Murphy and Silverman and Woods suggested various approximated forms of the multi-dimensional Kalman filtering set of equations. These algorithms are devised based on the nature of the functioning of the Raster Scan. There is tremendous potential for future research in Kalman filtering, to improve the approximated versions in order to yield better results in image processing.

5.2 FUTURE RESEARCH

We discussed the new form of two-dimensional Kalman filtering as applied to image processing. Reduced update Kalman filter algorithm has performed very satisfactorily and has helped in solving the two main problems, namely; excessive computations and storage.

5.2.1 Image Restoration

The concept of reduced updating could be extended to the problems related to image restoration. In image restoration, the
observation equation is modified to reflect a transformation on the signal as,

\[ r(m,n) = \sum_{k} O_{kl} S(m-k, n-l) + V(m,n) \]  \hspace{1cm} (5.1)

For a particular support of the distorting system \( O \), Eq. (5.1) can be put in the form of Eq. (4.16). The observation model can be designed by using one of the techniques discussed earlier.

5.2.2 BOUNDARY CONDITIONS

The initial condition problem for 1-D Kalman filtering is to select an optimal initial estimate of the state, along with its associated initial error covariance. When the data set is finite, a boundary value problem consisting of both initial and final conditions must be considered. If one has a finite length of section of an infinite-length data set, final conditions are ignored, thus enabling one to reduce the general boundary condition problem to an initial value problem.

The 2-D boundary conditions problem is more complicated because the data scan causes the boundaries to be encountered repetitively in the course of filtering, at the beginning and end of each scan line. We assumed the random boundary condition, but this work could be extended to develop various algorithms by inserting known boundary values or by updating the boundary values each time, though this will modify the definition of global state vector.
5.2.3 Adaptive 2-D Kalman Filtering

An adaptive filtering scheme could be devised in which joint estimation and identification would be performed. Dynamical system could be sectionalized by "L" time-invariant models forming "L" Kalman filters each tuned to one of the models, where each model would be sensitive to certain change in the pattern in an image.

5.3 SUMMARY

The objective of this work is to implement one-dimensional Kalman filtering algorithm to image processing, and to extend it to two-dimensions. The performance of these algorithms is measured by computing signal to noise ratio at various stages of estimation. Some of the image enhancement techniques were implemented, which eventually form the pre-processing scheme along with the filtering algorithms.

We solved the problems of excessive computations, and storage associated with the direct extension of 1-D Kalman filtering algorithm to 2-D. Defining the image model by a set of 12 parameters for (2x2) order model based on the linear predictive technique works efficiently for the Kalman filtering algorithm. This set of parameters have strong potential for studies in pattern classification. Interactive image enhancement technique, as discussed in Chapter II, has a wide range of applications for on-line processing.
5.3.1 Image Enhancement

1. Neighbourhood averaging does reduce the noise in an image, but causes blurring of features.

2. We confirmed that gradient computation methods are useful in image sharpening problems.

3. Interactive histogram modification works efficiently for image enhancement. We observed significant improvement in the subjective quality of an image.

5.3.2 One-Dimensional Kalman Filter

1. Development of an algorithm based on "orthogonality principle" leads to the formulation of a one-dimensional optimum Kalman filter.

2. Noise contaminated images processed through the 1-D Kalman filter. Estimated images are free of noise, but the features are blurred. Significant improvement in signal to noise ratio is obtained.

3. Edge-preserving 1-D Kalman filter algorithm, in which the observation vector is augmented with extra structural information, works efficiently in reducing the noise as well as retaining information about features in an estimated image.

5.3.3 Two-Dimensional Kalman Filter

1. We defined the Markov image model on a nonsymmetric half-plane (NSHP) region.

2. Development of reduced update Kalman filter in two-
dimensions. This algorithm requires fewer computations and less storage compared to the direct extension of 1-D set of equations in 2-D.

(3) A new concept of global state vector, local state vector and NSHP filter support is introduced for the implementation of RUKF in 2-D.

(4) Identification of a dynamical model based on the linear predictive technique. A set of 12 parameters for (2x2) order model represents the complete dynamics of image data.

(5) Two-dimensional reduced update Kalman pre-processing, RUKP, scheme incorporating histogram modifier, RUKF and feature (edge) extraction works satisfactorily.

5.4 CONCLUSIONS

Kalman filtering algorithms developed in this work are based on certain assumptions so that the algorithm becomes mathematically tractable and some of the physical processes could be simulated on the computers. The assumptions are:

(1) The measurement noise and the image data is uncorrelated and they are additive

(2) Measurement noise is white in nature

(3) The autocorrelation function is exponentially decaying as the shift in the position of data increases.

(4) The autocorrelation function is separable.

After the implementation of Kalman filtering algorithms for image data, we observed that the estimation based on the above
assumptions is still effective as we obtained significant improvement in signal to noise ratio. More efficient algorithms in 2-D Kalman filtering could be developed, for the spatial-variant data, based on multiple model estimation or adaptive estimation.
*** APPENDICES ***
APPENDIX A

FOUR QUADRANTS FOR 2-D FILTERING

A.1) QUARTER PLANES:

In this Appendix we will discuss some useful sets of admissible regions. In the Fig. (A1) we will identify + with min and - with max., then the factor ++ is min-min phase.

The four quadrants as shown in Fig. (A1) could be defined as follows:

\[ R_{++} = \{m > 0, n > 0\} \]

\[ R_{+-} = \{m \leq 0, n > 0\} \]

\[ R_{-+} = \{m > 0, n < 0\} \]

\[ R_{--} = \{m < 0, n < 0\} \]

A.2) NONSYMMETRIC HALF-PLANE:

A non-symmetric half plane is a region of the form \( \{m > 0, n > 0\} \cup \{m > 0, n < 0\} \) or \( \{m > 0, n < 0\} \cup \{m > 0, n > 0\} \) or their rotations as shown in Fig. (A2).
Fig. A1: Quarter Plane Regions

Fig. A2: Nonsymmetric Half-plane
APPENDIX B

IDENTIFICATION OF DYNAMIC MODEL

B.1) STOCHASTIC PROCESS:

A discrete stochastic linear dynamic system can be represented as,

\[ S_{i+1} = FS_i + GU_i \]  \hspace{1cm} (B1)

\[ r_i = HS_i + U_i \]  \hspace{1cm} (B2)

where

- \( S \) state vector \((nx1)\)
- \( F \) transition matrix \((nxn)\)
- \( G \) input matrix \((nxP)\)
- \( H \) output matrix \((rxn)\)
- \( U_i \) Gaussian noise \((Px1)\)
- \( r_i \) observation \((rx1)\)
- \( U_i \) measurement error \((rx1)\)

we assume that,

\[ E\{U_i\} = 0, \quad E\{U_iU_j^T\} = \sigma_s^2 \delta_{ij} \]

\[ E\{U_i\} = 0, \quad E\{U_iU_j^T\} = \sigma_n^2 \delta_{ij} \]

\[ E\{U_iU_j^T\} = 0, \text{ for all } i \text{ and } j \]
B.2) SCALAR INPUT SCALAR OUTPUT (SISO SYSTEMS):

Let \( r = P = 1 \), and define
\[
D_k = E \{ r_1 r_{1-k} \} \quad (B3)
\]

An expression for \( D_k \) can easily be obtained from (B1) and (B2)
\[
D_k = \begin{cases} 
HPH^T + \sigma_n^2 & , \quad k = 0 \\
HF^k PH^T & , \quad k > 0
\end{cases} \quad (B4)
\]

where
\[
P = E \{ S_1 S_1^T \} = FF^T + GG^T \quad (B5)
\]

Using (B4) for \( k = 1,2,\ldots,n \)
\[
\begin{bmatrix} 
D_1 \\
D_2 \\
\vdots \\
D_n 
\end{bmatrix} = BPH^T \quad (B6)
\]
where,

\[ B = \begin{bmatrix}
    H F \\
    \vdots \\
    \vdots \\
    \vdots \\
    H F^n \\
\end{bmatrix} \]  \hspace{1cm} (B7)

From (B6) we get,

\[ PH^T = B^{-1} \]  \hspace{1cm} (B8)

Now consider (B4) for \( k = n+1 \)

\[ D_{n+1} = H F^{n+1} B^{-1} \]

This can be simplified as,

\[ D_{n+1} = - \sum_{i=1}^{n} F_i D_i \]  \hspace{1cm} (B10)

Generalizing this result we get,
\[ D_{n+j} = - \sum_{i=1}^{n} F_i D_{i+j-1}, \quad j > 1 \]  

Eq. (B11) can also be written as,

\[
\begin{bmatrix}
D_{n+1} \\
\vdots \\
D_{2n}
\end{bmatrix}
= -
\begin{bmatrix}
D_1 & D_2 & \cdots & D_n \\
\vdots & \ddots & \vdots & \vdots \\
D_n & \cdots & D_{2n-1}
\end{bmatrix}
\begin{bmatrix}
F_1 \\
F_2 \\
\vdots \\
F_n
\end{bmatrix}
\]  

(B12)

Inverting we get,

\[
\begin{bmatrix}
F_1 \\
F_2 \\
\vdots \\
F_n
\end{bmatrix}
= -
\begin{bmatrix}
D_1 & D_2 & \cdots & D_n \\
\vdots & \ddots & \vdots & \vdots \\
D_n & \cdots & D_{2n-1}
\end{bmatrix}^{-1}
\begin{bmatrix}
D_{n+1} \\
\vdots \\
D_{2n}
\end{bmatrix}
\]  

(B13)

The reasonable estimate for \( D_k \) is

\[
\hat{D}_k = \frac{1}{N} \sum_{i=k}^{n} r_i r_{i-k}
\]  

(B14)

This leads to the simple way of estimating the parameters of the dynamic model as

\[
\begin{bmatrix}
F_1 \\
F_2 \\
\vdots \\
F_n
\end{bmatrix}
= -
\begin{bmatrix}
\hat{D}_1 & \hat{D}_2 & \cdots & \hat{D}_n \\
\vdots & \ddots & \vdots & \vdots \\
\hat{D}_n & \cdots & \hat{D}_{2n-1}
\end{bmatrix}
\begin{bmatrix}
D_{n+1} \\
\vdots \\
D_{2n}
\end{bmatrix}
\]  

