GARDNER, Steven Ray, 1942-
A GENERALIZED TEMPLATE MATCHING ALGORITHM
FOR PICTORIAL PATTERN RECOGNITION.

The Ohio State University, Ph.D., 1970
Computer Science

University Microfilms, A XEROX Company, Ann Arbor, Michigan
A GENERALIZED TEMPLATE MATCHING
ALGORITHM FOR PICTORIAL PATTERN RECOGNITION

DISSERTATION
Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By
Steven Ray Gardner, B.E.E., M.S.E.E.

* * * * *

The Ohio State University
1970

[Adviser's signature]
Department of Electrical Engineering
PLEASE NOTE:

Some pages have small and indistinct type.
Filmed as received.

University Microfilms
The research undertaken in this dissertation is concerned with the following problem. Given a two-dimensional image of an unknown object, it is desired to develop an algorithm for a digital computer which will determine the class to which the image belongs as well as its size, translation and rotation in the field of view with respect to some coordinate system. The major portion of the research is concerned with the estimation of the size, translation and rotation parameters associated with the image, while the classification problem is only a minor objective.

The algorithm which is developed uses the coordinates of a set of discrete points on the boundary of the image for both estimation and classification. The data acquisition system is simulated for testing the algorithm since it is
not vital in the development of the algorithm.

The images which are presented to the algorithm include ellipses, rectangles, convex crescents, and concave crescents. These images were chosen from a practical viewpoint (physical objects possess these images) and from a theoretical viewpoint (the images are not representable by an analytical equation except for the ellipse).

The algorithm which is developed solves the stated problem in the following manner. First of all, a template for each of the four patterns is generated by the computer and stored; that is to say, if the computer is given a set of parameters it can produce the corresponding template for each pattern. The algorithm next fits each stored template to the data points of the incoming image, automatically adjusting the parameters in such a manner that an appropriate error function is minimized. The parameters are adjusted by making use of nonlinear regression analysis. After each of the templates has been fitted to the data points, the template is chosen which yields the smallest least squares fit.

A total of three different error functions are investigated in this research, one of which is found to give very good recognition capabilities (good parameter estimates), even in the presence of moderate
levels of noise. The estimates for the rotation parameter are especially good, a quality not common to many pattern recognition systems.
ACKNOWLEDGEMENTS

The author gratefully acknowledges the encouragement, advice, and guidance provided to him throughout the course of this research by his adviser, Professor H. Hemami. Professors R. B. McGhee and K. J. Breeding are also to be thanked for their many helpful suggestions and for serving as readers of this manuscript.

This research was supported in part by the National Aeronautics and Space Administration under Grant No. NGR-36-008-076 and in part by the United States Air Force Office of Scientific Research under Grant No. AFOSR-70-1901.

The experimental results reported in the work were obtained on the IBM 360/75 Computer located at the Instructional and Research Computer Center, The Ohio State University.

The author would like to take this opportunity to thank his wife, Karen, for her patience and understanding during the course of his graduate studies. He is especially indebted to her for typing this manuscript.
VITA

September 15, 1942  Born - Columbus, Ohio

1965  B.E.E., The Ohio State University, Columbus, Ohio

1966  M.Sc., Electrical Engineering, The Ohio State University, Columbus, Ohio

1968  Teaching Associate, Mathematics Department, The Ohio State University, Columbus, Ohio

1969 - 1970  Research Associate, The Ohio State University Research Foundation, Communications and Control Systems Laboratory, Department of Electrical Engineering, Columbus, Ohio

FIELDS OF STUDY

Major Field: Electrical Engineering


Studies in Network Synthesis and Communication Theory: Professors W. C. Davis and C. E. Warren

Studies in Coding Theory: Professor R. Lackey

Studies in Physics: Professor W. H. Shaffer

Studies in Mathematics: Professor S. Drobot
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>VITA</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF PLATES</td>
<td>x</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Formulation of the Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Organization</td>
<td>6</td>
</tr>
<tr>
<td>II. SURVEY OF PATTERN RECOGNITION</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>8</td>
</tr>
<tr>
<td>2.2 The Pattern Recognition Process</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Related Pattern Recognition Techniques</td>
<td>21</td>
</tr>
<tr>
<td>2.4 Summary</td>
<td>22</td>
</tr>
<tr>
<td>III. NONLINEAR REGRESSION ANALYSIS</td>
<td>24</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>24</td>
</tr>
<tr>
<td>3.2 Representation of the Ellipse Pattern</td>
<td>24</td>
</tr>
<tr>
<td>3.3 Parameter Estimation Problem</td>
<td>33</td>
</tr>
<tr>
<td>3.4 Summary</td>
<td>50</td>
</tr>
<tr>
<td>IV. RECOGNITION OF ELLIPSES</td>
<td>51</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>51</td>
</tr>
<tr>
<td>4.2 Statement of Problem</td>
<td>52</td>
</tr>
<tr>
<td>4.3 Implementation of the Parameter Estimation Schemes</td>
<td>54</td>
</tr>
<tr>
<td>4.4 Results</td>
<td>59</td>
</tr>
<tr>
<td>4.5 Fitting an Ellipse Template to Ellipse Data Points</td>
<td>63</td>
</tr>
<tr>
<td>4.6 Implementation of the Ellipse Template</td>
<td>69</td>
</tr>
<tr>
<td>4.7 Summary</td>
<td>69</td>
</tr>
</tbody>
</table>
V. RECOGNITION OF RECTANGLES

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>83</td>
</tr>
<tr>
<td>5.2 Fitting an Ellipse to a Rectangle</td>
<td>84</td>
</tr>
<tr>
<td>5.3 Parameter Estimation for Noise-Free Rectangular Patterns</td>
<td>86</td>
</tr>
<tr>
<td>5.4 Parameter Estimation for Noisy Rectangular Patterns</td>
<td>92</td>
</tr>
<tr>
<td>5.5 Fitting a Rectangle Template to Rectangle Data Points</td>
<td>96</td>
</tr>
<tr>
<td>5.6 Implementation of the Rectangle Template</td>
<td>103</td>
</tr>
<tr>
<td>5.7 Summary</td>
<td>103</td>
</tr>
</tbody>
</table>

VI. RECOGNITION OF CRESCENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Introduction</td>
<td>122</td>
</tr>
<tr>
<td>6.2 Recognition of Convex Crescents</td>
<td>123</td>
</tr>
<tr>
<td>6.3 Recognition of Concave Crescents</td>
<td>129</td>
</tr>
<tr>
<td>6.4 Implementation of the Crescent Templates</td>
<td>135</td>
</tr>
<tr>
<td>6.5 Summary</td>
<td>136</td>
</tr>
</tbody>
</table>

VII. CONCLUSIONS, SUMMARY OF RESULTS, AND EXTENSIONS                     | 140  |

APPENDIX

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Generation of Data Points</td>
<td>149</td>
</tr>
<tr>
<td>B. Discussion of Criterion Functions</td>
<td>158</td>
</tr>
<tr>
<td>C. Parameter Estimation Using First and Second Moments</td>
<td>163</td>
</tr>
<tr>
<td>D. Regression Matrix for Template Matching Algorithm</td>
<td>172</td>
</tr>
<tr>
<td>E. Computer Program</td>
<td>183</td>
</tr>
</tbody>
</table>

BIBLIOGRAPHY                                                                | 215  |
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Ellipse Parameter Estimates Obtained by the One Step Minimization Method</td>
<td>74</td>
</tr>
<tr>
<td>2. Ellipse Parameter Estimates Obtained by the Iterative Minimization Scheme</td>
<td>75</td>
</tr>
<tr>
<td>3. Ellipse Parameter Estimates Obtained by the Template Matching Method</td>
<td>76</td>
</tr>
<tr>
<td>4. Parameters of Ellipses Fitted to Rectangles by the One Step Minimization Method (8 Data Points)</td>
<td>107</td>
</tr>
<tr>
<td>5. Parameters of Ellipses Fitted to Rectangles by the One Step Minimization Method (20 Data Points)</td>
<td>108</td>
</tr>
<tr>
<td>6. Parameters of Ellipses Fitted to Rectangles by the One Step Minimization Method (48 Data Points)</td>
<td>109</td>
</tr>
<tr>
<td>7. Parameters of Ellipses Fitted to Rectangles by the One Step Minimization Method (100 Data Points)</td>
<td>110</td>
</tr>
<tr>
<td>8. Parameters of Ellipses Fitted to Rectangles by the Iterative Minimization Method (8 Data Points)</td>
<td>111</td>
</tr>
<tr>
<td>9. Parameters of Ellipses Fitted to Rectangles by the Iterative Minimization Method (20 Data Points)</td>
<td>112</td>
</tr>
<tr>
<td>10. Parameters of Ellipses Fitted to Rectangles by the Iterative Minimization Method (48 Data Points)</td>
<td>113</td>
</tr>
<tr>
<td>Table</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>11. Parameters of Ellipses Pitted to Rectangles by the Iterative Minimization Method (100 Data Points)</td>
<td>114</td>
</tr>
<tr>
<td>12. Rectangle Parameter Estimates Obtained by the Template Matching Method</td>
<td>115</td>
</tr>
<tr>
<td>13. Convex Crescent Parameter Estimates Obtained by the Template Matching Method</td>
<td>138</td>
</tr>
<tr>
<td>14. Concave Crescent Parameter Estimates Obtained by the Template Matching Method</td>
<td>139</td>
</tr>
<tr>
<td>15. Values for the Sum-Squared Error Criterion Function Obtained by Matching Each Template to an Ellipse Image (48 Data Points)</td>
<td>145</td>
</tr>
<tr>
<td>16. Values for the Sum-Squared Error Criterion Function Obtained by Matching Each Template to a Rectangle Image (48 Data Points)</td>
<td>146</td>
</tr>
<tr>
<td>17. Values for the Sum-Squared Error Criterion Function Obtained by Matching Each Template to a Convex Crescent Image (48 Data Points)</td>
<td>147</td>
</tr>
<tr>
<td>18. Values for the Sum-Squared Error Criterion Function Obtained by Matching Each Template to a Concave Crescent Image (48 Data Points)</td>
<td>148</td>
</tr>
<tr>
<td>19. Comparison of Sum-Squared Error Criterion Functions Associated with the One Step Minimization Method and the Iterative Minimization Scheme</td>
<td>162</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Components of a Pattern Recognition System</td>
<td>10</td>
</tr>
<tr>
<td>2.</td>
<td>Example of an Ellipse</td>
<td>26</td>
</tr>
<tr>
<td>3.</td>
<td>Example of a Rotated and Translated Ellipse</td>
<td>26</td>
</tr>
<tr>
<td>4.</td>
<td>Relation of $\theta$ and $\rho$</td>
<td>26</td>
</tr>
<tr>
<td>5.</td>
<td>Parameters of a Rotated and Translated Ellipse</td>
<td>53</td>
</tr>
<tr>
<td>6.</td>
<td>Determination of Ellipse Model Point Associated with a Given Data Point</td>
<td>64</td>
</tr>
<tr>
<td>7.</td>
<td>Parameters of a Rotated and Translated Rectangle</td>
<td>85</td>
</tr>
<tr>
<td>8.</td>
<td>Determination of Rectangle Model Point Associated with a Given Data Point</td>
<td>97</td>
</tr>
<tr>
<td>9.</td>
<td>Example of a Convex Crescent</td>
<td>124</td>
</tr>
<tr>
<td>10.</td>
<td>Determination of Convex Crescent Model Point Associated with a Given Data Point</td>
<td>124</td>
</tr>
<tr>
<td>11.</td>
<td>Example of a Concave Crescent</td>
<td>130</td>
</tr>
<tr>
<td>12.</td>
<td>Determination of Concave Crescent Model Point Associated with a Given Data Point</td>
<td>130</td>
</tr>
<tr>
<td>13.</td>
<td>Example of an Ellipse</td>
<td>151</td>
</tr>
<tr>
<td>14.</td>
<td>Data Points Associated with an Ellipse Having No Rotation or Translation</td>
<td>151</td>
</tr>
<tr>
<td>15.</td>
<td>Data Points Associated with a Rotated and Translated Ellipse</td>
<td>151</td>
</tr>
<tr>
<td>16.</td>
<td>Example of a Rectangle</td>
<td>154</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>17.</td>
<td>Example of a Concave Crescent</td>
<td>154</td>
</tr>
<tr>
<td>18.</td>
<td>Example of a Convex Crescent</td>
<td>154</td>
</tr>
<tr>
<td>19.</td>
<td>Principal Axes for an Ellipse</td>
<td>168</td>
</tr>
<tr>
<td>20.</td>
<td>Principal Axes for a Rectangle</td>
<td>168</td>
</tr>
<tr>
<td>21.</td>
<td>Principal Axes for a Concave Crescent</td>
<td>170</td>
</tr>
<tr>
<td>22.</td>
<td>Principal Axes for a Convex Crescent</td>
<td>170</td>
</tr>
<tr>
<td>Plate</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>I.</td>
<td>Ellipses Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.0$)</td>
<td>77</td>
</tr>
<tr>
<td>II.</td>
<td>Ellipses Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.1$)</td>
<td>78</td>
</tr>
<tr>
<td>III.</td>
<td>Ellipses Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.2$)</td>
<td>79</td>
</tr>
<tr>
<td>IV.</td>
<td>Ellipses Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.3$)</td>
<td>80</td>
</tr>
<tr>
<td>V.</td>
<td>Ellipses Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.4$)</td>
<td>81</td>
</tr>
<tr>
<td>VI.</td>
<td>Ellipses Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.5$)</td>
<td>82</td>
</tr>
<tr>
<td>VII.</td>
<td>Rectangles Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.0$)</td>
<td>116</td>
</tr>
<tr>
<td>VIII.</td>
<td>Rectangles Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.1$)</td>
<td>117</td>
</tr>
<tr>
<td>IX.</td>
<td>Rectangles Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.2$)</td>
<td>118</td>
</tr>
<tr>
<td>X.</td>
<td>Rectangles Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.3$)</td>
<td>119</td>
</tr>
<tr>
<td>XI.</td>
<td>Rectangles Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.4$)</td>
<td>120</td>
</tr>
<tr>
<td>XII.</td>
<td>Rectangles Fitted to Data Points by the Iterative Minimization Method ($\sigma = 0.5$)</td>
<td>121</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

1.1 Formulation of the Problem

The research undertaken in this dissertation is concerned with the following problem. Given a two-dimensional image of an unknown object, it is desired to develop an algorithm for a digital computer which will determine the class to which the image belongs as well as its size, translation and rotation in the field of view with respect to some coordinate system. This problem actually consists of both the classical pattern classification problem of selecting the class of objects to which the unknown template belongs as well as the problem of precise estimation of its characterizing parameters, namely, its size and its translation and rotation with respect to a frame of reference. This latter problem shall be referred to as a pattern recognition problem in this research.

The problem is motivated by a variety of present industrial needs and by the new rendezvous and guidance
techniques required for space travel and exploration. For instance, consider the problem of an industrial robot attaching screws to an engine block located on an assembly belt. The robot must identify the block, its position on the belt, and finally the location and size of the holes in the block. Or, an industrial robot might be required to automatically handle parallelopiped or cylindrical packages, moving them from one location to another. Concerning space exploration, an important problem is that of automatic docking of two spacecraft where the translation and rotation of one craft with respect to the other must be precisely known. Also, the problem of automatic acquisition of a landmark (airstrip) or heavenly body (the moon, for instance) for navigation purposes is important.

In all of the above applications, extreme precision in the estimation of the parameters of size, translation, and rotation is required. On the other hand, the number of classes of objects in any particular application is not large and is well under control. Consequently, the classification problem is not crucial from an application point of view, and therefore it is only a minor objective of this research while the estimation problem comprises the major portion of the research. From a theoretical point of view, it is interesting that the algorithm which is developed is capable of both estimation and classification.
While some of the cases discussed above are three-dimensional in nature, this research is concerned with only two-dimensional images. Since the emphasis is on the development of a general algorithm, the mechanics of how the image is generated in a real world situation is not considered here.

The algorithm which is developed uses the coordinates of a set of discrete points on the boundary of the image for both classification and estimation. Therefore, it is assumed that the data acquisition system has the ability to obtain an image of the object in question, and enhance the edge of the image by some optical or electro-optical mechanism. The final image then consists of a closed contour in the field of view corresponding to the boundary of the image. It is then assumed that some scanning mechanism is available which will extract a discrete set of data points from this boundary for use with the algorithm. This entire data acquisition process is simulated for testing the algorithm since it is not vital in the development of the algorithm. Obviously, the coordinates of the boundary points obtained by any physical data acquisition system will contain noise to some degree. Therefore, the simulation for the data acquisition system produces boundary points whose coordinates are corrupted by additive noise (in particular,
zero mean, gaussian noise). Multiplicative noise that would remove or smear part of the boundary is not considered. The effect of this type of noise would be analogous to a case where not all of the image is in the field of view.

Based on the applications cited earlier, several classes of patterns were considered desirable. These include ellipses (a circle, such as a hole, viewed at an oblique angle), rectangles (images of packages, buildings, or airstrips), and crescents (images of planets which may be either concave or convex).

These classes of patterns have several other desirable properties in addition to the fact that they closely approximate the image of familiar objects. For example, the ellipse is described analytically by one equation, while a rectangle and crescent are not. The ellipse and rectangle are convex shapes, while a crescent may be either convex or concave.

The goal of the computer algorithm which is developed here is to derive the desired parameters of the object and its class from the coordinates of a set of points lying on the image boundary. The algorithm achieves this goal in the following manner. First of all, a template for each of the four patterns is generated by the computer and stored; that is to say, if the computer is given a set of
parameters it can produce the corresponding template for each pattern. Four templates are stored in the computer since a concave crescent and a convex crescent are considered to be two separate patterns. The algorithm next fits each stored template to the data points of the incoming image, automatically adjusting the parameters in such a manner that an appropriate error function is minimized. The parameters are adjusted by making use of nonlinear regression analysis. After each of the four templates has been fitted to the data points, the template is chosen which yields the smallest least squares fit.

A total of three different error functions are investigated in this research, one of which is found to give very good recognition capabilities, even in the presence of small amounts of noise.

At this point a remark should be made concerning the title of this dissertation. The term "template matching" has been used by others [1] in describing their pattern recognition schemes, the term referring to recognition based upon correlation techniques. In this research, the template matching algorithm uses correlation in the sense discussed above, i.e., a template is compared to a set of data points and a "goodness of fit" is computed. The template is then moved around in the reference frame, in a systematic manner, until the goodness of fit reaches
a maximum (in reality, until an error function reaches a minimum). The resulting template is then chosen as the template from which the data points came.

This template matching algorithm is generalized in the sense that both the size and shape of the template, as well as its position and orientation, are adjusted simultaneously to obtain a best fit to an image composed of discrete points, rather than a continuous contour.

1.2 Organization

The content of this research is divided into the next five chapters. Chapter II gives a brief survey of the history of the problem at hand. In particular, it discusses the general pattern recognition process and relates it to the algorithm which is developed in this research. Other pattern recognition schemes which use boundary information are also presented. Chapter III develops the basic nonlinear regression analysis approach which is utilized by the computer algorithm. This involves minimizing a squared-error criterion function, the error being either linear or nonlinear in the parameters. The results of using the algorithm for ellipses, rectangles, and crescents are presented in Chapters IV, V, and VI, respectively. Chapters IV and V discuss the results of using three different error functions while Chapter VI discusses the use of only one
error function. The reason for this is given in these chapters. Chapter VII discusses the pattern classification capabilities of the algorithm as well as a general evaluation of the algorithm in its ability to classify patterns and estimate their parameters. Also a few potential future research areas are presented. Finally, several appendices are also included, one of which contains a documentation and explanation of the computer program.
2.1 Introduction

The intent of this chapter is to discuss the basic structure of pattern recognition systems in general and to relate this structure to the algorithm which is developed in this research. The chapter concludes by presenting some other pattern recognition schemes which base their decisions on boundary information.

Thus, from the outset, it should be noted that this chapter does not contain a comprehensive review of the state of the art in pattern recognition, but rather it relates the general pattern recognition system to the developed algorithm. Nagy [2] has done a rather thorough review in the state of the art in pattern recognition, in which he states that "Perhaps the reason why relatively few reviews are published in the domain of pattern recognition is that prospective reviewers realize at the outset that pattern recognition is hardly a cohesive discipline in its own right. At best, it is a vast collection of highly varied problems. Any technique which contributes to the solution of any
of these problems can therefore be considered as part and parcel of pattern recognition."

Pattern recognition has been defined by Stevens [3] as "...a process requiring a decision as to which specific one of a plurality of possible input patterns was in fact 'sensed' by a suitable scanning-detection mechanism." Selfridge [4] states that "...pattern recognition involves classifying configurations of data into classes of equivalent significance so that very many different configurations all belong in the same equivalence class." These definitions apply to the recognition of two-dimensional patterns as well as to the recognition of three-dimensional spatial patterns, acoustical patterns [5], and to weather forecasting [6] and medical diagnoses [7].

2.2 The Pattern Recognition Process

A pattern recognition system can generally be divided into three components or stages. These include

1. Transducer or Pattern Transformation Device
2. Preprocessor or Information Selector
3. Classifier or Recognition Device

A block diagram illustrating these components is shown in Figure 1.
Figure 1. Components of a Pattern Recognition System

The source pattern (the two-dimensional pattern which is to be recognized) is operated on by the transducer to yield an input pattern which is in the format required by the preprocessor. The preprocessor then operates on the input pattern, extracting information which may be used by the classifier to decide into which class or category the source pattern belongs. These three stages or operations generally are not independent in most pattern recognition systems. Quite often the transduction and preprocessing steps are performed simultaneously. Generally the classification stage is separate from the other two steps, although the actual mechanics of the classification stage will certainly depend on the type of information which the preprocessor is extracting.

In any case, all pattern recognition systems employ these three operations in one way or another. Each of these steps will now be considered in more detail and related to the algorithm to be developed in this dissertation.

2.2.1 Transducer

Most pattern recognition systems employ either a
magnetic or an optical scanner for the transforming device. In the simplest of systems no scanning device is used at all since recognition could be based on matching an optical reproduction of the source pattern with a collection of templates representing known patterns. The template which gave the best match (as determined by the template allowing the maximum amount of light to pass through it and the source pattern, for instance) would then be considered to represent the same pattern as the source pattern [8].

Magnetic scanners are generally used in systems which use source patterns which have been printed with an ink containing iron oxide. These systems find their widest use in character recognition, especially in the recognition of printed characters and numerals on bank checks [9]. The characters are styled in such a manner that each character has a unique voltage versus time waveform when scanned by the reading head. This waveform is then used for the input to the preprocessor.

The flying-spot scanner is by far the most common scanner used in optical scanning systems. The single-slit finds less use since it can be used only for a limited number of patterns [10]. The flying-spot scanner is quite often used in such a manner as to quantize the source pattern into an n×m matrix, where each element of the matrix has a quantized value corresponding to the light intensity of that region in the source pattern.
Often the source pattern is quantized into a binary pattern; that is, the matrix is composed of 1's and 0's representing black and white, respectively. A flying-spot scanner may also be used so as to give a large variety of scan patterns; these scan configurations then are used to test various classification methods [11].

The transducer portion of the pattern recognition scheme which is developed in this research corresponds to an electro-optical system which extracts the coordinates of a set of points which lie on the boundary of the image of the unknown object. This system consists of an optical system to record the image, an edge enhancement mechanism to sharpen the boundary of the image, and a scanning mechanism to extract discrete points which lie on this boundary. This entire process is simulated in this research since it is not vital to the development of the recognition algorithm and since the appropriate electro-optical equipment was not available.

2.2.2 Preprocessor

The role of the preprocessor is to obtain a set of numbers for each source pattern which the transducer presents to it. If these numbers are denoted by \( x_1, x_2, \ldots, x_m \), then the vector \( X \) having components \( x_1, x_2, \ldots, x_m \) is called the pattern (or descriptor or
feature) vector $X$. Notice then that the pattern vector may be represented by a point in an $m$-dimensional measurement or sample space. Generally $m$ will be less than the dimension of the source pattern which is presented to the preprocessor, which for an $n \times n$ matrix representation would be $n^2$. The preprocessor, in fact, is generally designed from one of two points of view. "Either one attempts to transform the sample space in such a manner that the members of each class exhibit less variability and the relative separation between the classes is increased, thus allowing the use of a simpler decision mechanism, or one reduces the dimensionality of the sample space, permitting the application of more complicated decision schemes." [1]

Now suppose that the source patterns which the recognition system is to classify belong to one of $R$ classes or categories. Then an ideal preprocessor would be one that would assign a unique vector, or set of numbers, to any source pattern which belonged to a given class. Classification would then be quite simple since it would merely consist of determining which of the $R$ unique vectors was identical to the vector representation of the source pattern. As one might expect, preprocessing techniques have not nearly advanced to this stage. Instead, the preprocessor generally
produces a scatter of pattern vectors in the $m$-dimensional measurement space for each class of source patterns. As mentioned previously, much attention is focused on designing a preprocessor such that each class has the smallest scatter as possible while different classes are spread as far from each other as possible. (Generally the measure of distance is taken to be Euclidean distance). As a matter of fact, it may not be possible to design a preprocessor such that vectors from different classes do not intermingle at all. The overall recognition scheme then will definitely not be error free in its classifications.

In order for the preprocessor to present the classifier a simpler set of information from which to make its decision, it seems reasonable to expect that the preprocessor should extract only important or significant features from the source pattern. In fact, Selfridge [4] states that "Pattern recognition is the extraction of the significant features from a background of irrelevant detail." In character recognition, for instance, significant features might include the detection of the "presence or absence of line segments of certain slopes and lengths, intersections, corners, and arcs of certain curvatures." [12] The presence of any of these features could be denoted by a "1" while
their absence could be denoted by a "0". Other features may have nonbinary values, indicating the degree to which a feature exists. All these numbers then become the component values for the pattern vector, \( X \), of the source pattern. One should note that many researchers in the pattern recognition field consider the feature extraction stage of their recognition system as the most significant.

"Unfortunately," Nilsson [12] states, "there seems to be no general theory to help guide our search for the relevant features in any given recognition problem. The design of feature extractors is largely an empirical matter following different ad hoc rules found to be useful in each special situation. One learns little, if anything, about how to design visual feature extractors from a study of successful acoustic feature extractors. This lack of guiding principles makes the study of biological prototypes especially interesting and relevant."

Concerning this last point, Kazmierczak and Steinbuch [13] state that "the human visual system is capable of selecting features or criteria from a pattern where the statement of the description would be independent of registration, skew, size, contrast, deformation, or other noise effects."

It should be pointed out that many times operations are performed on the source pattern (\( n \times n \) grid or matrix) preliminary to the feature extraction process. The
operations do nothing to reduce the dimension of the measurement space, but rather they "clean up" the grid so that feature extraction is made easier. These operations might include such things as speck removal, gap filling, thickening, thinning, and edging. Notice that these operations are in a sense filtering noise out of the grid. These operations, as well as others, are discussed in detail by Dineen [14] and by Nilsson [12].

The preprocessor portion of the pattern recognition scheme which is developed in this research corresponds to that part of the algorithm which computes the error associated with each data point. When the "radial distance" error function is used, each data point has two errors associated with it. One error corresponds to the x-distance between it and the computer template and the other error corresponds to the y-distance between it and the computer template. Thus, if there are \( N \) boundary points, then the feature vector \( X \) corresponds to the \( 2N \) error vector.

2.2.3 Classifier

For an extensive survey of classification techniques the interested reader is referred to recent articles by Nagy [2] and by Ho and Agrawala [15]. In this section some of the more important aspects of classification
schemes shall be emphasized. Their relation to the algorithm which is presented in this dissertation shall also be discussed.

As was mentioned earlier, the preprocessor has not been developed to the point where it can transform all the source patterns representing a given class into a single point (or vector) in measurement space. But rather, it tries to transform the various source patterns of a given class into some "small" region of points in measurement space, representing a small scatter of vectors. It is common practice then to associate a probability density with the scatter of pattern points for each class, given by \( p(X|C_i) \). That is, \( p(X|C_i) \) is the probability density function of the pattern vector \( X \) given that class \( i \) was the class of the source pattern presented to the recognition system. Once these densities are known, then decision theory may be applied to obtain the "optimum" classifier. If by optimum it is meant the minimization of misclassifications, then it can be shown that the optimum classifier chooses that class for which \( P(C_i|X) \) is the largest \([16],[17]\).

Using Bayes rule yields

\[
P(C_i|X) = \frac{p(X|C_i)P(C_i)}{p(X)} \tag{2-1}
\]
Since \( p(X) \) is independent of the class \( C_i \), the above rule is the same as choosing that class for which

\[
p(X|C_i)P(C_i) \quad i = 1, \ldots, R
\]

is the largest. Here \( P(C_i) \) is the a priori probability that class \( C_i \) was presented to the recognition system. If the a priori probabilities are all equal, then one merely chooses the class having the largest \( p(X|C_i) \).

If the scatter of pattern vectors for each class may be represented by a gaussian distribution, then the rule for optimum classification becomes quite simple. The optimum classifier merely computes (16).

\[
f_i(X) = - \frac{1}{2} X^T K_i^{-1} X + X^T K_i^{-1} P_i - \frac{1}{2} P_i^T K_i^{-1} P_i + \log P(C_i) - \frac{1}{2} \log |K_i| \quad i = 1, \ldots, R
\]

for \( i = 1, \ldots, R \), and assigns \( X \) to that class having the largest \( f_i(X) \). Here \( K_i \) is the covariance matrix of the gaussian distribution for each class, while \( P_i \) is the mean vector for each class. One notes that the functions \( f_i(X) \) are quadratic in the pattern vector \( X \). This decision rule then essentially partitions the measurement space with quadratic surfaces, where these surfaces separate the pattern vectors \( X \) which belong to one class from the pattern vectors belonging to other classes.

The decision rule becomes simpler if it is assumed
that all a priori probabilities $P(C_i)$ are equal, and that the components of the pattern vectors are statistically independent and have equal variances; i.e., $K_i = kI$, where $k$ is a constant and $I$ is the identity matrix.

Then the functions $f_i(X)$ for the optimum decision rule become

$$f_i(X) = X \cdot P_i - \frac{1}{2} P_i \cdot P_i \quad i = 1, \ldots, R \quad (2-4)$$

where $X \cdot P_i$ is the dot product of $X$ and $P_i$. These functions are linear in the pattern vector $X$, and thus the measurement space is partitioned by hyperplanes that bisect the line segments which connect pairs of mean vectors of the various classes.

Thus it can be seen that if a preprocessor were able to represent each class with a scatter of pattern vectors which had a gaussian distribution, then the design of the classification stage would be complete. In fact, even if the form of the probability densities of the pattern vectors which were generated by the preprocessor were known a priori, still one would have to estimate the unknown parameters (mean vector and covariance matrix) from pattern vectors whose classes are known. Abramson and Braverman [18] and Keehn [19] discuss methods for estimating mean vectors and covariance matrices from sample patterns. However, most generally the probability
densities are not gaussian by any stretch of the imagination.

Several classification techniques have been developed which try to estimate directly the parameters of the probability densities from a set of known patterns (design set) or which try to partition the measurement space without directly estimating the probability density parameters. These techniques are adaptive in that they update their classification rules as more data is presented. Notice that the decision rule discussed previously was non-adaptive. Several of these techniques are discussed by Nilsson [12].

Some of the more important classification techniques which fall into this category have been developed by Fix and Hodges [20], Cover and Hart [21], Stark et al [22], Firschein and Fischler [23], Ball and Hall [24], Bonner [25], Mattson and Damon [26], Rosen and Hall [27], as well as others [28], [29], [30].