(B15)
APPENDIX C

DEGRADATION MODEL

The degradation process will be modelled as an operator, H, which together with an additive noise u(m,n) operates on an input image S(m,n) to produce the degraded image r(m,n). It is assumed that our knowledge about u(m,n) is limited to information of a statistical nature.

\[ r(m,n) = H S(m,n) + u(m,n) \]  \hspace{1cm} (C1)

There is a set of properties which are satisfied by H. H is linear.
\[ H [k_1 S(m,n) + k_2 S(0,L)] = \]

\[ k_1 H S(m,n) + k_2 H S(0,L) \]

\[(C2)\]

\[ k_1, k_2 \text{ are constants.} \]

The property of additivity,

\[ H [S(m,n) + S(0,L)] = H S(m,n) + H S(0,L) \]

\[(C3)\]

The property of space-invariance,

\[ H S(m-\alpha, j-\beta) = r S(m-\alpha, j-\beta) \]

\[(C4)\]

This definition indicates that the response at any point in the image depends only on the value of the input at that point and not on the position of the point.
APPENDIX D

TEMPLATES FOR EDGES

Edge detecting templates are based on the concept of computing a gradient at a pixel to detect the transition between the gray-levels. Various templates could be designed depending upon the picture and certain vital features in the picture. Templates as shown in Fig. (D1) were tried on noisy and the estimated pictures. The results are certainly encouraging as we can see from Fig. (D2) that Kalman processor works efficiently in reduction of noise and the estimated picture causes significant improvement in quality to be used for further analysis.

\[
\begin{array}{ccc}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{array}
\]

\[
\begin{array}{ccc}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1 \\
\end{array}
\]

\[
\begin{array}{ccc}
0 & -1 & 2 \\
1 & 0 & -1 \\
-2 & 1 & 0 \\
\end{array}
\]

\[
\begin{array}{ccc}
2 & -1 & 0 \\
-1 & 0 & 1 \\
0 & 1 & -2 \\
\end{array}
\]

Fig. D1 Edges Templates
Fig. (D2) Edges of 'DISK-PACK'
REFERENCES


FILENAME: SHARP. PROGRAM TO SHARPEN THE DESIRED IMAGE OF 128x128 SIZE. ROBERT'S CROSS-DIFFERENCE OPERATOR IS USED.

DIMENSION IP(128,128)
DIMENSION NAME(5), IFILE(5)
DATA IBLK, NSIZE=128, NLINE=128.

1000 ACCEPT "EDGES TO BE SHARP(1), OTHERWISE(0) =": IS
IF(IS.EQ.0) GO TO 501
ACCEPT "INTENSITY LEVELS FOR EDGES, (LE. 255) =": IEDG

501 DO 1 I=1,5
1 NAME(I)=IBLK
WRITE(10,2)
2 FORMAT(1X, 'INPUT IMAGE FILENAME:'), Z
READ(11,3)(NAME(I), I=1,5)
3 FORMAT(5A2)
OPEN 1, NAME, LEN=2*NSIZE, REC=NLINE
DO 4 I=1,128
4 READ(1)(IP(I,J), J=1,128)
CLOSE 1
ACCEPT "THRESHOLD INT. LEVEL =", THRS
ISTART=1
DO 5 I=1,127
IEND=I
DO 6 I=1,127
CALL DEBLUR(ISTART, IEND, THRS, IS, IEDG, IP)
6 IEND=IEND+1
7 ISTART=ISTART+1
ACCEPT "WISH TO STORE THE IMAGE? YES(1), NO(0) =": IWISH
IF(IWISH.EQ.0) GO TO 500
DO 7 I=1,5
8 IFILE(I)=IBLK
WRITE(10,8)
8 FORMAT(1X, 'OUTPUT IMAGE FILENAME:'), Z
READ(11,9)(IFILE(I), I=1,5)
9 FORMAT(5A2)
OPEN 2, IFILE, LEN=2*NSIZE, REC=NLINE
DO 10 I=1,128
10 WRITE(2)(IP(I,J), J=1,NSIZE)
CLOSE 2
500 ACCEPT "PROGRAM STATUS: STOP(0), CONTINUE(NE. 0) =": ISTA
IF(ISTA.NE.0) GO TO 1000
STOP
END
FILENAME: SMOOTH, program to smoothen the desired image
by using the simple averaging technique using 3 x 3 matrix.

DIMENSION IP(128, 128).
DIMENSION NAME(10), IFILE(10).
DATA IBLNK /1/.

100 FORMAT( ' ', A, ' input filename : ', Z).
READ(1, 101)(NAME(I), I = 1, 10).

102 FORMAT( ' ', A).
OPEN(1, NAME, LEN = 128, REC = NLINE).
DO 103 I = 1, 128.
READ(1, IP(I, J)), J = 1, 128.
CLOSE 1.

FINDING THE AVERAGE INTENSITY LEVEL OF THE IMAGE.

SUM = 0.
DO 200 J = 1, 128.
DO 200 K = 1, 128.
SUM = SUM + IP(J, K).
END.

GRELV = SUM / 512.
WRITE(10, 201).

201 FORMAT( ' average gray level intensity in the image : ', GRELV )
WRITE(10, 202).

COMPUTATION FOR SMOOTHING.

ISTART = 1.
IEND = 3.
DO 1 II = 1, 128.
ISTAR = 1.
IEND = 3.
DO 2 JJ = 1, 128.
CALL AVERAG(ISTART, IEND, ISTAR, IEND, 1, THRESH, IP).
ISTAR = ISTAR + 1.
IEND = IEND + 1.
END.

IMAGE AFTER SMOOTHING.

ACCEPT 'WISH TO STORE IMAGE? YES(NE, 0), NO(0) -> ', Iwish.
IF(IWISH, EQ, 0) GO TO 500.
DO 104 I = 1, 10.
IFILE(I) = IBLNK.
WRITE(10, 105).

105 FORMAT( ' output filename : ', Z).
READ(11, 106)(IFILE(I), I = 1, 10).

106 FORMAT( ' open 2, IFILE, LEN = 2, NLINE, REC = NLINE).
DO 107 I = 1, 128.
WRITE(2, (1, J)) J = 1, NSIZE.
CLOSE 2.

500 ACCEPT 'PROGRAM STATUS: STOP(0), CONTINUE(NE, 0) -> ', ISTA.
IF(ISTA, NE, 0) GO TO 1000.
STOP.
FILENAME: GRADIENT. PROGRAM DETECTS THE EDGES AND STORES THEM OF THE DESIRED IMAGE USING SOBEL'S OPERATOR.

DIMENSION IPE(128,128), IP(3,128)
DIMENSION NAME(5), IFILE(5)
DATA IBLK/1/

500 DO 1 I=1,5
   NAME(I)=IBLK
   IFILE(I)=IBLK
   WRITE(10,2)
2 FORMAT(1X, 'FILENAME OF PICTURE: ', Z)
   READ(1,3)(IFILE(I), I=1,5)
3 FORMAT(5A2)
   WRITE(10,4)
4 FORMAT(1X, 'FILENAME FOR GRADIENTS IN PICTURE: ', Z)
   READ(1,5)(NAME(I), I=1,5)
   ACCEPT 'THRESHOLD INTENSITY LEVEL= ', ITHR
   ACCEPT 'INT. LEVEL FOR EDGES = ', IEDGE
   OPEN 1, FILE, LEN=256, REC=128
   OPEN 2, NAME, LEN=256, REC=128
   DO 5 I=1,128
      DO 5 J=1,128
5      IPE(I,J)=0

500 DO 6 I1=1,128
   DO 6 L1=1,128
6      READ(1)(IP(LL), LL=L1,L2), L2=1,128)
   IS2=1
   IE2=1
   DO 7 IS2=1,128
      I1=IP(3, IS2)
      I2=IP(3, IE2)
      I3=IP(1, IS2)
      I4=IP(1, IE2)
      I5=IP(2, IS2)
      I6=IP(2, IE2)
      I7=IP(1,1,1)
      SUMX=I1+2*I2+I3-I4+2*I5-I6
      SUMY=I5+2*I6+I7-I4-2*I3-I2
      IFF1=ABS(SUMX)
      IFF2=ABS(SUMY)
      IGRAD=IFF1+IFF2
      IF (IGRAD.GE.ITHR) GO TO 900
   GO TO 300
300 IPE(IS1+1, IS2+1)=IEDGE
300.
300 CONTINUE
   IS2=IS2+1
   IE2=IE2+1
   CONTINUE
   IS1=IS1+1.
   IE1=IE1+1
   BACKSPACE 1
   BACKSPACE 1
5 CONTINUE
   DO 11 I=1,128
11   WRITE(2)(IPE(I,J), J=1,128)
   CLOSE 1
   CLOSE 2
   ACCEPT 'WISH TO COMPUTE GRADS. OF ANOTHER PICTURE? Y-L N-N ' IF(IN. EQ. 1) GO TO 500
}
FILENAME: PROOF. PROGRAM FOR GENERATING THE PROBABILITY
DENSITY FUNCTION FOR THE DESIRED SPECIFICATIONS.
SUBROUTINES READ : -PLTEKS,
SUBROUTINE PROOF(THETA,THETAR,EM, EH, PZ)
DIMENSION PZ(256)
PZ(1)=0

CALCULATION OF INTERMITTANT BREAK POINTS.

PYE=3. 1415926
B=TAN((90-THETA)*PYE)/180. 0
C=TAN((90-THETAR)*PYE)/180. 0
XMAXL=1. 0/(B*EM+1. 0)
YMAXL=B*XMAXL
XMAX2=(EM/(1. 0-EM)+C)/(1. 0/(1. 0-EM)+C)
YMAX2=(1. 0/(1. 0-EM)*XMAX2-(EM/(1. 0-EM))

CALCULATION OF THE SLOPES.

SLOPE1=B
SLOPE2=(EH-YMAX1)/(EM-XMAX1)
SLOPE3=(EH-YMAX2)/(XMAX2-EM)
SLOPE4=C

PROB. DENSITY UPTO THE FIRST BREAK POINT.

I=2
XSTEP=1. 0/255. 0
PZ(I)=SLOPE1*XSTEP
XSTEP=XSTEP+1. 0/255. 0
I=I+1
IF( XSTEP .GT. XMAXL) GO TO 2
GO TO 1

PROB. DENSITY UPTO THE SECOND BREAK POINT.