The classification portion of the algorithm which is developed in this research corresponds to that part of the algorithm which computes and minimizes the sum-squared error criterion function. That is, the classifier stage computes the dot product of the error vector times itself which is supplied by the preprocessor. A nonlinear regression analysis technique then is used to
minimize this sum-squared error criterion function with respect to the parameters. The estimates for the parameters for each of the four template classes is then accomplished, and the unknown pattern is classified as belonging to that class which corresponds to the smallest criterion function.

2.3 Related Pattern Recognition Techniques

The concept of using boundaries for pattern recognition and classification has been known for some time [31]. The subjects of edge enhancement and contour tracing have been reviewed by Levine [1]. Specifically, pattern recognition using contour analysis and chain encoding of the pattern boundary have been considered [32], [33]. The question of sufficiency of the discrete set of points on the boundary for a unique and adequate description of the pattern is not treated at all in the literature, and is not considered in this research. However, some thoughts along these lines can be found in Montanari [34], [35].

The linguistic approach for representation of line drawings in strings and manipulation of such strings has been discussed by Narasimhan [36], Clowes [37], and Breeding [38]. An excellent review in the linguistic approach is given by Zavalishin [39].

Brill [40] has used contours to recognize characters
and at the same time measure rotation. Borel [41] has used contours to recognize more general shapes such as aircraft. These two techniques are translational invariant. Zahn [42] has also developed a technique for representing the contour of a pattern.

Pattern recognition may also be accomplished optically in an analog fashion as opposed to the digital technique developed in this research. Vander Lugt [43] was one of the first to discuss spatial filtering using coherent light. In addition to being expensive, this system suffers due to the fact that it is not rotation invariant, i.e., it cannot measure rotation.

Paton [44] has developed a recognition scheme for chromosomes which involves fitting a quadratic curve to the unknown pattern. His work differs from this research in that he has defined a slightly different criterion function and does not constrain the fitted quadratic to be an ellipse, nor does he consider non-quadratic templates.

2.4 Summary

This chapter discusses the basic components of a pattern recognition system and relates them to the computer algorithm which is presented in this dissertation. These components include the transducer, the preprocessor,
and the classifier stages. The chapter concludes with a discussion of some related pattern recognition schemes which also base their decisions exclusively on boundary information.
3.1 Introduction

This chapter discusses in some detail the basic nonlinear regression analysis scheme which is used in the algorithm. Since the first class of patterns considered are ellipses, in Section 3.2 the representation of ellipse patterns is discussed. In Section 3.3 the parameter estimation problem is formulated, both in terms of an error function which is linear in the parameters and another error function which is nonlinear in the parameters. The procedures which are used to minimize these error functions are discussed in Sections 3.3.1 through 3.3.6.

3.2 Representation of the Ellipse Pattern

The equation of an ellipse whose major and minor axes are coincident with the w,z-coordinate axes, as shown in Figure 2, is given by

\[ g(w,z) = \frac{w^2}{r_x^2} + \frac{z^2}{r_y^2} - 1 = 0 \]  
(3-1)
or

\[ g(w,z) = e_1 w^2 + e_1 z^2 - 1 = 0 \]  \hspace{1cm} (3-2)

where

\[ e_1 = \frac{1}{r^2_x} \quad \text{and} \quad e_2 = \frac{1}{r^2_y} \]  \hspace{1cm} (3-3)

\[ 2r_x = \text{diameter of the ellipse in the } w\text{-direction}, \]

\[ 2r_y = \text{diameter of the ellipse in the } z\text{-direction}. \]

If one wishes to express the equation of this ellipse with respect to an \( x,y \)-coordinate system as shown in Figure 3, the following transformation holds.

\[ w = (x-A) \cos \theta + (y-B) \sin \theta \]  \hspace{1cm} (3-4)

\[ z = - (x-A) \sin \theta + (y-B) \cos \theta \]  \hspace{1cm} (3-5)

The equation for the ellipse in the \( x,y \)-coordinate system then becomes

\[ F(x,y) = g(w,z) \]

\[ = e_1 [(x-A) \cos \theta + (y-B) \sin \theta]^2 \]

\[ + e_2 [-(x-A) \sin \theta + (y-B) \cos \theta]^2 - 1 \]

\[ = [e_1 \cos^2 \theta + e_2 \sin^2 \theta] x^2 + [2(e_1 - e_2) \cos \theta \sin \theta] xy \]

\[ + [e_1 \sin^2 \theta + e_2 \cos^2 \theta] y^2 \]

\[ + [-2e_1 \cos \theta (A \cos \theta + B \sin \theta) - 2e_2 \sin \theta (A \sin \theta - B \cos \theta)] x \]

\[ + [-2e_1 \sin \theta (A \cos \theta + B \sin \theta) + 2e_2 \cos \theta (A \sin \theta - B \cos \theta)] y \]

\[ + [e_1 (A \cos \theta + B \sin \theta)^2 + e_2 (A \sin \theta - B \cos \theta)^2] - 1 \]

\[ \hspace{1cm} (3-6) \]
Figure 2. Example of an Ellipse

Figure 3. Example of a Rotated and Translated Ellipse

Figure 4. Relation of $\theta$ and $\rho$
Equation (3-6) contains five parameters which completely describe the ellipse. These parameters are $e_1$, $e_2$, $A$, $B$, and $\theta$. One notes that Equation (3-6) is a nonlinear function of these parameters.

Equation (3-6) may be transformed into a linear function of a new set of parameters via a nonlinear transformation of the parameters. To this end, let the original parameters be denoted by the vector $\mathbf{c}$ and the new parameters by the vector $\mathbf{p}$, i.e.,

$$
\mathbf{c} = \begin{bmatrix}
e_1 \\
e_2 \\A \\
B \\\theta
\end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix}
p_1 \\
p_2 \\
p_3 \\
p_4 \\
p_5
\end{bmatrix}
$$

and denote the nonlinear transformation by $\mathbf{\psi}$, i.e.,

$$
\mathbf{c} = \mathbf{\psi}(\mathbf{p})
$$

In order to derive the nonlinear transformation of Equation (3-8) one may rewrite Equation (3-6) as

$$
F(x,y) = \rho_1 x^2 + \rho_2 xy + \rho_3 y^2 + \rho_4 x + \rho_5 y + \rho_6 - 1 = 0 \quad (3-9)
$$

where

$$
\begin{align*}
\rho_1 &= e_1 \cos^2 \theta + e_2 \sin^2 \theta \\
\rho_2 &= 2(e_1 - e_2) \cos \theta \sin \theta \\
\rho_3 &= e_1 \sin^2 \theta + e_2 \cos^2 \theta \\
\rho_4 &= -2A(e_1 \cos^2 \theta + e_2 \sin^2 \theta) - 2B(e_1 - e_2) \cos \theta \sin \theta \\
\rho_5 &= -2B(e_1 \sin^2 \theta + e_2 \cos^2 \theta) - 2A(e_1 - e_2) \cos \theta \sin \theta
\end{align*}
$$

(3-10) - (3-14)
\[ \rho_6 = A^2(e_1 \cos^2 \theta + e_2 \sin^2 \theta) + B^2(e_1 \sin^2 \theta + e_2 \cos^2 \theta) + 2AB(e_1 - e_2) \cos \theta \sin \theta \]  

Equations (3-10) through (3-15) may be manipulated to obtain the values of the parameters \( e_1, e_2, A, B, \) and \( \theta \) in terms of the parameters \( \rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \) and \( \rho_6 \).

From Equations (3-10), (3-11), and (3-13)

\[ \rho_4 = -2A\rho_1 - B\rho_2 \]  

while Equations (3-11), (3-12) and (3-14) yield

\[ \rho_5 = -A\rho_2 - 2B\rho_3 \]  

Solving Equations (3-16) and (3-17) simultaneously for \( A \) and \( B \) gives

\[ A = \frac{-2\rho_3 \rho_4 + \rho_2 \rho_5}{4\rho_1 \rho_3 - \rho_2^2} \]  

\[ B = \frac{-2\rho_1 \rho_5 + \rho_2 \rho_4}{4\rho_1 \rho_3 - \rho_2^2} \]

From Equations (3-10) and (3-12) one obtains

\[ \rho_1 + \rho_3 = e_1 + e_2 \]  

and \( \rho_1 - \rho_3 = (e_1 - e_2)(\cos^2 \theta - \sin^2 \theta) \)

\[ = (e_1 - e_2) \cos 2\theta \]

while Equation (3-11) yields

\[ \rho_2 = (e_1 - e_2) \sin 2\theta \]
Equations (3-21) and (3-22) then give

\[ \tan 2\theta = \frac{\rho_2}{\rho_1 - \rho_3} \]  
(3-23)

or

\[ \theta = \frac{1}{2} \tan^{-1} \frac{\rho_2}{\rho_1 - \rho_3} \]  
(3-24)

If a right triangle having sides of length \( \rho_2 \) and \( \rho_1 - \rho_3 \) is formed, then the hypotenuse has a length \( \sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2} \) as shown in Figure 4. From Figure 4 it is apparent that

\[ \sin 2\theta = \frac{\rho_2}{\sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2}} \]  
(3-25)

Then from Equations (3-22) and (3-25) one obtains

\[ \sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2} = e_1 - e_2 \]  
(3-26)

and then Equations (3-20) and (3-26) yield

\[ e_1 = \frac{1}{2} \left[ \rho_1 + \rho_3 + \sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2} \right] \]  
(3-27)

\[ e_2 = \frac{1}{2} \left[ \rho_1 + \rho_3 - \sqrt{\rho_2^2 + (\rho_1 - \rho_3)^2} \right] \]  
(3-28)

Since only five independent parameters are required to fully specify an ellipse, it is reasonable to expect that Equation (3-9) may be simplified. Dividing Equation (3-9) by \( (\rho_6 - 1) \) gives

\[ F(x,y) = p_1 x^2 + p_2 xy + p_3 y^2 + p_4 x + p_5 y + 1 = 0 \]  
(3-29)
where

\[ p_i = \frac{\rho_i}{\rho_s - 1} \quad \text{for } i = 1, 2, 3, 4, 5 \]  

(3-30)

From Equations (3-18), (3-19), and (3-24) one observes that \( A, B, \) and \( \theta \) are ratios of the "\( \rho \)" parameters where both numerators and denominators are of the same order. Thus, the denominator of Equation (3-30) will cancel, making Equations (3-18), (3-19), and (3-24) the same function of the "\( \rho \)" parameters as of the "\( \rho \)" parameters. Therefore

\[ A = \frac{-2p_3p_4 + p_2p_5}{4p_1p_3 - p_2^2} \]  

(3-31)

\[ B = \frac{-2p_1p_5 + p_2p_4}{4p_1p_3 - p_2^2} \]  

(3-32)

\[ \theta = \frac{1}{2} \tan^{-1} \frac{p_2}{p_1 - p_3} \]  

(3-33)

This is not the case for Equations (3-27) and (3-28), however. They become

\[ e_1 = (\rho_s - 1) \left[ \frac{1}{2} (p_1 + p_3 + \sqrt{p_2^2 + (p_1 - p_3)^2}) \right] \]  

(3-34)

\[ e_2 = (\rho_s - 1) \left[ \frac{1}{2} (p_1 + p_3 - \sqrt{p_2^2 + (p_1 - p_3)^2}) \right] \]  

(3-35)
From Equations (3-10), (3-12), and (3-15)

\[ \rho_6 = A^2 \rho_1 + B^2 \rho_3 + AB \rho_2 \]  

(3-36)

Equations (3-30) and (3-36) then give

\[ \rho_6 - 1 = (A^2 \rho_1 + B^2 \rho_3 + AB \rho_2) - 1 \]

\[ = [(\rho_6 - 1)(A^2 \rho_1 + B^2 \rho_3 + AB \rho_2)] - 1 \]  

(3-37)

or

\[ \rho_6 - 1 = \frac{1}{A^2 \rho_1 + B^2 \rho_3 + AB \rho_2 - 1} \]  

(3-38)

Making use of Equations (3-31) and (3-32) further reduces Equation (3-38) to

\[ \rho_6 - 1 = \frac{1}{\frac{p_1 p_5^2 + p_3 p_4^2 - p_2 p_4 p_5}{4 p_1 p_3 - p_2^2} - 1} \]

\[ = \frac{4 p_1 p_3 - p_2^2}{p_1 p_5^2 + p_3 p_4^2 + p_2^2 - p_2 p_4 p_5 - 4 p_1 p_3} \]  

(3-39)

By substituting Equation (3-39) into Equations (3-34) and (3-35) the parameters \( e_1 \) and \( e_2 \) become functions only of the "p" parameters.
\[
e_1 = \frac{4p_1p_3 - p_2^2}{p_1p_5^2 + p_3p_4^2 + p_2^2 - p_2p_4p_5 - 4p_1p_3} \times \left[ \frac{1}{2}(p_1 + p_3 + \sqrt{p_2^2 + (p_1 - p_3)^2}) \right]
\]

\[
e_2 = \frac{4p_1p_3 - p_2^2}{p_1p_5^2 + p_3p_4^2 + p_2^2 - p_2p_4p_5 - 4p_1p_3} \times \left[ \frac{1}{2}(p_1 + p_3 - \sqrt{p_2^2 + (p_1 - p_3)^2}) \right]
\]

If the number found from computing Equation (3-39) is negative then the expressions for \(e_1\) and \(e_2\) as given by Equations (3-40) and (3-41), respectively, should be interchanged as seen by referring to Equations (3-21), (3-22), and (3-24).

The derivation of the transformation

\[
c = \psi(p) = \begin{bmatrix}
\psi_1(p) \\
\psi_2(p) \\
\psi_3(p) \\
\psi_4(p) \\
\psi_5(p)
\end{bmatrix}
\]

has now been completed, with the components \(\psi_1, \psi_2, \psi_3, \psi_4,\) and \(\psi_5\) given by Equations (3-40), (3-41), (3-31), (3-32), and (3-33), respectively.

It was shown that an ellipse may be expressed as a linear function of a set of five parameters or as a
nonlinear function of another set of five parameters as given by Equations (3-29) and (3-6), respectively. The two sets of parameters are related by the transformation given by Equation (3-8). It is now appropriate to investigate methods whereby the unknown parameter vector for an ellipse may be estimated after points on the ellipse have been measured.

3.3 Parameter Estimation Problem

3.3.1 The Error Formulation

In order to estimate the five parameters that represent the size and orientation of the ellipse one may write Equations (3-29) or (3-6) as

\[
F(x_i, y_i; \zeta_0) = 0
\]  

(3-43)

where \((x_i, y_i)\) is any point on the ellipse and

\[
\zeta_0 = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \end{bmatrix}
\]

(3-44)

is the parameter vector which characterizes the ellipse.

If Equation (3-29) is used for describing the ellipse, then

\[
\zeta_0 = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix}
\]

(3-45)
If one measures any point on this ellipse and computes \( F \) using some other parameter vector, \( \zeta \), the value for \( F \) will not be zero, but rather it will be equal to an error, \( \epsilon \). That is,

\[
F(x_i, y_i; \zeta) = \epsilon \tag{3-46}
\]

Likewise, if one makes an error in the measurement of a point on this ellipse, then the computed value for \( F \) using the true parameter vector, \( \zeta_0 \), is again non-zero, representing an error, \( \xi \). That is,

\[
F(\tilde{x}_i, \tilde{y}_i; \zeta_0) = \xi \tag{3-47}
\]

where \((\tilde{x}_i, \tilde{y}_i)\) is now a noisy measurement point.

Now as the boundary of the pattern is traced, a sequence of coordinates \((x_i, y_i)\) becomes available. Given Equation (3-46) and the coordinates of the boundary \((x_i, y_i), i = 1, 2, ..., \) one can estimate a vector \( \tilde{\zeta} \) of the true parameter vector \( \zeta_0 \).

The estimation will be based on minimizing an appropriate function of the error in Equation (3-46). The simplest of these functions appears to be the sum-squared of the error. If \( N \) points on the boundary are available, the sum-squared error is given by
\[ \phi(\hat{x}, \hat{y}; \hat{\zeta}) = \sum_{i=1}^{N} [F(\hat{x}_i, \hat{y}_i; \hat{\zeta}) - F(x_i, y_i; \zeta_0)]^2 \]

\[ = \sum_{i=1}^{N} [F(\hat{x}_i, \hat{y}_i; \hat{\zeta})]^2 \quad (3-48) \]

For convenience, the tilde on \( x_i \) and \( y_i \) will be eliminated from now on. It will be understood that \( (x_i, y_i) \) represent noisy measurement points. Let

\[ \text{Min}\phi(\hat{x}, \hat{y}, \hat{\zeta}) = \phi(\hat{x}, \hat{y}, \zeta_e) \quad (3-49) \]

where \( \hat{x}, \hat{y} \) are \( N \)-dimensional vectors consisting of the \( N \) points which were measured on the ellipse boundary.

By defining an \( N \times 1 \) error vector, \( \hat{e} \)

\[ \hat{e} = F(\hat{x}, \hat{y}; \zeta) = \begin{bmatrix} F(x_1, y_1; \zeta) \\ \vdots \\ F(x_N, y_N; \zeta) \end{bmatrix} \quad (3-50) \]

the sum-squared error may be conveniently written as

\[ \phi(x, y; \zeta) = \hat{e}^T \hat{e} \quad (3-51) \]

where \( T \) denotes transposition.

It should be noted that the criterion function, \( \phi \), corresponding to the sum-squared error is dependent upon whether \( F \) given in Equation (3-50) corresponds to Equation (3-6) or to Equation (3-29). The resulting criterion
functions are not identical. This point is discussed in greater detail in Appendix B.

3.3.2 Error Minimization by Linear Regression Analysis

When the ellipse is given by Equation (3-29) then it is a linear function of the parameter vector, \( \hat{p} \), and the minimization of the sum-squared error reduces to a simple result. The error may be written as

\[ \hat{e} = \hat{F}(x, y; \hat{p}) = M\hat{p} + \hat{1} \]  

(3-52)

where \( M \) is the \( N \times 5 \) matrix

\[
M = \begin{bmatrix}
    x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots \\
    x_N^2 & x_N y_N & y_N^2 & x_N & y_N
\end{bmatrix}
\]  

(3-53)

and \( \hat{1} \) is an \( N \times 1 \) vector containing all 1's. Equation (3-53) indicates that the elements of \( M \) are simply functions of the measured points on the ellipse.

The sum-squared error, \( \phi \), becomes

\[
\phi(\hat{x}, \hat{y}; \hat{p}) = \hat{e}^T \hat{e} \\
= (M\hat{p} + \hat{1})^T (M\hat{p} + \hat{1})
\]  

(3-54)

which is a positive definite quadratic form in the coordinates of the trial parameter vector, \( \hat{p} \). Therefore, one merely needs to compute all of the partial derivatives of \( \phi \) with respect to the components of \( \hat{p} \) and equate them
to zero to find the unique minimum value of $\phi$. Expanding Equation (3-54) gives

$$
\phi = (M^\dagger p)^T (M^\dagger p) + (M^\dagger p)^T \mathbf{I} + \mathbf{I}^T M^\dagger p + \mathbf{I}^T \mathbf{I}
$$

$$
= p^TM^T M^\dagger p + 2p^TM^T \mathbf{I} + N
$$

(3-55)

and therefore

$$
\frac{\partial \phi}{\partial p} \equiv \nabla \phi = 2M^T M^\dagger p + 2M^T \mathbf{I}
$$

(3-56)

and

$$
\frac{\partial \phi}{\partial p} \bigg|_{p=p^e} = 0 = 2M^T M^\dagger p^e + 2M^T \mathbf{I}
$$

(3-57)

or

$$
M^T M^\dagger p^e = -M^T \mathbf{I}
$$

(3-58)

The matrix $M^TM$ may be inverted, assuming $M$ is of full rank, to give

$$
p^e = -[M^TM]^{-1} M^T \mathbf{I}
$$

(3.59)

This estimate is then the "least squares estimate" of the true parameter vector, $p^e$, and shall be referred to as the one step minimization method.

It should be noted that the simple expression for $p^e$ as given in Equation (3-59) would not have resulted had the criterion function been something other than quadratic in the parameter vector components, since the
differentiation would have yielded a nonlinear relation in the parameter vector components.

If the parameter vector, \( \hat{c}_0 \), is to be estimated directly from Equation (3-6) some other technique than Equation (3-59) must be employed since the parameters enter Equation (3-6) in a nonlinear manner, making the criterion function, \( \phi \), no longer quadratic in the parameters. A complete automatic computer algorithm to estimate parameter vectors for this sort of problem has been developed by R. B. McGhee [45], [46] and shall be utilized here. This is an iterative minimization scheme rather than a one step minimization scheme such as was associated with the linear regression analysis. The essence of this computer algorithm is discussed below.

3.3.3 Gauss-Newton Iteration

If the nonlinear response vector, \( \hat{F} \), has its Taylor series expansion truncated after the linear term

\[
\hat{F}(c_1 + \Delta c) = \hat{F}(c_1) + \left. \frac{\partial \hat{F}}{\partial c} \right|_{c=c_1} \Delta c
\]  

(3-60)

or

\[
\hat{F}(c_1 + \Delta c) = \hat{F}(c_1) + Z \Delta c
\]  

(3-61)
then the criterion function, $\hat{\phi}$, associated with this response function is

$$\hat{\phi}(\hat{x}, \hat{y}; \Delta \hat{c}, \Delta \hat{c}_1) = \hat{F}^T \hat{F}$$

(3-63)

where $\hat{F}$ is defined in Equation (3-60) and

$$\hat{\phi}(\Delta \hat{c}) = (\hat{e} + Z \Delta \hat{c})^T (\hat{e} + Z \Delta \hat{c})$$

(3-64)

which is a quadratic form in $\Delta \hat{c}$. Expanding Equation (3-64) gives

$$\hat{\phi}(\Delta \hat{c}) = \hat{e}^T \hat{e} + \hat{e}^T Z \Delta \hat{c} + \Delta \hat{c}^T Z^T \hat{e} + \Delta \hat{c}^T Z^T Z \Delta \hat{c}$$

(3-65)

If Equation (3-65) is differentiated with respect to $\Delta \hat{c}$ and the result equated to zero, the minimizing value of $\Delta \hat{c}$ becomes

$$\left. \frac{\partial \hat{\phi}}{\partial \Delta \hat{c}} \right|_{\Delta \hat{c} = \Delta \hat{c}_1} = 0 = 2 \hat{e}^T + 2 \hat{e}^T Z \Delta \hat{c}_1$$

(3-66)

or

$$\Delta \hat{c}_1 = -(Z^T Z)^{-1} Z^T \hat{e}$$

(3-67)
assuming that $z^T z$ is nonsingular. The matrix

$$S = z^T z \quad (3-68)$$

is referred to as the regression matrix due to the similarity of this method to linear regression analysis.

The normal equation for iteration, Equation (3-66), is linear in $\Delta \hat{c}$ only because function $F$ was linearized and a quadratic criterion function was chosen.

Since

$$\nabla \phi (c) = \frac{\partial \phi}{\partial \hat{c}} \bigg| \begin{array}{c} \hat{c} = c_1 \\ \frac{\partial \phi}{\partial \hat{c}} \bigg| \begin{array}{c} \frac{\partial \phi}{\partial \hat{c}} \\ \hat{c} = c_1 \end{array} \end{array} \bigg| \begin{array}{c} \frac{\partial \phi}{\partial \hat{c}} \\ \hat{c} = c_1 \end{array} = 2 \frac{\partial e}{\partial \hat{c}}^T e = 2z^T e \quad (3-69)$$

Equation (3-67) becomes

$$\Delta \hat{c}_1 = -\frac{1}{2} S^{-1} \nabla \phi (c) \equiv \beta_1 \quad (3-70)$$

The new value for the parameter vector, $\hat{c}$, then is

$$\hat{c}_2 = \hat{c}_1 + \Delta \hat{c}_1 \quad (3-71)$$

upon which a new iteration may then be initiated. This procedure is referred to as the "Gauss-Newton" iteration method [45].

As mentioned previously, Equation (3-70) is based on the assumption that linearizing function $F$, Equation (3-60), is valid. Since, in fact, this may be completely invalid, it is quite possible that the sequence of parameter vector estimates, given by Equation (3-71),
will not converge to \( \hat{c}_0 \). The necessary and sufficient conditions for the convergence of the Gauss-Newton procedure may be derived; however, the test is generally complicated enough that in practice one simply computes \( \phi(\hat{c}_i + \delta_i) \) at each step to see if an improvement results.

It can be shown that the Gauss-Newton iteration always converges when binary scale factor adjustment is used [45]. When this technique is used, \( \hat{c}_{i+1} \) is found from

\[
\hat{c}_{i+1} = \hat{c}_i + 2^{-k} \delta_i
\]  

(3-72)

where \( k \) is the first non-negative integer which reduces \( \phi \). However, experimental results show that the rate of convergence can be quite slow. For this reason, the "modified" Gauss-Newton procedure will not be used. The Gauss-Newton iteration enjoys its greatest success as a terminal iterative technique, where the current parameter vector is "close" to its minimizing value.

3.3.4 Newton-Raphson Iteration

When the Gauss-Newton iteration fails to give a reduced value for the criterion function, \( \phi \), then direct gradient techniques may be appropriate. This eliminates the necessity of inverting the matrix, \( S \), and also makes it possible to handle parameter range constraints in a straightforward manner.
The gradient technique to be employed is the method of steepest descent, in which case the parameter change vector, $\Delta \hat{c}$, is directly proportional to the negative gradient of the criterion function, $\phi$.

$$\Delta \hat{c}_i = -k \nabla \phi(\hat{c}_i)$$

(3-72)

where

$$k > 0$$

(3-73)

Thus, $\Delta \hat{c}_i$ is in the direction of the greatest rate of decrease of $\phi$. The next parameter vector estimate then becomes

$$\hat{c}_{i+1} = \hat{c}_i + \Delta \hat{c}_i$$

(3-74)

which may then be used to perform another iteration.

Before Equation (3-72) can be utilized, it is necessary to choose some value for the scale factor, $k$. The "Newton-Raphson" method may be used to obtain a value for $k$. Essentially it is based on taking the linear portion of the Taylor series expansion of the criterion function, $\phi$, and extrapolating this to zero. More precisely, suppose that $\phi$ is a sufficiently smooth function of $\hat{c}$ such that it may be represented locally by the Taylor series

$$\phi(\hat{c} + \Delta \hat{c}) = \phi(\hat{c}) + \nabla \phi^T \Delta \hat{c} + O(\Delta \hat{c}^2)$$

(3-75)

where $O(\Delta \hat{c}^2)$ represents all the terms in the series which are quadratic or higher order in $\Delta \hat{c}$. Then for small $\Delta \hat{c}$, $O(\Delta \hat{c}^2)$ may be ignored, and
\[ \phi(\hat{c} + \Delta \hat{c}) = \phi(\hat{c}) + \nabla \phi^T \Delta \hat{c} \]  

or

\[ \phi(\hat{c} + \Delta \hat{c}) = \phi(\hat{c}) - k |\nabla \phi|^2 \]  

using Equation (3-72). Extrapolating \( \phi \) to zero, for which \( k=k^0 \), gives

\[ 0 = \phi(\hat{c}_1) - k_1 |\nabla \phi(\hat{c}_1)|^2 \]  

or

\[ k_1 = \frac{\phi(\hat{c}_1)}{|\nabla \phi(\hat{c}_1)|^2} \]  

The corresponding parameter change vector, \( \Delta \hat{c}_i \), at each state of iteration is then, from Equation (3-72),

\[ \Delta \hat{c}_i = - \frac{\phi(\hat{c}_i) \nabla \phi(\hat{c}_i)}{|\nabla \phi(\hat{c}_i)|^2} \]  

It may well be the case that iteration based on Equation (3-80) will give a value for \( \phi(\hat{c} + \Delta \hat{c}) \) which is larger than \( \phi(\hat{c}) \). This simply means that \( \Delta \hat{c} \) is so large that linear extrapolation of \( \phi \) to zero is, in fact, invalid. Equation (3.75) guarantees that for some \( 0 \leq k \leq k^0 \), the criterion function, \( \phi \), will be reduced, however. Thus it is desired to find some \( k=k^* \) such that

\[ \min_{k>0} \phi(\hat{c}_i - k_1 \nabla \phi(\hat{c}_i)) = \phi(\hat{c}_i - k^*_1 \nabla \phi(\hat{c}_i)) \]  

(3-81)
The next parameter vector estimate is then

\[ \hat{\mathbf{c}}_{i+1} = \hat{\mathbf{c}}_i - k_i \nabla \phi(\hat{\mathbf{c}}_i) \]  

(3-82)

This iteration scheme is called the "optimum gradient method".

It is, of course, not feasible to search over all values of \( k \) on a computer, but it is quite feasible to perform a binary search over the range

\[ 0 \leq k \leq \frac{\phi}{|\nabla \phi|^2} = k^0 \]  

(3-83)

which may be considered a "suboptimum gradient method". Assuming that \( \phi \) is continuous and \( \nabla \phi \neq 0 \), Equation (3-75) guarantees that there exists an \( n \) such that

\[ \phi(\hat{\mathbf{c}}_i - \frac{1}{2^n} k^0 \nabla \phi) < \phi(\hat{\mathbf{c}}_i) \]  

(3-84)

and therefore a binary search procedure always produces a convergent sequence of values for \( \phi \). A simple algorithm may be constructed to find the minimizing value for \( n \) as follows. First of all, compute \( \Delta \hat{\mathbf{c}}_i^0 \) from Equation (3-80). Then, for \( n = 0, 1, \ldots \), evaluate

\[ \phi_i^n = \phi(\hat{\mathbf{c}}_i + \frac{1}{2^n} \Delta \hat{\mathbf{c}}_i^0) \]  

(3-85)

Once a value for \( n \) is reached, say \( n = m \), such that

\[ \phi_{i,m-1}^n > \phi_i^m < \phi_{i,m+1}^n \]  

(3-86)
and

\[
\phi_i^m < \phi_i \tag{3-87}
\]

then take for the new value of \( \bar{c} \)

\[
\bar{c}_i + 1 = \bar{c}_i + \frac{1}{2^m} \Delta \phi_i \tag{3-88}
\]

If Equation (3-87) is not satisfied, continue increasing \( n \) until Equation (3-86) is again satisfied and then check Equation (3-87) once again. Continue this until both Equations (3-86) and (3-87) are satisfied. Equation (3-88) is then the new value for \( \bar{c} \).

A further refinement may be incorporated into this algorithm by fitting a quadratic function to the points \( \phi_i^{m-1}, \phi_i^m \) and \( \phi_i^{m+1} \). Letting \( \phi_i^{m-1} = \phi_0, \phi_i^m = \phi_1 \) and \( \phi_i^{m+1} = \phi_2 \), it is straightforward to show that the minimum of this fitted quadratic function occurs at

\[
k_{\text{quad}} = \frac{3k_i^* \phi_2 - 5\phi_1 + 4\phi_0}{4 \phi_2 - 3\phi_1 + 2\phi_0} \tag{3-89}
\]

where

\[
k_i^* = \frac{1}{2^m} k_i^0 \tag{3-90}
\]

If

\[
\phi(c_i - k_{\text{quad}} \hat{v}) < \phi(c_i - k_i^* \hat{v}) \tag{3-91}
\]
then the parameter change vector is

$$\Delta \hat{c}_i = - k_{quad} \vec{\nabla}_\phi (\hat{c}_i) \quad (3-92)$$

Otherwise,

$$\Delta \hat{c}_i = - k_i \vec{\nabla}_\phi (\hat{c}_i) \quad (3-93)$$

3.3.5 Range Constraints

It is necessary to place constraints on the parameters since we assume the ellipse to be the field of view. These are range constraints, in which each component of the parameter vector is independently restricted to lie within some specified interval on the number scale. Thus, each component, $c_i$, must satisfy

$$c_{i\text{ min}} \leq c_i \leq c_{i\text{ max}} \quad (3-94)$$

where $c_{i\text{ min}}$ and $c_{i\text{ max}}$ are the lower and upper limits of the allowed range, respectively. This then means that the parameter vector, $\hat{c}$, which minimizes the criterion function, $\phi$, must be in a hypercube in parameter space.

Considering the problem at hand, one notes that in order for Equation (3-2) to represent an ellipse it is necessary that $e_1$ and $e_2$ (or $c_1$ and $c_2$, respectively) be positive. Likewise, the finite field of view of the optical equipment places constraints on $e_1$ and $e_2$ as well
as the translations A and B (or $c_3$ and $c_s$, respectively). Since an ellipse is symmetric about its two axes, the rotation angle, $\theta = c_s$, may be constrained to lie in the first quadrant.

Since the gradient-descent method discussed earlier is valid only on the interior of the 5-dimensional constraint region, $R$, it is necessary to use a different strategy when a constraint boundary is encountered during a gradient-descent. The method to be used is called the gradient-projection method. If a constraint boundary should be encountered, this method projects the gradient onto the constraint surface and then travels in the negative direction of the projected gradient until a minimum for $\phi$ is found. The actual minimum for $\phi$ may either be located on the interior of $R$ or on a constraint boundary of $R$. In the latter case the projection of the gradient will have all of its components equal to zero at the final iteration.

The mechanization of the gradient-projection method may be performed in three steps.

1. Check each component of the current parameter vector estimate, $\hat{c}$, to see whether it is within the allowed range or if it lies at the lower or upper end of the range.
2. If any component lies on either extreme of its range, and if the negative of the corresponding gradient component points out of the constraint region, then set this component of the gradient equal to zero. Leave all other components of the gradient at their true value.

3. The resulting vector, $\tilde{\mathbf{v}}_p$, is the desired projected gradient.

In order to find the optimum step size, it is necessary to find the maximum scale factor which can be applied to $-\tilde{\mathbf{v}}_p$ without violating a range constraint. To this end, suppose that $\frac{\partial \phi_p}{\partial c_j}$ is positive. This means that $c_j$ can be reduced in value without violating a range constraint. Let $k_j^0$ be the largest scale factor that can be applied to the negative $j^{th}$ gradient component without violating the $j^{th}$ range constraint. Then $k_j^0$ satisfies

$$c_j - k_j^0 \frac{\partial \phi_p}{\partial c_j} = c_j^{\text{min}}$$

which gives

$$k_j^0 = \frac{c_j - c_j^{\text{min}}}{\frac{\partial \phi_p}{\partial c_j}}$$

Likewise, for the negative components of $\tilde{\mathbf{v}}_p$ it follows
that

$$k_j^0 = \frac{c_j - c_j^{\text{max}}}{\frac{\partial \phi_p}{\partial c_j}}$$ \hspace{1cm} (3-97)

The maximum scale factor, $k_0$, is found from

$$k_0 = \min_{j} \{k_j^0\}$$ \hspace{1cm} (3-98)

where the $k_j^0$ are defined only for the non-zero components of $v_p$.

The maximum step size for the parameter change vector now becomes

$$\Delta c_j = - k_0 v_p$$ \hspace{1cm} (3-99)

This value may then be used as the maximum step size for the binary search procedure discussed earlier.

3.3.6 Global Optima

Both the Gauss-Newton method and the Newton-Raphson method are suited for determining local minima since they make use only of local information. If more than one minimum is contained within constraint region, $R$, then it is desirable to find the smallest of all of these minima. Such a minimum is called a global minimum. Of course, the only way to find the global minimum with certainty is to exhaustively search the entire constrained parameter space. Since this is not feasible or practical
on a computer, one must choose some method whereby a
given confidence level is attained that the minimizing
parameter vector obtained is associated with a value of
ϕ which is smaller than some specified per cent of the
points in R. Such a method is that of uniform random
searching in which parameter vectors are chosen at random
(with a uniform distribution for each component) and their
corresponding criterion functions are evaluated. The
parameter vector, \( \hat{c} \), being associated with the smallest
value for \( \phi \) is then used to initiate a local minimization
[45].

3.4 Summary

The objectives of this chapter have been threefold. First, the equation of a pattern (in particular, an
ellipse) has been developed as a linear function of
tive parameters and as a nonlinear function of a
different set of five parameters. An error function
is then defined for either of these functions. Finally,
minimization techniques are presented which minimize
the sum-squared error function corresponding to the
appropriate equation for the ellipse, yielding an estimate
for the characterizing parameters of the ellipse.
CHAPTER IV
RECOGNITION OF ELLIPSES

4.1 Introduction

This chapter is concerned with the recognition of patterns whose boundaries are in the form of an ellipse. The term "recognition" is used in the sense that the five parameters which characterize an ellipse are to be estimated. After the five parameters have been estimated, the associated ellipse is then judged to be the ellipse from which the data points originated.

Section 4.2 discusses the statement of the problem and the general approach which is to be pursued when the error function is simply the equation for an ellipse. Two minimization schemes are presented which use this error function. The various factors which are associated with the implementation of these two minimization schemes are discussed in Section 4.3. The results which were obtained from these two minimization schemes are presented in Section 4.4.

Section 4.5 investigates another recognition scheme in which the associated error function is the radial
distance (as measured from the origin of the fitted ellipse, or "template") for the data points from the ellipse template. This scheme uses first and second moments of the data points to provide an initial estimate for the parameter vector. The results of using this recognition scheme (using the same data points as in Section 4.4) is discussed in Section 4.6.

Finally, Section 4.7 contains a brief summary of the results and conclusions which have been reached in this chapter.

4.2 Statement of Problem

The parameter estimation schemes were first applied to the recognition of elliptical patterns. Elliptical patterns were selected first because they are a somewhat complex pattern and yet their boundary may be represented analytically. Furthermore, most of the ground work for the estimation of the parameters of an ellipse has been laid in Chapter III.

As was discussed in Chapter III, an ellipse, located in a plane, may be fully characterized by five parameters. Two parameters, $e_1$ and $e_2$, are necessary to specify the size and shape of an ellipse, while three parameters are necessary to specify its position and orientation in the plane. Figure 5 shows a typical ellipse which has been translated and rotated with respect to the reference
With respect to the \( w,z \)-coordinate system, this ellipse may be expressed analytically by Equation (4-1)

\[
e_1 w^2 + e_2 z^2 = 1 \tag{4-1}
\]

where

\[
e_1 = \frac{1}{r_x^2} \quad \text{and} \quad e_2 = \frac{1}{r_y^2} \tag{4-2}
\]

The parameters \( r_x \) and \( r_y \) are respectively called the \( w \)-axis radius and the \( z \)-axis radius of the ellipse. The parameters which are actually estimated are \( e_1 \) and \( e_2 \), which are related to \( r_x \) and \( r_y \) by Equation (4-2). The \( x \)-and \( y \)-translation parameters are denoted by \( A \) and \( B \), respectively.
and the rotation parameter is denoted by $\theta$. These parameters are all shown in Figure 5.

The parameter vector, $\tilde{c}$, which characterizes an ellipse is given by Equation (4-3).

$$\tilde{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ A \\ B \\ \theta \end{bmatrix}$$

(4-3)

It is, then, the intent of the next two sections to determine the feasibility of recognizing an elliptical planar pattern by estimating its associated parameter vector $\tilde{c}$, and to determine whether the one step minimization method or the iterative minimization scheme does the better job of performing this parameter estimation task.

4.3 Implementation of the Parameter Estimation Schemes

To simulate an ellipse in the field of view, points which lie on the boundary of an ellipse are artificially generated by the subroutine denoted by DATA. The logic by which this subroutine selects the data points is discussed in Appendix A. After the subroutine DATA is provided with parameter vector $\tilde{c}_0$, it generates data points which lie on the boundary of an ellipse which is characterized by $\tilde{c}_0$. This parameter vector was arbitrarily chosen to be
This corresponds to an ellipse which has a \( w \)-axis radius and a \( z \)-axis radius equal to 2.0 and 1.0, respectively. In addition, the ellipse has been translated one unit in the \( x \)-direction and two units in the negative \( y \)-direction, and rotated 0.5 radians.

Thus, in the absence of measurement noise, one would expect the estimate for \( \hat{c} \) to be exactly \( \hat{c}_o \).

While the parameter vector \( \hat{c}_o \) was held fixed, two other parameters were varied to determine their effect on the accuracy of the parameter estimation schemes.

One of these variable parameters was the number of data points which were used to represent the boundary of the ellipse. Ten data points were chosen for a sparse distribution of points on the boundary, while one hundred data points were chosen for a dense distribution of points on the boundary. Intermediate values for the number of data points were chosen as 20 and 50.

The other variable parameter was the amount of noise which was added to the data points to simulate the effect of measurement noise or other errors. The noise samples, which are generated on the digital computer, have a gaussian distribution. The mean and standard deviation.
of these noise samples may be independently specified. In all cases the mean was chosen to be zero, while the standard deviation was either 0.0, 0.1, 0.2, 0.3, 0.4, or 0.5. The maximum value for the standard deviation of the noise sample, 0.5, was one-half of the z-axis radius of the noiseless ellipse. Noise samples having a standard deviation larger than 0.5 result in the data points having such a large scatter that they no longer even remotely resemble the boundary of an ellipse. In fact, physical systems which correspond to the higher values of standard deviation (0.3 - 0.5) would have limited practical utility, but it is of interest to investigate the reliability of the parameter estimation schemes for high noise levels, and to develop bounds on the performance of such systems.

The pattern which is to be recognized is required to lie within some bounds since in a physical situation the optical system would have a finite field of view. The field of view was arbitrarily chosen to be a square measuring eight units on a side. The boundary of the ellipse which is characterized by the parameter vector \( \mathbf{c}_0 \) given by Equation (4-4) lies entirely within this field of view.

A remark should be made at this point. If the noise which is added to the data points has a large standard
deviation, it is possible that some of the resulting noisy data points will fall outside of the field of view. When this situation arises, those noisy data points which fall outside of the field of view are still regarded as valid data points in the simulation. Physically, in an actual landmark tracking or automatic docking situation, noise may be classified into two general categories. The first category consists of noise associated with measurement errors. These include grid quantization errors, detector or sensor errors, and transmission errors. In any of these cases the coordinates of a data point (which is in the optical field of view) will be in error, and if the true data point is near the boundary of the field of view then it is possible that the noisy, or measured, data point will have coordinates which lie outside of the field of view. This situation is contrasted to the second-category into which noise may be classified, which may be termed "masking" noise for lack of a better name. This kind of noise corresponds to a case in which the field of view is partially covered with clouds or to a case in which the optical system is very badly out of focus. The sensor will be unable to detect the data points which are masked, or obscured, due to either of these situations, and therefore these data points are in essence, discarded. This
masking noise has the effect, therefore, of shrinking the field of view.

Thus, it can be seen that the noise which is being simulated corresponds to measurement noise rather than "masking" noise.

The numerical values utilized in simulation experiments for the range constraints for the five ellipse parameters are as follows:

\[

e_1 \in (0, 0.0625, 16.0) \\
e_2 \in (0, 0.0625, 16.0) \\
A \in (-4.0, 4.0) \\
B \in (-4.0, 4.0) \\
\theta \in (0, 1.57)
\]

These range constraints permit the fitted ellipse to have either of its radii range in size from 1/4 unit to 4 units (i.e., the maximum diameter is constrained to be no larger than the dimensions of the field of view). In addition, the center of the fitted ellipse is permitted to lie anywhere within the field of view, while the rotation angle is constrained to lie in the first quadrant due to the symmetry of an ellipse.

The estimation process has to be started with an arbitrary initial parameter vector, \( \bar{c}_e \). The initial guess for the parameter vector was
\[
\hat{c}_e = \begin{bmatrix} 0.5 \\ 1.3 \\ 0.7 \\ -2.5 \\ 0.8 \end{bmatrix}
\] (4-6)

which corresponds to an ellipse having a w-axis radius and a z-axis radius equal to 1.414 and 0.877, respectively.

The parameter vector \( \hat{c}_e \) was chosen such that its components were "close" in value to the components of \( \hat{c}_o \) and yet not so close as to make the estimation problem trivial.

After performing a local minimization using \( \hat{c}_e \) as the initial estimate for \( \hat{c}_o \), four more local minimizations are executed with the initial estimate in each case being found by the RANSER (random search) subroutine. [45] Thus, a total of five trial local minimizations are carried out. The number of random searches for each trial local minimization was set equal to 100.

4.4 Results

Both the one step minimization method and the iterative minimization scheme which are outlined in Chapter III were employed to estimate the parameters of the given ellipse. The results which were obtained by using these two schemes are shown in Tables 1 and 2, respectively. It should be pointed out that the data points are exactly the same for both minimization schemes, permitting a meaningful comparison to be made. The results which are tabulated in
Table 2 are also shown pictorially in Plates I, II, III, IV, V, and VI. In these plates the ellipse having a solid line boundary corresponds to the parameter vector $\mathbf{c}_o$. The "+" symbols correspond to the noisy data points arising from the solid line boundary. The ellipse which is fitted to these noisy data points, and characterized by $\mathbf{c}_e$, is represented by the dashed line boundary. The x and y-radii correspond to the w-axis and z-axis radii, respectively.

A comparison of Tables 1 and 2 shows that for noise levels below $\sigma = 0.4$ the two minimization schemes produced results which were quite similar. Referring to these tables or to Plate I one notes that for noiseless data points the parameter vector is estimated precisely, that is, $\mathbf{c}_e = \mathbf{c}_o$. This is a criterion which any good recognition scheme should fulfill, of course.

Plate II shows the results which were obtained for $\sigma = 0.1$. Special note should be made concerning the accuracy with which the rotation component of the parameter vector was estimated. It is seen that the largest error is less than three degrees, while for three of the four cases this error is considerably less than one degree.

A brief comment concerning the expression for the error should be made at this point. In most instances it is more convenient and meaningful to express an error in percentage rather than absolute terms. Such is the
case here. Since an ellipse is symmetric about both its vertical and horizontal axes, its angular position is unique only in the first quadrant, i.e., 0 to 90 degrees. The percentage error may then be defined as the ratio of the absolute error to 90 degrees. With the percentage error so defined, one can see that for $\sigma = 0.1$ the maximum error is approximately three percent for the rotation parameter, which is quite good considering the fact that the reference rotation angle is not constrained to be small.

The results for $\sigma = 0.2$ are shown in Plate III. Here again one notes that the error in estimating the rotation parameter is quite good. In fact, ignoring the 10 data point case, the maximum error is still less than three percent. In the 10 data point case the error is approximately seven percent, which is still reasonable considering the scarcity of data points and the noise level. Another observation which can be made from both Plates II and III is that the estimates for the parameter vector become better as more data points are used, a situation which intuitively seems reasonable.

When the noise level reaches $\sigma = 0.3$, as shown in Plate IV, the fitted ellipses begin to differ from the reference ellipses to a larger, and perhaps unacceptable, extent. The scatter of the data points is such that an
accurate fit cannot be realized by either of the parameter estimation schemes. However, it should be pointed out that the estimate for the rotation parameter is still respectable, except for the 10 data point case. In the other cases the maximum error in the rotation parameter estimate is less than nine percent, and for the 100 data point case this error is approximately three percent (for the iterative minimization scheme).

For higher noise levels ($\sigma = 0.4$ and 0.5) the one step minimization method runs into serious difficulties. Table 1 shows two instances in which the one step minimization method was unable to fit an ellipse to the data points. In both of these instances the estimate $c_e$ was found to have one of its first two components a negative number. This means that the one step minimization method actually fit a hyperbola to the given data points rather than an ellipse.

For relatively high levels of noise ($\sigma = 0.4$ and 0.5) Table 2 shows that the iterative minimization scheme also exhibits an undesirable characteristic; that is, it has a tendency to select values for the first two components of $c_e$ which are at the boundary of their respective range constraints. When this is the case the resulting fitted ellipse is actually a circle having a radius equal to four units, as shown in Plates V and VI. These plates
indicate that the iterative minimization scheme has attempted to cluster all of the data points along a small portion of the boundary of the fitted ellipse (or circle), with approximately one half of the data points on either side of the fitted ellipse's boundary.

4.5 Fitting an Ellipse Template to Ellipse Data Points

This section shall consider a recognition scheme in which an ellipse template, which is specified by its associated parameter vector, is placed over the data points (conceptually, that is). A measure of the physical misalignment of the template with the data points is defined, and this misalignment error is used in the sum-squared error criterion function associated with the nonlinear regression analysis technique discussed in Chapter III. The parameters of the template are automatically adjusted, in an iterative fashion, until this criterion function is minimized.

This recognition scheme shall now be discussed in greater detail. First of all, the ellipse template which is initially chosen to fit the data points is obtained by computing the first and second moments of the data points. From these moments an initial guess is supplied for the five parameters which define the ellipse template. This procedure is mechanized in the ESTIMT subroutine, and is
discussed in more detail in Appendix C.

The parameter vector which is to be estimated has slightly different components than was the case in Sections 4.2, 4.3, and 4.4. Specifically, the first two components of the parameter vector are now taken to be the two radii of an ellipse, rather than the reciprocal of the square of the respective radii. Thus

\[ c = \begin{bmatrix} \frac{1}{x_A} \\ \frac{1}{x_B} \\ \theta \end{bmatrix} \]  

(4-7)

The error associated with fitting a given ellipse template to a set of data points may be clarified by considering Figure 6.

---

Figure 6. Determination of Ellipse Model Point Associated with a Given Data Point
The ellipse template shown in Figure 6 is either the initial guess for the ellipse to which the data points belong, or it is the template corresponding to the current estimate for the parameter vector. Only one data point is shown for illustrative purposes.

The w,z-reference frame, which corresponds to the principal axes for the ellipse template, is related to the x,y-reference frame by the transformation

\[ \begin{align*}
    w &= (x - A) \cos \theta + (y - B) \sin \theta \\
    z &= -(x - A) \sin \theta + (y - B) \cos \theta 
\end{align*} \]  

(4-8)

where A, B, and \( \theta \) are the current estimates for the translation and rotation parameters. Thus, a data point having coordinates \((x_d, y_d)\) in the x,y-reference frame has coordinates \((w_d, z_d)\) in the w,z-reference frame given by

\[ \begin{align*}
    w_d &= (x_d - A) \cos \theta + (y_d - B) \sin \theta \\
    z_d &= -(x_d - A) \sin \theta + (y_d - B) \cos \theta 
\end{align*} \]  

(4-9)

If a radial line is drawn from the origin of the w,z-reference frame to a given data point, as shown in Figure 6, then the angle of inclination of this line with respect to the w-axis is \( \psi \), which is given by

\[ \tan \psi = \frac{z_d}{w_d} \]  

(4-10)
The point of intersection of this line with the ellipse template also has the same angle of inclination. Thus

\[ \tan \psi = \frac{z_m}{w_m} \]  \hspace{1cm} (4-11)

where the "m" subscript denotes that the point is on the template (or model). Combining Equations (4-10) and (4-11) then gives

\[ z_m = w_m \times \frac{z_d}{w_d} \]  \hspace{1cm} (4-12)

All that remains is to find the coordinates of this point of intersection with respect to the five parameters. To this end, recall that the equation of the ellipse template is given by

\[ \frac{w_m^2}{r_x^2} + \frac{z_m^2}{r_y^2} = 1 \]  \hspace{1cm} (4-13)

Substituting Equation (4-12) into Equation (4-13) yields

\[ \frac{w_m^2}{r_x^2} + \frac{w_m^2 \left( \frac{z_d}{w_d} \right)^2}{r_y^2} = 1 \]  \hspace{1cm} (4-14)
or

\[ w_m = \pm \frac{r_x r_y}{\sqrt{r_y^2 + r_x^2 \left( \frac{z_d}{w_d} \right)^2}} \quad (4-15) \]

The ± sign which is associated with Equation (4-15) is resolved when it is known in which quadrant the data point lies. Specifically,

\[ w_m = + \frac{r_x r_y}{\sqrt{r_y^2 + r_x^2 \left( \frac{z_d}{w_d} \right)^2}} \quad \text{if} \quad \begin{cases} 0 \leq \psi \leq \frac{\pi}{2} \\ \frac{3\pi}{2} \leq \psi < 2\pi \end{cases} \quad (4-16) \]

and

\[ w_m = - \frac{r_x r_y}{\sqrt{r_y^2 + r_x^2 \left( \frac{z_d}{w_d} \right)^2}} \quad \text{if} \quad \frac{\pi}{2} < \psi < \frac{3\pi}{2} \quad (4-17) \]

The corresponding z-coordinate of the model point is then given by Equation (4-12).

The x,y-coordinates of the model point may be determined from the inverse transformation of Equation (4-9). That is,

\[ x_m = w_m \cos \theta - z_m \sin \theta + A \]
\[ y_m = w_m \sin \theta + z_m \cos \theta + B \quad (4-18) \]
Equation (4-18) gives the x,y-coordinates of the model point which corresponds to the data point \((x_d, y_d)\). The error is defined to be the difference in the x-and y-coordinates of these two points. Therefore

\[ e_i = (x_d - x_m)_i \]
\[ e_{i+N} = (y_d - y_m)_i \]  

(4-19)

where the subscript "i" denotes a given data point, and \(N\) is the total number of data points. The sum-squared error criterion function is then

\[ \phi = \sum_{i=1}^{2N} e_i^2 \]

(4-20)

The error function has now been defined in terms of the five parameters which characterize the ellipse template. After the appropriate partial derivatives of the coordinates \((x_m, y_m)\) are computed, the Gauss-Newton or Newton-Raphson minimization scheme may be used to determine the "best" fitting ellipse template. This is discussed in more detail in Appendix D, where a listing of these partial derivatives is given.
4.6 Implementation of the Ellipse Template

The recognition scheme which is discussed in the previous section was employed to fit an ellipse template to the same sets of data points which are considered in Section 4.4 for the other two recognition schemes. Table 3 lists the parameter estimates which were obtained for each set of data points.

Comparing Table 3 with Tables 1 and 2, one notes that for low noise levels all of the recognition schemes yield results which are quite similar. However, for high noise levels ($\sigma = 0.4$ and 0.5) Table 3 shows that fitting an ellipse template to the data points results in much better estimates for the parameters. Due to the manner in which the error function is defined for the ellipse template, no erratic estimates occurred for these high noise levels as they did using the other two recognition schemes.

4.7 Summary

This chapter has investigated the merits of "recognizing" a planar elliptical pattern, whose boundary points are given, by estimating the values for the five parameters which characterize an ellipse. The ellipse which was to be recognized was permitted to have arbitrary size and shape, as well as arbitrary position and
orientation so long as it was located within the specified field of view.

As was pointed out in Sections 4.4, and 4.6, all three minimization schemes provided essentially the same results for the estimate of the parameter vector associated with the reference ellipse when the noise level was below $\sigma = 0.4$. For the lower noise levels ($\sigma = 0.0$, 0.1, and 0.2) these estimates were quite good, and special note was made concerning the accuracy with which the rotation parameter was estimated. Excluding the 10 data point case for $\sigma = 0.2$, the rotation parameter was never more than three percent in error, which is a remarkable result. Unfortunately, no other schemes exist presently with which these results can be compared.

For $\sigma = 0.3$ the estimate for the rotation parameter was still respectable, but the other parameters were not estimated accurately enough to yield a fitted ellipse which approximated the reference ellipse to an acceptable degree.

Both minimization schemes discussed in Section 4.4 displayed undesirable characteristics for very high noise levels ($\sigma = 0.4$ and 0.5). The one step minimization method had a tendency to fit a hyperbola to the data points rather than an ellipse (characterized by a negative value for one of the first two components of the parameter vector).
while the iterative minimization scheme had a tendency to fit a constrained ellipse to the data points \( r_x = r_y = 4.0 \).

The fact that both minimization schemes failed to accurately estimate the parameters associated with the reference ellipse for high noise levels does not distract from their usefulness. In practice one would regard a system corresponding to \( \sigma = 0.3, 0.4 \) and 0.5 as having an unacceptable level of noise and hence would demand a better design for the system. Upon viewing Plates IV, V, and VI one sees that it would be very difficult, if not impossible, to develop a recognition scheme that could accurately recognize an ellipse from the given scatter of data points (this includes a human being as a "pattern recognizer").

As a minor point, it should be mentioned that the undesirable characteristics of the two minimization schemes (for high noise) which were previously mentioned can be corrected to some extent. The iterative minimization scheme can be improved if the range constraints on the first two components of \( \mathbf{c}_e \) are further restricted after the data points become known. One simple procedure is to construct a rectangle, having sides parallel to the \( x,y \)-axes, that encloses all the data points and that has at least one data point lying on each of its sides. One would expect the fitted ellipse to have neither of
its diameters larger than the diagonal of this "bounding" rectangle. Thus, the two radii are constrained to be no larger than one half of this diagonal, and so the lower bounds on the range constraints for the first two components of $\mathbf{c}_e$ are modified accordingly.

For the one step minimization method, constraints could be specified so that the fitted pattern is forced to be an ellipse. However, the simplicity of the one step minimization method involved the fact that it was an unconstrained minimization scheme. Since the unconstrained one step minimization method was unable to always fit an ellipse to the data points (and for reasons given in Chapter V) the iterative minimization scheme was considered the better of these two schemes. Consequently, it is used in Chapter V to determine its usefulness in recognizing non-elliptical patterns.

Another point which might be noted concerning the one step minimization method is that this scheme tends to estimate the larger radius, $r_x$, much less accurately than the smaller radius, $r_y$. This can be seen in Table 1. This is not a property of the iterative minimization scheme, however, giving more support for its use.

While the recognition scheme which fits an ellipse template to the data points produces very good results for the low noise cases (as do the other recognition schemes),
it also does a reasonable job for the high noise cases. At any rate, at no time does it fit a constrained circle to the data points as occurred with the other iterative minimization scheme. Also, by the very nature of the recognition scheme, it appears that non-elliptical templates may be generated by the computer and fitted to data points which belong to the corresponding non-elliptical patterns. Thus, it appears that the template matching algorithm has the most versatility and generality. This shall be explored in more detail in the succeeding two chapters.
TABLE 1. ELLIPSE PARAMETER ESTIMATES OBTAINED BY ONE STEP MINIMIZATION METHOD

[Reference ellipse parameters are: $e_1=0.25$ ($r_x=2.0$), $e_2=1.00$ ($r_y=1.0$), $A=1.00$, $B=-2.00$, $\theta=0.50$]

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>N</th>
<th>$e_1$</th>
<th>$e_2$</th>
<th>$A$</th>
<th>$B$</th>
<th>$\theta$</th>
<th>$r_x$</th>
<th>$r_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>10</td>
<td>0.250035</td>
<td>1.000063</td>
<td>0.999918</td>
<td>-1.999968</td>
<td>0.499943</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.0</td>
<td>20</td>
<td>0.249987</td>
<td>0.999831</td>
<td>0.999987</td>
<td>-2.000008</td>
<td>0.499964</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.0</td>
<td>50</td>
<td>0.249962</td>
<td>1.000428</td>
<td>1.000113</td>
<td>-2.000046</td>
<td>0.500288</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.0</td>
<td>100</td>
<td>0.250081</td>
<td>1.00146</td>
<td>0.999868</td>
<td>-1.999961</td>
<td>0.499766</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.1</td>
<td>10</td>
<td>0.226588</td>
<td>0.985690</td>
<td>1.133521</td>
<td>-2.017937</td>
<td>0.505674</td>
<td>2.101</td>
<td>1.007</td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
<td>0.247580</td>
<td>1.019186</td>
<td>0.986980</td>
<td>-2.036948</td>
<td>0.500796</td>
<td>2.010</td>
<td>0.991</td>
</tr>
<tr>
<td>0.1</td>
<td>50</td>
<td>0.230059</td>
<td>0.966594</td>
<td>1.061745</td>
<td>-2.000900</td>
<td>0.545271</td>
<td>2.085</td>
<td>1.017</td>
</tr>
<tr>
<td>0.1</td>
<td>100</td>
<td>0.240472</td>
<td>1.021042</td>
<td>0.960186</td>
<td>-1.995705</td>
<td>0.504009</td>
<td>2.039</td>
<td>0.990</td>
</tr>
<tr>
<td>0.2</td>
<td>10</td>
<td>0.250661</td>
<td>1.091854</td>
<td>1.046996</td>
<td>-1.926707</td>
<td>0.610062</td>
<td>1.997</td>
<td>0.957</td>
</tr>
<tr>
<td>0.2</td>
<td>20</td>
<td>0.205477</td>
<td>0.870532</td>
<td>1.019036</td>
<td>-2.167258</td>
<td>0.457846</td>
<td>2.206</td>
<td>1.072</td>
</tr>
<tr>
<td>0.2</td>
<td>50</td>
<td>0.245096</td>
<td>1.081746</td>
<td>1.029216</td>
<td>-2.026368</td>
<td>0.482127</td>
<td>2.020</td>
<td>0.961</td>
</tr>
<tr>
<td>0.2</td>
<td>100</td>
<td>0.222750</td>
<td>1.104007</td>
<td>0.993413</td>
<td>-2.054868</td>
<td>0.527154</td>
<td>2.119</td>
<td>0.952</td>
</tr>
<tr>
<td>0.3</td>
<td>10</td>
<td>0.122579</td>
<td>1.141891</td>
<td>0.510788</td>
<td>-2.504557</td>
<td>0.728425</td>
<td>2.856</td>
<td>0.936</td>
</tr>
<tr>
<td>0.3</td>
<td>20</td>
<td>0.241930</td>
<td>1.038012</td>
<td>1.264826</td>
<td>-1.812599</td>
<td>0.547318</td>
<td>2.033</td>
<td>0.982</td>
</tr>
<tr>
<td>0.3</td>
<td>50</td>
<td>0.187579</td>
<td>0.965065</td>
<td>1.005195</td>
<td>-2.067051</td>
<td>0.611433</td>
<td>2.309</td>
<td>1.018</td>
</tr>
<tr>
<td>0.3</td>
<td>100</td>
<td>0.141655</td>
<td>0.976113</td>
<td>0.911619</td>
<td>-2.123045</td>
<td>0.518779</td>
<td>2.657</td>
<td>1.012</td>
</tr>
<tr>
<td>0.4</td>
<td>10</td>
<td>0.156311</td>
<td>0.662240</td>
<td>0.965569</td>
<td>-1.958733</td>
<td>0.386550</td>
<td>2.529</td>
<td>1.229</td>
</tr>
<tr>
<td>0.4</td>
<td>20</td>
<td>0.037682</td>
<td>1.751601</td>
<td>0.351396</td>
<td>-2.735868</td>
<td>0.661454</td>
<td>5.151</td>
<td>0.756</td>
</tr>
<tr>
<td>0.4</td>
<td>50</td>
<td>0.029652</td>
<td>0.997244</td>
<td>3.404136</td>
<td>-1.132561</td>
<td>0.461635</td>
<td>5.807</td>
<td>1.001</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
<td>0.081115</td>
<td>0.644221</td>
<td>2.139103</td>
<td>-1.733687</td>
<td>0.542781</td>
<td>3.511</td>
<td>1.246</td>
</tr>
<tr>
<td>0.5</td>
<td>20</td>
<td>0.137709</td>
<td>0.967893</td>
<td>0.618680</td>
<td>-2.566473</td>
<td>0.499484</td>
<td>2.695</td>
<td>1.016</td>
</tr>
<tr>
<td>0.5</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>100</td>
<td>0.075127</td>
<td>1.015188</td>
<td>1.068787</td>
<td>-2.383365</td>
<td>0.507140</td>
<td>3.648</td>
<td>0.992</td>
</tr>
</tbody>
</table>