I=XMAXL+255+0. 5
XSTEP=1. 0/255. 0
PZ(I)=SLOPE2*XSTEP+YMAX1
XSTEP=XSTEP+1. 0/255. 0
I=I+1
IF( XSTEP+XMAX1). GT. EM) GO TO 4
GO TO 3

PROB. DENSITY UPTO THIRD BREAK POINT.

I=EM+255
XSTEP=1. 0/255. 0
PZ(I)=SLOPE3*(XMAX2-EM-XSTEP)+YMAX2
I=I+1
XSTEP=XSTEP+1. 0/255. 0
IF( XSTEP+EM). GT. XMAX2) GO TO 6
GO TO 5

PROB. DENSITY UPTO THE LAST POINT.

I=XMAX2+255
XSTEP=0. 75/255. 0
PZ(I)=SLOPE4*(1. 0-XMAX2-XSTEP)
IP(I,Gt.255) GO TO 8
IF(CXSTEP+XMAX2).GT.1.0) GO TO 8
GO TO 7
8 PZ(256)=0
RETURN
END

******************************************************************************
FILENAME: HISTGM. HISTOGRAM EQUALIZATION OF THE DESIRED
IMAGE, NOT ADAPTIVE.
SUBROUTINES: READ, PLTEKS, POINT.
******************************************************************************
DIMENSION IP(128,128), PR(256), X(255), S(255)
DIMENSION NAME(10), IFILE(10)
DATA IBLK/DD"/"
OPEN 0,"FILE"
1000 ACCEPT" # OF PIXELS PER LINE =",N.SIZE
ACCEPT" # OF SAMPLED LINES =",N.LINE
DO 100 I=1,10
100 NAME(I)=IBLK.
WRITE(10,200)
200 FORMAT(1X, 'INPUT FILENAME:', 2)
READ(11,200)NAME(I), I=1,10
300 FORMAT(10X3)
OPEN 1,NAME,LEN=2*N.SIZE, REC=N.LINE
DO 400 I=1,128
400 READ(1)(IP(I,J), J=1,N.SIZE)
CLOSE 1

NORMALIZING GRAY LEVEL INTENSITIES.

IBIG=IP(1,1)
ISMALL=IP(1,1)
DO 1 I=1,128
1 DO 1 J=1,128
 IF(IP(I,J).GT.IBIG) IBIG=IP(I,J)
 IF(IP(I,J).LT.ISMALL) ISMALL=IP(I,J)
 CONTINUE
IDIFF=IBIG-ISMALL
DO 2 I=1,128
2 DO 2 J=1,128
 A=(255.0(IP(I,J)-ISMALL))/IDIFF
 IP(I,J)=A

CALCULATIONS FOR PROB. DENSITY FUNCTION.

DO 3 K=1,256
PR(K)=0.
3 X(K)=(K-1)
DO 110 N=1,256
NK=0
DO 104 I=1,128
104 DO 104 J=1,128
 IF(IP(I,J).EQ.N-1) NK=NK+1
104 CONTINUE
110 PR(N)=NK/N/16384.0
COMPUTATION OF TRANSFORMED GRAY LEVEL INTENSITIES.

\[ S(i) = PR(i) \]
\[ DO 105 J = 2, 256 \]
\[ S(J) = S(J-1) + PR(J) \]
\[ BIG = S(1) \]
\[ SMALL = S(1) \]
\[ DO 6 J = 2, 256 \]
\[ IF(BIG.GT.S(J)) BIG = S(J) \]
\[ IF(SMALL.GT.S(J)) SMALL = S(J) \]
\[ CONTINUE \]

NORMALIZING OF TRANSFORMED GRAY LEVEL INTENSITIES.

\[ DIFF = BIG - SMALL \]
\[ DO 11 I = 1, 256 \]
\[ S(I) = (S(I) - SMALL) / DIFF * 255.0 \]

TRANSFORMED IMAGE.

\[ DO 8 I = 1, 128 \]
\[ DO 8 J = 1, 128 \]
\[ IP(I, J) = S(IP(I, J)) \]

STORE THE TRANSFORMED IMAG.

TYPE ""
ACCEPT"WISH TO STORE THE TRANSFORMED IMAGE. ?YES(N.E. 0), NO(0) - >", JST
IF(JST.EQ.0) GO TO 700
DO 500 I = 1, 128
IFILE(I) = IBLK
WRITE(10, 600)
\[ IFILE(I) = IBLK \]
\[ WRITE(10, 600) \]
\[ OPEN 2, IFILE, LEN=2*NSIZE, REC=NLINE \]
\[ OPEN 2, IFILE, LEN=2*NSIZE, REC=NLINE \]
\[ WRITE(2)(IP(I, J), J = 1, NSIZE) \]
\[ WRITE(2)(IP(I, J), J = 1, NSIZE) \]
\[ CLOSE 2 \]
\[ CLOSE 2 \]
\[ CALL PLTEKS(X, PR, 255) \]
\[ CALL PLTEKS(X, S, 256) \]
\[ CONTINUE \]
\[ CONTINUE \]
\[ IF(IWT.NE.0) GO TO 1000 \]
\[ IF(IWT.NE.0) GO TO 1000 \]
STOP
END
FILENAME: DISTRIST
PROGRAM TO MODIFY THE HISTOGRAM OF THE DESIRED IMAGE BY USING DIRECT HISTOGRAM SPECIFICATION METHOD.
DESIRED PROBABILITY DENSITY FUNCTION FOR THE GRAY LEVELS IN THE IMAGE IS DEFINED AND GENERATED BY USING 'JOY-STICK' TECHNIQUE.
SUBROUTINES READ, WRITE, FLEKS, POINT.
******************************************************************************************************
******************************************************************************************************
DIMENSION IP(128, 128), PR(256), X(256), S(256), PZ(256), V(256).
DIMENSION X(256), Y(256)
DIMENSION NAME(I), IFILE(I)
DATA IBLNK, / '/
OPEN 0, "TITLE"
ACCEPT "# OF PIXELS PER LINE = " NSIZE
ACCEPT "# OF TOTAL LINES = " NLINE
TYPE " DATA FOR GENERATING PROB. DENSITY."
TYPE "
ACCEPT "LEFT ANGLE OF ROTATION OF PROB. = " 'THETAL
ACCEPT "RIGHT ANGLE OF ROTATION OF PROB. = " 'THETAR
ACCEPT "BREAK-POINT FOR UNITY PROB. DENSITY (BST. 0-1) = " 'EM
EM=1.0
N= NLINE+NSIZE
DO 100 I=1, 1
NAME(I)=IBLNK
WRITE(10, 200)
200 FORMAT (' ', 2X, 'INPUT FILENAME = ', Z)
READ(11, 300) (NAME(I), I=1, 10)
300 FORMAT (10A2)
OPEN 1, NAME, LEN=2*N SIZE, REC=N LINE
DO 400 I=1, 128
400 READ(1) (IP(I, J), J=1, NSIZE)
CLOSE 1

NORMALISING GRAY LEVEL INTENSITIES.

IBIG=IP(1, 1)
ISmall=IP(1, 1)
DO 1 I=1, 128
DO 1 J=1, 128
IF(IP(I, J).GT. IBIG) IBIG=IP(I, J)
IF(IP(I, J).LT. ISmall) ISmall=IP(I, J)
1 CONTINUE
IDiff=IBIG-ISmall
DO 2 I=1, 128
DO 2 J=1, 128
IP(I, J)=(255.0+IP(I, J)-ISmall)/IDiff
2 CONTINUE

CALCULATIONS FOR PROBABILITY DENSITY FUNCTION.

DO 3 K=1, 256
X(K)=(K-1)
NK=0.0
KL=K-1
DO 4 I=1, 128
DO 4 J=1, 128
IF(IP(I, J).EQ. KL) NK=NK+1
4 CONTINUE
PR(K)=NK/16384.0

COMPUTATION OF TRANSFORMED GRAY LEVEL INTENSITIES.
NORMALISING OF THE TRANSFORMED GRAY LEVEL INTENSITIES.

DIFF = BIG - SMALL
DO 11 I = 1, 256
   S(I) = (S(I) - SMALL) / DIFF * 255.0

GENERATION OF THE DESIRED PROB. DENSITY FUNCTION.

CALL PRODF(THETAL, THETAR, EM, EH, PZ)

NORMALISING OF GRAY LEVELS STAGE TWO.

V(1) = PZ(1)
DO 401 I = 2, 256
   V(I) = V(I-1) + PZ(I)
   BIG = v(I)
   SMALL = v(I)
   DO 350 I = 2, 256
      IF(v(I) .GT. BIG) BIG = v(I)
      IF(v(I) .LT. SMALL) SMALL = v(I)
   CONTINUE
   DIFF = BIG - SMALL
   DO 352 I = 1, 256
      V(I) = (V(I) - SMALL) / DIFF * 255.0
   352

TRANSFORMED IMAGE STAGE TWO.