*Unable to fit ellipse.
TABLE 2. ELLIPSE PARAMETER ESTIMATES OBTAINED BY ITERATIVE MINIMIZATION SCHEME
[Reference ellipse parameters are: $e_1=0.25 \ (r_x=2.0), \ e_2=1.00 \ (r_y=1.0)$,
$A=1.00, \ B=-2.00, \ \theta=0.50$]

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$N$</th>
<th>$e_1$</th>
<th>$e_2$</th>
<th>$A$</th>
<th>$B$</th>
<th>$\theta$</th>
<th>$r_x$</th>
<th>$r_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>10</td>
<td>0.25000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-2.00000</td>
<td>0.50000</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.0</td>
<td>20</td>
<td>0.25000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-2.00000</td>
<td>0.50000</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.0</td>
<td>50</td>
<td>0.25000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-2.00000</td>
<td>0.50000</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.0</td>
<td>100</td>
<td>0.25000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>-2.00000</td>
<td>0.50000</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.0</td>
<td>10</td>
<td>0.23309</td>
<td>0.91828</td>
<td>1.12770</td>
<td>-2.00796</td>
<td>0.50372</td>
<td>2.071</td>
<td>1.044</td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
<td>0.25481</td>
<td>0.97564</td>
<td>0.98513</td>
<td>-2.03016</td>
<td>0.50099</td>
<td>1.981</td>
<td>1.012</td>
</tr>
<tr>
<td>0.1</td>
<td>50</td>
<td>0.24045</td>
<td>0.90386</td>
<td>1.05204</td>
<td>-1.99382</td>
<td>0.54868</td>
<td>2.039</td>
<td>1.052</td>
</tr>
<tr>
<td>0.1</td>
<td>100</td>
<td>0.25103</td>
<td>0.95177</td>
<td>0.95558</td>
<td>-1.98599</td>
<td>0.50581</td>
<td>1.996</td>
<td>1.025</td>
</tr>
<tr>
<td>0.2</td>
<td>10</td>
<td>0.26219</td>
<td>0.98635</td>
<td>1.04461</td>
<td>-1.91310</td>
<td>0.60783</td>
<td>1.953</td>
<td>1.007</td>
</tr>
<tr>
<td>0.2</td>
<td>20</td>
<td>0.23523</td>
<td>0.65429</td>
<td>0.99513</td>
<td>-2.13235</td>
<td>0.46860</td>
<td>2.062</td>
<td>1.236</td>
</tr>
<tr>
<td>0.2</td>
<td>50</td>
<td>0.28036</td>
<td>0.86192</td>
<td>1.02352</td>
<td>-1.98737</td>
<td>0.49220</td>
<td>1.889</td>
<td>1.077</td>
</tr>
<tr>
<td>0.2</td>
<td>100</td>
<td>0.26151</td>
<td>0.86500</td>
<td>0.97935</td>
<td>-2.01866</td>
<td>0.54554</td>
<td>1.955</td>
<td>1.075</td>
</tr>
<tr>
<td>0.3</td>
<td>10</td>
<td>0.17273</td>
<td>0.78372</td>
<td>0.56804</td>
<td>-2.39295</td>
<td>0.80087</td>
<td>2.406</td>
<td>1.130</td>
</tr>
<tr>
<td>0.3</td>
<td>20</td>
<td>0.32107</td>
<td>0.62374</td>
<td>1.13235</td>
<td>-1.81673</td>
<td>0.61739</td>
<td>1.765</td>
<td>1.266</td>
</tr>
<tr>
<td>0.3</td>
<td>50</td>
<td>0.23160</td>
<td>0.65681</td>
<td>0.99362</td>
<td>-1.99064</td>
<td>0.63811</td>
<td>2.078</td>
<td>1.234</td>
</tr>
<tr>
<td>0.3</td>
<td>100</td>
<td>0.22484</td>
<td>0.60177</td>
<td>0.88051</td>
<td>-2.01215</td>
<td>0.54895</td>
<td>2.109</td>
<td>1.289</td>
</tr>
<tr>
<td>0.4</td>
<td>10</td>
<td>0.16435</td>
<td>0.53921</td>
<td>0.80733</td>
<td>-1.93123</td>
<td>0.30713</td>
<td>2.467</td>
<td>1.362</td>
</tr>
<tr>
<td>0.4</td>
<td>20</td>
<td>0.20117</td>
<td>0.74273</td>
<td>0.79921</td>
<td>-2.26838</td>
<td>0.70981</td>
<td>2.230</td>
<td>1.160</td>
</tr>
<tr>
<td>0.4</td>
<td>50</td>
<td>0.06250</td>
<td>0.06250</td>
<td>-0.55830</td>
<td>1.00064</td>
<td>0.11353</td>
<td>4.000</td>
<td>4.000</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td>0.06250</td>
<td>0.06250</td>
<td>-1.12144</td>
<td>1.08820</td>
<td>0.12380</td>
<td>4.000</td>
<td>4.000</td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
<td>0.10609</td>
<td>0.55384</td>
<td>1.85071</td>
<td>-1.87407</td>
<td>0.49956</td>
<td>3.070</td>
<td>1.344</td>
</tr>
<tr>
<td>0.5</td>
<td>20</td>
<td>0.26380</td>
<td>0.44969</td>
<td>0.74672</td>
<td>-2.38593</td>
<td>0.64307</td>
<td>1.947</td>
<td>1.491</td>
</tr>
<tr>
<td>0.5</td>
<td>50</td>
<td>0.06250</td>
<td>0.06250</td>
<td>-0.04203</td>
<td>1.33321</td>
<td>0.13402</td>
<td>4.000</td>
<td>4.000</td>
</tr>
<tr>
<td>0.5</td>
<td>100</td>
<td>0.06250</td>
<td>0.06250</td>
<td>-1.35350</td>
<td>0.90174</td>
<td>0.14893</td>
<td>4.000</td>
<td>4.000</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>N</td>
<td>$r_x$</td>
<td>$r_y$</td>
<td>$A$</td>
<td>$B$</td>
<td>$\theta$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>---------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>10</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>20</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>50</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>100</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>10</td>
<td>2.124</td>
<td>0.983</td>
<td>1.083</td>
<td>-2.008</td>
<td>0.542</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
<td>1.943</td>
<td>1.013</td>
<td>0.992</td>
<td>-2.021</td>
<td>0.494</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>50</td>
<td>2.033</td>
<td>1.028</td>
<td>1.044</td>
<td>-1.998</td>
<td>0.537</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>100</td>
<td>1.985</td>
<td>1.003</td>
<td>0.950</td>
<td>-1.984</td>
<td>0.501</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>10</td>
<td>1.974</td>
<td>0.950</td>
<td>1.057</td>
<td>-1.958</td>
<td>0.609</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>20</td>
<td>2.047</td>
<td>1.124</td>
<td>1.014</td>
<td>-2.059</td>
<td>0.499</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>50</td>
<td>1.844</td>
<td>1.002</td>
<td>1.038</td>
<td>-1.976</td>
<td>0.452</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>100</td>
<td>1.877</td>
<td>1.012</td>
<td>0.979</td>
<td>-2.017</td>
<td>0.520</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>10</td>
<td>2.492</td>
<td>0.952</td>
<td>0.666</td>
<td>-2.267</td>
<td>0.796</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>20</td>
<td>1.742</td>
<td>1.063</td>
<td>1.089</td>
<td>-1.835</td>
<td>0.398</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>50</td>
<td>1.931</td>
<td>1.113</td>
<td>0.954</td>
<td>-2.001</td>
<td>0.629</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>100</td>
<td>1.905</td>
<td>1.160</td>
<td>0.928</td>
<td>-2.001</td>
<td>0.498</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>10</td>
<td>2.362</td>
<td>1.299</td>
<td>0.867</td>
<td>-2.009</td>
<td>0.398</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>20</td>
<td>1.982</td>
<td>0.937</td>
<td>0.765</td>
<td>-2.590</td>
<td>0.914</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>50</td>
<td>1.956</td>
<td>1.121</td>
<td>1.149</td>
<td>-2.011</td>
<td>0.586</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td>1.854</td>
<td>1.022</td>
<td>1.138</td>
<td>-1.991</td>
<td>0.581</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
<td>3.981</td>
<td>1.185</td>
<td>2.341</td>
<td>-1.433</td>
<td>0.561</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>20</td>
<td>1.548</td>
<td>1.287</td>
<td>0.732</td>
<td>-2.247</td>
<td>0.263</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>50</td>
<td>2.022</td>
<td>1.032</td>
<td>0.734</td>
<td>-2.017</td>
<td>0.304</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>100</td>
<td>1.752</td>
<td>1.215</td>
<td>0.988</td>
<td>-2.080</td>
<td>0.591</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE 3. ELLIPSE PARAMETER ESTIMATES OBTAINED BY TEMPLATE MATCHING METHOD

[Reference ellipse parameters are: $e_1=0.25$ ($r_x=2.0$), $e_2=1.00$ ($r_y=1.0$), $A=1.00$, $B=-2.00$, $\theta=0.50$]
PLATE I. ELLIPSES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD (\( \sigma = 0.0 \))
PLATE II. ELLIPSES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD ($\sigma = 0.1$)

(a) FIELD OF VIEW

(b) FIELD OF VIEW

(c) FIELD OF VIEW

(d) FIELD OF VIEW

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE

REFERENCE ELLIPSE

LEAST-SQUARES ELLIPSE
PLATE III. ELLIPSES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD ($\sigma = 0.2$)

(a) Reference Ellipse
- Major Axis: 2.000
- Minor Axis: 1.503
- Translation: 1.000
- Rotation in Degrees: 20.912

(b) Least-Squares Ellipse
- Major Axis: 2.000
- Minor Axis: 1.513
- Translation: 1.000
- Rotation in Degrees: 20.869

(c) Reference Ellipse
- Major Axis: 2.000
- Minor Axis: 1.503
- Translation: 1.000
- Rotation in Degrees: 20.912

(d) Least-Squares Ellipse
- Major Axis: 2.000
- Minor Axis: 1.513
- Translation: 1.000
- Rotation in Degrees: 20.869
PLATE IV. ELLIPSES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD (σ = 0.3)

(a) Reference Ellipse

- Major Axis: 2.000
- Minor Axis: 1.000
- Translation: 1.000
- Rotation in Degrees: 29.998

(b) Least-Squares Ellipse

- Major Axis: 2.000
- Minor Axis: 1.000
- Translation: 1.000
- Rotation in Degrees: 29.997

(c) Reference Ellipse

- Major Axis: 2.000
- Minor Axis: 1.000
- Translation: 1.000
- Rotation in Degrees: 34.193

(d) Least-Squares Ellipse

- Major Axis: 2.000
- Minor Axis: 1.000
- Translation: 1.000
- Rotation in Degrees: 29.998

Field of View

50 Data Points

Standard Deviation: 0.3
PLATE V. ELLIPSES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD ($\sigma = 0.4$)

(a) REFERENCE ELLIPSE
- $a = 2.000$
- $b = 1.000$
- Translation: 0.000
- Rotation in degrees: 28.648

(b) LEAST-SQUARES ELLIPSE
- $a = 2.000$
- $b = 1.000$
- Translation: 0.000
- Rotation in degrees: 28.648

(c) REFERENCE ELLIPSE
- $a = 2.000$
- $b = 1.000$
- Translation: 0.000
- Rotation in degrees: 28.648

(d) LEAST-SQUARES ELLIPSE
- $a = 2.000$
- $b = 1.000$
- Translation: 0.000
- Rotation in degrees: 28.648
PLATE VI. ELLIPSES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD ($\sigma = 0.5$)

(a) 10 data points

- Reference Ellipse
  - Semi-major axis: 1.000
  - Semi-minor axis: 1.000
  - Rotation in degrees: 29.503

- Least-Squares Ellipse
  - Semi-major axis: 1.000
  - Semi-minor axis: 1.000
  - Rotation in degrees: 29.503

(b) 20 data points

- Reference Ellipse
  - Semi-major axis: 1.000
  - Semi-minor axis: 1.000
  - Rotation in degrees: 29.503

- Least-Squares Ellipse
  - Semi-major axis: 1.000
  - Semi-minor axis: 1.000
  - Rotation in degrees: 29.503

(c) 50 data points

- Reference Ellipse
  - Semi-major axis: 1.000
  - Semi-minor axis: 1.000
  - Rotation in degrees: 29.503

- Least-Squares Ellipse
  - Semi-major axis: 1.000
  - Semi-minor axis: 1.000
  - Rotation in degrees: 29.503

(d) 100 data points

- Reference Ellipse
  - Semi-major axis: 1.000
  - Semi-minor axis: 1.000
  - Rotation in degrees: 29.503

- Least-Squares Ellipse
  - Semi-major axis: 1.000
  - Semi-minor axis: 1.000
  - Rotation in degrees: 29.503
5.1 Introduction

The objective of this chapter is to investigate the feasibility of employing either of two schemes to recognize rectangular planar patterns. Again, by the term "recognition" is meant the estimation of the five parameters which characterize a rectangle having arbitrary size and shape, as well as arbitrary position (translation and rotation) in a planar field of view.

In Section 5.2, one recognition scheme is discussed which involves using the equation for an ellipse as the error function. As discussed in Chapter III, this involves two different criterion functions, depending upon whether the equation for the ellipse is a linear function of its five parameters (Equation 3-29) or is a nonlinear function of a different set of five parameters (Equation 3-6). Therefore, two different minimization methods are considered with respect to this recognition scheme, the same as was done in Chapter IV. The results which are obtained from these
two minimization methods for rectangles having known parameter vectors (no noise) are then analyzed in Section 5.3. The iterative minimization method (corresponding to the error function which is nonlinear in its parameters) is then employed in Section 5.4 to estimate the parameters associated with rectangles whose boundary points are corrupted with noise, and the resulting estimates are discussed.

Section 5.5 presents the other recognition scheme which is investigated. This recognition scheme involves fitting a rectangle template to the data points rather than an ellipse. The resulting error function is the radial distance which a given data point is from the template. This error function is also nonlinear in the five parameters which characterize a rectangle. The results of using this recognition scheme (using the same data points as in Section 5.4) is discussed in Section 5.6.

Finally, Section 5.7 contains a brief summary of the results and conclusions which have been reached in this chapter.

5.2 Fitting an Ellipse to a Rectangle

In order to further test the recognition techniques developed for elliptical objects, a second class of patterns was considered. Rectangular patterns were chosen for this purpose because they are simple geometric patterns and yet do not possess a simple analytic representation. Furthermore, a rectangle has several
properties in common with an ellipse. Both of these patterns are convex, and both are symmetrical about two orthogonal axes. Thus, a rectangle may be characterized by a set of five parameters in a manner quite similar to an ellipse. Figure 7 shows a rectangle which has been translated and rotated with respect to the x,y-coordinate system. This rectangle is characterized by its two radii \( R_x \) and \( R_y \), and by the x-and y-translation of its center \((A', B')\), respectively), as well as by its rotation \( \theta' \).

\[ \begin{align*}
\text{Figure 7. Parameters of a Rotated and Translated Rectangle} \\
\end{align*} \]

Therefore, the parameter vector characterizing a rectangle may be expressed as
One strategy which was used to recognize rectangular patterns was to present a number of different size and shape rectangles to both the one step minimization method and the iterative minimization scheme, and to determine what relationship, if any, existed between the parameter vectors of the fitted ellipses and the parameter vectors of the corresponding rectangles. This strategy is motivated by the fact that the noiseless data points which lie on the boundary of a rectangle may be considered as being noisy data points which originally belonged on the boundary of some ellipse. If a relationship can be found between the parameter vectors of the fitted ellipses and the parameter vectors of the corresponding rectangles, then it will be possible to compute the parameter vector of an unknown rectangle after the parameter vector of its associated fitted ellipse is determined.

5.3 Parameter Estimation for Noise-Free Rectangular Patterns

The results of using the one step minimization method to recognize a rectangle are shown in Tables 4, 5, 6, and 7. Each table corresponds to a different number of
data points on the boundary of the rectangle, the number of data points being 8, 20, 48, or 100, respectively. These data points were generated by the DATA subroutine which is described in Appendix A.

A total of twenty rectangles were to be recognized. Ten of the rectangles have a w-axis radius equal to one unit of length, with the z-axis radius varying from 0.1 to 1.0 units of length in increments of 0.1. The other ten rectangles are exactly twice the dimensions of the first ten. The translation and rotation parameters of the rectangles were chosen to be the same as those which were used for the ellipse which was discussed in Chapter IV. Thus

\[
\begin{align*}
  c_j^1 &= \text{x-translation} = A' = 1.0 \\
  c_j^2 &= \text{y-translation} = B' = -2.0 \\
  c_j^3 &= \text{rotation in radians} = \theta' = 0.5
\end{align*}
\]

In the tables the radii of the rectangles are denoted by \( R \) while the radii of the fitted ellipses are denoted by \( r \). The radii of the fitted ellipses are, of course, related to the first two components of their characterizing parameter vector, \( \mathbf{c} \), by Equation (5-3).

\[
\begin{align*}
  r_x &= \sqrt{\frac{1}{c_1}} \\
  r_y &= \sqrt{\frac{1}{c_2}}
\end{align*}
\]
Inspection of Tables 4, 5, 6, and 7 indicates that the one step minimization method is not very effective in recognizing the rectangles. For the most part the translation and rotation parameters of the fitted ellipses are quite close in value to the corresponding parameters of the given rectangles. However, there is no recognizable correspondence between the two radii of the fitted ellipses and the respective radii of the rectangles.

A remark should be made at this point regarding the results which one would intuitively expect to obtain. First of all, since an ellipse and rectangle have similar geometric properties (convexity and symmetry about two orthogonal axes) one would expect that the known rectangles and the corresponding fitted ellipses would have identical coordinates for their centers, as well as identical rotation angles. On the other hand, it is difficult to predict the exact relation between each radius of a known rectangle and the corresponding radius of the fitted ellipse. However, again due to symmetry, one would expect that the ratio of the radii of a fitted ellipse would be nearly equal to the ratio of the radii of the corresponding known rectangle, being more or less independent of the rectangle's size. The one step minimization method did not possess this property, however. If a recognition scheme does have this property, then the constant of
proportionality relating the size of the rectangle to the size of the corresponding fitted ellipse may be determined experimentally.

Another weakness of the one step minimization method is that it is unable to fit any ellipse to some of the given rectangles. This might be expected, however, since the one step minimization was unable to fit an ellipse to an ellipse under high noise conditions as was pointed out in Chapter IV. Thus, the one step minimization method does not appear to be a good method for estimating the parameters of a rectangle.

The iterative minimization scheme was next employed to estimate the parameters of these same rectangles. The same values were used for the range constraints as were used in the recognition of ellipses in Chapter IV. Tables 8, 9, 10, and 11 show the results of using this scheme. The initial estimate used for the parameter vector of the fitted ellipse for the ten larger rectangles was

\[ c_e = \begin{bmatrix} 0.25 \\ 1.00 \\ 0.7 \\ -2.5 \\ 0.8 \end{bmatrix} \]  

(5-4)

Inspection of Tables 8, 9, 10, and 11 indicates that the iterative minimization scheme is quite effective in estimating the parameters of a rectangle. It is seen
that the ratio $r_y/r_x$ corresponding to the radii of the fitted ellipse is equal (to within three decimal places) to the radii $R_y/R_x$ of the rectangle which is to be recognized. Thus the ratio of the radii of the fitted ellipses gives a direct indication of the shape of the rectangles to which they are fitted.

Tables 8, 9, 10, and 11 also indicate the size of the rectangles may be determined to a reasonable degree of accuracy. When eight data points on the boundary of the rectangle are given, Table 8 shows that $R_x = 0.775 r_x$, meaning that the rectangles' radii are 0.775 times the length of the fitted ellipses' radii. Likewise, Tables 9, 10, and 11 show that the scale factor, $R_x/r_x$ is equal to 0.830, 0.842, and 0.845 for 20, 48, and 100 data points on the boundary of the rectangle, respectively.

One can see that the scale factor does not change appreciably when more than 48 data points are given. If one assumes that the scale factor associated with 100 data points is essentially the same as the scale factor associated with an infinite number of data points (which seems reasonable in light of the above results) then it is possible to compare the scale factor associated with a finite number of data points with the scale factor associated with a continuous representation of the rectangle. For eight data points this ratio
is $0.775/0.845 = 0.917$, which means that the estimated size of the rectangle is only $91.7\%$ of the size of the actual rectangle, although it has exactly the same shape as the actual rectangle. For 20 data points this ratio increases to $0.830/0.845 = 0.983$ and for 48 data points the ratio is $0.842/0.845 = 0.997$. Thus, if the rectangle is represented by twenty or more data points on its boundary, then one need merely multiply the radii of the fitted ellipse by the factor $0.845$ to obtain the radii of the corresponding rectangle, having assurance that this rectangle will be at least within $2\%$ of the size of the actual rectangle.

Tables 8, 9, 10, and 11 show another very desirable property of the iterative minimization scheme, namely, the translation and rotation parameters of the fitted ellipses have exactly the same values (to within three decimal places) as the corresponding parameters of the given rectangles. Therefore, only the first two components of the fitted ellipse's parameter vector need to be transformed in order to obtain the parameter vector of the rectangle, and this transformation is a simple scale change.

Thus, the desired relationship between the parameter vector of the fitted ellipse and the parameter vector of the associated rectangle has now been determined. If the
final estimate for the parameter vector of the fitted ellipse is given by

\[ c_e^+ = \begin{bmatrix} e_1 \\ e_2 \\ A \\ B \\ \theta \end{bmatrix} \]  \hspace{1cm} (5-5)

then the estimate for the parameter vector of the rectangle which is to be recognized is

\[ c_e^+ = \begin{bmatrix} R_W \\ R_Z \\ A' \\ B' \\ \theta' \end{bmatrix} = \begin{bmatrix} \frac{k}{\sqrt{e_1}} \\ \frac{k}{\sqrt{e_2}} \\ A \\ B \\ \theta \end{bmatrix} \]  \hspace{1cm} (5-6)

where \( k \), the scale factor, is a function of the number of data points. If the number of data points is 20 or greater, \( k \) may be taken to be 0.845.

5.4 Parameter Estimation for Noisy Rectangular Patterns

Since the iterative minimization scheme was able to effectively recognize rectangles, the question naturally arises as to how well it can recognize rectangles which are represented by noisy data points. In order to determine this a total of twenty-four different cases were considered, as was done with the ellipse in Chapter IV.

The noiseless rectangle, whose parameters are to be estimated, is characterized by the following parameter vector.
\[
\begin{bmatrix}
R_W \\
R_Z \\
A' \\
B' \\
\theta'
\end{bmatrix}
= \begin{bmatrix}
2.00 \\
1.00 \\
1.00 \\
-2.00 \\
0.50
\end{bmatrix}
\tag{5-7}
\]

The last three components have the same value as they did for the ellipse considered in Chapter IV. The noise levels are also the same as they were previously, namely, \( \sigma = 0.0, 0.1, 0.2, 0.3, 0.4, \) and \( 0.5. \) Also, all of the parameters associated with the iterative minimization scheme were given the same values as they had in Chapter IV, with the exception of the initial parameter vector estimate \( \hat{c_e}. \) It is

\[
\begin{bmatrix}
e_1 \\
e_2 \\
A \\
B \\
\theta
\end{bmatrix}
= \begin{bmatrix}
0.25 \\
1.00 \\
0.70 \\
-2.50 \\
0.80
\end{bmatrix}
\tag{5-8}
\]

Plates VII, VIII, IX, X, XI, and XII show the results of estimating the parameter vector of a rectangle using the iterative minimization scheme. Before discussing these results it should be pointed out that the radii of the rectangle were computed by using the scale factor associated with the appropriate number of data points. Thus, for example, for the six cases in which the rectangle was represented by 20 data points the scale factor which was used was 0.830, and not 0.845. By
doing this, any error in the estimated parameter vector is due to the noisy data points.

An examination of Plate VII reveals that the x-radii and y-radii of the fitted rectangles differ in the third decimal place from the corresponding radii of the reference rectangles. This is due to rounding off the scale factors to the third decimal place. This error is entirely negligible compared to the error resulting from the noisy data points.

In some of the plates there are not as many noisy data points as the number which is indicated. This is due to the fact that some of the noisy data points fell outside of the field of view. As before, these data points are considered to be valid points for the iterative minimization scheme to use, being analogous to measurement noise.

An inspection of Plates VII, VIII, IX, X, XI, and XII shows that fitting an ellipse to a set of data points belonging on the boundary of a rectangle is an effective method by which to estimate the parameters of the rectangle when the noise level is within reasonable limits ($\sigma = 0.0$, 0.1, and 0.2).

Referring to Plate VIII($\sigma = 0.1$) one sees that the rectangles which correspond to the fitted ellipses (the dashed line rectangles) resemble the reference rectangles
very closely except in the eight data point case. It would seem that eight data points, when corrupted by noise, simply are too sparse in number for the iterative minimization scheme to yield a good estimate for the reference rectangle's parameter vector. However, it should be noted that the error in estimating the rotation angle for the eight data point case is quite acceptable, being approximately two percent. In the other three cases this error is approximately one percent or less.

For $\sigma = 0.2$ it can be seen in Plate IX that the parameter vector estimates are beginning to deteriorate, but for the 20, 48 and 100 data point cases these estimates are still acceptable by most standards. In particular, it is seen that for these three cases the error in the rotation angle estimate is no greater than approximately four percent, which is rather small considering the scatter of the data points.

When the noise level reaches $\sigma = 0.3$, the overall effectiveness of the iterative minimization scheme becomes questionable. The fitted rectangles have a tendency to be larger than the reference rectangles. However, one good point which can be made is that the estimate for the rotation angle in all four cases does not exceed four percent, which means that this estimate has not been affected to any extent by the increase in noise level.
from $\alpha = 0.2$ to $\alpha = 0.3$.

Plates XI and XII indicate that for high noise levels ($\sigma = 0.4$ and 0.5) the recognition capability of the iterative minimization scheme has completely deteriorated. Some improvement could be achieved for the cases in which the fitted rectangles have radii equal to their constraint value. In these cases the fitted rectangle is a square which concentrates the data points in one of its corners, with approximately one half of the data points on the inside of the square and one half on the outside. This situation is very similar to that which occurred in the recognition of ellipses under high noise conditions, and it can be remedied in exactly the same manner as described in Chapter IV.

5.5 Fitting a Rectangle Template to Rectangle Data Points

The last three sections have discussed the feasibility of fitting an ellipse to a set of data points which lie on the boundary of a rectangle, and the manner in which the parameters of the least squares ellipse are related to the parameters of the rectangle. This section shall present a different recognition scheme, which involves fitting a rectangle template to the data points directly, rather than indirectly as was done previously.
To facilitate presenting this recognition scheme, consider Figure 8.

The rectangle template which is illustrated in Figure 8 is determined either by an intelligent guess or, as was the case, by the ESTIMT subroutine which bases its guess on the values of the first and second moments of the data points as discussed in Appendix C. After having obtained
the initial template, it is necessary to determine which side of the rectangle template shall be associated with any given data point, and to compute the associated error.

In Figure 8 it is seen that the \( w,z \)-coordinate system is broken into four sectors, the boundary of each sector going through the origin of the \( w,z \)-reference frame and through a corner of the rectangle template. A reasonable solution to the data point correspondence problem is to associate any data point in a given sector with the side of the rectangle template which is in that sector. This may be mathematically carried out in the following manner.

The coordinates of any of the data points may be found with respect to the \( w,z \)-reference frame from the transformation

\[
\begin{align*}
   w_i &= (x_i - A) \cos \theta + (y_i - B) \sin \theta \\
   z_i &= -(x_i - A) \sin \theta + (y_i - B) \cos \theta
\end{align*}
\] (5-9)

The angle of inclination, \( \psi_i \), of any data point \( (w_i,z_i) \) with respect to the \( w \)-axis is given by

\[
\psi_i = \tan^{-1} \frac{z_i}{w_i}
\] (5-10)

The decision as to which side of the rectangle template a given data point \( (w_i,z_i) \) will be associated is made on the basis of which sector contains \( \psi_i \). Thus
where Side 1 is in Sector I, Side 2 is in Sector II, and so on.

Once the appropriate side of the rectangle template has been determined for a given data point, it remains to compute the error, which is the radial distance from the given data point to the side of the rectangle template. The error associated with data points lying in Sector I shall be discussed first.

Referring to Figure 8, one notes that in the \( w, z \)-reference frame the equation for Side 1 is

\[
\psi_i = \tan^{-1} \left( \frac{-r_y}{r_x} \right) \quad (5-11)
\]

\[
\psi_i = \tan^{-1} \left( \frac{r_y}{r_x} \right) \quad (5-12)
\]

\[
\psi_i = \tan^{-1} \left( \frac{-r_y}{r_x} \right) \quad (5-13)
\]

\[
\psi_i = \tan^{-1} \left( \frac{r_y}{r_x} \right) \quad (5-14)
\]

where Side 1 is in Sector I, Side 2 is in Sector II, and so on.

Once the appropriate side of the rectangle template has been determined for a given data point, it remains to compute the error, which is the radial distance from the given data point to the side of the rectangle template. The error associated with data points lying in Sector I shall be discussed first.

Referring to Figure 8, one notes that in the \( w, z \)-reference frame the equation for Side 1 is

\[
w_m = r_x \quad (5-15)
\]

where the "m" subscript denotes that this is the \( w \)-coordinate for the template (or model). A radial line going from the origin of the \( w, z \)-reference frame to
a data point \((w_d, z_d)\) in Sector I has an angle of inclination, \(\psi\), given by

\[
\tan \psi = \frac{z_d}{w_d}
\]

(5-16)

This line intersects Side 1 at \((w_m, z_m)\). Thus

\[
\tan \psi = \frac{z_m}{w_m}
\]

(5-17)

Substituting Equation (5-16) into Equation (5-17) yields

\[
zm = w_m \times \frac{z_d}{w_d}
\]

(5-18)

Therefore, the \(w, z\)-coordinates of the model point which lies on Side 1 of the rectangle template (for a data point lying in Sector I) are given by

\[
w_m = r_x
\]

(5-19)

\[
z_m = r_x \times \frac{z_d}{w_d}
\]

and the corresponding \(x, y\)-coordinates are found by substituting Equation (5-19) into the transformation

\[
x_m = w_m \cos \theta - z_m \sin \theta + A
\]

\[
y_m = w_m \sin \theta + z_m \cos \theta + B
\]

(5-20)

The error is then the difference in the \(x\)-and \(y\)-coordinates
of the data point and the model point. Thus

\[ e_i = (x_d - x_m)_i \]

\[ e_{i+N} = (y_d - y_m)_i \]

and

\[ \phi = \sum_{i=1}^{2N} e_i^2 \]

\[ = \sum_{i=1}^{N} \left\{ (x_d - x_m)_i^2 + (y_d - y_m)_i^2 \right\} \]

where the "i" subscript denotes each data point.

The error as given by Equation (5-21) is a function of the five parameters \( r_x, r_y, A, B, \) and \( \theta \). Note that the first two parameters are now the radii, rather than the reciprocal of the square of the radii as is the case when fitting an ellipse to the rectangle data points as discussed in Section 5.3 and 5.4.

The \( w,z \)-coordinates of the other three sides of the rectangle template may be found in a similar manner. For a data point \((w_d,z_d)\) lying in Sector II, the corresponding model point \((w_m,z_m)\) is

\[ w_m = r_y \times \frac{w_d}{z_d} \]

\[ z_m = r_y \]

(5-23)
Likewise, for a data point \((w_d, z_d)\) lying in Sector III, the corresponding model point \((w_m, z_m)\) is

\[
\begin{align*}
  w_m &= -r_x \\
  z_m &= -r_x \times \frac{z_d}{w_d}
\end{align*}
\]  

(5-24)

Finally, for a data point \((w_d, z_d)\) lying in Sector IV, the corresponding model point \((w_m, z_m)\) is

\[
\begin{align*}
  w_m &= -r_y \times \frac{w_d}{z_d} \\
  z_m &= -r_y
\end{align*}
\]

(5-25)

The coordinates of the model points given by Equations (5-23), (5-24), and (5-25) are referred to the \(x, y\)-reference frame via the transformation given by Equation (5-20). The associated error and criterion function is then given by Equations (5-21) and (5-22), respectively.

Now that the error for any data point has been computed in terms of the five parameters that characterize the rectangle template, it is only necessary to compute the partial derivatives of the coordinates \((x_m, y_m)\) with respect to these parameters to permit the Gauss-Newton or the Newton-Raphson minimization scheme to minimize the sum-squared error criterion function (Equation (5-21)) by iterative adjustments of the parameters, as
explained in Appendix D where a documentation of these partial derivatives is presented.

5.6 Implementation of the Rectangle Template

The recognition scheme which is discussed in the previous section was employed to fit a rectangle template to the same sets of data points which are considered in Section 5.4 for the iterative minimization scheme (ellipse template). Table 12 lists the parameter estimates which were obtained for each set of data points.

Comparing Table 12 with Plates VII, VIII, IX, X, XI, and XII, one notes that for low noise levels the two recognition schemes yield results which are very similar. However, for high noise levels ($\sigma = 0.4$ and $0.5$) it is seen that fitting a rectangle template to the data points directly results in much better estimates for the parameters. In particular, consider Plate XI for $\sigma = 0.4$ and 48 data points. The best fitting rectangle template (given in Table 12) is a very good fit to the data points, while the rectangle resulting from ellipse template is a totally unacceptable fit.

5.7 Summary

This chapter investigated the feasibility of utilizing either the one step minimization method or the iterative minimization scheme to estimate the parameters of a
rectangle when noise free data points lying on the rectangle's boundary are given. If the parameters are estimated with sufficient precision, then the rectangle has been "recognized" correctly.

In addition, another recognition scheme was proposed which fit a rectangle template to the data points directly, rather than first fitting an ellipse to the data points as the one step minimization method and the iterative minimization scheme do.

It was found that the one step minimization method was completely inadequate in its capability to estimate the parameters of given rectangles. While it did do a reasonable job in estimating the translation and rotation parameters, the two major shortcomings of this method were:

1. the ellipse which was fitted to the data points did not have the same shape as the given rectangle, i.e., the ratio of the radii of the fitted ellipse was not identical to the corresponding ratio of the radii of the given rectangle, and

2. the size of the fitted ellipse did not double when the size of the given rectangle doubled.
The iterative minimization scheme, on the other hand, did not have these shortcomings. Not only did it estimate the translation and rotation parameters very precisely, but the ellipse which it fitted to the data points had radii whose ratio was identical to that of the given rectangle, and this ratio was independent of the size of the given rectangle. It was, therefore, possible to experimentally determine a scale factor relating the size of the fitted ellipse's radii to the radii of the given rectangle.

Since the iterative minimization scheme had the capability to precisely estimate the parameters of a rectangle whose boundary points were noise free, the next step was to determine the degradation in the parameter vector estimates in situations for which the data points were noisy. Reference to Plates VII, VIII, and XI indicates that for moderate levels of noise (σ = 0.0, 0.1, and 0.2) the iterative minimization scheme did a very satisfactory job of recognizing the rectangles. Special notice should be taken concerning the accuracy with which the rotation angle was estimated. Excluding the eight data point case, this error was never greater than four percent for these moderate noise levels.
The recognition scheme produced results of questionable value for noise level $\sigma = 0.3$. Although the rotation angle was still estimated with good precision (maximum of four percent error), the size of the fitted rectangle tended to be larger than the size of the reference rectangle. It can be said that $\sigma = 0.3$ represents the maximum noise level for which the iterative minimization scheme produces useful results for the particular set of rectangles investigated.

For larger noise levels ($\sigma = 0.4$ and 0.5) the iterative minimization scheme was not able to do a satisfactory job of recognizing the rectangles at all. This is not at all surprising, since even a human being would have difficulty trying to fit a rectangle to the data as shown on Plates XI and XII.

The recognition scheme which fit a rectangle template to the data points directly was somewhat superior to the iterative minimization scheme which fit an ellipse to the data points, however. While the estimates were essentially the same for low noise levels ($\sigma = 0.0$, 0.1, and 0.2), the rectangle template resulted in a reasonably good fit for the higher noise levels also (for 48 and 100 data points). Also, the template matching scheme (using the radial distance of the data points to the template as the error criterion) may be generalized to other patterns, as discussed in the next chapter.
### Table 4. Parameters of Ellipses Fitted to Rectangles by the One Step Minimization Method (8 Data Points)

<table>
<thead>
<tr>
<th>$R_x$</th>
<th>$R_y$</th>
<th>$R_y/R_x$</th>
<th>$r_x$</th>
<th>$r_y$</th>
<th>$r_y/r_x$</th>
<th>$A$</th>
<th>$B$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.1</td>
<td>0.100</td>
<td>1.331</td>
<td>0.077</td>
<td>0.058</td>
<td>1.152</td>
<td>-1.917</td>
<td>0.496</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.200</td>
<td>1.768</td>
<td>0.202</td>
<td>0.114</td>
<td>0.962</td>
<td>-2.023</td>
<td>0.499</td>
</tr>
<tr>
<td>1.0</td>
<td>0.3</td>
<td>0.300</td>
<td>1.755</td>
<td>0.298</td>
<td>0.170</td>
<td>0.996</td>
<td>-2.007</td>
<td>0.499</td>
</tr>
<tr>
<td>1.0</td>
<td>0.4</td>
<td>0.400</td>
<td>1.773</td>
<td>0.396</td>
<td>0.223</td>
<td>1.003</td>
<td>-2.008</td>
<td>0.499</td>
</tr>
<tr>
<td>1.0</td>
<td>0.5</td>
<td>0.500</td>
<td>1.799</td>
<td>0.491</td>
<td>0.273</td>
<td>1.006</td>
<td>-2.016</td>
<td>0.498</td>
</tr>
<tr>
<td>1.0</td>
<td>0.6</td>
<td>0.600</td>
<td>1.834</td>
<td>0.585</td>
<td>0.319</td>
<td>1.007</td>
<td>-2.018</td>
<td>0.497</td>
</tr>
<tr>
<td>1.0</td>
<td>0.7</td>
<td>0.700</td>
<td>1.886</td>
<td>0.675</td>
<td>0.358</td>
<td>1.010</td>
<td>-2.025</td>
<td>0.496</td>
</tr>
<tr>
<td>1.0</td>
<td>0.8</td>
<td>0.800</td>
<td>1.957</td>
<td>0.761</td>
<td>0.389</td>
<td>1.014</td>
<td>-2.034</td>
<td>0.495</td>
</tr>
<tr>
<td>1.0</td>
<td>0.9</td>
<td>0.900</td>
<td>2.063</td>
<td>0.842</td>
<td>0.408</td>
<td>1.017</td>
<td>-2.045</td>
<td>0.493</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>1.000</td>
<td>2.234</td>
<td>0.915</td>
<td>0.410</td>
<td>1.020</td>
<td>-2.060</td>
<td>0.491</td>
</tr>
<tr>
<td>2.0</td>
<td>0.2</td>
<td>0.100</td>
<td>3.517</td>
<td>0.201</td>
<td>0.057</td>
<td>0.995</td>
<td>-2.005</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4</td>
<td>0.200</td>
<td>3.540</td>
<td>0.395</td>
<td>0.112</td>
<td>1.003</td>
<td>-2.007</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.6</td>
<td>0.300</td>
<td>3.670</td>
<td>0.584</td>
<td>0.160</td>
<td>1.008</td>
<td>-2.017</td>
<td>0.499</td>
</tr>
<tr>
<td>2.0</td>
<td>0.8</td>
<td>0.400</td>
<td>3.918</td>
<td>0.761</td>
<td>0.194</td>
<td>1.014</td>
<td>-2.034</td>
<td>0.499</td>
</tr>
<tr>
<td>2.0</td>
<td>1.0</td>
<td>0.500</td>
<td>4.481</td>
<td>0.915</td>
<td>0.204</td>
<td>1.021</td>
<td>-2.060</td>
<td>0.498</td>
</tr>
<tr>
<td>2.0</td>
<td>1.2</td>
<td>0.600</td>
<td>6.801</td>
<td>1.030</td>
<td>0.151</td>
<td>1.010</td>
<td>-2.109</td>
<td>0.497</td>
</tr>
<tr>
<td>2.0</td>
<td>1.4</td>
<td>0.700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>0.800</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>1.8</td>
<td>0.900</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Not able to fit ellipse to data.

Note: $R$ corresponds to reference rectangles' radii

$r$ corresponds to fitted ellipses' radii

All reference rectangles have $A = 1.0$, $B = -2.0$, and $\theta = 0.5$
TABLE 5. PARAMETERS OF ELLIPSES FITTED TO RECTANGLES
BY THE ONE STEP MINIMIZATION METHOD
(20 DATA POINTS)

<table>
<thead>
<tr>
<th>$R_x$</th>
<th>$R_y$</th>
<th>$R_y/R_x$</th>
<th>$r_x$</th>
<th>$r_y$</th>
<th>$r_y/r_x$</th>
<th>$A$</th>
<th>$B$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.1</td>
<td>0.100</td>
<td>0.699</td>
<td>0.056</td>
<td>0.080</td>
<td>1.238</td>
<td>-1.865</td>
<td>0.498</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.200</td>
<td>1.326</td>
<td>0.209</td>
<td>0.158</td>
<td>0.988</td>
<td>-2.007</td>
<td>0.500</td>
</tr>
<tr>
<td>1.0</td>
<td>0.3</td>
<td>0.300</td>
<td>1.330</td>
<td>0.314</td>
<td>0.236</td>
<td>1.003</td>
<td>-2.002</td>
<td>0.500</td>
</tr>
<tr>
<td>1.0</td>
<td>0.4</td>
<td>0.400</td>
<td>1.335</td>
<td>0.418</td>
<td>0.313</td>
<td>1.003</td>
<td>-2.004</td>
<td>0.499</td>
</tr>
<tr>
<td>1.0</td>
<td>0.5</td>
<td>0.500</td>
<td>1.340</td>
<td>0.520</td>
<td>0.388</td>
<td>1.004</td>
<td>-2.008</td>
<td>0.498</td>
</tr>
<tr>
<td>1.0</td>
<td>0.6</td>
<td>0.600</td>
<td>1.347</td>
<td>0.622</td>
<td>0.462</td>
<td>1.005</td>
<td>-2.012</td>
<td>0.497</td>
</tr>
<tr>
<td>1.0</td>
<td>0.7</td>
<td>0.700</td>
<td>1.354</td>
<td>0.722</td>
<td>0.533</td>
<td>1.008</td>
<td>-2.016</td>
<td>0.496</td>
</tr>
<tr>
<td>1.0</td>
<td>0.8</td>
<td>0.800</td>
<td>1.364</td>
<td>0.819</td>
<td>0.600</td>
<td>1.011</td>
<td>-2.022</td>
<td>0.494</td>
</tr>
<tr>
<td>1.0</td>
<td>0.9</td>
<td>0.900</td>
<td>1.377</td>
<td>0.914</td>
<td>0.664</td>
<td>1.014</td>
<td>-2.029</td>
<td>0.491</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>1.000</td>
<td>1.394</td>
<td>1.005</td>
<td>0.721</td>
<td>1.018</td>
<td>-2.038</td>
<td>0.487</td>
</tr>
<tr>
<td>2.0</td>
<td>0.2</td>
<td>0.100</td>
<td>2.643</td>
<td>0.209</td>
<td>0.079</td>
<td>1.017</td>
<td>-1.992</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4</td>
<td>0.200</td>
<td>2.671</td>
<td>0.418</td>
<td>0.156</td>
<td>1.002</td>
<td>-2.005</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.6</td>
<td>0.300</td>
<td>2.695</td>
<td>0.622</td>
<td>0.231</td>
<td>1.005</td>
<td>-2.012</td>
<td>0.499</td>
</tr>
<tr>
<td>2.0</td>
<td>0.8</td>
<td>0.400</td>
<td>2.729</td>
<td>0.819</td>
<td>0.300</td>
<td>1.011</td>
<td>-2.022</td>
<td>0.499</td>
</tr>
<tr>
<td>2.0</td>
<td>1.0</td>
<td>0.500</td>
<td>2.789</td>
<td>1.004</td>
<td>0.360</td>
<td>1.018</td>
<td>-2.038</td>
<td>0.498</td>
</tr>
<tr>
<td>2.0</td>
<td>1.2</td>
<td>0.600</td>
<td>2.902</td>
<td>1.168</td>
<td>0.402</td>
<td>1.029</td>
<td>-2.063</td>
<td>0.497</td>
</tr>
<tr>
<td>2.0</td>
<td>1.4</td>
<td>0.700</td>
<td>3.165</td>
<td>1.289</td>
<td>0.407</td>
<td>1.046</td>
<td>-2.104</td>
<td>0.495</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>0.800</td>
<td>4.429</td>
<td>1.309</td>
<td>0.296</td>
<td>1.062</td>
<td>-2.187</td>
<td>0.492</td>
</tr>
<tr>
<td>2.0</td>
<td>1.8</td>
<td>0.900</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Not able to fit ellipse to data.

Note: $R$ corresponds to reference rectangles' radii
$r$ corresponds to fitted ellipses' radii

All reference rectangles have $A = 1.0$, $B = -2.0$, and $\theta = 0.5$. 
### TABLE 6. PARAMETERS OF ELLIPSES FITTED TO RECTANGLES
#### BY THE ONE STEP MINIMIZATION METHOD

(48 DATA POINTS)

<table>
<thead>
<tr>
<th>R_x</th>
<th>R_y</th>
<th>R_y/R_x</th>
<th>r_x</th>
<th>r_y</th>
<th>r_y/r_x</th>
<th>A</th>
<th>B</th>
<th>θ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.1</td>
<td>0.100</td>
<td>1.307</td>
<td>0.111</td>
<td>0.085</td>
<td>0.640</td>
<td>-2.206</td>
<td>0.515</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.200</td>
<td>1.246</td>
<td>0.207</td>
<td>0.166</td>
<td>1.001</td>
<td>-2.001</td>
<td>0.503</td>
</tr>
<tr>
<td>1.0</td>
<td>0.3</td>
<td>0.300</td>
<td>1.277</td>
<td>0.318</td>
<td>0.249</td>
<td>1.001</td>
<td>-2.002</td>
<td>0.501</td>
</tr>
<tr>
<td>1.0</td>
<td>0.4</td>
<td>0.400</td>
<td>1.282</td>
<td>0.424</td>
<td>0.331</td>
<td>1.001</td>
<td>-2.005</td>
<td>0.500</td>
</tr>
<tr>
<td>1.0</td>
<td>0.5</td>
<td>0.500</td>
<td>1.284</td>
<td>0.529</td>
<td>0.412</td>
<td>1.003</td>
<td>-2.007</td>
<td>0.499</td>
</tr>
<tr>
<td>1.0</td>
<td>0.6</td>
<td>0.600</td>
<td>1.288</td>
<td>0.632</td>
<td>0.491</td>
<td>1.004</td>
<td>-2.010</td>
<td>0.498</td>
</tr>
<tr>
<td>1.0</td>
<td>0.7</td>
<td>0.700</td>
<td>1.292</td>
<td>0.735</td>
<td>0.569</td>
<td>1.006</td>
<td>-2.013</td>
<td>0.497</td>
</tr>
<tr>
<td>1.0</td>
<td>0.8</td>
<td>0.800</td>
<td>1.299</td>
<td>0.835</td>
<td>0.643</td>
<td>1.009</td>
<td>-2.018</td>
<td>0.495</td>
</tr>
<tr>
<td>1.0</td>
<td>0.9</td>
<td>0.900</td>
<td>1.307</td>
<td>0.933</td>
<td>0.714</td>
<td>1.012</td>
<td>-2.024</td>
<td>0.491</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>1.000</td>
<td>1.317</td>
<td>1.028</td>
<td>0.781</td>
<td>1.015</td>
<td>-2.031</td>
<td>0.486</td>
</tr>
<tr>
<td>2.0</td>
<td>0.2</td>
<td>0.100</td>
<td>2.537</td>
<td>0.211</td>
<td>0.083</td>
<td>0.977</td>
<td>-2.014</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4</td>
<td>0.200</td>
<td>2.564</td>
<td>0.424</td>
<td>0.165</td>
<td>1.002</td>
<td>-2.004</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.6</td>
<td>0.300</td>
<td>2.576</td>
<td>0.633</td>
<td>0.246</td>
<td>1.004</td>
<td>-2.010</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.8</td>
<td>0.400</td>
<td>2.598</td>
<td>0.835</td>
<td>0.321</td>
<td>1.009</td>
<td>-2.018</td>
<td>0.499</td>
</tr>
<tr>
<td>2.0</td>
<td>1.0</td>
<td>0.500</td>
<td>2.635</td>
<td>1.028</td>
<td>0.390</td>
<td>1.015</td>
<td>-2.032</td>
<td>0.498</td>
</tr>
<tr>
<td>2.0</td>
<td>1.2</td>
<td>0.600</td>
<td>2.702</td>
<td>1.203</td>
<td>0.445</td>
<td>1.025</td>
<td>-2.052</td>
<td>0.497</td>
</tr>
<tr>
<td>2.0</td>
<td>1.4</td>
<td>0.700</td>
<td>2.845</td>
<td>1.343</td>
<td>0.472</td>
<td>1.040</td>
<td>-2.086</td>
<td>0.495</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>0.800</td>
<td>3.323</td>
<td>1.401</td>
<td>0.422</td>
<td>1.065</td>
<td>-2.152</td>
<td>0.492</td>
</tr>
<tr>
<td>2.0</td>
<td>1.8</td>
<td>0.900</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Not able to fit ellipse to data.

**Note:** R corresponds to reference rectangles' radii

r corresponds to fitted ellipses' radii

All reference rectangles have A = 1.0, B = -2.0, and θ = 0.5
TABLE 7. PARAMETERS OF ELLIPSES FITTED TO RECTANGLES
BY THE ONE STEP MINIMIZATION METHOD
(100 DATA POINTS)

<table>
<thead>
<tr>
<th>Rx</th>
<th>Ry</th>
<th>R_y/R_x</th>
<th>r_x</th>
<th>r_y</th>
<th>r_y/r_x</th>
<th>A</th>
<th>B</th>
<th>θ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.1</td>
<td>0.100</td>
<td>0.491</td>
<td>0.044</td>
<td>0.090</td>
<td>0.722</td>
<td>-2.141</td>
<td>0.506</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2</td>
<td>0.200</td>
<td>1.256</td>
<td>0.213</td>
<td>0.170</td>
<td>1.015</td>
<td>-1.992</td>
<td>0.503</td>
</tr>
<tr>
<td>1.0</td>
<td>0.3</td>
<td>0.300</td>
<td>1.265</td>
<td>0.320</td>
<td>0.253</td>
<td>0.995</td>
<td>-2.004</td>
<td>0.501</td>
</tr>
<tr>
<td>1.0</td>
<td>0.4</td>
<td>0.400</td>
<td>1.269</td>
<td>0.425</td>
<td>0.335</td>
<td>1.000</td>
<td>-2.004</td>
<td>0.500</td>
</tr>
<tr>
<td>1.0</td>
<td>0.5</td>
<td>0.500</td>
<td>1.273</td>
<td>0.530</td>
<td>0.416</td>
<td>1.002</td>
<td>-2.006</td>
<td>0.499</td>
</tr>
<tr>
<td>1.0</td>
<td>0.6</td>
<td>0.600</td>
<td>1.277</td>
<td>0.635</td>
<td>0.497</td>
<td>1.004</td>
<td>-2.009</td>
<td>0.498</td>
</tr>
<tr>
<td>1.0</td>
<td>0.7</td>
<td>0.700</td>
<td>1.281</td>
<td>0.737</td>
<td>0.575</td>
<td>1.006</td>
<td>-2.013</td>
<td>0.496</td>
</tr>
<tr>
<td>1.0</td>
<td>0.8</td>
<td>0.800</td>
<td>1.287</td>
<td>0.838</td>
<td>0.651</td>
<td>1.008</td>
<td>-2.017</td>
<td>0.495</td>
</tr>
<tr>
<td>1.0</td>
<td>0.9</td>
<td>0.900</td>
<td>1.295</td>
<td>0.937</td>
<td>0.724</td>
<td>1.011</td>
<td>-2.023</td>
<td>0.491</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>1.000</td>
<td>1.305</td>
<td>1.032</td>
<td>0.791</td>
<td>1.015</td>
<td>-2.030</td>
<td>0.486</td>
</tr>
<tr>
<td>2.0</td>
<td>0.2</td>
<td>0.100</td>
<td>2.552</td>
<td>0.214</td>
<td>0.084</td>
<td>0.995</td>
<td>-2.005</td>
<td>0.497</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4</td>
<td>0.200</td>
<td>2.543</td>
<td>0.425</td>
<td>0.167</td>
<td>1.003</td>
<td>-2.004</td>
<td>0.499</td>
</tr>
<tr>
<td>2.0</td>
<td>0.6</td>
<td>0.300</td>
<td>2.556</td>
<td>0.634</td>
<td>0.248</td>
<td>1.005</td>
<td>-2.009</td>
<td>0.499</td>
</tr>
<tr>
<td>2.0</td>
<td>0.8</td>
<td>0.400</td>
<td>2.577</td>
<td>0.838</td>
<td>0.325</td>
<td>1.008</td>
<td>-2.018</td>
<td>0.499</td>
</tr>
<tr>
<td>2.0</td>
<td>1.0</td>
<td>0.500</td>
<td>2.611</td>
<td>1.032</td>
<td>0.395</td>
<td>1.015</td>
<td>-2.030</td>
<td>0.498</td>
</tr>
<tr>
<td>2.0</td>
<td>1.2</td>
<td>0.600</td>
<td>2.672</td>
<td>1.209</td>
<td>0.452</td>
<td>1.024</td>
<td>-2.050</td>
<td>0.497</td>
</tr>
<tr>
<td>2.0</td>
<td>1.4</td>
<td>0.700</td>
<td>2.801</td>
<td>1.353</td>
<td>0.483</td>
<td>1.039</td>
<td>-2.083</td>
<td>0.495</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>0.800</td>
<td>3.215</td>
<td>1.417</td>
<td>0.441</td>
<td>1.064</td>
<td>-2.146</td>
<td>0.492</td>
</tr>
<tr>
<td>2.0</td>
<td>1.8</td>
<td>0.900</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>1.000</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

*Not able to fit ellipse to data.

Note: R corresponds to reference rectangles' radii
r corresponds to fitted ellipses' radii
All reference rectangles have A = 1.0, B = -2.0, and θ = 0.5
TABLE 8. PARAMETERS OF ELLIPSES FITTED TO RECTANGLES
BY THE ITERATIVE MINIMIZATION METHOD
(8 DATA POINTS)

<table>
<thead>
<tr>
<th>R_x</th>
<th>R_y</th>
<th>R_y/R_x</th>
<th>r_x</th>
<th>r_y</th>
<th>r_y/r_x</th>
<th>A</th>
<th>B</th>
<th>θ</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>0.2</td>
<td>0.100</td>
<td>2.582</td>
<td>0.258</td>
<td>0.100</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4</td>
<td>0.200</td>
<td>2.582</td>
<td>0.516</td>
<td>0.200</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.6</td>
<td>0.300</td>
<td>2.582</td>
<td>0.775</td>
<td>0.300</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.8</td>
<td>0.400</td>
<td>2.582</td>
<td>1.003</td>
<td>0.400</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.0</td>
<td>0.500</td>
<td>2.582</td>
<td>1.291</td>
<td>0.500</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.2</td>
<td>0.600</td>
<td>2.582</td>
<td>1.549</td>
<td>0.600</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.4</td>
<td>0.700</td>
<td>2.582</td>
<td>1.807</td>
<td>0.700</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>0.800</td>
<td>2.582</td>
<td>2.066</td>
<td>0.800</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.8</td>
<td>0.900</td>
<td>2.582</td>
<td>2.324</td>
<td>0.900</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>1.000</td>
<td>2.582</td>
<td>2.582</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.454</td>
</tr>
</tbody>
</table>

r = ellipse radii
R = rectangle radii

Scale Factor $k_s = \frac{R_y}{R_x} = \frac{r_y}{r_x} = 0.775$

8 data points
### TABLE 9. PARAMETERS OF ELLIPSES FITTED TO RECTANGLES BY THE ITERATIVE MINIMIZATION METHOD (20 DATA POINTS)

<table>
<thead>
<tr>
<th>$R_x$</th>
<th>$R_y$</th>
<th>$R_y/R_x$</th>
<th>$r_x$</th>
<th>$r_y$</th>
<th>$r_y/r_x$</th>
<th>$A$</th>
<th>$B$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>0.2</td>
<td>0.100</td>
<td>2.409</td>
<td>0.241</td>
<td>0.100</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4</td>
<td>0.200</td>
<td>2.409</td>
<td>0.482</td>
<td>0.200</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.6</td>
<td>0.300</td>
<td>2.409</td>
<td>0.723</td>
<td>0.300</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.8</td>
<td>0.400</td>
<td>2.409</td>
<td>0.963</td>
<td>0.400</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.0</td>
<td>0.500</td>
<td>2.409</td>
<td>1.204</td>
<td>0.500</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.2</td>
<td>0.600</td>
<td>2.409</td>
<td>1.445</td>
<td>0.600</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.4</td>
<td>0.700</td>
<td>2.409</td>
<td>1.686</td>
<td>0.700</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>0.800</td>
<td>2.409</td>
<td>1.927</td>
<td>0.800</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.8</td>
<td>0.900</td>
<td>2.409</td>
<td>2.168</td>
<td>0.900</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>1.000</td>
<td>2.409</td>
<td>2.409</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
</tbody>
</table>

$r$ - ellipse radii  
$R$ - rectangle radii  

Scale Factor  

\[ k = \frac{R_x}{r_x} = \frac{R_y}{r_y} = 0.830 \]

20 data points
TABLE 10. PARAMETERS OF ELLIPSES FITTED TO RECTANGLES
BY THE ITERATIVE MINIMIZATION METHOD
(48 DATA POINTS)

<table>
<thead>
<tr>
<th>Rx</th>
<th>R_y</th>
<th>R_y/R_x</th>
<th>r_x</th>
<th>r_y</th>
<th>r_y/r_x</th>
<th>A</th>
<th>B</th>
<th>θ</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>0.2</td>
<td>0.100</td>
<td>2.374</td>
<td>0.237</td>
<td>0.100</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4</td>
<td>0.200</td>
<td>2.374</td>
<td>0.475</td>
<td>0.200</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.6</td>
<td>0.300</td>
<td>2.374</td>
<td>0.712</td>
<td>0.300</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.8</td>
<td>0.400</td>
<td>2.374</td>
<td>0.950</td>
<td>0.400</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.0</td>
<td>0.500</td>
<td>2.374</td>
<td>1.187</td>
<td>0.500</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.2</td>
<td>0.600</td>
<td>2.374</td>
<td>1.424</td>
<td>0.600</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.4</td>
<td>0.700</td>
<td>2.374</td>
<td>1.662</td>
<td>0.700</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>0.800</td>
<td>2.374</td>
<td>1.899</td>
<td>0.800</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.8</td>
<td>0.900</td>
<td>2.374</td>
<td>2.137</td>
<td>0.900</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>1.000</td>
<td>2.374</td>
<td>2.374</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.864</td>
</tr>
</tbody>
</table>

r - ellipse radii
R - rectangle radii

Scale Factor \( k_{x,y} = \frac{R_x}{r_x} = \frac{R_y}{r_y} = 0.842 \)

48 data points
TABLE 11. PARAMETERS OF ELLIPSES FITTED TO RECTANGLES
BY THE ITERATIVE MINIMIZATION METHOD
(100 DATA POINTS)

<table>
<thead>
<tr>
<th>$R_x$</th>
<th>$R_y$</th>
<th>$R_y/R_x$</th>
<th>$r_x$</th>
<th>$r_y$</th>
<th>$r_y/r_x$</th>
<th>$A$</th>
<th>$B$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>0.2</td>
<td>0.100</td>
<td>2.368</td>
<td>0.237</td>
<td>0.100</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.4</td>
<td>0.200</td>
<td>2.368</td>
<td>0.474</td>
<td>0.200</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.6</td>
<td>0.300</td>
<td>2.368</td>
<td>0.710</td>
<td>0.300</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.8</td>
<td>0.400</td>
<td>2.368</td>
<td>0.947</td>
<td>0.400</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.0</td>
<td>0.500</td>
<td>2.368</td>
<td>1.184</td>
<td>0.500</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.2</td>
<td>0.600</td>
<td>2.368</td>
<td>1.421</td>
<td>0.600</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.4</td>
<td>0.700</td>
<td>2.368</td>
<td>1.658</td>
<td>0.700</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.6</td>
<td>0.800</td>
<td>2.369</td>
<td>1.895</td>
<td>0.800</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>1.8</td>
<td>0.900</td>
<td>2.368</td>
<td>2.131</td>
<td>0.900</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>1.000</td>
<td>2.368</td>
<td>2.368</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
</tbody>
</table>

$r$ - ellipse radii  
$R$ - rectangle radii  

Scale Factor $k_{uv} = \frac{R_x}{r_x} = \frac{R_y}{r_y} = 0.845$ for 100 data points
TABLE 12. RECTANGLE PARAMETER ESTIMATES OBTAINED BY
THE TEMPLATE MATCHING METHOD

[Reference rectangle parameters are: \( r_x=2.0, r_y=1.0, A=1.0, B=-2.0, \theta=0.5 \)]