DO 355 I = 1, NLINE
   DO 353 J = 1, NSIZE
      IP(I, J) = (IP(I, J) + 1)
   353
   DO 354 I = 1, 256
      IF(IP(I, J) .EQ. IV(L)) GO TO 356
   354
   356
   IP(I, J) = L - 1
   GO TO 353

CONTINUE

ACCEPT "WISH TO STORE THE TRANSFORMED IMAGE? YES/NO. NO(0)-->", ISTE
IF (ISTE .EQ. 0) GO TO 700

DO 500 I = L, 10
   IFFILE(I) = IELNK
   WRITE(10, 600)
   600   FORMAT(‘IX, ’TRANSFORMED IMAGE FILENAME: ’, Z)
   READ(11, 651)(IFFILE(I), I = 1, 10)
   651   FORMAT(10A2)
   OPEN 2, IFFILE/LEN=2*NSIZE, REC=NLINE
   DO 652 I = 1, 128
   652   WRITE(2)(IP(I, J), J = 1, NSIZE)
   CLOSE 2
   CALL PLTEK(X, PR, 256)
   CALL PLTEK(X, S, 256)
   CALL PLTEK(X, PZ, 256)
   CALL PLTEK(X, IV, 256)

CLOSE 1
ACCEPT "DO YOU WANT TO CONTINUE? Y/N: "
FILENAME: NOISY PROGRAM TO ADD GAUSSIAN NOISE TO THE DESIRED IMAGE.
SUBROUTINES READ, := GAUSS, RANDB

DIMENSION IP(64,128), NAME(5), IFILE(5)
M1=128
M2=128
CT=FLOAT(M1+M2)
WRITE(10,2)

10 FORMAT(1X, 'FILENAME OF ORIGINAL IMAGE: ',Z)
READ(11,3)(NAME(I), I=1, 5)
WRITE(10,4)

4 FORMAT(1X, 'FILENAME FOR NOISY IMAGE: ',Z)
READ(11,3)(IFILE(I), I=1, 5)
OPEN 1, NAME, LEN=255, REC=1 & 3
OPEN 2, IFILE, LEN=255, REC=128
SUM=0.0
Do 5 I=1, 2
Do 6 J=1, 64

5 READ(1,5)(IP(J,K), K=1, 128)
Do 7 J=1, 64
Do 7 K=1, 128
SUM=SUM+FLOAT(IP(J,K))
CONTINUE

A=SUM/CT
REWIND 1
SUM=0.0
Do 8 I=1, 2
Do 9 J=1, 64

8 READ(1,5)(IP(J,K), K=1, 128)
Do 10 J=1, 64
Do 10 K=1, 128
Y=FLOAT(IP(J,K))

9 SUM=SUM+(Y-AVE)**2
CONTINUE

VARSIG=SUM/CT
TYPE" "
TYPE"MEAN OF ORIGINAL IMAGE =", AVE
TYPE"VAR. OF SIGNAL IN IMAGE =", VARSIG
TYPE" 
ACCEPT"SIGNAL TO NOISE RATIO =", SNR
VARNOI=VARSIG/(10.**((SNR/10.))
SD=ROUND(VARNOI)
TYPE"VAR. OF NOISE < FOR GIVEN SNR =", VARNOI
REWIND 1
IX=1
AM=0
KFG=2548
KP=10
IT=1027
KNT=1
Do 15 I=1, 2
Do 16 J=1, 64

15 READ(1,5)(IP(J,K), K=1, 128)
Do 17 J=1, 64
Do 17 K=1, 128

16 CALL GAUSS(IX, IT, SD, AM, PN)
IF AM .LT. 0.0 OR PN .GT. 255.0 GO TO 55
KNT=KNT+1
IF(KNT.GT.KFG) GO TO 19
KP=KP-1
IT=2*KP+3
CONTINUE
IP(J,K)=IP(J,K)+RN
DO 15 J=1,64
15 WRITE(2,IP(J,K),K=1.128)
ACCEPT"WISH TO ADD NOISE IN ANOTHER PICTURE?Y-1,N-0:-",IN
IF(IN.EQ.1) GO TO 100
STOP
END

**********************************************************************
FILENAME:-ODKF. ONE-DIMENSIONAL KALMAN ESTIMATION
TECHNIQUE. HORIZONTAL CORR. IS CONSIDERED FOR THE
DEFINITION OF STATE VECTOR MODEL.
INITIAL CONDITIONS:
XHAT(0)=0.
COVAR(0)=VAR. OF SIGNAL.

SUBROUTINES RECOMMENDED: GAUSS, RANDB, MEAN, VAR, CORCO

**********************************************************************
DIMENSION IX(16,128), IV(16,128)
DIMENSION IP(128), IF(128), IFI(128)
DIMENSION NAME(5), IFIL(5), IFIL(5)
REAL IXHAT(128), IX(128), IV(128), AMP(128), COVAR(128)
M1=128
M2=128
WRITE(1,10)
10 FORMAT(1X, 'FILENAME OF THE ORIGINAL IMAGE: ',Z)
READ(1,11)(NAME(I), I=1,5)
WRITE(10,12)
12 FORMAT(1X, 'FILENAME OF NOISY IMAGE: ',Z)
READ(11,11)(IFIL(1), I=1,5)
WRITE(10,13)
13 FORMAT(1X, 'FILENAME OF ESTIMATED IMAGE: ',Z)
READ(11,11)(IFIL(1), I=1,5)

OPEN 1, NAME, LEN=256, REC=128
OPEN 2, IFIL, LEN=256, REC=128
OPEN 3, IFIL, LEN=256, REC=128
SUMX=0.
DO 1 I=1,8
   DO 2 J=1,16
      READ(1)(IX(J,K), K=1,M2)
   DO 1 J=1,16
   DO 1 K=1,128
1 SUMX=SUMX+IX(J,K)
   CT=M1*M2
   SIMEANX=SUMX/CT
   WRITE(2,*) IX(J,K), I=1,8, J=1,16, K=1,M2
WRITE(10,14)
14 FORMAT(1X, 'FILENAME OF NOISY IMAGE: ',Z)
READ(11,11)(IFIL(1), I=1,5)
WRITE(10,15)
15 FORMAT(1X, 'FILENAME OF ESTIMATED IMAGE: ',Z)
READ(11,11)(IFIL(1), I=1,5)

C

TYPE"MEAN OF THE ORIGINAL IMAGE:=". SMAX
REWIND 1
SUMXX=0.
DO 3 I=1,8
DO 3 J=1,16
31 READ(1)(IX(J,K), K=1,M2)
DO 3 J=1,16
DO 3 K=1,M2
X=IX(J,K)
3 SUMXX=SUMXX+(X-SMAX)**2
VARXX=SUMXX/CT

C

TYPE"VARIANCE OF THE SIGNAL IN IMAGE:=". VARSX
REWIND 1
ACCEPT "SIGNAL TO NOISE RATIO FOR NOISY IMAGE=", SNR
VARNOI=VARSX/(10**((SNR/10)))
SD=SQR(T(VARNOI))
AM=0.0
IS=1
KP=9
IT=515

...  
DO 4 I=1,8
DO 4 J=1,16
4 READ(1)(IX(J,K), K=1,M2)
DO 5 J=1,16
DO 5 K=1,128
56 CALL GAUSS(IS, IT, SD, AM, V)
IF(V.LT.0. OR V.GT.255) GO TO 56
5 IY(J,K)=IX(J,K)+V
KP=KP+1
IT=IT+1
4 DO 4 J=1,15
WRITE(2)(IY(J,K), K=1,M2)
REWIND 1
REWIND 2
VARN=SD**2

C

DO 9 I=1,128
READ(1)(IPK(K), K=1, M2)
READ(2)(IPN(K), K=1, M2)
DO 301 L=1,128
IXX(L)=IPK(L)
IYY(L)=IPN(L)
301 CALL MEAN(IXX(1:2, SMAXXX)
CALL MEAN(IYY, M2, SMAXYY)
CALL VAR(IXX, M2, SMAXXX, VARSXX)
CALL VAR(IYY, M2, SMAXXX, VARSYY)
CALL CORCO(IXX, IYY, SMAXXX, SMAXYY, VARSXX, VARSYY, M2, HRCO)
IXHAT(1)=0.
COVAR(1)=VARSX
DO .5 K=2, M2
   K1=K-1
   AMP(K1)=(HORCO*COVAR(K1))/COVAR(K1)+VARN
   IXHAT(K)=(HORCO-AMP(K1))*IXHAT(K1)+AMP(K1)*IVY(K1)
   PARTL=((HORCO-AMP(K1)**2)*COVAR(K1)+1.0-HORCO**2)*VARSX
   PART2=(AMP(K1)**2)*VARN
   COVAR(K)=PART1+PART2
   CONTINUE
   DO 302 L=1, M2
   IPE(L)=IXHAT(L)
   WRITE(3)<IPE(J2), J2=1, M2>
   302
   WRITE(3)<IPE(J2), J2=1, M2>
   WRITE(3)<IPE(J2), J2=1, M2>
   WRITE(3)<IPE(J2), J2=1, M2>

C

CLOSE 3
CLOSE 2
CLOSE 1

ACCEPT "WISH TO ESTIMATE ANOTHER IMAGE FROM NOISY IMAGE?Y-1, N-0:-"", I,
IF(W.NE.0) GO TO 100
STOP
END

FILENAME: -EDGETEMP. PROGRAM TO USE VARIOUS
EDGE TEMPLATES TO MATCH SOME FEATURES OF THE
GIVEN IMAGE. THIS TECHNIQUE IS PROBLEM ORIENTED
AS TEMPLATES COULD BE DESIGNED TO MATCH CERTAIN
FEATURES IN THE GIVEN IMAGE.

SUBROUTINES READ: -VECTOR1:

DIMENSION IPE(128,128), NAME(5), IFILE(5)
DIMENSION IP3(128), W3(3,3), W2(3,3)
DATA IBLK/, /
M1=128
M2=128
100
DO 1 I=1, 5
   NAME(I)=IBLK
   WRITE(10,2)
1
   IFILE(I)=IBLK
   WRITE(10,2)
2 FORMAT(1X, 'FILENAME OF PICTURE:-', Z)
   READ(11,3)<NAME(I), I=1, 5>
   FORMAT(5A2)
3 FORMAT(1X, 'FILENAME FOR EDGES:-', Z)
   READ(11,200)<IFILE(I), I=1, 5>
   FORMAT(5A2)
4 FORMAT(1X, 'FILENAME FOR EDGES:-', Z)
   OPEN 1, NAME, LEN=255, REC=128
   OPEN 2, IFILE, LEN=255, REC=128
   CALL 'VECTOR1(W3,W2)
   ACCEPT"THRESHOLD INTENSITY LEVEL="', THRSH
   DO 555 J=1, 128
   DO 555 I=1, 128
   555 IPE(I,J)=0
   IS1=1
   IE1=3
   DO 6 I=1, 128
   DO 6 5 11=1, 128
   6550 I=1, 3
800    READ(1)X(I,J),J=1,128
     IS2=1
     IE2=3
     DO 10 I4=1,128
     SUMH=0
     SUMW=0
     K1=1
     DO 7 I2=1,3
     K2=1
     DO 8 I3=IS2,IE2
     SUMH1=SUMH+W1(KL,K2)*X(I2,I3)
     SUMW2=SUMW2+W2(KL,K2)*X(I2,I3)
     K2=K2+1
     8   CONTINUE
     K1=K1+1
     CONTINUE
     IF1=ABS(SUMH1)
     IF2=ABS(SUMW2)
     IGRAD=IF1+IF2
     IF(IGRAD.GE.THRESH) GO TO 310
     GO TO 330
     310   IPE(IS1+1,IE2+1)=IGRAD
     320   IS2=IS2+1
     IE2=IE2+1
     CONTINUE
     BACKSPACE 1
     BACKSPACE 1
     IS1=IS1+1
     IE1=IE1+1
     CONTINUE
     DO 1000 L1=1,MI
     1000  WRITE(2)*IPE(L1,L2),L2=1,128
     CLOSE 1
     CLOSE 2
     ACCEPT"WANT TO DETECT GRADS. OF ANOTHER PICTURE: ":IW
     IF(IW.EQ.4) GO TO 100
     STOP
     END
FILENAME: LSPI. THIS PROGRAM IS FOR ON-LINE IDENTIFICATION OF THE PARAMETERS OF THE DYNAMIC SYSTEM FOR KALMAN FILTER. THE PARAMETERS OBTAINED ARE BASED ON THE METHOD OF MINIMIZATION OF MEAN SQUARED ERROR USING LINEAR PREDICTIVE APPROACH.