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>N</th>
<th>( r_x )</th>
<th>( r_y )</th>
<th>A</th>
<th>B</th>
<th>( \theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>8</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.0</td>
<td>20</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.0</td>
<td>48</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.0</td>
<td>100</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.1</td>
<td>8</td>
<td>1.990</td>
<td>1.036</td>
<td>1.028</td>
<td>-1.949</td>
<td>0.486</td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
<td>2.087</td>
<td>1.007</td>
<td>1.015</td>
<td>-2.024</td>
<td>0.521</td>
</tr>
<tr>
<td>0.1</td>
<td>48</td>
<td>2.027</td>
<td>0.968</td>
<td>0.978</td>
<td>-2.087</td>
<td>0.523</td>
</tr>
<tr>
<td>0.1</td>
<td>100</td>
<td>1.976</td>
<td>1.008</td>
<td>0.973</td>
<td>-1.961</td>
<td>0.504</td>
</tr>
<tr>
<td>0.2</td>
<td>8</td>
<td>1.860</td>
<td>1.122</td>
<td>0.920</td>
<td>-1.990</td>
<td>0.404</td>
</tr>
<tr>
<td>0.2</td>
<td>20</td>
<td>2.097</td>
<td>1.115</td>
<td>0.939</td>
<td>-2.089</td>
<td>0.566</td>
</tr>
<tr>
<td>0.2</td>
<td>48</td>
<td>2.072</td>
<td>1.068</td>
<td>1.034</td>
<td>-1.998</td>
<td>0.479</td>
</tr>
<tr>
<td>0.2</td>
<td>100</td>
<td>2.043</td>
<td>0.972</td>
<td>1.007</td>
<td>-2.007</td>
<td>0.468</td>
</tr>
<tr>
<td>0.3</td>
<td>8</td>
<td>1.978</td>
<td>1.275</td>
<td>1.250</td>
<td>-2.213</td>
<td>0.466</td>
</tr>
<tr>
<td>0.3</td>
<td>20</td>
<td>1.923</td>
<td>0.879</td>
<td>0.973</td>
<td>-2.020</td>
<td>0.574</td>
</tr>
<tr>
<td>0.3</td>
<td>48</td>
<td>1.963</td>
<td>1.046</td>
<td>0.975</td>
<td>-2.003</td>
<td>0.472</td>
</tr>
<tr>
<td>0.3</td>
<td>100</td>
<td>1.885</td>
<td>1.074</td>
<td>1.090</td>
<td>-1.860</td>
<td>0.539</td>
</tr>
<tr>
<td>0.4</td>
<td>8</td>
<td>2.188</td>
<td>0.911</td>
<td>0.702</td>
<td>-1.619</td>
<td>0.638</td>
</tr>
<tr>
<td>0.4</td>
<td>20</td>
<td>1.936</td>
<td>1.151</td>
<td>1.072</td>
<td>-1.930</td>
<td>0.441</td>
</tr>
<tr>
<td>0.4</td>
<td>48</td>
<td>1.890</td>
<td>1.065</td>
<td>1.076</td>
<td>-1.979</td>
<td>0.502</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td>2.008</td>
<td>1.083</td>
<td>1.180</td>
<td>-1.992</td>
<td>0.520</td>
</tr>
<tr>
<td>0.5</td>
<td>8</td>
<td>2.158</td>
<td>1.220</td>
<td>1.322</td>
<td>-2.214</td>
<td>0.774</td>
</tr>
<tr>
<td>0.5</td>
<td>20</td>
<td>2.276</td>
<td>1.541</td>
<td>0.824</td>
<td>-1.639</td>
<td>1.068</td>
</tr>
<tr>
<td>0.5</td>
<td>48</td>
<td>2.088</td>
<td>1.056</td>
<td>1.010</td>
<td>-1.854</td>
<td>0.309</td>
</tr>
<tr>
<td>0.5</td>
<td>100</td>
<td>1.801</td>
<td>1.235</td>
<td>0.906</td>
<td>-2.174</td>
<td>0.600</td>
</tr>
</tbody>
</table>
PLATE VII. RECTANGLES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD (σ = 0.0)
PLATE VIII. RECTANGLES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD (σ = 0.1)

FIELD OF VIEW

(a) Reference Rectangle
- Radius: 2.000
- Angle: 0.079
- Translation: 1.000
- Rotation: 0.000

(b) Least-Squares Rectangle
- Radius: 2.000
- Angle: 0.079
- Translation: 1.000
- Rotation: 0.000

FIELD OF VIEW

(c) Reference Rectangle
- Radius: 2.000
- Angle: 1.057
- Translation: 1.000
- Rotation: 0.000

(d) Least-Squares Rectangle
- Radius: 1.917
- Angle: 1.057
- Translation: 1.000
- Rotation: 0.000
PLATE IX. RECTANGLES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD (σ = 0.2)

FIELD OF VIEW

10 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE RECTANGLE
S-RADIUS = 2.000
S-ANGLE = 1.000
S-TRANSLATION = 1.000
MOTION IN DEGREES = 29.000

LEAST-SQUARES RECTANGLE
S-RADIUS = 2.000
S-ANGLE = 1.000
S-TRANSLATION = 1.000
MOTION IN DEGREES = 29.000

(a)

FIELD OF VIEW

10 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE RECTANGLE
S-RADIUS = 2.000
S-ANGLE = 1.000
S-TRANSLATION = 1.000
MOTION IN DEGREES = 29.000

LEAST-SQUARES RECTANGLE
S-RADIUS = 2.000
S-ANGLE = 1.000
S-TRANSLATION = 1.000
MOTION IN DEGREES = 29.000

(b)

FIELD OF VIEW

10 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE RECTANGLE
S-RADIUS = 2.000
S-ANGLE = 1.000
S-TRANSLATION = 1.000
MOTION IN DEGREES = 29.000

LEAST-SQUARES RECTANGLE
S-RADIUS = 2.000
S-ANGLE = 1.000
S-TRANSLATION = 1.000
MOTION IN DEGREES = 29.000

(c)

FIELD OF VIEW

10 DATA POINTS
STANDARD DEVIATION = 0.2

REFERENCE RECTANGLE
S-RADIUS = 2.000
S-ANGLE = 1.000
S-TRANSLATION = 1.000
MOTION IN DEGREES = 29.000

LEAST-SQUARES RECTANGLE
S-RADIUS = 2.000
S-ANGLE = 1.000
S-TRANSLATION = 1.000
MOTION IN DEGREES = 29.000

(d)
PLATE X. RECTANGLES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD (σ = 0.3)

FIELD OF VIEW

REFERENCE RECTANGLE
- MINUS = 2.000
- SCALE = 1.000
- TRANSLATION = 1.000
- ROTATION IN DEGREES = 28.640

LEAST-SQUARES RECTANGLE
- MINUS = 2.000
- SCALE = 1.000
- TRANSLATION = 1.000
- ROTATION IN DEGREES = 27.904

FIELD OF VIEW

REFERENCE RECTANGLE
- MINUS = 2.000
- SCALE = 1.000
- TRANSLATION = 1.000
- ROTATION IN DEGREES = 28.640

LEAST-SQUARES RECTANGLE
- MINUS = 2.000
- SCALE = 1.000
- TRANSLATION = 1.000
- ROTATION IN DEGREES = 27.904

FIELD OF VIEW

REFERENCE RECTANGLE
- MINUS = 2.000
- SCALE = 1.000
- TRANSLATION = 1.000
- ROTATION IN DEGREES = 28.640

LEAST-SQUARES RECTANGLE
- MINUS = 2.000
- SCALE = 1.000
- TRANSLATION = 1.000
- ROTATION IN DEGREES = 27.904
PLATE XI. RECTANGLES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD (σ = 0.4)

(a) REFERENCE RECTANGLE
1. RADIUS = 2.000
2. TRANSLATION = 0.000
3. ROTATION IN DEGREES = 20.048

LEAST-SQUARES RECTANGLE
1. RADIUS = 2.000
2. TRANSLATION = 0.000
3. ROTATION IN DEGREES = 20.048

(b) REFERENCE RECTANGLE
1. RADIUS = 2.000
2. TRANSLATION = 0.000
3. ROTATION IN DEGREES = 67.732

LEAST-SQUARES RECTANGLE
1. RADIUS = 2.000
2. TRANSLATION = 0.000
3. ROTATION IN DEGREES = 20.048

(c) REFERENCE RECTANGLE
1. RADIUS = 2.000
2. TRANSLATION = 0.000
3. ROTATION IN DEGREES = 20.048

LEAST-SQUARES RECTANGLE
1. RADIUS = 1.085
2. TRANSLATION = 0.000
3. ROTATION IN DEGREES = 20.452

(d) REFERENCE RECTANGLE
1. RADIUS = 2.000
2. TRANSLATION = 0.000
3. ROTATION IN DEGREES = 20.452

LEAST-SQUARES RECTANGLE
1. RADIUS = 1.065
2. TRANSLATION = 0.000
3. ROTATION IN DEGREES = 12.449
PLATE XII. RECTANGLES FITTED TO DATA POINTS BY THE ITERATIVE MINIMIZATION METHOD (σ = 0.5)

(a) REFERENCE RECTANGLE
- RADIUS: 6.000
- TRANSLATION: 1.000
- ROTATION IN DEGREES: 29.500

(b) LEAST-SQUARES RECTANGLE
- RADIUS: 6.120
- TRANSLATION: 0.999
- ROTATION IN DEGREES: 29.110

(c) REFERENCE RECTANGLE
- RADIUS: 6.000
- TRANSLATION: 1.000
- ROTATION IN DEGREES: 29.500

(d) LEAST-SQUARES RECTANGLE
- RADIUS: 6.120
- TRANSLATION: 0.999
- ROTATION IN DEGREES: 29.110
CHAPTER VI
RECOGNITION OF CRESCENTS

6.1 Introduction

This chapter presents the procedure which was developed to recognize both convex and concave crescents. One normally associates a crescent with the moon when it is in its first phase, i.e., a concave crescent. However, the term crescent as referred to in this research shall be synonymous with the image of the moon during any of its phases, which includes both concave and convex patterns. The criterion function corresponding to the sum of the squared radial distances of the data points to the template was the only criterion function which was considered since it was the only one which had the capability to associate a given data point with the appropriate side of a crescent template. Section 6.2 discusses the procedure for fitting a convex crescent template to a set of data points, as well as deriving the expression for the error function in terms of the five parameters. Section 6.3 proceeds through a similar discussion for a concave crescent template. Section 6.4 then presents the results of fitting these
templates to data points which lie on the boundary of the appropriate crescent.

6.2 Recognition of Convex Crescents

A convex crescent is illustrated in Figure 9. This crescent is composed of two sides as shown. Since the crescent corresponds to the image of a heavenly body, it is clear that Side 1 is a semicircle, which is simply a special case of an ellipse in which both radii are equal. Side 2 is a semiellipse, in which the larger radius, \( r_x \), is the equal to the radius of the semicircle while the smaller radius may have any value between zero and \( r_x \).

Once the initial guess for the five parameters characterizing the convex crescent template is specified (this is given by the ESTIMT subroutine) it is necessary to determine to which side a given data point is to be associated, and to determine the corresponding error in the radial distance from the data point to the template. After a procedure has been given for these two steps, the sum-squared error criterion function may be determined, and iterations continue until this criterion function is minimized, yielding the best fitting convex crescent template.

To this end, consider the convex crescent template which is illustrated in Figure 10. In Figure 10 are four data points, each of which is in a different
Figure 9. Example of a Convex Crescent

Figure 10. Determination of Convex Crescent Model Point Associated with a Given Data Point
quadrant of the \( w,z \)-coordinate system. The coordinates of any of the data points may be found with respect to the \( w,z \)-coordinate system from the transformation

\[
\begin{align*}
    w_i &= (x_i-A) \cos \theta + (y_i-B) \sin \theta \\
    z_i &= -(x_i-A) \sin \theta + (y_i-B) \cos \theta
\end{align*}
\] (6-1)

The angle of inclination, \( \psi_i \), of any data point \((w_i,z_i)\) with respect to the \( w \)-axis is given by

\[
\psi_i = \tan^{-1} \frac{z_i}{w_i}
\] (6-2)

The data point correspondence problem may be solved in the following manner. If a data point lies in the first or second quadrant of the \( w,z \)-reference frame, then it is to be associated with the semicircle side of the template. Data points lying in the other two quadrants are to be associated with the semiellipse side of the template. In terms of the angle \( \psi_i \), this means

\[
\begin{align*}
    (x_i,y_i) &\sim \text{semicircle} \quad \text{for} \quad 0 \leq \psi_i \leq \pi \\
    (x_i,y_i) &\sim \text{semiellipse} \quad \text{for} \quad \pi < \psi_i < 2\pi
\end{align*}
\] (6-3)

The radial distance from a given data point to the appropriate side of the template must now be derived. This distance will then be the error associated with this data point for this template. This derivation is quite
similar to that for an ellipse, but will be carried out for completeness.

With respect to the \(w, z\)-reference frame the equation for the semicircle side of the template (or model) is

\[
\frac{w_m^2}{r_x^2} + \frac{z_m^2}{r_x^2} = 1 \quad (0 \leq \psi \leq \pi)
\]  

\[\text{(6-4)}\]

while the equation for the semiellipse side of the template is

\[
\frac{w_m^2}{r_x^2} + \frac{z_m^2}{r_y^2} = 1 \quad (\pi < \psi < 2\pi)
\]  

\[\text{(6-5)}\]

It is seen that Equation (6-4) is just a special case of Equation (6-5) with \(r_y\) set equal to \(r_x\).

In order to determine the model point \((w_m, z_m)\) (point on the template) one must, conceptually, draw a radial line from the origin of the \(w, z\)-reference frame to a given data point, \((w_d, z_d)\), and find the point of intersection of this radial line with the template. This radial line makes the angle \(\psi\) with the \(w\)-axis where

\[
\tan \psi = \frac{z_d}{w_d}
\]  

\[\text{(6-6)}\]

which is the same as Equation (6-2) except that the "i"
subscript has been dropped. Since the model point must also lie on this radial line,

\[ \tan \psi = \frac{z_m}{w_m} \]  (6-7)

Therefore

\[ z_m = w_m \tan \psi \]  (6-8)

or

\[ z_m = w_m \frac{z_d}{w_d} \]  (6-9)

Equation (6-9) and Equations (6-4) or (6-5) give two relations which the coordinates of the model point must satisfy. Substituting Equation (6-9) into Equation (6-5) yields

\[ \frac{w_m^2}{r_x^2} + \frac{w_m^2 (\frac{z_d}{w_d})^2}{r_y^2} = 1 \]  (6-10)

Solving for \( w_m \) gives

\[ w_m = \pm \frac{r_x}{\sqrt{1 + (\frac{z_d}{w_d})^2}} \]  (6-11) \[ \text{if } 0 \leq \psi \leq \frac{\pi}{2} \]

\[ w_m = \pm \frac{r_x}{\sqrt{1 + (\frac{z_d}{w_d})^2}} \]  (6-12) \[ \text{if } \frac{\pi}{2} < \psi < \pi \]
where Equations (6-11) and (6-12) are special cases of Equations (6-14) and (6-13), respectively, with \( r_y = r_x \).

The associated z-coordinate of the model point is then given by Equation (6-9).

The \( x, y \)-coordinates of the model point may be determined from the following transformation.

\[
\begin{align*}
  x_m &= w_m \cos \theta - z_m \sin \theta + A \\
  y_m &= w_m \sin \theta + z_m \cos \theta + B
\end{align*}
\]

The error is then the difference in the \( x \) -and \( y \)-coordinates of the data point and the model point.

Thus

\[
\begin{align*}
  e_i &= (x_d - x_m)_i \\
  e_{i+N} &= (y_d - y_m)_i
\end{align*}
\]
and

\[ \phi = \sum_{i=1}^{2N} e_i^2 \]

where the "i" subscript denotes each data point.

The partial derivatives of the coordinates \((x_m, y_m)\) with respect to the five parameters \(r_x, r_y, A, B, \) and \(\theta\) and the error as given by Equation (6-16) may now be used in the Gauss-Newton or the Newton-Raphson minimization scheme to vary the five parameters in such a manner as to produce a minimum value for the sum-squared error criterion function (Equation 6-17). Appendix D contains a detailed listing of these partial derivatives as well as the manner in which they enter the algorithm.

6.3 Recognition of Concave Crescents

A concave crescent is illustrated in Figure 11. One can see that it is similar to a convex crescent in that it has one side which is a semicircle, and one side which is a semiellipse. In fact, the primary difference in the scheme which is employed to recognize a concave crescent from that which is used to recognize a convex
Figure 11. Example of a Concave Crescent

Figure 12. Determination of Concave Crescent Model Point Associated with a Given Data Point
crescent is the method by which the point correspondence problem is solved.

Figure 12 illustrates a concave crescent template in the x,y-reference frame. It can be seen that a radial line which is drawn from the origin of the w,z-coordinate system to any data point which lies in the third or fourth quadrant of the w,z-reference frame will not intersect the template. Therefore, it is necessary to discard (for this iteration) any data point lying in the third or fourth quadrant of the w,z-reference frame. If the angle $\psi$ is defined the same as it was previously defined, then all data points are discarded for which $\pi < \psi < 2\pi$.

For those data points which lie in the first or second quadrant of the w,z-reference frame, it is still necessary to determine which side of the concave crescent template will be associated with each of these data points. A sensible procedure is to determine the radial distance that a given data point is from each of the two sides, and then associate the data point with the side corresponding to the smaller of these two distances.

The equation for the error associated with each valid data point is the same as for the convex crescent template as given by Equation (6-16), where
\[ w_m = + \frac{r_x}{\sqrt{1 + \left( \frac{z_d}{w_d} \right)^2}} \quad \text{for } 0 \leq \psi \leq \frac{\pi}{2} \quad (6-18) \]

or

\[ w_m = - \frac{r_x}{\sqrt{1 + \left( \frac{z_d}{w_d} \right)^2}} \quad \text{for } \frac{\pi}{2} < \psi \leq \pi \quad (6-19) \]

if the data point lies closer to the semicircle side of the template, and

\[ w_m = + \frac{r_x r_y}{\sqrt{r_y^2 + r_x^2 \left( \frac{z_d}{w_d} \right)^2}} \quad \text{for } 0 \leq \psi \leq \frac{\pi}{2} \quad (6-20) \]

or

\[ w_m = - \frac{r_x r_y}{\sqrt{r_y^2 + r_x^2 \left( \frac{z_d}{w_d} \right)^2}} \quad \text{for } \frac{\pi}{2} < \psi \leq \pi \quad (6-21) \]

if the data point lies closer to the semiellipse side of the template. The z-coordinate of the model point is given by Equation (6-9), and repeated below.

\[ z_m = w_m \times \frac{z_d}{w_d} \quad (6-22) \]

It should be made clear that at the next iteration a concave crescent template corresponding to the new value for the parameter vector will cause, in general, a new set of data points to be discarded. The valid data points for this new template may include some of the discarded
data points from the previous template. However, as the parameter estimates become close to the value which minimizes the criterion function, nearly all of the data points which are rejected remain the same.

Two further points should also be clarified at this time. First of all, it is necessary to slightly modify the criterion function so that the discarded data points are taken into account. The criterion function has been defined as

\[ \phi = \sum_{i=1}^{k} \left( (x_d - x_m)_i^2 + (y_d - y_m)_i^2 \right) \]  

(6-23)

where there are k retained data points, and \(0 \leq k \leq N\). This criterion function will automatically become smaller as more data points are discarded since fewer terms will be added. The iteration scheme would therefore have a tendency to retain very few data points, causing the best fitting template to be a very poor fit to the entire group of data points. Thus it is necessary to define the criterion function in terms of a squared error per retained data point. Hence

\[ \phi_{\text{new}} = \frac{\phi}{k} \]  

(6-24)

where \(\phi\) is given by Equation (6-23).
The second point which needs to be clarified involves the procedure to be followed when a concave crescent template is being fitted to a set of data points which lie on the boundary of one of the other three patterns. Quite typically when this is the case the number of discarded data points becomes equal to nearly one half of the total number of data points. It seems reasonable to reject the results of fitting a concave crescent template to a set of data points when the number of discarded data points exceeds twenty percent of the total number of data points.

Although it may not be obvious to the casual reader, there is one degenerate case for either a convex or a concave crescent in which both templates fail to fit the data points in an acceptable manner. This degenerate case occurs when \( r_y = 0 \). It is obvious that a radial line going from the origin of the \( w,z \)-reference frame to a data point will not intersect the semiellipse side of the template when that side has degenerated into a straight line. No attempt was made to modify the recognition procedures to handle this degenerate case, although one possibility is the following scheme. Determine the number of data points which lie within a small, specified arc about the positive and negative \( w \)-axis. If this number exceeds a specified percent (40\%, for example) of the total number of data points,
then it is concluded that these points lie on the \( w \)-axis and hence \( r_y = 0 \).

6.4 Implementation of the Crescent Templates

Table 13 shows the results of fitting a convex crescent template to data points which are associated with a convex crescent characterized by

\[
\begin{bmatrix}
  r_x \\
  r_y \\
  A \\
  B \\
  \theta
\end{bmatrix} = \begin{bmatrix}
  2.0 \\
  1.0 \\
  1.0 \\
  -2.0 \\
  0.5
\end{bmatrix} \quad (6-25)
\]

Once again, patterns containing 10, 20, 50, and 100 data points are considered, as well as noise levels of \( \sigma = 0.0, 0.1, 0.2, 0.3, 0.4, \) and 0.5. The initial guess for the parameters of the template was determined by the ESTIMT subroutine.

Table 13 shows that for noiseless data points an exact fit was accomplished, which, of course, is an absolute requirement for any worthwhile recognition scheme. While the estimates for low noise levels \( (\sigma = 0.1 \text{ and } 0.2) \) are quite satisfactory by practically any standard, one can see that the estimates for higher noise levels deteriorate more rapidly than for the ellipse or rectangle case. This is to be expected, however, since a noisy convex crescent may be easily confused with an ellipse.
Table 14 shows the results of fitting a concave crescent template to data points which are associated with a concave crescent characterized by

$$\begin{bmatrix} r_x \\ r_y \\ A \\ B \\ \theta \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.0 \\ 1.0 \\ -2.0 \\ 0.5 \end{bmatrix}$$

(6-26)

The same numbers of data points and the same noise levels are used for the concave crescent pattern as are used for the convex crescent pattern.

Once again, a perfect fit is attained when the data points are noiseless. Furthermore, very good fits are achieved for low noise levels ($\sigma = 0.1$ and 0.2), especially in the 50 and 100 data point cases. The estimates for $\sigma = 0.3$ begin to have questionable merit, while the estimates for high noise levels ($\sigma = 0.4$ and 0.5) cannot be considered of much value. This has demonstrated, however, that for reasonable noise levels it is possible to fit a template to a set of data points which belong to a pattern which is not convex.

6.5 Summary

The results of this chapter, as given in Tables 13 and 14, show that crescents may be recognized with good accuracy for low noise levels by the template matching
scheme which is developed in this chapter. The template matching scheme has therefore been shown to be a powerful recognition scheme since it has the ability to recognize both convex and concave patterns whose boundaries need not be given by an analytical equation.
TABLE 13. CONVEX CRESCENT PARAMETER ESTIMATES OBTAINED BY THE TEMPLATE MATCHING METHOD

[Reference crescent parameters are: \( r_x = 2.0 \), \( r_y = 1.0 \), \( A = 1.0 \), \( B = -2.0 \), \( \theta = 0.5 \)]

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( N )</th>
<th>( r_x )</th>
<th>( r_y )</th>
<th>( A )</th>
<th>( B )</th>
<th>( \theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>10</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.0</td>
<td>20</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.0</td>
<td>50</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.0</td>
<td>100</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.1</td>
<td>10</td>
<td>2.052</td>
<td>0.824</td>
<td>1.076</td>
<td>-2.121</td>
<td>0.319</td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
<td>1.978</td>
<td>1.005</td>
<td>1.004</td>
<td>-1.995</td>
<td>0.532</td>
</tr>
<tr>
<td>0.1</td>
<td>50</td>
<td>1.972</td>
<td>1.006</td>
<td>0.974</td>
<td>-2.006</td>
<td>0.491</td>
</tr>
<tr>
<td>0.1</td>
<td>100</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.2</td>
<td>10</td>
<td>1.820</td>
<td>1.078</td>
<td>0.964</td>
<td>-1.880</td>
<td>0.552</td>
</tr>
<tr>
<td>0.2</td>
<td>20</td>
<td>2.223</td>
<td>0.950</td>
<td>1.026</td>
<td>-2.142</td>
<td>0.434</td>
</tr>
<tr>
<td>0.2</td>
<td>50</td>
<td>1.906</td>
<td>1.266</td>
<td>0.960</td>
<td>-1.835</td>
<td>0.503</td>
</tr>
<tr>
<td>0.2</td>
<td>100</td>
<td>1.960</td>
<td>1.087</td>
<td>1.002</td>
<td>-1.947</td>
<td>0.584</td>
</tr>
<tr>
<td>0.3</td>
<td>10</td>
<td>2.202</td>
<td>0.677</td>
<td>1.091</td>
<td>-1.973</td>
<td>0.537</td>
</tr>
<tr>
<td>0.3</td>
<td>20</td>
<td>1.890</td>
<td>1.277</td>
<td>0.864</td>
<td>-1.631</td>
<td>0.932</td>
</tr>
<tr>
<td>0.3</td>
<td>50</td>
<td>1.834</td>
<td>1.244</td>
<td>0.828</td>
<td>-1.870</td>
<td>0.589</td>
</tr>
<tr>
<td>0.3</td>
<td>100</td>
<td>1.614</td>
<td>2.060</td>
<td>0.661</td>
<td>-1.534</td>
<td>1.571</td>
</tr>
<tr>
<td>0.4</td>
<td>10</td>
<td>2.455</td>
<td>0.611</td>
<td>0.951</td>
<td>-2.350</td>
<td>0.310</td>
</tr>
<tr>
<td>0.4</td>
<td>20</td>
<td>1.931</td>
<td>1.124</td>
<td>0.926</td>
<td>-2.123</td>
<td>0.620</td>
</tr>
<tr>
<td>0.4</td>
<td>50</td>
<td>2.025</td>
<td>1.171</td>
<td>1.087</td>
<td>-1.970</td>
<td>0.843</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td>1.749</td>
<td>1.331</td>
<td>0.970</td>
<td>-1.737</td>
<td>0.551</td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
<td>2.276</td>
<td>1.376</td>
<td>1.144</td>
<td>-2.306</td>
<td>0.872</td>
</tr>
<tr>
<td>0.5</td>
<td>20</td>
<td>1.723</td>
<td>1.617</td>
<td>0.585</td>
<td>-1.778</td>
<td>0.377</td>
</tr>
<tr>
<td>0.5</td>
<td>50</td>
<td>2.260</td>
<td>1.109</td>
<td>0.885</td>
<td>-2.097</td>
<td>0.486</td>
</tr>
<tr>
<td>0.5</td>
<td>100</td>
<td>1.763</td>
<td>1.637</td>
<td>0.827</td>
<td>-1.719</td>
<td>0.496</td>
</tr>
</tbody>
</table>
**TABLE 14. CONCAVE CRESCENT PARAMETER ESTIMATES OBTAINED BY THE TEMPLATE MATCHING METHOD**

[Reference crescent parameters are: \( r_x=2.0, r_y=1.0, A=1.0, B=-2.0, \theta=0.5 \)]

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( N )</th>
<th>( r_x )</th>
<th>( r_y )</th>
<th>( A )</th>
<th>( B )</th>
<th>( \theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>10</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.0</td>
<td>20</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.0</td>
<td>50</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.0</td>
<td>100</td>
<td>2.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-2.000</td>
<td>0.500</td>
</tr>
<tr>
<td>0.1</td>
<td>10</td>
<td>1.794</td>
<td>0.841</td>
<td>1.094</td>
<td>-1.793</td>
<td>0.342</td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
<td>2.008</td>
<td>1.011</td>
<td>0.960</td>
<td>-2.043</td>
<td>0.495</td>
</tr>
<tr>
<td>0.1</td>
<td>50</td>
<td>1.980</td>
<td>1.024</td>
<td>1.008</td>
<td>-2.023</td>
<td>0.508</td>
</tr>
<tr>
<td>0.1</td>
<td>100</td>
<td>2.090</td>
<td>1.110</td>
<td>1.041</td>
<td>-2.053</td>
<td>0.489</td>
</tr>
<tr>
<td>0.2</td>
<td>10</td>
<td>1.877</td>
<td>1.032</td>
<td>0.883</td>
<td>-1.913</td>
<td>0.485</td>
</tr>
<tr>
<td>0.2</td>
<td>20</td>
<td>2.344</td>
<td>1.240</td>
<td>1.109</td>
<td>-2.343</td>
<td>0.559</td>
</tr>
<tr>
<td>0.2</td>
<td>50</td>
<td>2.163</td>
<td>1.087</td>
<td>1.081</td>
<td>-2.119</td>
<td>0.401</td>
</tr>
<tr>
<td>0.2</td>
<td>100</td>
<td>2.231</td>
<td>1.271</td>
<td>1.089</td>
<td>-2.274</td>
<td>0.493</td>
</tr>
<tr>
<td>0.3</td>
<td>10</td>
<td>3.219</td>
<td>1.281</td>
<td>2.090</td>
<td>-2.993</td>
<td>1.220</td>
</tr>
<tr>
<td>0.3</td>
<td>20</td>
<td>2.038</td>
<td>0.900</td>
<td>1.110</td>
<td>-1.756</td>
<td>0.453</td>
</tr>
<tr>
<td>0.3</td>
<td>50</td>
<td>1.890</td>
<td>0.942</td>
<td>0.843</td>
<td>-1.907</td>
<td>0.781</td>
</tr>
<tr>
<td>0.3</td>
<td>100</td>
<td>2.088</td>
<td>1.141</td>
<td>1.020</td>
<td>-2.009</td>
<td>0.612</td>
</tr>
<tr>
<td>0.4</td>
<td>10</td>
<td>2.516</td>
<td>1.425</td>
<td>0.954</td>
<td>-2.476</td>
<td>0.390</td>
</tr>
<tr>
<td>0.4</td>
<td>20</td>
<td>2.223</td>
<td>1.191</td>
<td>1.076</td>
<td>-2.283</td>
<td>0.565</td>
</tr>
<tr>
<td>0.4</td>
<td>50</td>
<td>2.153</td>
<td>0.854</td>
<td>0.931</td>
<td>-1.651</td>
<td>0.508</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td>1.717</td>
<td>0.744</td>
<td>0.829</td>
<td>-1.685</td>
<td>0.201</td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
<td>2.743</td>
<td>1.031</td>
<td>1.466</td>
<td>-2.925</td>
<td>0.690</td>
</tr>
<tr>
<td>0.5</td>
<td>20</td>
<td>2.194</td>
<td>0.636</td>
<td>0.271</td>
<td>-1.191</td>
<td>0.113</td>
</tr>
<tr>
<td>0.5</td>
<td>50</td>
<td>2.493</td>
<td>1.110</td>
<td>0.985</td>
<td>-2.146</td>
<td>0.429</td>
</tr>
<tr>
<td>0.5</td>
<td>100</td>
<td>3.365</td>
<td>2.091</td>
<td>1.715</td>
<td>-3.293</td>
<td>0.195</td>
</tr>
</tbody>
</table>
CHAPTER VII

CONCLUSIONS, SUMMARY OF RESULTS, AND EXTENSIONS

The research undertaken in this dissertation has investigated the feasibility of recognizing planar patterns when a discrete set of data points associated with their boundaries are given. The planar patterns which are considered include ellipses, rectangles, and crescents, the crescents being either convex or concave. Recognition is taken in the sense of estimating the five parameters which characterize both the size and shape of the given patterns, as well as their position (translation and rotation) with respect to a given reference frame. The parameter estimation is performed by an algorithm based upon nonlinear regression analysis.

Three different error functions are investigated. It is found that when the error represents the radial distance from the given data points to the template which is being fitted to them, then excellent recognition results are obtained for all of the patterns which are considered. This was not the case for the other two error functions, however.

140
Since it has been shown that the "radial distance" error yields very good results when a template is fitted to data points which lie on the boundary of a pattern which is the same as the template (even under reasonably high noise conditions), it is now appropriate to demonstrate that this recognition scheme can also distinguish between patterns of different classes.

To this end, each template was fitted to a given pattern and the resulting sum-squared error criterion function for each template was computed. The template corresponding to the smallest criterion function was then selected as the pattern to which the data points belong. The one exception to this rule is that the criterion function corresponding to the concave crescent template is disregarded if more than twenty percent of the data points are discarded.