SUBROUTINES REQD.: - MATINV.

DIMENSION IPIC(128, 128), NAME(5), IFILE(5), LL(12), MM(12)
REAL MODL(16)
REAL P(12, 12), S1(12), Q(12), PINV(12, 12)
INTEGER MODISZ(269)
DATA IBLK/"/
ACCEPT"# OF COLUMNS IN IMAGE =", M1
ACCEPT"# OF ROWS IN IMAGE =", M2
ACCEPT FIRST PIXEL ON A SCAN LINE (1<NMIN<128) =", MMIN
ACCEPT LAST PIXEL ON A SCAN LINE (6<NMAX<128) =", MMAX
ACCEPT"# OF FIRST SCAN LINE (1<NMIN<128) =", NM1N
ACCEPT"# OF LAST SCAN LINE (3<NMAX<128) =", NM1X
TYPE "
TYPE DATA FOR SPECIFYING THE PARAMETERS OF MODEL"
TYPE "
IQ=2
IK=2
IP=2
TYPE PARAMETERS OF THE MODEL =", IQ, IK, IP
DO 1 I=1, 5
IFILE(I)=IBLK
1 NAME(I)=IBLK
WRITE(10, 2)
2 FORMAT(1X, 'FILENAME OF IMAGE:-', Z)
READ(1L3)(NAME(I), I=1, 5)
3 FORMAT(5A2)
WRITE(10, 3)
302 FORMAT(1X, 'FILENAME FOR MODEL PARAMETERS:-', Z)
READ(1L303)(IFILE(I), I=1, 5)
303 FORMAT(5A2)

SET UP PROGRAM CONSTANTS AND INITIALIZATIONS

IP1=IP+1
ISZA1=IK+IQ+1
NRPP=IP+ISZA1+IQ+1
NRP=NRPP-1
ITEM=IP+ISZA1+IQ+128+129
MSIZE=MMAX-MMIN+1
NSIZE=NMAX-NMIN+1
MMIN=MMIN+IK
MMAX=MMAX-IQ
NM1N=NMIN+IP
IMPTS=(MMAX-MMIN+1)*(NM1X-NMIN+1)
OPEN 1, NAME, LEN=256, REC=128
OPEN 2, IFILE, LEN=64, REC=1
DO 5 I=1, 128
5 READ(1)(IFIC(K), K=1, M2)
CLOSE 1
DO 10 I=1, NRP
Q(I)=0.
MODL(I)=0.
DO 10 J=1, NRP
PK(I, J)=0.

WARNING: USE AT YOUR OWN RISK.
FILL UP THE POINTER ARRAY

DO 20 IPM=1,ITEMP
IPMM=IPM-129
MODISZ(IPM)=MOD(IPMM,ISZAI)
IF(MODISZ(IPM),EQ.0) MODISZ(IPM)=ISZAI
CONTINUE

COMPUTATION OF MODEL COEFFICIENTS

DO 30 N=NMIN, NMAX
DO 30 M=MMIN, NMAX
SMN=FLOAT(IPIC(N,M))

FILL UP LOCAL STATE VECTOR S1

DO 40 I=1,NRP
IL=I-1
L=IP-IL/ISZAI
K=MODISZ(I+129)
K=I0+1-K
MMK=M-K
MNLM=N-L
S1(I)=FLOAT(IPIC(NML, MMK))

COMPUTATION OF CORRELATION VECTOR AND COEFF. MATRIX

DO 50 I=1,NRP
G(I)=G(I)+S1(I)*SMN
DO 50 J=1,NRP
P(I,J)=P(I,J)+S1(I)*S1(J)
PINV(J,I)=P(I,J)
CONTINUE

CALL MATINV(PINV,12, D, LL, MM)
DO 60 I=1,NRP
DO 60 J=1,NRP
MODL(I)=MODL(I)+PINV(I,J)*G(J)
CONTINUE

COMPUTATION OF PLANT VARIANCE

DO 70 N=NMIN, NMAX
DO 70 M=MMIN, NMAX
SMN=FLOAT(IPIC(N,M))
SUM=0
DO 80 I=1,NRP
IL=I-1
L=IP-IL/ISZAI
K=MODISZ(I+129)
K=I0+1-K
MMK=M-K
MNLM=N-L
SUM=SUM+MODL(I)*FLOAT(IPIC(NML, MMK))
CONTINUE
VPN=VPN+(SMN-SUM)**2
CONTINUE
VPN=VPN/IMGPTS
MODL(NRPP)=VPN
DO 200 I=14,16
MODL(I)=0.0
WRITE(2) (MODL(I), I=1,16)
100 FORMAT(4(E15.6,'/'))
ACCEPT "WISH TO CONTINUE?YES(NE.0), NO(0) :-)", IW
IF (IW.NE.0) GO TO 1000
STOP
END
FILENAME: RU3F. THIS PROGRAM FILTERS THE IMAGE WHICH IS
CONTAMINATED BY WHITE NOISE USING THE PREDEIGNED STEADY
STATE GAIN VALUES. RU3F COMPUTE MMSE ESTIMATES
SUBJECT TO THE CONSTRAINT OF UPDATING A FIXED REGAIN
OF STATE VECTOR CALLED 'LOCAL STATE VECTOR'.

DIMENSION IPIN(128,128), NAME(5), IFILE(5), NFILE(5), JFILE(5)
REAL GAIN(16), MODL(16)
INTEGER MODDISZ(256)
DATA IBLK / '/
DATA IQ, IK, IP/2, 2, 2/
DATA LIQ, LIK, LIP/3, 3, 2/
DATA IQP, IKP/7, 7/
DATA IUPD, IUPOX/10, 20/
ACCEPT "VARIANCE OF IMAGE :", VARSIG
ACCEPT "VARIANCE OF NOISE :", VARNO
ACCEPT "PLANT VARIANCE :", SGH
ACCEPT "SCALING FACTOR(SGH) :", SCLSGH
DO 1 I=1, 5
NAME(I)=IBLK
NFILE(I)=IBLK
IFILE(I)=IBLK
JFILE(I)=IBLK
WRITE(10, 2)
FORMAT (1X, 'FILENAME OF THE NOISY IMAGE :', Z)
READ (11, 3) NAME(I), I=1, 5
FORMAT(5A2)
WRITE(10, 4)
FORMAT(1X, 'FILENAME OF MODEL PARAMETERS :', Z)
READ (11, 3) IFILE(I), I=1, 5
WRITE(10, 2000)
FORMAT(1X, 'FILENAME OF GAIN, COEFFS. :', Z)
READ (11, 3) NFILE(I), I=1, 5
WRITE(10, 5)
FORMAT(1X, 'FILENAME FOR FILTERED IMAGE :', Z)
READ (11, 3) JFILE(I), I=1, 5
OPEN 1, NAME, LEN=255, REC=128
OPEN 2, IFILE, LEN=64, REC=1
OPEN 3, JFILE, LEN=256, REC=128
OPEN 4, NFILE, LEN=72, REC=1

DO 5 I=1, 128
READ (1) (IPN(I, J), J=1, 128)
READ (2) (MODL(I), I=1, 16)
READ (4) (GAIN(I), I=1, 16)

SET UP PROGRAM CONSTANTS
IP1=IP+1
LIP1=LIP+1
ISZ1=IK+IQ
ISZ1L=ISZ1+1
LISZ1=LIZ1+LIQ+1
NRPP=IP*ISZ1+IQ+1
NR=NRPP-1
LRPP=LIP+ISZ1L+LIQ+1
MIN=LIQ+1
MINX=128-LIK
MIN=LIP+1
FILL UP THE POINTER ARRAY

DO 110 IPM=1,ITEMP
IPMH=IPM-129
MODISZ(IPM)=MOD(IPMH,ISZA1)
IF(MODISZ(IPM).EQ.0) MODISZ(IPM)=ISZA1
110 CONTINUE

LOOP ON RONS

DO 6000 h=NMIN,NMAX
NO=N-LIP
NL=N
NO100=NO+100
NL100=NL+100
N128=(N-1)

LOOP ON COLUMNS

DO 5000 M=MMIN,MMAX
MO=N-LIQ
ML=M+LIK
MO100=MO+100
ML100=ML+100
MP1=M+1
YMN=FLOAT(IPN(N128,M))

PREDICTION

SUM=0.
DO 4100 I=1,NRP
IL=I-I
L=IF(IL.ISZA1.
K=MODISZ(I+1)+129)
K=IQ+1-K
MMK=M-K
NML=M-L

IF(NMK.LT.MMIN. OR. MMK.GT.MMAX. OR. NML.LT.MMIN) GO TO 4100
SUM=SUM+MODL(I)*FLOAT(IPN(NML,MMK))
4100 CONTINUE
IPN(N128,M)=SUM+0.5
PRDEFR=YNM-FLOAT(IPN(N128,M))

KALMAN FILTER EQUATIONS

DO 4220 L100=NO100,NL100
L=L100-100
LM128=(L-1)
LMA1=(L-NO)*LISZA1+1
DO 4210 K100=M0100,ML100
K=K100-100
IF(L.EQ.NL. AND. K.GT.M) GO TO 4210
IF(K<.LT.MMIN. OR. K.GT.MMAX. OR. L.LT.MMIN) GO TO 4210
KMMN=K-MO
IARG=LMA1+KMMN
IPN(LM128,K)=FLOAT(IPN(LM128,K))++GAIN(IARG)*PRDEFR
4210 CONTINUE
4220 CONTINUE
5000 CONTINUE
6000 CONTINUE

END OF LOOP ON ROWS

4000 DO 1000 I=1,128
4000 WRITE(3)(IPN(I,J),J=1,128)
CLOSE 1
CLOSE 2
CLOSE 3
STOP
END
FILENAME: RUKD. THIS PROGRAM DESIGNS THE KALMAN FILTER AND COMPUTES THE STEADY STATE GAIN MATRIX FOR THE NON-SYMMETRIC HALF-PLANE (NSHP) LOCAL STATE VECTOR USING ---- REDUCED UPDATE PROCEDURES

SUBROUTINES REQ'D: FIND, FINDA, FINDB, FINDE, FINDF

DIMENSION MODISZ(269), MODLSZ(269)
INTEGER 2, INDEXA(5), INDEXB(5), INDEXC(5), INDEXD(5)
REAL GAIN(28), MODL(13), YSA1(20, 20), A(4590), CWK(2025), AO(4590)
DATA IQ, IK, IP/2, 2, 2/
DATA LIQ,LIK,LIP/3, 3, 2/
DATA IUPD, IUPDX/10, 20/
DATA VARSIG, VARI0, SCLSGW/1811.0, 907.0, 1.0/
DATA SGW/100.00/
DATA ISE, JSE/10, 20/
DATA MODL(1), MODL(2)/-0.529667E-01, -0.554307E-01/
DATA MODL(3), MODL(4)/0.269529E00, 0.391073E-01/
DATA MODL(5), MODL(6)/0.298016E-02, -0.964986E-02/
DATA MODL(7), MODL(8)/-0.110690E00, 0.422228E00/
DATA MODL(9), MODL(10)/0.433340E-01, 0.336155E-01/
DATA MODL(11), MODL(12)/0.731367E-01, 0.346117E00/
DATA MODL(13)/0.357840E07/
COMMON /1/ KQ, IQP, K129, IGPSZ, IP1SG, IPP, MODISP(269), MODIPI(269), J1SIZ

I6
IKP=7
WRITE(6,1)
FORMAT(1, 'MODEL SUPPORT PARAMETERS(R++)')
WRITE(6,2) IQ, IK, IP
WRITE(6,3)
FORMAT(1, 'LOCAL STATE VECTOR PARAMETERS(L++)')
WRITE(6,2) LIQ, LK, LIP
WRITE(6,4)
FORMAT(1, 'COVARIANCE UPDATE PARAMETERS(T++)')
WRITE(6,5) IQP, IKP
WRITE(6,6)
FORMAT(1, 'STEADY-STATE PARAMETERS')
WRITE(6,7) IUPD, IUPDX
WRITE(6,8)
FORMAT(1, 'SIGNAL AND NOISE PARAMETERS')
WRITE(6,999) (J, MODL(J), J=1,12)
999 FORMAT(1, 'MODL(12) = E18.6')

PROGRAM CONSTANTS AND INITIALIZATIONS

NMAX=IUPD
MMAX=IUPDX
MINQ=-IQ
IUPDXK=IUPDX+1K
ISZAI=ISZA+1
LISZA=LIK+1 TN
ISZAP=IKP+IQP
ISZA1P=ISZAP+1
IP1=IP+1
LIP1=LIP+1
IPP=LIP1
LIP2=LIP1
IP1SC=IPP*IPP
IPE1=LIP1+128
IGPSZ=ISZA1P*IP1SG
ISZL=(128+ISZAP)*IGPSZ
ISZS=ISZA1P*IGPSZ
IP2=IPP+1
KO=IQP+1
K129=129-KO
KK=IUPDX+IKP+IQP
NRPP=IP*ISZA1+IQ+1
LNRPP=LIP*LISZA1+LIQ+1
NRPP=NRPP-1
NMC=IUPDX
IUPDX2=IUPDX/2
ITEMP=IP*ISZA1+IQ+128+129
IUPDSZ=(IUPDX-1)*IGPSZ
IP1IPD=IUPD+IP128
IP1JSE=JSE+IP128
IPLIP1=IPP-LIP1
LIP2M1=LIP2-LIP1
1ULIP=IUPDX-LIP
1ULIP=IUPDX+LIP1
IP1IP1=IPP*(IPP-1)
J=1

DO 100 I=IPP,IP1SG,IPP
INDEXA(J)=I
INDEXB(J)=IP1IP1+J
J=J+1
100 CONTINUE
IPTR1=IPP
ISTART=1
IBIGIN=1
DO 110 I=1,LNRPP
GAIN(I)=0.0

FILL UP THE POINTER ARRAY
DO 120 IPM=1,ITEMP
IPM=IPM-129
MODISZ(IPM)=MOD(IPM,ISZA1)
IF(MODISZ(IPM).EQ.0)MODISZ(IPM)=ISZA1
MODLSZ(IPM)=MOD(IPM,LISZA1)
IF(MODLSZ(IPM).EQ.0)MODLSZ(IPM)=LISZA1
MODISP(IPM)=MOD(IPM,ISZA1P)
IF(MODISP(IPM).EQ.0)MODISP(IPM)=ISZA1P
MODIP1(IPM)=MOD(IPM,IPP)
IF(MODIP1(IPM).EQ.0)MODIP1(IPM)=IPP
120 CONTINUE

MODIX=MODISP(IUPDX+128)
MODL(NRPP)=0.
CCH: CCH=CCH+CHC

C
C
FILL UP THE ERROR COVARIANCE ARRAY.

DO 130 I=1,4590
  A(I)=0.

130
  DO 150 I=1,4590,IGPSZ
  DO 140 J=1,IP150,IP2
     IARG=I+J-1
     A(IARG)=VARNO1
     ANI(IARG)=VARNO1
     CONTINUE

150 CONTINUE

LOOP OVER THE ROWS OF IMAGE(STEADY-STATE REGION)

DO 6000 N=1,NMAX

SET UP ROW CONSTANTS

WRITE(6,9995) N
9995 FORMAT(1A,','LOOP N=',13,
N0=N-LIP
NL=N+LIP2M1
NLP=N+1LIP1
MOP=NO
MO100=NO+100
NL100=NL+100
MOP100=MOP+100
NLP100=NLP+100
IAB=1
NP1=N+1
IP1N=N+IP128

DO 5110 I=1,ISZS
CW(I)=A(I)
5110 CONTINUE
IA=ISZS+1

LOOP OVER THE COLUMNS

DO 5000 M=1,MMAX

SET UP COLUMN CONSTANTS

MQ=M-LIQ
ML=M+LIK
MOP=M-1QP
MLP=M+IKP
MO100=M0+100
ML100=ML+100
MOP100=MOP+100
MOP100=MOP+100
MP1=M+1

IF(N.EQ.NMAX.ANU.M.GT.IUPDX2) GO TO 6000
XF(M.EQ.1) GO TO 4130
K1=M-1
IR=MODISP(KI+129)
IARG=(IR-1)*ITAPS+1
C
DO 4110 I=IARG,IBUT
IABI=IABI+1
A(IABI)=CWK(I)
4110 CONTINUE
IARG1=IARG-1
IBUT=IA+IGPSZ-1
DO 4120 I=IA,IBUT
IARG1=IARG1+1
CWK(IARG1)=A(I)
4120 CONTINUE
IA=IA+IGPSZ
IAB=IAB+IGRSZ
4130 CONTINUE
C
ERROR COVARIANCE PREDICTION
C
DO 4230 L100=MOP100,MEP100
L=L100-100
IP1L=L+IP128
DO 4220 K100=MOP100,MLP100
K=K100-100
IF(K.LT.HNIG.OR.K.GT.IUPDXK).GO TO 4220
IF(L.LT.EQ.N.AND.K.GE.M.AND.K.LE.IUPDX).GO TO 4220
SUM=0.
CALL FIND(M,N,K,L,IF1,IP1N,IP1L)
DO 4210 IARG=1,NKP
KO=MCISZ(IARG+129)
KO=KO+1-KO
IL=IARG-1
LP=IL-IL/ISZAL
MMO=M-KO
NMF=N-LP
IP1NMP=NMP+IP128
CALL FIND(MMO,NMF,K,L,IF2,IP1NMP,IP1L)
SUM=SUM+MODL(IARG)*CWK(IF2)
4210 CONTINUE
CWK(IF1)=SUM
4220 CONTINUE
4230 CONTINUE
C
ERROR COVARIANCE PREDICTION AT(M,N)
C
SUM=0.
DO 4310 IARG=1,NKP
K=MODISZ(IARG+129)
K=K+1-K
IL=IARG-1
L=IL-IL/ISZAL
MMK=M-K
NML=N-L
IP1NML=NML+IP128
CALL FIND(M,N,MMK,NML,IF2,IP1N,IP1NML)
SUM=SUM+MODL(IARG)*CWK(IF2)
4310 CONTINUE
CALL FIND(M,N,M,N,IFA1,IP1N,IP1N)
CWK(IFA1)=SUM+SGW
C
SYB1=CWK(IFA1)
GAINFC=1./(SYB1+SGV)
DO 4420 L100=ND100,NL100
L=L100-100
IP1L=L+IP128
LMNO=L-NO
LMA1=LMNO*LIBZAI+1
DO 4410 K100=M0100,ML100
K=K+00-100
IF(K.LE.KND1 OR.K.GT.IUDPXX) GO TO 4410
IF(L.EQ.N AND.K.GE.MP1.AND.K.LE.IUDPXX) GO TO 4410
CALL FINDU(M,N,K,L,IF2,IP1N,IP1L)
KMM0=K-MO
I=LMA1+KMM0
GAIN(I)=CWK(IF2)*GAINFO
44.0 CONTINUE
4420 CONTINUE