Tables 15, 16, 17, and 18 show the results of fitting each template to data points associated with the boundary of an ellipse, a rectangle, a convex crescent, and a concave crescent, respectively. The criterion functions which are listed in these tables are normalized by dividing the sum-squared error by the number of retained data points, where the number of retained data points is the same as the total number of data points except when the concave crescent template is being used.
Inspection of Table 15 indicates that the ellipse template fitted the ellipse data points better than any of the other templates, even when the noise level is extremely high ($\sigma = 0.5$). While for higher noise levels ($\sigma = 0.3$ and 0.5) the concave crescent template possesses the smallest value for the criterion function, it should be noted that in these two cases considerably more than twenty percent of the data points have been discarded, and hence the results are rejected.

When the data points are associated with the boundary of a rectangle, Table 16 shows that the rectangle template corresponds to the smallest value for the criterion function for all noise levels considered (disregarding the concave crescent template).

The results of fitting the four templates to data points associated with the boundary of a convex crescent is shown in Table 17. It can be seen that the convex crescent template fits the data points the best for low noise levels ($\sigma = 0.0$ and 0.1). For high noise levels, an ellipse template results in a better least squares fit. This is to be expected, however, since a noisy convex crescent characterized by a two to one ratio of its radii resembles an ellipse to a certain extent.

Finally, Table 18 shows that a concave crescent template results in the best fit when the data points do,
indeed, lie on the boundary of a concave crescent. Special notice should be taken concerning the small percentage of data points which are discarded when the data points actually do belong to a concave crescent.

Thus, it has been shown that the recognition scheme developed in this research has the capability to distinguish each pattern from the other three, and to give a very accurate estimate for the five parameters characterizing the pattern.

While only ellipses, rectangles, and crescents were considered in this research, it appears very feasible that the recognition scheme could be extended to other planar patterns which are more complicated to describe, i.e., patterns which require more than five parameters to totally specify them. In such cases one might have to resort to numerical differentiation if the analytic expressions for the appropriate partial derivatives are very complicated, or difficult to derive.

Furthermore, while all of the results contained in this research relate to two-dimensional patterns, it appears that the basic approach is applicable to three dimensional image analysis as well. Specifically, it seems feasible that a computational procedure could be developed which would be capable of producing a replica of the image of a given three-dimensional object produced
by a particular optical system with a specified spatial relationship to the object. From such a synthetic image, it ought to be straightforward to extract boundary points which could then be compared to the boundary points of the real scene. Iterative adjustment of the spatial parameters, used to generate synthetic images, could then be accomplished by the nonlinear regression program included in this research so as to optimize the fit of the synthetic image to the real image. Such an extension of the present work would permit use of optically derived guidance information in such difficult tasks as automatic orbital rendezvous and docking of spacecraft.

Although noisy data points are presented to the recognition scheme developed here, it has been pointed out that it is additive, uncorrelated gaussian noise which is considered. Further work needs to be done using data which has multiplicative noise or correlated noise to determine the recognition scheme's capability in these cases. Finally, the recognition scheme needs to be tested on real data obtained by some physical electro-optical system.
TABLE 15. VALUES FOR THE SUM-SQUARED ERROR CRITERION FUNCTION OBTAINED BY MATCHING EACH TEMPLATE TO AN ELLIPSE IMAGE (48 DATA POINTS)

<table>
<thead>
<tr>
<th>σ</th>
<th>Template</th>
<th>Discarded Data Points</th>
<th>φ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>Ellipse</td>
<td>0</td>
<td>1.542 x 10^{-12}</td>
</tr>
<tr>
<td>0.0</td>
<td>Rectangle</td>
<td>0</td>
<td>2.536 x 10^{-2}</td>
</tr>
<tr>
<td>0.0</td>
<td>Crescent (Concave)</td>
<td>23</td>
<td>6.378 x 10^{-3}</td>
</tr>
<tr>
<td>0.0</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>2.162 x 10^{-2}</td>
</tr>
<tr>
<td>0.1</td>
<td>Ellipse</td>
<td>0</td>
<td>1.176 x 10^{-2}</td>
</tr>
<tr>
<td>0.1</td>
<td>Rectangle</td>
<td>0</td>
<td>3.828 x 10^{-2}</td>
</tr>
<tr>
<td>0.1</td>
<td>Crescent (Concave)</td>
<td>21</td>
<td>4.497 x 10^{-2}</td>
</tr>
<tr>
<td>0.1</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>3.716 x 10^{-2}</td>
</tr>
<tr>
<td>0.3</td>
<td>Ellipse</td>
<td>0</td>
<td>1.165 x 10^{-1}</td>
</tr>
<tr>
<td>0.3</td>
<td>Rectangle</td>
<td>0</td>
<td>1.298 x 10^{-1}</td>
</tr>
<tr>
<td>0.3</td>
<td>Crescent (Concave)</td>
<td>16</td>
<td>7.797 x 10^{-2}</td>
</tr>
<tr>
<td>0.3</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>1.706 x 10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Ellipse</td>
<td>0</td>
<td>2.948 x 10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Rectangle</td>
<td>0</td>
<td>3.038 x 10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Crescent (Concave)</td>
<td>20</td>
<td>1.750 x 10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>3.478 x 10^{-1}</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Template</td>
<td>Discarded Data Points</td>
<td>$\phi$</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------</td>
<td>-----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>0.0</td>
<td>Ellipse</td>
<td>0</td>
<td>$3.839 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.0</td>
<td>Rectangle</td>
<td>0</td>
<td>$1.567 \times 10^{-1}$</td>
</tr>
<tr>
<td>0.0</td>
<td>Crescent (Concave)</td>
<td>25</td>
<td>$8.158 \times 10^{-3}$</td>
</tr>
<tr>
<td>0.0</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>$4.950 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.1</td>
<td>Ellipse</td>
<td>0</td>
<td>$5.759 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.1</td>
<td>Rectangle</td>
<td>0</td>
<td>$2.058 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.1</td>
<td>Crescent (Concave)</td>
<td>23</td>
<td>$1.665 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.1</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>$8.319 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.3</td>
<td>Ellipse</td>
<td>0</td>
<td>$9.803 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.3</td>
<td>Rectangle</td>
<td>0</td>
<td>$6.819 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.3</td>
<td>Crescent (Concave)</td>
<td>26</td>
<td>$4.162 \times 10^{-2}$</td>
</tr>
<tr>
<td>0.3</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>$1.372 \times 10^{-1}$</td>
</tr>
<tr>
<td>0.5</td>
<td>Ellipse</td>
<td>0</td>
<td>$2.780 \times 10^{-1}$</td>
</tr>
<tr>
<td>0.5</td>
<td>Rectangle</td>
<td>0</td>
<td>$2.618 \times 10^{-1}$</td>
</tr>
<tr>
<td>0.5</td>
<td>Crescent (Concave)</td>
<td>21</td>
<td>$1.378 \times 10^{-1}$</td>
</tr>
<tr>
<td>0.5</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>$2.966 \times 10^{-1}$</td>
</tr>
</tbody>
</table>
TABLE 17. VALUES FOR THE SUM-SQUARED ERROR CRITERION
FUNCTION OBTAINED BY MATCHING EACH TEMPLATE
TO A CONVEX CRESCENT IMAGE (48 DATA POINTS)

<table>
<thead>
<tr>
<th>σ</th>
<th>Template</th>
<th>Discarded Data Points</th>
<th>φ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>Ellipse</td>
<td>0</td>
<td>4.405×10^{-3}</td>
</tr>
<tr>
<td>0.0</td>
<td>Rectangle</td>
<td>0</td>
<td>2.936×10^{-2}</td>
</tr>
<tr>
<td>0.0</td>
<td>Crescent (Concave)</td>
<td>24</td>
<td>1.205×10^{-12}</td>
</tr>
<tr>
<td>0.0</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>1.977×10^{-12}</td>
</tr>
<tr>
<td>0.1</td>
<td>Ellipse</td>
<td>0</td>
<td>1.355×10^{-2}</td>
</tr>
<tr>
<td>0.1</td>
<td>Rectangle</td>
<td>0</td>
<td>4.622×10^{-2}</td>
</tr>
<tr>
<td>0.1</td>
<td>Crescent (Concave)</td>
<td>24</td>
<td>1.321×10^{-2}</td>
</tr>
<tr>
<td>0.1</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>1.156×10^{-2}</td>
</tr>
<tr>
<td>0.3</td>
<td>Ellipse</td>
<td>0</td>
<td>8.696×10^{-2}</td>
</tr>
<tr>
<td>0.3</td>
<td>Rectangle</td>
<td>0</td>
<td>1.181×10^{-1}</td>
</tr>
<tr>
<td>0.3</td>
<td>Crescent (Concave)</td>
<td>19</td>
<td>5.204×10^{-2}</td>
</tr>
<tr>
<td>0.3</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>9.982×10^{-2}</td>
</tr>
<tr>
<td>0.5</td>
<td>Ellipse</td>
<td>0</td>
<td>1.938×10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Rectangle</td>
<td>0</td>
<td>2.440×10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Crescent (Concave)</td>
<td>20</td>
<td>1.028×10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>2.034×10^{-1}</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Template</td>
<td>Discarded Data Points</td>
<td>$\phi$</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------</td>
<td>-----------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>0.0</td>
<td>Ellipse</td>
<td>0</td>
<td>1.785x10^{-1}</td>
</tr>
<tr>
<td>0.0</td>
<td>Rectangle</td>
<td>0</td>
<td>1.624x10^{-1}</td>
</tr>
<tr>
<td>0.0</td>
<td>Crescent (Concave)</td>
<td>1</td>
<td>9.960x10^{-13}</td>
</tr>
<tr>
<td>0.0</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>1.376x10^{-1}</td>
</tr>
<tr>
<td>0.1</td>
<td>Ellipse</td>
<td>0</td>
<td>1.904x10^{-1}</td>
</tr>
<tr>
<td>0.1</td>
<td>Rectangle</td>
<td>0</td>
<td>1.994x10^{-1}</td>
</tr>
<tr>
<td>0.1</td>
<td>Crescent (Concave)</td>
<td>0</td>
<td>9.845x10^{-3}</td>
</tr>
<tr>
<td>0.1</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>1.392x10^{-1}</td>
</tr>
<tr>
<td>0.3</td>
<td>Ellipse</td>
<td>0</td>
<td>1.957x10^{-1}</td>
</tr>
<tr>
<td>0.3</td>
<td>Rectangle</td>
<td>0</td>
<td>1.699x10^{-1}</td>
</tr>
<tr>
<td>0.3</td>
<td>Crescent (Concave)</td>
<td>6</td>
<td>5.081x10^{-2}</td>
</tr>
<tr>
<td>0.3</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>1.550x10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Ellipse</td>
<td>0</td>
<td>2.389x10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Rectangle</td>
<td>0</td>
<td>3.043x10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Crescent (Concave)</td>
<td>3</td>
<td>1.134x10^{-1}</td>
</tr>
<tr>
<td>0.5</td>
<td>Crescent (Convex)</td>
<td>0</td>
<td>2.109x10^{-1}</td>
</tr>
</tbody>
</table>
APPENDIX A
GENERATION OF DATA POINTS

The data points which are presented to the parameter estimation algorithm are not physically measured points since no equipment was available for this purpose. The entire data acquisition process is instead simulated by a digital computer. The following sections briefly explain how the data points which lie on the boundary of an ellipse, a rectangle, or a crescent are generated, as well as how noisy data points may be generated. The subroutine which generates the data points is denoted by DATA.

Ellipse

The generation of data points lying on the boundary of an ellipse shall be considered first. An ellipse, as shown in Figure 13, may be expressed analytically by Equation A-1.

\[ e_1 w^2 + e_2 z^2 = 1 \]  \hspace{1cm} (A-1)

where

\[ e_1 = \frac{1}{r_x^2} \quad \text{and} \quad e_2 = \frac{1}{r_y^2} \]  \hspace{1cm} (A-2)
Thus, if one is given the two parameters $e_1$ and $e_2$ then the corresponding ellipse in the $w-z$ plane is completely specified. It is desired to represent this ellipse's boundary by some finite number of points. For a given number of data points, say $N$, there are infinitely many different ways in which these points may be positioned on the ellipse's boundary. However, it seems quite unrealistic to have the data points very dense on one portion of the boundary and very sparse, or nonexistent, on the remaining portion of the boundary. Perhaps the most realistic situation is for the data points to be uniformly distributed on the boundary of the ellipse. This would require the distance between any two adjacent data points to be $L/N$, where $L$ is the length of the boundary of the ellipse. This particular distribution of the data points on the boundary of the ellipse was not used, however, because of the complex computer programming which would be involved and because in practice the physical measuring equipment probably would not select the data points in precisely this manner anyway.

The method which was employed to select data points on the boundary of the ellipse consists of dividing the $w$-axis diameter of the ellipse into $N/2$ equal length segments when $N$ data points are desired. This, of course, requires $N$ to be an even number. This procedure is shown
Figure 13. Example of an Ellipse

Figure 14. Data Points Associated with an Ellipse Having No Rotation or Translation

Figure 15. Data Points Associated with a Rotated and Translated Ellipse
in Figure 14 for $N = 10$. The $N$ data points then comprise those points on the boundary of the ellipse whose $w$-coordinates are the same as the $w$-coordinates of the end points of the segments of the $w$-axis diameter.

After the $w,z$-coordinates of the data points which lie on the boundary of the ellipse have been determined, it is necessary to find the coordinates of these same data points with respect to the reference $x,y$-coordinate system. These two coordinate systems are shown in Figure 15.

Once the $x$-and $y$-translation (denoted by $A$ and $B$, respectively) of the center of the ellipse and the rotation (denoted by $\theta$) of the ellipse are specified, then the $x,y$-coordinates of the data points having $w,z$-coordinates $(w_i,z_i)$ are

$$x_i = w_i \cos \theta - z_i \sin \theta + A \quad \text{(A-3)}$$
$$y_i = w_i \sin \theta + z_i \cos \theta + B \quad \text{(A-4)}$$

The $x,y$-coordinates of the data points are then taken as the coordinates of the data points which represent the ellipse whose parameters are now to be estimated.

**Rectangle**

The data points which lie on the boundary of a given rectangle are generated in a slightly different manner than those of an ellipse. Figure 16 shows a rectangle
whose center is at the origin of the $w,z$-coordinate system.

The rectangle's dimension in the $w$-direction is $2r_x$, while its dimension in the $z$-direction is $2r_y$. The dimensions $r_x$ and $r_y$ may be thought of as "radii" of the rectangle.

The data points are selected such that one quarter of the total number of data points lie on each side of the rectangle. This requires $N$ to be divisible by four. The data points are further restricted to be equally spaced along each side. Therefore, the spacing between data points which lie on the vertical boundaries is $8r_y/N$ while the spacing between data points which lie on the horizontal boundaries is $8r_x/N$. Once the $w,z$-coordinates of all the data points lying on the boundary of the rectangle are found, their corresponding $x,y$-coordinates may be determined from Equations (A-3) and (A-4).

Crescent

The data points which lie on the boundary of a crescent are generated in a manner quite similar to those of an ellipse. Figure 17 shows a crescent whose center is at the origin of the $w,z$-coordinate system. The two "sides" of a crescent are simply half circumferences of ellipses. The outer side is a special ellipse, namely, a circle. The data points lying on each side are selected in exactly the
Figure 16. Example of a Rectangle

Figure 17. Example of a Concave Crescent

Figure 18. Example of a Convex Crescent
same way as they are for an ellipse. The outer side is characterized by the radii \( r_x \) and \( r_x \) while the inner side is characterized by the radii \( r_x \) and \( r_y \). Each side of the crescent will possess \( N/2 \) data points.

Since a crescent may, in general, possess a shape which is either concave, as in Figure 17, or convex, as in Figure 18, a parameter is included in the subroutine which generates crescent data points to specify whether the inner side has positive or negative z-coordinates. This parameter is given the name CONVEX.

**Noise**

The preceding discussion has briefly explained how the data points corresponding to either an ellipse, a rectangle, or a crescent are generated, being given the parameters \( e_1, e_2, A, B, \theta \) or \( r_x, r_y, A, B, \theta \). These data points fall exactly on the boundary of the appropriate pattern. Since there is no error in the coordinates of these data points, they may be considered as noiseless data points.

In a realistic system, however, one would expect that the measurement points would not exactly overlay the boundary of the pattern from which they came. This error may be due to several different reasons. For instance, if the field of view has been slightly clouded over, or defocused, then the boundary of the pattern is no longer
precise and the exact coordinates of points lying on the boundary can only be estimated. Even if the field of view is clear there is still the possibility that the electronic equipment associated with the optical system can commit errors, be they internal or transmission errors. Furthermore, there is always the quantization error associated with analog to digital conversion. All of these errors may be considered as forms of noise.

Therefore, in order for the artificially generated data points to realistically correspond to physically measured data points it is necessary to degrade the artificially generated data points by corrupting them with some type of noise. A detailed analysis of the physical system would be required in order to know the exact nature of the actual noise; i.e., its distribution and whether it is additive, multiplicative, or whatever. In this study no particular physical system was considered; therefore, the noise samples were assumed to be additive, statistically independent, gaussian noise samples. How well this artificial noise resembles the actual noise in a physical system was not considered.

The gaussian noise was generated by the subroutine GAUSS which is in the library of the IBM 360/75 at The Ohio State University Computer Center. The subroutine permits one to specify both the mean and the standard
deviation of the gaussian noise samples which it is to generate. The subroutine makes use of another library subroutine called RANDU which generates uniformly distributed random numbers in the range 0-1. The subroutine GAUSS approximates a gaussian random variable by adding together twelve uniform random variables, making use of the Central Limit Theorem.

Since it is assumed that the noise is additive gaussian, the numbers which are generated by GAUSS are simply added independently to each coordinate of the data points. Thus, if the coordinates of a noiseless data point are given by \((X_i, Y_i)\), then the coordinates of the corresponding noisy data point are \((x_i, y_i)\), where

\[
x_i = x_i + n_j \tag{A-5}
\]

\[
y_i = y_i + n_{j+1} \tag{A-6}
\]

where \(n_j\) and \(n_{j+1}\) are two consecutive noise samples.
APPENDIX B

DISCUSSION OF CRITERION FUNCTIONS

It is important to recognize the fact that the criterion function used in the case where the error function is a linear function of a set of parameters (one step minimization method) is not identical (even to within a scale factor) to the criterion function which results when the error function is a nonlinear function of a different set of parameters (iterative minimization scheme).

To be more specific, suppose that the criterion function resulting from the linear error function (Equation 3-53) is denoted by $\phi_L$ and written as

$$\phi_L(x^*, y^*, \hat{p}) = \hat{e}^T \hat{e}$$

$$= \sum_{i=1}^{N} \left( p_1 x_i^2 + p_2 x_i y_i + p_3 y_i^2 + p_4 x_i + p_5 y_i + 1 \right)^2 \quad (B-1)$$

The minimization of $\phi_L$ is taken with respect to the parameter vector $\hat{p} = (p_1, p_2, p_3, p_4, p_5)^T$. If $\hat{p}^*$ is that value of $\hat{p}$ which results in $\phi_L$ attaining its unique minimum value, $\phi_L^*$, then
The ellipse parameter vector \( \mathbf{c} = (e_1, e_2, A, B, \theta)^T \) is then found from Equation (3-42), that is,

\[
\mathbf{c}^* = \frac{1}{p^*} \quad (B-3)
\]

The criterion function resulting from the nonlinear error function, denoted by \( \phi_N \), may be written as

\[
\phi_N(x, y; \mathbf{p}(\mathbf{c})) = e^T \mathbf{e}
\]

\[
= \sum_{i=1}^{N} \left( \rho_1 x_i^2 + \rho_2 x_i y_i + \rho_3 y_i^2 + \rho_4 x_i + \rho_5 y_i + \rho_6 + 1 \right)^2 \quad (B-4)
\]

where the vector \( \mathbf{p} = (\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6)^T \) is a nonlinear function of the ellipse parameter vector \( \mathbf{c} \) with which the minimization of \( \phi_N \) is taken. The vector \( \mathbf{p}(\mathbf{c}) \) is given by Equations (3-10 through 3-15).

Note that \( \phi_N \) may be factored

\[
\phi_N = \sum_{i=1}^{N} (\rho_6 - 1)^2 \left[ \frac{\rho_1}{\rho_6 - 1} x_i^2 + \frac{\rho_2}{\rho_6 - 1} x_i y_i + \frac{\rho_3}{\rho_6 - 1} y_i^2 + \frac{\rho_4}{\rho_6 - 1} x_i \\
+ \frac{\rho_5}{\rho_6 - 1} y_i + 1 \right]^2 \quad (B-5)
\]

\[
\phi_N = (\rho_6 - 1)^2 \phi_L \quad (B-6)
\]
Since \( \hat{p} \) is a function of \( \hat{p} \) (Equation 3-39), the factor \((p_0 - 1)^2\) may be represented by some function, \( g(\hat{p}) \), to give

\[
\phi_N(x, y; \hat{p}) = g(\hat{p}) \phi_L(x, y; \hat{p}) \tag{B-7}
\]

If \( \hat{p}^{**} \) is the value of \( \hat{p} \) for which \( \phi_N \) attains its minimum value, denoted by \( \phi_N^* \), then

\[
\phi_N^*(x, y; \hat{p}^{**}) = \min_{\hat{p}} \phi_N(x, y; \hat{p}) \tag{B-8}
\]

and so

\[
\phi_N^*(x, y; \hat{p}^{**}) \leq g(\hat{p}^*) \phi_L^*(x, y; \hat{p}^*) \tag{B-9}
\]

One should note that if \( \phi_N \) is divided by the factor \((p_0 - 1)^2\) before the minimization with respect to \( \hat{c} \) is taken, then the two criterion functions would be identical and both minimization techniques would yield the same value for the minimizing parameter vector (assuming no boundaries are encountered).

A comparison of \( \phi_L^* \) and \( \phi_N^* \) is made in Table 19. The right hand side of Equation (B-9) is also tabulated, being denoted by \( \phi^* \). The values of the criterion functions are those that were obtained by using the two minimization schemes on an ellipse which is characterized by

\[
\hat{c}_0 = \begin{bmatrix}
0.25 \\
1.00 \\
1.00 \\
-2.00 \\
0.50
\end{bmatrix} \tag{B-10}
\]
The parameter vector estimates which correspond to these values for the criterion functions are shown in Tables 1 and 2 in Chapter IV.

It is interesting to note that in every case $\phi^*_N \leq \phi^*$ in Table 19, as Equation (B-9) implies.
TABLE 19. COMPARISON OF SUM-SQUARED ERROR CRITERION FUNCTIONS ASSOCIATED WITH THE ONE STEP MINIMIZATION METHOD AND THE ITERATIVE MINIMIZATION SCHEME [Reference ellipse parameters are: \( r_x = 2.0, r_y = 1.0, A = 1.0, B = -2.0, \theta = 0.5 \)]

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( N )</th>
<th>( \phi^* )</th>
<th>( \phi )</th>
<th>( \phi^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>10</td>
<td>0.1564x10^-7</td>
<td>0.2496x10^-6</td>
<td>0.6555x10^-1</td>
</tr>
<tr>
<td>0.0</td>
<td>20</td>
<td>0.2379x10^-7</td>
<td>0.3795x10^-6</td>
<td>0.1846x10^-1</td>
</tr>
<tr>
<td>0.0</td>
<td>50</td>
<td>0.6631x10^-6</td>
<td>0.1060x10^-4</td>
<td>0.7957x10^-1</td>
</tr>
<tr>
<td>0.0</td>
<td>100</td>
<td>0.8263x10^-6</td>
<td>0.1319x10^-4</td>
<td>0.4480x10^-1</td>
</tr>
<tr>
<td>0.1</td>
<td>10</td>
<td>0.1757x10^-1</td>
<td>0.3219x10^0</td>
<td>0.2911x10^0</td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
<td>0.1747x10^-1</td>
<td>0.3099x10^0</td>
<td>0.2913x10^0</td>
</tr>
<tr>
<td>0.1</td>
<td>50</td>
<td>0.8430x10^-1</td>
<td>0.1313x10^1</td>
<td>0.1191x10^1</td>
</tr>
<tr>
<td>0.1</td>
<td>100</td>
<td>0.1697x10^0</td>
<td>0.2710x10^1</td>
<td>0.2449x10^1</td>
</tr>
<tr>
<td>0.2</td>
<td>10</td>
<td>0.2264x10^-1</td>
<td>0.3992x10^0</td>
<td>0.3467x10^0</td>
</tr>
<tr>
<td>0.2</td>
<td>20</td>
<td>0.1611x10^0</td>
<td>0.2566x10^1</td>
<td>0.1687x10^1</td>
</tr>
<tr>
<td>0.2</td>
<td>50</td>
<td>0.2627x10^0</td>
<td>0.5528x10^1</td>
<td>0.3989x10^1</td>
</tr>
<tr>
<td>0.2</td>
<td>100</td>
<td>0.5216x10^0</td>
<td>0.1164x10^2</td>
<td>0.8214x10^1</td>
</tr>
<tr>
<td>0.3</td>
<td>10</td>
<td>0.9620x10^-1</td>
<td>0.2193x10^1</td>
<td>0.1232x10^1</td>
</tr>
<tr>
<td>0.3</td>
<td>20</td>
<td>0.3844x10^0</td>
<td>0.6326x10^1</td>
<td>0.2912x10^1</td>
</tr>
<tr>
<td>0.3</td>
<td>50</td>
<td>0.7211x10^0</td>
<td>0.1151x10^2</td>
<td>0.6331x10^1</td>
</tr>
<tr>
<td>0.3</td>
<td>100</td>
<td>0.1881x10^1</td>
<td>0.3245x10^2</td>
<td>0.1480x10^2</td>
</tr>
<tr>
<td>0.4</td>
<td>10</td>
<td>0.2027x10^0</td>
<td>0.9334x10^0</td>
<td>0.5916x10^0</td>
</tr>
<tr>
<td>0.4</td>
<td>20</td>
<td>0.2268x10^0</td>
<td>0.1817x10^2</td>
<td>0.6042x10^1</td>
</tr>
<tr>
<td>0.4</td>
<td>50</td>
<td>0.1097x10^1</td>
<td>0.3412x10^2</td>
<td>0.9126x10^1</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td>0.1544x10^1</td>
<td>0.1190x10^3</td>
<td>0.1520x10^2</td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
<td>0.6232x10^-1</td>
<td>0.7164x10^0</td>
<td>0.5555x10^0</td>
</tr>
<tr>
<td>0.5</td>
<td>20</td>
<td>0.4839x10^0</td>
<td>0.1388x10^2</td>
<td>0.4421x10^1</td>
</tr>
<tr>
<td>0.5</td>
<td>50</td>
<td>0.1257x10^1</td>
<td>0.7153x10^2</td>
<td>0.8891x10^1</td>
</tr>
<tr>
<td>0.5</td>
<td>100</td>
<td>0.3031x10^1</td>
<td>0.1048x10^3</td>
<td>0.2317x10^2</td>
</tr>
</tbody>
</table>
APPENDIX C

PARAMETER ESTIMATION USING FIRST AND SECOND MOMENTS

The computer algorithm which is developed in this dissertation requires that an initial guess for the parameter vector for each template be provided before any iterations can proceed. The technique which is used to supply this initial guess requires that the first and second moments of the data points be computed. This task is performed in the subroutine called ESTIMT. The determination of the initial guess for the two radii parameters and the rotation parameter shall be discussed first.

Initial Guess for Radii Parameters and Rotation Parameter

Once the unknown pattern comes into the field of view of the optical system, the coordinates of the data points which lie on the pattern's boundary are known with respect to the x,y-reference frame. The first step is then to determine the x- and y-centroids \((x_c, y_c)\) of the data points, where

\[ x_c = \frac{1}{N} \sum_{i=1}^{N} x_i \]  

\[ y_c = \frac{1}{N} \sum_{i=1}^{N} y_i \]  

(C-1)
\[ y_c = \frac{1}{N} \sum_{i=1}^{N} y_i \]  

where \((x_i, y_i)\) are the coordinates of the data points and \(N\) is the total number of data points. A new \(u,v\)-reference frame may be defined whose axes are parallel to the \(x,y\)-reference frame and whose origin is at \((x_c, y_c)\). Thus, the \(u,v\)-coordinates of the point \((x_i, y_i)\) are

\[ u_i = x_i - x_c \]  
\[ v_i = y_i - y_c \]

The second moments of inertia of the data points with respect to the \(u,v\)-reference frame (assuming the data points have unity "mass") are

\[ I_{uu} = \frac{1}{N} \sum_{i=1}^{N} u_i^2 \]  
\[ I_{uv} = \frac{1}{N} \sum_{i=1}^{N} u_i v_i \]  
\[ I_{vv} = \frac{1}{N} \sum_{i=1}^{N} v_i^2 \]

and these moments may be considered to be coefficients in the quadratic form

\[ Q(u,v) = I_{uu} u^2 + 2I_{uv}uv + I_{vv}v^2 \]
It is now desirable to find a new $w, z$-reference frame whose origin coincides with the $u, v$-reference frame but which is rotated so that the cross second moment of the data points is zero. The $w, z$-reference frame is related to the $u, v$-reference frame by the transformation

$$
u = w \cos \theta - z \sin \theta$$ (C-9)

$$v = w \sin \theta + z \cos \theta$$ (C-10)

and the corresponding quadratic form is

$$Q(w, z) = I_1 w^2 + I_2 z^2$$ (C-11)

The coefficients $I_1$ and $I_2$, which are called the principal moments of inertia, and the rotation angle $\theta$ of the principal axes $w, z$-reference frame may be determined by considering the matrix representation of Equation (C-8).

$$Q(u, v) = [u \ v] \begin{bmatrix} I_{uu} & I_{uv} \\ I_{uv} & I_{vv} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$ (C-12)

The principal moments of inertia are simply the eigenvalues which are associated with the matrix given in Equation (C-12). These may be found by equating the determinant

$$\begin{vmatrix} I_{uu} - I & I_{uv} \\ I_{uv} & I_{vv} - I \end{vmatrix}$$ (C-13)
to zero to give

\[ I = I_1 = \frac{1}{2} \left[ I_{uu} + I_{vv} + \sqrt{(I_{uu} - I_{vv})^2 + 4 I_{uv}^2} \right] \]  
(C-14)

\[ I = I_2 = \frac{1}{2} \left[ I_{uu} + I_{vv} - \sqrt{(I_{uu} - I_{vv})^2 + 4 I_{uv}^2} \right] \]  
(C-15)

and

\[ \theta = \tan^{-1} \frac{I_1 - I_{uu}}{I_{uv}} \]  
(C-16)

It should be noted that for an ellipse or a rectangle the rotation angle is unique only in the first quadrant, whereas for a crescent the rotation angle is unique in the entire plane. Thus when fitting a crescent template to the data points one must utilize the subroutine SUMSQR to determine whether \( \theta, \theta + \frac{\pi}{2}, \theta + \pi, \text{ or } \theta + \frac{3\pi}{2} \) gives the best fit, where \( \theta \) is obtained from Equation (C-16). The angle giving the best fit is then the initial guess for the rotation angle. Equation (C-16) gives the initial guess for the rotation angle directly for the ellipse and rectangle templates.

After having obtained the principal moments of inertia for the data points, these are compared to four tables which are stored in the computer. Each table contains the principal moments of inertia for one of the templates for various combinations of size and shape (i.e., for various
combinations of $r_x$ and $r_y$). The largest sized template in each table corresponds to the size of the field of view of the optical system. The entry in each table which most closely matches the principal moments of inertia of the data points is then selected. These four entries then represent the initial guess for the two radii for each of the four templates which are to be fitted to the data points.