ERRD COVARIANCE UPDATE

DO 4580 L100=NOP100,NLP100
L=L100-100
IP1L=L+IP128
LN=MODIP1(IP1L)
LNZA1=(LN-1)*LIBZAI
IFINDA=1
IFINDB=1
DO 4560 K100=M0P100,MLP100
K=K100-100
IF(K.LE.KMIQ OR.K.GT.IUDPXX) GO TO 4560
IF(L.EQ.N AND.K.GE.MP1.AND.K.LE.IUDPXX) GO TO 4560
KM=K-MO
CALL FINDU(M,N,K,L,IF2,IP1N,IP1L)
GO TO 4550
4541 CONTINUE
C IF(K.LT.M) GO TO 4544
C
C IF(IFINDB.EQ.1) GO TO 4542
IF2=IRPPP1+(K-M)*IP1SQ
GO TO 4550
4542 CALL FINDU(M,N,K,L,IF2,IP1N,IP1L,IRPPP1,IFINDB)
GO TO 4550
C
C 4544 IF(IFINDA.EQ.1) GO TO 4548
IF(K.GE.KD) GO TO 4546
IF2=(K+IQP-1)*I6PSZ+(M-K)*IP1SQ+IBPP
GO TO 4550
4546 IF2=(MODISP(K+129)-1)*I6PSZ+(M-K)*IP1SQ+IBPP
GO TO 4550
4548 CALL FINDU(M,N,K,L,IF2,IP1N,IP1L,IBPP,IFINDA)
C
C 4550 CONTINUE
DO 4540 J100=NO100,NL100
J=J100-100
IP1J=J+IP128
JMNO=J-NO
JN=MODIP1(IP1J)
JNZAI=(JN-1)*LIBZAI
IF(INDF=1)
DO 4520 I100=M100, ML100
I=I00-100
IF(I.LE.MINQ.OR.I.GT.IUPDXK) GO TO 4520
IF(J.EQ.N.AND.I.LE.MPI1.AND.I.LE.IUPDX) GO TO 4520
IF(K.LT.HQ.OR.K.GT.HL) GO TO 4502
IM=I-MO

MAPJ=JNZAI+1M
MAPK=LNZAI+Km
IF(MAPKL.LT.MAPIJ) GO TO 4520
CONTINUE
GO TO 4512
CONTINUE
IF(K.LT.I) GO TO 4508

IF.( FINDF.EQ.1) GO TO 4506
IF(I.GE.KO) GO TO 4504
IF1=(I+IQP-1)*IGPSZ+(K-I)*IP1SQ+IBPC
GO TO 4512
IF1=(MODISP(I+K129)-1)*IGPSZ+(K-I)*IP1SQ+IBPC
GO TO 4512
CALL FINDF(I,J,K,L,IF1,IP1J,IP1L,IBPC,IFINDF)
GO TO 4512

IF.( FINDE.EQ.1) GO TO 4510
IF1=IRSPPI+(I-K)*IP1SQ
GO TO 4512
CALL FINDE(I,J,K,L,IF1,IP1J,IP1L,IRSPPI,IFINDE)
CONTINUE

CALL FIND(I,J,K,L,IF1,IP1J,IP1L)
IMMO=I-MO
IARG=JMAI+IMMD
CWK(IF1)=CWK(IF1)-GAIN(IARG)*CWK(IF2)
IF(I.EQ.ISE.AND.J.EQ.JSE.AND.K.EQ.ISE.AND.L.EQ.JSE) GO TO 45
GO TO 4518
IST08=ISE-1QP
IST09=ISE+1KP
IF(M.GE.IST08.AND.M.LE.IST09) GO TO 4516
GO TO 4518
CONTINUE
CALL FINDE(ISE,JSE,ISE,JSE,ITST,IP1JISE,IP1JSE)
WRITE(6,951) M,N,ITST,CWK(ITST)
FORMAT(1X,'M,N,ITST,CWK(ITST)=R(ISE,JSE,ISE,JSE)='
951 . 315,E15.5)
CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE
YSA1(M,N)=CWK(IFAI)
IF(N.LE.LIP) GO TO 4720
IF(M.LE.LIQ) GO TO 4720
IST01=M-LIQ
IST02=N-LIF
IP1ST2=IST02+IP128
CALL FIND(IST01,IST02,IST01,IST02,IFS,IP1ST2,IP1ST2)
1 I3, 10X, 'N-LIP=', I3, 10X, 'CWK(INM,NM,N=','E13.5)
IF(M.LT.IUPDX) GO TO 4720
JJ=LJP
II=LIQ-1
DO 4710 I=1,LIQ
NMJJ=N-JJ
MMII=M-II
IP1NMJ=NMMJJ+IP128
CALL FINL(MMII, NMJJ, MMII, NMJJ, ITEMP, IP1NMJ, IP1NMJ)
9060 FORMAT('1X,ITEMP=', IS, 10X, 'CWK(ITEMP)=', E13.5, 10X,
1 'MMII=', I3, 10X, 'NMJJ=', I3,/)
II=II-1
4710 CONTINUE
4720 CONTINUE
C
5000 CONTINUE
C END OF LOOP ON COLUMNS
C
IR=MODIX
IF(IR.NE.ISZA1P) GO TO 5220
IST01=IUPDSZ
DO 5210 I=1,ISZS
IST01=IST01+1
A(IST01)=CWK(I)
5210 CONTINUE
GO TO 5250
5220 CONTINUE
ITEMP=IGPSZ*(ISZA1P-IR)
IST02=IR*IGPSZ
IST03=IST02+ITEMP
DO 5230 I=1,IST02
IST03=IST03+1
A(IST03)=CWK(I)
5230 CONTINUE
IST04=IR*IGPSZ+1
IST05=IUPDSZ
IBUT=IST04+ITEMP-1
DO 5240 I=IST04,IBUT
IST05=IST05+1
A(IST05)=CWK(I)
5240 CONTINUE
5250 CONTINUE
C
C IPTR2=IPTR1
DO 5310 I=1,IPP
INDEXC(I)=INDEXA(IPTR2)
INDEXD(I)=INDEXB(IPTR2)
IF(IPTR2.EQ.IPP) IPTR2=0
IPTR2=IPTR2+1
5310 CONTINUE
IPTR1=IPTR1-1
IF(IPTR1.EQ.0) IPTR1=IPP
DO 5330 J=1,4590,1IPSQ
IBASE=J-1
K=1
DO 5320 I=ISTART,IPSQ,IPP
IBASE=I+IBASE
IBINK=IBASE+INDEXC(K)
A(IBINK)=A0(IB1NK)
K=K+1
K=0.
DO 5350 I=IBIGIN+4590,IP1SQ
KIP1SQ=K*IP1SQ
DO 5340 J=1,IPP
IARG1=I+J-1
IARG2=KIP1SQ+INDEX(J)
A(IARG1)=A0(IARG2)
5340 CONTINUE
K=K+1
5350 CONTINUE
ISTART=ISTART+1
IBIGIN=IBIGIN+IPP
IF(ISTART.LE.IPP) GO TO 5360
ISTART=1
IBIGIN=1
5360 CONTINUE
6000 CONTINUE
C
END OF LOOP ON ROWS.
C
ISTORE=IQP+1
D.: 7010 IHORZ=1,ISTORE
IBUT=UPDX2-IHORZ+1
CALL FIND(UPDX2,UPD,IBUT,UPD,ITEMPA,IP1PD,IP1PD)
IBUT=IHORZ-1
WRITE(6,9070) IBUT1,CWK(ITEMPA)
9070 FORMAT(1X,'HORIZ,DIS FROM DIAG COV',I5,20X,E15.4)
7010 CONTINUE
C
WRITE(6,9991)
9991 FORMAT('GAIN COEFFS FOR KALMAN FILTER')
WRITE(6,9992) (I,GAIN(I),I=1,18)
9992 FORMAT('GAIN (',I2,')=',E12.5)
STOP
END
DIMENSION IP(16,128)
DIMENSION NAME(5),IFILE(5)
DATA IBLK/1/
M1=128
M2=128

DO 1 I=1,5
  NAME(I)=IBLK
  IFILE(I)=IBLK
  WRITE(10,2)
1 FORMAT(1X,'FILENAME OF ORIGINAL IMAGE: ',Z)
  READ(11,3)(NAME(I),I=1,5)
2 FORMAT(5A2)
  WRITE(10,4)
4 FORMAT(1X,'FILENAME FOR NEW IMAGE: ',Z)
  READ(11,5)(IFILE(I),I=1,5)
5 FORMAT(5A2)
  OPEN 1 ,NAME,LEN=256,REC=128
  OPEN 2,IFILE,LEN=256,REC=128
  SUM=0.
  DO 6 I=1,8
    DO 7 J=1,16
5     READ(1)(IP(J,K),K=1,M2)
6     DO 9 J=1,16
       DO 9 K=1,M2
9       SUM=SUM+FLOAT(IP(J,K))
8     CONTINUE
7     CT=FLOAT(M1*M2)
     SMEAN=SUM/CT
  TYPE" "
  TYPE"MEAN OF THE ORIGINAL IMAGE: ",SMEAN
  TYPE" "
  REWIND 1
  VAR=0.
  DO 10 I=1,8
    DO 11 J=1,16
10       READ(1)(IP(J,K),K=1,M2)
11     DO 13 J=1,16
       DO 13 K=1,M2
13       X=FLOAT(IP(J,K))
12       VAR=VAR+(X-SMEAN)**2
9     CONTINUE
8     SVAR=VAR/CT
  TYPE" "
  TYPE"VARIANCE OF THE ORIGINAL IMAGE: ",SVAR
  TYPE" ".
  REWIND 1
  ACCEPT"DESIRED MEAN FOR NEW IMAGE =",DMEAN
  ACCEPT"DESIRED VARIANCE FOR NEW IMAGE=",DVAR
  SCALE=SQR(DVAR/SVAR)
  DO 15 I=1,8
    DO 16 J=1,16
15     READ(1)(IP(J,K),K=1,M2)
16     DO 18 J=1,16
6     DO 18 K=1,M2
18       Y=(SCALE*(FLOAT(IP(J,K))-SMEAN))+DMEAN
     IP(J,K)=IFIX(Y)
50  IP(J,K)=0.
   GO TO 18
60  IF(IP(J,K), GT, 255) GO TO 80
   GO TO 18
80  IP(J,K)=255
18  CONTINUE
   DO 19 J=1, M2
19  WRITE(2),(IP(J,K), K=1, M2)
15  CONTINUE
   CLOSE 1
   CLOSE 2
   ACCEPT"WISH TO CONTINUE? YES(NE. 0), NO(0):->", IW
   IF(IW.NE. 0) GO TO 100
   STOP
   END
FILENAME: FIND. SUBROUTINE FIND TO BEUSED WITH
RUKD TO LOCATE THE PROPER POINTS FOR THE COMPUTA-
TION OF COVARIANCE.

SUBROUTINE FIND(M,N,K,L,IF1,IP1N,IP1L)
COMMON/B1/K0,IOPI,K129,IGFSZ,IP1SQ,IP1,MODISP(269),
1MODIP1(269),ISIZ

IF(K,LT,M)GO TO 6
IF(M,GE,KO)GO TO 7
IR=M+IOPI
GO TO 8

IR=MODISP(M+K129)
IB=K-M
IBR=MODIP1(IP1N)
IBC=MODIP1(IP1L)~

IF(K,EQ,M.AND.IBC,LT,IBR)GO TO 5
GO TO 9

ISTORE=IBR.
IBR=IBC
IBC=ISTORE
GO TO 9

IF(K,GE,K0)GO TO 10
IR=K+IOPI
GO TO 11

IR=MODISP(K+K129)

IB=M-K
IBR=MODIP1(IP1L)
IBC=MODIP1(IP1N)

IF1=(IR-1)*IGFSZ+IB*IP1SQ+(IBR-1)*IP1+IBC
RETURN
END
FILENAME: FINDA.

SUBROUTINE FINDA(M, N, K, L, IF1, IP1N, IP1L, IBPC, IFINDA)
COMMON/B1/KO, IQP, K129, IGPSZ, IP1SQ, IP1, MODISP(269),
IBPC, MODIP1(269), ISZS

IF(K.GE.KO) GO TO 10
   IR=K+IQP
   GO TO 11

10   IR=MODISP(K+K129)-
11   IB=M-K
    IBR=MODIP1(IP1L)
    IBC=MODIP1(IP1N)
    IBPC=(IBR-1)*IP1SQ+IBPC

IF1=(IR-1)*1GPSZ+IB*IP1SQ+IBPC
IFINDA=0
RETURN
END
SUBROUTINE FINDB(M,N,K,L,IF1,IP1N,IP1L,IRSP1,IFINDB)
COMMON/B1/KO,IGP,K129,IGPSZ,IP1SG,IP1,MODISP(269),
MODIP1(269),1SZS

IF(M.GE.KO) GO TO 7
IR=M+IGP
GO TO 8

7 IR=MODISP(M+K129)
IB=K-M
IBR=MODIP1(IP1N)
IBC=MODIP1(IP1L)
IRSP1=(IR-1)*IGPSZ+(IBR-1)*IP1+IBC

IF1=IRSP1+IB*IP1SG
IFINDB=0
RETURN
END
C
***************
FILENAME=-FINNE.
***************
SUBROUTINE FINDE(M,N,K,L,IF1,IP1N,IP1L,IRSP1,FINDE)
COMMON/B1/KO,IQP,K129,IGPSZ,IP1SQ,IP1,MODISP(269),
  1 MODIP1(269),ISIZ
C
IF(K.GE.KO) GO TO 10
IR=K+IQP
GO TO 11
C
10 IR=MODISP(K+K129)
  IB=M-K
  IBR=MODIP1(IP1L).
  IBC=MODIP1(IP1N)
  IRSP1=(IR-1)*IGPSZ+(IB-1)*IP1+IBC
C
IF1=IRSP1+IB+IP1SQ
IFINDE=0.
RETURN
END
**FILENAME: FINDF**

**SUBROUTINE FINDF(K,N,K,L,IF1,IP1N,IP1L,IBPC,IFINDF)**

**COMMON/K1/K0, IQP, K129, IGPSZ, IP1SQ, IP1, MODISP(269),**

1 MODIP1(269), ISZS

IF(M.GE.K0) GO TO 7
IR=M+IQP
GO TO 8

7IR=MODISP(K+K129)
IB=K-M
IBC=MODIP1(IP1L)
IBR=MODIP1(IP1N)
IBPC=(IBR-1)*IP1+IBC

IF1=IBPC+(IR-1)*IGPSZ+IB*IP1SQ
IFINDF=0
RETURN
END
FILE NAME: AVERAGE. SUBROUTINE FOR THE SMOOTHING PROGRAM.

SUBROUTINE AVERAGE(ISTART, IEND, ISTAR, IEND1, THRESH, IP)
DIMENSION IP(120, 120)
IAVE=0
DO 10 J=ISTART, IEND
  DO 10 K=ISTAR, IEND1
    IAVE=IAVE+IP(J, K)
    IAVE=(IAVE-IP(IEND-1, ISTAR+1))/9
  DO 20 J=ISTART, IEND
    DO 20 K=ISTAR, IEND1
      DIFF=ABS(IP(J, K)-IAVE)
      IF(DIFF .GT. THRESH) IP(J, K)=IAVE
    20 IP(J, K)=IP(J, K)
RETURN
END
FILENAME: DEBLUR. PROGRAM TO COMPUTE GRADIENTS
OF THE DESIRED IMAGE USING ROBERTS AND SOBELS
METHOD.

SUBROUTINE DEBLUR(ISTART, IEND, THRESH, IEDI, IS, IP)
DIMENSION IP(32, 32)
IF1=IP(ISTART, IEND)-IP(ISTART+1, IEND+1)
IFF1=ABS(IF1)
IF2=IP(ISTART+1, IEND)-IP(ISTART, IEND+1)
IFF2=ABS(IF2)
IGRAD=(IFF1+IFF2)
IF(IGRAD GT THRESH) GO TO 1
IP(ISTART, IEND)=IP(ISTART, IEND)
GO TO 2
1 IF(IS EQ 0) GO TO 3
IP(ISTART, IEND)=IEDG
GO TO 2
3 IP(ISTART, IEND)=IGRAD
RETURN
END
FILENAME: VECTOR1. GRAD. DETECTION VECTOR.

SUBROUTINE VECTOR1(W1, W2)
DIMENSION W1(3,3), W2(3,3)
DO 1 I=1,3
  DO 2 J=1,3
    W1(I,J)=0.
    W2(I,J)=0.
  2 CONTINUE
  W1(1,2)=W1(1,2)*2
  DO 3 J=1,3
    W1(3,J)=W1(1,J)*(-1)
    W2(J,1)=W1(J,1)
    W2(J,3)=W2(J,1)*(-1)
  3 CONTINUE
WRITE(10,4)
  4 FORMAT(1X, 'HORIZONTAL VECTOR')
WRITE(10,5) W2
  5 FORMAT(3(I2, ', ','))
WRITE(10,6)
  6 FORMAT(1X, 'VERTICAL VECTOR')
WRITE(10,7) W1
  7 FORMAT(3(I2, ', ','))
RETURN
END
SUBROUTINE VECTOR(W1, W2, W3, W4)
DIMENSION W1(3,3), W2(3,3), W3(3,3), W4(3,3)
DO 1 I=1,3
DO 1 J=1,3
W1(I, J)=-1
W2(I, J)=-1
W3(I, J)=-1
W4(I, J)=-1
1 CONTINUE
DO 2 J=1,3
W1(2, J)=W1(2, J)*(-2)
2 CONTINUE
DO 3 J=1,3
W2(J, K1)=W2(J, K1)*(-2)
3 CONTINUE
K1=K1-1
DO 4 J=1,3
W3(J, 2)=W3(J, 2)*(-2)
4 CONTINUE
DO 5 J=1,3
W4(J, J)=W4(J, J)*(-2)
5 CONTINUE
WRITE(10, 6)
6 FORMAT(1X, 'VECTOR FOR HORIZONTAL EDGE')
WRITE(10, 7) W1
7 FORMAT(3(I2, ', ',))
WRITE(10, 8)
8 FORMAT(1X, 'VECTOR FOR EDGE INCLINED AT +45 DEG')
WRITE(10, 9) W2
9 FORMAT(3(I2, ', ',))
WRITE(10, 10)
10 FORMAT(1X, 'VECTOR FOR VERTICAL EDGE')
WRITE(10, 11) W3
11 FORMAT(3(I2, ', ',))
WRITE(10, 12)
12 FORMAT(1X, 'VECTOR FOR EDGE INCLINED AT -45 DEG')
WRITE(10, 13) W4
13 FORMAT(3(I2, ', ',))
RETURN
END
SUBROUTINE CORCO(IX, IY, ETAX, ETAY, XSEGMA, YSEGMA, M2, RHO)
DIMENSION IX(128), IY(128)
SUM=0
DO 1 J=1, M2
    X=FLOAT(IX(J))
    Y=FLOAT(IY(J))
    S1=X-ETAX
    S2=Y-ETAY
    SUM=SUM+S1*S2
1    D1=XSEGMA*YSEGMA
    RHO=SUM/<(SORT(D1))*M2>
RETURN
END
FILENAME: MEAN. PROGRAM TO COMPUTE MEAN OF ONE ROW IN IMAGE.

SUBROUTINE MEAN (IX, M2, ETA)
DIMENSION IX(128)
SUM=0.
DO 1 J=1, M2
  X=FLOAT(IX(J))
  SUM=SUM+X
1 ETA=SUM/M2
RETURN
END
VITA AUCTORIS

1952 Born at Bombay, India (December 4th)

1973 Completed the Degree of B.E. (Electrical)
Regional College of Engineering, Nagpur, India

1975 Completed the Degree of M.E. (Electrical)
University of Bombay, Bombay, India

1975-77 Served as Design Engineer in Rectifiers Division
of Bharat Heavy Electrical Ltd., India

1977-79 Served as Design Engineer in Electronic Controls
Division of Larsen and Toubro Ltd., Bombay, India

1982 Serving as Design Engineer in Canadian Controlled
Power Ltd., Windsor, Ontario (A subsidiary of
Controlled Power Co., Troy, Michigan, USA)

1980 – present

Candidate for the Degree of M.A.Sc. (Systems) at
University of Windsor, Windsor, Ontario, Canada