It should be mentioned that the tables containing the principal moments of inertia for each of the four templates were obtained using noiseless data points generated by the DATA subroutine (48 points for the rectangle template, 50 points for the other templates). When the data points are obtained by some physical scanning mechanism, then it may be necessary to construct new tables. This will be known only after real data becomes available on which to test this algorithm.

Initial Guess for Translation Parameters

Figures 19 and 20 illustrate data points which lie on the boundary of an ellipse and a rectangle, respectively. The $w,z$-reference frame corresponds to the principal axes. It can be seen that if the data points are symmetrically located around the boundaries, then the centers of these two patterns correspond exactly to the $x$- and $y$-centroids of the data points. In general, of course, the boundaries
Figure 19. Principal Axes for an Ellipse

Figure 20. Principal Axes for a Rectangle
will be sampled in a manner such that the data points will not be symmetrically spaced. However, the $x$- and $y$-centroids were found to be a good initial guess for the $x$- and $y$-translations of an ellipse or a rectangle. Thus

\[
A_{\text{guess}} = x_c
\]  \hspace{1cm} (C-17)

\[
B_{\text{guess}} = y_c
\]  \hspace{1cm} (C-18)

Figures 21 and 22 illustrate data points which lie on the boundary of a concave crescent and a convex crescent, respectively. It can be seen that in this case the $x$- and $y$-translation of the patterns does not correspond to the $x$- and $y$-centroids of the data points, which are denoted by $x_c$ and $y_c$, respectively, in Figures 21 and 22. From simple geometry one can see that the $x$- and $y$-translations are given by

\[
A_{\text{guess}} = x_c + h \sin \theta_{\text{guess}}
\]  \hspace{1cm} (C-19)

\[
B_{\text{guess}} = y_c - h \cos \theta_{\text{guess}}
\]  \hspace{1cm} (C-20)

where $\theta_{\text{guess}}$ is the initial guess for the rotation parameter. The length, $h$, is found in the following manner.

The data point having the largest positive $w$-coordinate and the data point having the largest negative $w$-coordinate are determined. The average of the $z$-coordinates of these two data points is then taken as the negative value for $h$. 
Figure 21. Principal Axes for a Concave Crescent

Figure 22. Principal Axes for a Convex Crescent
This procedure gives a very good initial guess for A and B when the data points are distributed over the entire boundary of the crescent (as would be expected) and when the noise is relatively low.
APPENDIX D

REGRESSION MATRIX FOR TEMPLATE MATCHING ALGORITHM

In Chapter III the gradient of the criterion function, \( \nabla \phi \), and the regression matrix, \( S \), were developed using the equation for an ellipse as the response function, and consequently the error function. When using the template matching scheme, however, the response function is simply the x- and y-coordinates of the boundary points of the template (or model), while the error is the difference between the x- and y-coordinates of the data points and the x- and y-coordinates of the corresponding model points.

The response vector, \( \mathbf{F} \), given in Equations (3-60) and (3-61) thus corresponds to

\[
\mathbf{F} = \begin{bmatrix}
  x_1 (x_1, y_1; c) \\
  \vdots \\
  x_m (x_m, y_m; c) \\
  y_1 (x_1, y_1; c) \\
  \vdots \\
  y_N (x_N, y_N; c)
\end{bmatrix}
\]

\[(D-1)\]
The matrix $Z$, given by Equation (3-62), then becomes

$$Z = \left[ \begin{array}{cccc} \frac{\partial x_1 (x_{1d}, y_{1d}; \hat{c})}{\partial c_1} & \frac{\partial x_1 (x_{1d}, y_{1d}; \hat{c})}{\partial c_5} \\ \vdots & \vdots \\ \frac{\partial y_{N_m} (x_{N_{d}}, y_{N_d}; \hat{c})}{\partial c_1} & \frac{\partial y_{N_m} (x_{N_{d}}, y_{N_d}; \hat{c})}{\partial c_5} \\ \end{array} \right]_{c=c_1}$$

which is now a $2N \times 5$ matrix. The linearized criterion function then becomes

$$\phi (\Delta \hat{c}) = (\hat{e} - Z \Delta \hat{c})^T (\hat{e} - Z \Delta \hat{c})$$

where

$$\hat{e} = \left[ \begin{array}{c} x_{1d} - x_{1m} \\ \vdots \\ x_{N_d} - x_{N_m} \\ y_{1d} - y_{1m} \\ \vdots \\ y_{N_d} - y_{N_m} \end{array} \right]$$

Differentiating Equation (D-3) with respect to $\Delta \hat{c}$ and equating the result to zero gives the minimizing value
for $\Delta c^\dagger$,

$$\frac{2\Phi}{\partial \Delta c^\dagger} \bigg|_{\Delta c = \Delta c_1} = 0 = -2Z^T \varepsilon + 2Z^T \Delta c_1^\dagger$$  \hspace{1cm} (D-5)

or

$$\Delta c_1^\dagger = S^{-1}Z^T \varepsilon$$  \hspace{1cm} (D-6)

where the regression matrix, $S$, is given by

$$S = Z^T Z$$  \hspace{1cm} (D-7)

with $Z$ given in Equation (D-2).

The gradient of the criterion function is

$$\nabla \Phi(c_1) = \frac{\partial \Phi}{\partial c^\dagger} \bigg|_{c = c_1} = -2 \left( \frac{\partial c^\dagger}{\partial \varepsilon} \right)^T \varepsilon = -2Z^T \varepsilon$$  \hspace{1cm} (D-8)

Thus, Equation (D-6) becomes

$$\Delta c_1^\dagger = -\frac{1}{2}S^{-1}\nabla \Phi(c_1) = \tilde{\beta}_1$$  \hspace{1cm} (D-9)

As seen by Equations (D-2, 6, 7, 8, and 9), the iterative minimization process may be carried out for the template matching scheme once the error vector and the partial derivatives of the model points have been computed. The error vector for each template, as a function of the five parameters, is developed in Chapters IV, V, and VI. The associated partial derivatives are given below.
Ellipse Template

\[
\frac{\partial x_m}{\partial r_x} = \frac{\partial w_m}{\partial r_x} \cos \theta - \frac{\partial z_m}{\partial r_x} \sin \theta
\]  \hspace{1cm} (D-10)

\[
\frac{\partial x_m}{\partial r_y} = \frac{\partial w_m}{\partial r_y} \cos \theta - \frac{\partial z_m}{\partial r_y} \sin \theta
\]  \hspace{1cm} (D-11)

\[
\frac{\partial x_m}{\partial \theta} = \left[ \frac{\partial w_m}{\partial \theta} - z_m \right] \cos \theta - \left[ \frac{\partial z_m}{\partial \theta} + w_m \right] \sin \theta
\]  \hspace{1cm} (D-12)

\[
\frac{\partial y_m}{\partial r_x} = \frac{\partial w_m}{\partial r_x} \sin \theta + \frac{\partial z_m}{\partial r_x} \cos \theta
\]  \hspace{1cm} (D-13)

\[
\frac{\partial y_m}{\partial r_y} = \frac{\partial w_m}{\partial r_y} \sin \theta + \frac{\partial z_m}{\partial r_y} \cos \theta
\]  \hspace{1cm} (D-14)

\[
\frac{\partial y_m}{\partial \theta} = \frac{\partial w_m}{\partial \theta} \sin \theta + \frac{\partial z_m}{\partial \theta} \cos \theta
\]  \hspace{1cm} (D-15)

\[
\frac{\partial y_m}{\partial A} = \frac{\partial w_m}{\partial A} \sin \theta + \frac{\partial z_m}{\partial A} \cos \theta
\]  \hspace{1cm} (D-16)

\[
\frac{\partial y_m}{\partial B} = \frac{\partial w_m}{\partial B} \sin \theta + \frac{\partial z_m}{\partial B} \cos \theta + 1
\]  \hspace{1cm} (D-17)
\[ \frac{\partial y_m}{\partial \theta} = \left[ \frac{\partial w_m}{\partial \theta} - z_m \right] \sin \theta + \left[ \frac{\partial z_m}{\partial \theta} + w_m \right] \cos \theta \]  
(D-19)

where

\[ \frac{\partial w_m}{\partial x} = \pm \frac{r_y}{\sqrt{r_y^2 + r_x^2} \tan^2 \psi} \frac{r_x r_y \tan^2 \psi}{\left( \sqrt{r_y^2 + r_x^2} \tan^2 \psi \right)^3} \]  
(D-20)

\[ \frac{\partial z_m}{\partial x} = \frac{\partial w_m}{\partial x} \tan \psi \]  
(D-21)

\[ \frac{\partial w_m}{\partial y} = \pm \frac{r_x}{\sqrt{r_y^2 + r_x^2} \tan^2 \psi} \frac{r_x r_y^2}{\left( \sqrt{r_y^2 + r_x^2} \tan^2 \psi \right)^3} \]  
(D-22)

\[ \frac{\partial z_m}{\partial y} = \frac{\partial w_m}{\partial y} \tan \psi \]  
(D-23)

\[ \frac{\partial w_m}{\partial A} = \mp \frac{r_x r_y z_d (w_d \sin \theta + z_d \cos \theta)}{w_d^3 \left( \sqrt{r_y^2 + r_x^2} \tan^2 \psi \right)^3} \]  
(D-24)

\[ \frac{\partial z_m}{\partial A} = \frac{\partial w_m}{\partial A} \tan \psi + \frac{w_m (w_d \sin \theta + z_d \cos \theta)}{w_d^2} \]  
(D-25)

\[ \frac{\partial w_m}{\partial B} = \mp \frac{r_x r_y z_d (-w_d \cos \theta + z_d \sin \theta)}{w_d^3 \left( \sqrt{r_y^2 + r_x^2} \tan^2 \psi \right)^3} \]  
(D-26)
\[
\frac{\partial z_m}{\partial B} = \frac{\partial w_m}{\partial B} \tan \psi + \frac{w_m (-w_d \cos \theta + z_d \sin \theta)}{w_d^2}
\]
(D-27)

\[
\frac{\partial w_m}{\partial \theta} = \pm \frac{r_x^3 r_y z_d (w_d^2 + z_d^2)}{w_d^3 \left( \sqrt{r_x^2 + r_y^2} \tan^2 \psi \right)^3}
\]
(D-28)

\[
\frac{\partial z_m}{\partial \theta} = \frac{\partial w_m}{\partial \theta} \tan \psi - \frac{w_m (w_d^2 + z_d^2)}{w_d^2}
\]
(D-29)

and

\[
\tan \psi = \frac{z_d}{w_d}
\]
(D-30)

The upper sign of the "\(\pm\)" or "\(\mp\)" sign is used when

\[0 \leq \psi \leq \frac{\pi}{2}\] or \[\frac{3\pi}{2} < \psi < 2\pi\], while the lower sign is used when \[\frac{\pi}{2} < \psi < \frac{3\pi}{2}\].

**Rectangle Template**

The corresponding partial derivatives which are associated with a rectangle template are given below.

If the data point \((w_d, z_d)\) lies in Sectors I or III, then

\[
\frac{\partial x_m}{\partial r_m} = \pm \cos \theta \mp \tan \psi \sin \theta
\]
(D-31)

\[
\frac{\partial x_m}{\partial r_y} = 0
\]
(D-32)
\[
\frac{\partial x_m}{\partial A} = \mp r_x \sin \theta \frac{\partial \tan \psi}{\partial A} + 1 \quad (D-33)
\]

\[
\frac{\partial x_m}{\partial B} = \mp r_x \sin \theta \frac{\partial \tan \psi}{\partial B} \quad (D-34)
\]

\[
\frac{\partial x_m}{\partial \theta} = \mp r_x \sin \theta \left[1 + \frac{\partial \tan \psi}{\partial \theta}\right] + \tan \psi \cos \theta \quad (D-35)
\]

\[
\frac{\partial y_m}{\partial r_x} = \mp \sin \theta \pm \tan \psi \cos \theta \quad (D-36)
\]

\[
\frac{\partial y_m}{\partial r_y} = 0 \quad (D-37)
\]

\[
\frac{\partial y_m}{\partial A} = \pm r_y \cos \theta \frac{\partial \tan \psi}{\partial A} \quad (D-38)
\]

\[
\frac{\partial y_m}{\partial B} = \pm r_x \cos \theta \frac{\partial \tan \psi}{\partial B} + 1 \quad (D-39)
\]

\[
\frac{\partial y_m}{\partial \theta} = \pm r_x \cos \theta \left[1 + \frac{\partial \tan \psi}{\partial \theta}\right] - \tan \psi \sin \theta \quad (D-40)
\]

where

\[
\frac{\partial \tan \psi}{\partial A} = \frac{w_d \sin \theta + z_d \cos \theta}{w_d} \quad (D-41)
\]
\[ \frac{\partial x_m}{\partial A} = r_x \sin \theta \frac{\partial \tan \psi}{\partial A} + 1 \]  
(D-33)

\[ \frac{\partial x_m}{\partial B} = r_x \sin \theta \frac{\partial \tan \psi}{\partial B} \]  
(D-34)

\[ \frac{\partial x_m}{\partial \theta} = r_x \sin \theta \left[ 1 + \frac{\partial \tan \psi}{\partial \theta} \right] + \tan \psi \cos \theta \]  
(D-35)

\[ \frac{\partial y_m}{\partial x} = \pm \sin \theta \pm \tan \psi \cos \theta \]  
(D-36)

\[ \frac{\partial y_m}{\partial y} = 0 \]  
(D-37)

\[ \frac{\partial y_m}{\partial A} = r_y \cos \theta \frac{\partial \tan \psi}{\partial A} \]  
(D-38)

\[ \frac{\partial y_m}{\partial B} = r_x \cos \theta \frac{\partial \tan \psi}{\partial B} + 1 \]  
(D-39)

\[ \frac{\partial y_m}{\partial \theta} = \pm r_x \cos \theta \left[ 1 + \frac{\partial \tan \psi}{\partial \theta} \right] - \tan \psi \sin \theta \]  
(D-40)

where

\[ \frac{\partial \tan \psi}{\partial A} = \frac{w_d \sin \theta + z_d \cos \theta}{w_d^2} \]  
(D-41)
\[ \frac{\partial \tan \psi}{\partial B} = \frac{-w_d \cos \theta + z_d \sin \theta}{w_d^2} \]  
\[ \frac{\partial \tan \psi}{\partial \theta} = -\frac{w_d^2 + z_d^2}{w_d^2} \]  

The upper sign of the "±" or "\( \mp \)" sign is used if the data point lies in Sector I, while the lower sign is used if the data point lies in Sector III.

If the data point \((w_d, z_d)\) lies in Sectors II or IV, then

\[ \frac{\partial x_m}{\partial x} = 0 \]  
\[ \frac{\partial x_m}{\partial y} = \pm \cot \psi \cos \theta \mp \sin \theta \]  
\[ \frac{\partial x_m}{\partial A} = \pm r_y \cos \theta \frac{\partial \cot \psi}{\partial A} + 1 \]  
\[ \frac{\partial x_m}{\partial B} = \pm r_y \cos \theta \frac{\partial \cot \psi}{\partial B} \]  
\[ \frac{\partial x_m}{\partial \theta} = \pm r_y \cos \theta \left[ \frac{\partial \cot \psi}{\partial \theta} - 1 \right] - \cot \psi \sin \theta \]
\[ \frac{\partial y_m}{\partial x} = 0 \] (D-49)

\[ \frac{\partial y_m}{\partial y} = \pm \cot \psi \sin \theta \pm \cos \theta \] (D-50)

\[ \frac{\partial y_m}{\partial A} = \pm r_y \sin \theta \frac{\partial \cot \psi}{\partial A} \] (D-51)

\[ \frac{\partial y_m}{\partial B} = \pm r_y \sin \theta \frac{\partial \cot \psi}{\partial B} + 1 \] (D-52)

\[ \frac{\partial y_m}{\partial \theta} = \pm r_y \sin \theta \left[ \frac{\partial \cot \psi}{\partial \theta} - 1 \right] + \cot \psi \cos \theta \] (D-53)

where

\[ \frac{\partial \cot \psi}{\partial A} = - \frac{w_d \sin \theta + z_d \cos \theta}{z_d^2} \] (D-54)

\[ \frac{\partial \cot \psi}{\partial B} = \frac{w_d \cos \theta - z_d \sin \theta}{z_d^2} \] (D-55)

\[ \frac{\partial \cot \psi}{\partial \theta} = \frac{w_d^2 + z_d^2}{z_d^2} \] (D-56)

The upper sign of the "\pm" or "±" sign is used if the data point lies in Sector II, while the lower sign is used if the data point lies in Sector IV.
Crescent Template

The corresponding partial derivatives which are associated with a crescent template are given below. If the data point \( (w_d, z_d) \) is associated with the semiellipse side of the crescent, then Equations (D-10) through (D-30) are used, employing the proper value of \( \psi \). On the other hand, if the data point \( (w_d, z_d) \) is associated with the semicircle side of the crescent, then Equations (D-10) through (D-19) are applicable, with \( 0 \leq \psi \leq \pi \), but Equations (D-20) through (D-30) become

\[
\frac{\partial w_m}{\partial r_x} = \pm \frac{1}{\sqrt{1 + \tan^2 \psi}} \tag{D-57}
\]

\[
\frac{\partial z_m}{\partial r_x} = \frac{\partial w_m}{\partial r_x} \tan \psi \tag{D-58}
\]

\[
\frac{\partial w_m}{\partial r_y} = 0 \tag{D-59}
\]

\[
\frac{\partial z_m}{\partial r_y} = 0 \tag{D-60}
\]

\[
\frac{\partial w_m}{\partial A} = \pm \frac{r_x z_d (w_d \sin \theta + z_d \cos \theta)}{w_d^3 \left(\sqrt{1 + \tan^2 \psi}\right)^3} \tag{D-61}
\]
\[
\frac{\partial z_m}{\partial A} = \frac{\partial w_m}{\partial A} \tan \psi + \frac{w_m (w_d \sin \theta + z_d \cos \theta)}{w_d^2}
\]  
(D-62)

\[
\frac{\partial w_m}{\partial B} = \pm \frac{r x z_d (-w_d \cos \theta + z_d \sin \theta)}{w_d^3 (\sqrt{1 + \tan^2 \psi})^3}
\]  
(D-63)

\[
\frac{\partial z_m}{\partial B} = \frac{\partial w_m}{\partial B} \tan \psi + \frac{w_m (-w_d \cos \theta + z_d \sin \theta)}{w_d^2}
\]  
(D-64)

\[
\frac{\partial w_m}{\partial \theta} = \pm \frac{r x z_d (w_d^2 + z_d^2)}{w_d^3 (\sqrt{1 + \tan^2 \psi})^3}
\]  
(D-65)

\[
\frac{\partial z_m}{\partial \theta} = \frac{\partial w_m}{\partial \theta} \tan \psi - \frac{w_m (w_d^2 + z_d^2)}{w_d^2}
\]  
(D-66)

and

\[
\tan \psi = \frac{z_d}{w_d}
\]  
(D-67)

The upper sign of the "±" or "\(\pm\)" sign is used when

\[0 \leq \psi \leq \frac{\pi}{2}\]

while the lower sign is employed when

\[\frac{\pi}{2} < \psi \leq \pi\]
A listing of the Fortran program which was used to estimate the five parameters associated with an ellipse, a rectangle, or a crescent is given in this appendix. The program has been broken down into a main program along with several subroutines, each of which has a specific function. Each subroutine is briefly discussed in the following paragraphs.

**MAIN Program**

The main program performs three functions. The first function is to read all the required input information for the overall program. Secondly, the main program calls the various subroutines in the correct sequence such that the iterations for the estimates of the parameters are correctly performed. Finally, the main program writes out the input information as well as the best estimate for the parameter vector.

The main program calls four subroutines which are different depending upon what pattern the data is to represent and what the computer generated model (or template) is to represent. These subroutines are DATA,
ESTINT, SUMSQ, and REGRES. The remaining subroutines are the same regardless of the data points and template which are being considered.

The main program requires the following inputs:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPAR</td>
<td>the number of parameters which are to be estimated.</td>
</tr>
<tr>
<td>NPOINT</td>
<td>the number of data points which lie on the boundary of the unknown pattern.</td>
</tr>
<tr>
<td>NTRIAL</td>
<td>the total number of independent local minimizations of the criterion function with respect to the parameter vector. NTRIAL &gt; 1.</td>
</tr>
<tr>
<td>MSMSQ</td>
<td>the maximum number of times which the LOCMIN subroutine may call the SUMSQ subroutine.</td>
</tr>
<tr>
<td>MRAND</td>
<td>the number of independent parameter vectors which are randomly selected by the RANSER subroutine.</td>
</tr>
<tr>
<td>NSET</td>
<td>the number of patterns whose parameters are to be estimated.</td>
</tr>
<tr>
<td>CE</td>
<td>the initial guess for the unknown parameter vector.</td>
</tr>
<tr>
<td>MINPAR</td>
<td>the vector corresponding to the lower limits for the parameters.</td>
</tr>
<tr>
<td>MAXPAR</td>
<td>the vector corresponding to the upper limits for the parameters.</td>
</tr>
<tr>
<td>SIGMA</td>
<td>the standard deviation of the gaussian noise which is added to the simulated data points generated in the DATA subroutine.</td>
</tr>
<tr>
<td>AVE</td>
<td>the average value of the gaussian noise associated with SIGMA.</td>
</tr>
</tbody>
</table>
ESTOP the upper limit for the absolute value of the error. The program is terminated if $|\text{error}|$ becomes larger than ESTOP.

EPHI the lower limit for DPHI. The program is terminated if DPHI becomes smaller than EPHI, where

$$d_\phi = \frac{\phi(c_i) - \phi(c_{i+1})}{\phi(c_i)}$$

EC the lower limit for DC. The program is terminated if DC becomes smaller than EC, where

$$d_c = \frac{\Delta c_i^T \Delta c_i}{\Delta c_i^T c_i}$$

EGRAD the lower limit for the squared magnitude of the gradient vector. The program is terminated if $|\nabla \phi(c_i)|^2$ becomes smaller than EGRAD.

EBDRY a constant which is used in the GRPREX subroutine to prevent division by zero.

**DATA Subroutine**

The purpose of the DATA subroutine is to artificially generate the data points which lie on the boundary of either an ellipse or a rectangle. Although the DATA subroutines corresponding to both an ellipse and a rectangle are shown in the listing, only one of them is included in the program when it is actually used. Appendix A gives more details as to how the data points are generated and how simulated noise is added to them.
**ESTIMT Subroutine**

The ESTIMT subroutine determines the initial guess for the parameter vector. It accomplishes this by computing the first and second moments of the data points. Appendix D discusses in more detail how the initial guess is obtained from these moments.

**SUMSQR Subroutine**

The SUMSQR subroutine simply evaluates the criterion function for a specific value of the parameter vector. It also has an instability indicator, KX, which is set to one if $|\text{error}|$ exceeds ESTOP.

**LOCMIN Subroutine**

The LOCMIN subroutine performs a local minimization of the criterion function with respect to the parameters. It does this by calling the next three subroutines. It also checks the various criteria for terminating the program.

**REGRES Subroutine**

The REGRES subroutine evaluates the criterion function (PHI), the gradient of the criterion function (GRADP), and the Gauss-Newton parameter change vector (BETA) for a specified parameter vector (C) which is supplied by the LOCMIN subroutine. A library subroutine (MINV) is used for matrix inversion.
GRASER Subroutine

The GRASER subroutine is called only when the Newton-Raphson method is used to determine the next value for the parameter vector. The GRASER subroutine finds the optimum binary scale factor by which to multiply $\Delta \mathbf{c}_i$.

GRPREX Subroutine

The GRPREX subroutine is called only when the full Newton-Raphson step ($\Delta \mathbf{c}_i = - \frac{\phi(\mathbf{c}_i) \mathbf{v} \phi(\mathbf{c}_i)}{|\mathbf{v} \phi(\mathbf{c}_i)|^2}$) violates a range constraint. It then projects the gradient onto the constraint surface, after which the GRASER subroutine finds the optimum binary scale factor for this projected gradient. The GRPREX subroutine has an output variable, KEXIT, which when set to one indicates that the parameter vector is on a constraint boundary of the constraint region, $\mathbf{R}$.

RANSER Subroutine

The RANSER subroutine selects a given number ($MRAND$) of parameter vectors randomly, using a uniform distribution, and determines that parameter vector which yields the smallest value for the criterion function. This parameter vector is then used as the initial guess for another local minimization. The RANSER subroutine uses a library subroutine, RANDU, for its uniform number generator.
C THE FOLLOWING FIVE SUBRUTINES ***** RA1N, LOCIN, CRASE, DPHI, DPHK, ***** ARE USED REGARDLESS OF WHICH TEMPLATE IS TO BE STO...
C IN THE COMPUTER.

DIMENSION C0(5), CE(5), CI(5), C(5)
REAL MINPAR(5), MAXPAR(5)
INTEGER CONVERG
COMMON/CO1/ESTOP
COMMON/CON2/NGAUS
COMMON/CO3/NLOC
COMMON/CO4/NK
COMMON/CO5/CONVEX

NGAUS=317757125
C IF THE COMPUTER STORED TEMPLATE IS A CONVEX OR CONCAVE CRESCENT,
C THE NEXT STATEMENT SHOULD BE 'CONVEX=1' OR 'CONVEX=0', RESPECTIVELY
C CONVEX=1
NRIT=0
K=1
NLOC=0
NPAR=5
READ (5,5002) (CO(I),I=1,NPAR), (MINPAR(I),I=1,NPAR), (MAXPAR(I),I=1,NPAR), EGRAD, EBDY
5002 FORMAT (5F16.8)
1 CALL SCL0K1
READ (5,5001) NPAR, NPOINT, NTRIAL, MSMSQ, MRAND, NSET
5001 FORMAT (6I1)
READ (5,5002) SIGMA, AVE, ESTOP, EPHI, EC
WRITE (6,5003)
5003 FORMAT (I1) /*---------------------------------------------*/ , T42, 'NONLINEAR SYSTEM PARAMETER ESTIMATION', /*---------------------------------------------*/
WRITE (6,5004)
5004 FORMAT (T9,'NPAR',T27,'NPOINT',T47,'NTRIAL',T67,'MSMSQ',T87,'MRAND' ,T107,'NRIT',T127,'NLOC',T147,'NSET',T167)
WRITE (6,5005) NPAR, NPOINT, NTRIAL, MSMSQ, MRAND, NSET
5005 FORMAT (I10,5(1X,I10))
WRITE (6,5006)
5006 FORMAT (I10,5(1X,I10))
WRITE (6,5007) SIGMA, AVE, ESTOP, EPHI, EC, EGRAD, EBDY
CALL DATA (CO,SIGMA,AVE,NPOINT)
CALL ESTINT (CE,NPOINT)
WRITE (6,5008)
5008 FORMAT (I10,5(1X,I10))
WRITE (6,5009) (I,CO(I),CE(I), MINPAR(I), MAXPAR(I), I=1,NPAR)
5009 FORMAT (120,T5,X4,C0, T5,X4, T5,X4, T5,X4, T5,X4, T5,X4, T5,X4, T5,X4, T5,X4, T5,X4)
MSMSQ=0
CALC SUMSQ (CE,PHI,XX,MSMSQ,NPOINT,NPAR)
IF (XX) 3,3,2
2 WRITE (6,5010)
5010 FORMAT (I10,5(1X,I10)) THE INITIAL PARAMETER ESTIMATES PRODUCE AN UNSATISFACTORY RESPONSE. DESCENT TO A MINIMUM WILL NOT BE CARRIED OUT.
GO TO 4
3 CALL LOCIN (CE,PHI, MINPAR, MAXPAR, EPHI, EC, EGRAD, EBDY, MSMSQ, NPOINT, NPAR)
NLOC=1
NITNTRIAL-1) R+R,4
4 IY=127456729
DO 7 I=2,NTRIAL

188
CALL RAMSER (C1,MINPAR,MAXPAR,IRANP,1Y,INPOINT,NPAR)
CALL LCHIN (C1,PHILOC,HINPAR,MAXPAR,PHI,EC,GRAD,EBRY,NSMSO,NPD
1 INT,NPAR)
IF(PHI-PHILOC)7,7,5
PHI=PHILOC
DO 6 J=1,NPAR
6 CE(J)=C1(J)
CONTINUE
WRITE (6,5011)
5011 FORMAT ('1'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
IF(KX) 9,9,21
IF(PHIREG-PHI) 10,22,22
DO 11 I=1,NPAR
C1(I)=C(I)
11  DELTAC(I)=RETA(I)
PHI=(PHI-PHIREG)/PHI
GO TO 13
12  DPHI=(PHI-PHIGRA)/PHI
PHI=PHIGRA
13  IF(DPHI-EPHI) 14,14,15
14  WRITE (6,5006) DPHI
5006 FORMAT (' THE NORMALIZED SUM-SQUARED ERROR CHANGE CRITERION IS SATISFIED. DPHI = ',E15.8)
GO TO 32
15  DELCSQ=0.
DO 16 I=1,NPAR
16  DELCSQ=DELCSQ+DELTAC(I)*DELTAC(I)
DC=DELCSQ/C50
IF(DC-EC) 17,17,18
17  WRITE (6,5007) DC
5007 FORMAT (' THE PARAMETER CHANGE ERROR CRITERION IS SATISFIED. DC = ',E15.8)
GO TO 32
18  IF(NSMSQ-MSMSQ) 1,19,19
19  WRITE (6,5008)
5008 FORMAT (' THE LOOP COUNT CRITERION IS SATISFIED. )
GO TO 32
20  CONTINUE
GO TO 23
21  CONTINUE
GO TO 23
22  CONTINUE
DO 24 I=1,NPAR
DELTAC(I)=-(PHI*GRADP(I))/GRADPS
24  C(I)=C(I)+DELTAC(I)
DO 26 I=1,NPAR
IF(C(I)-MINPAR(I)) 30,25,25
25  IF(C(I)-MAXPAR(I)) 26,26,30
26  CONTINUE
BINSCL=2.*NS
DO 28 I=1,NPAR
28  DELTAC(I)=DELTAC(I)/BINSCL
CALL GRASER (PHIINT,C1,NS,DELTAC,KX,NSMSQ,NPOINT,NPAR)
PHIGRA=PHIINT
NT=NS
NS=M0
NO=NT
IF (KX) 29,29,33
29  CONTINUE
GO TO 12
30  CONTINUE
CALL GRPRFX (C1,DELTAC,MINPAR,MAXPAR,GRADP,KEXIT,EBDNY,NPAR)
IF(KEXIT) 27,27,31
31  WRITE (6,5018) PHI
5018 FORMAT (' THE CONSTRAINED MINIMUM IS PHI = ',E15.8, AND THE PARAMETERS ARE '/'
C1(I)'/')
WRITE (6,5019) (I,C1(I),I=1,NPAR)
5019 FORMAT (I10,5X,E15.4)
32  CONTINUE
WRITE (6,5200) PHI
SUBROUTINE "GRASER" (PHI1, CI1, N, DELTAC, KX, NSMSO, NPOINT, NPAR)

DIMENSION CI(5), N, DELTAC(5), DELCMN(5)

DO 1 I = 1, NPAR
  CI(I) = CI1(I) + DELTAC(I)
  CALL SUMSOR (CI, PHI2, KX, NSMSO, NPOINT, NPAR)
  IF (KX) 3, 3, 2

2 CONTINUE
  GO TO 39

3 DO 4 I = 1, NPAR
  DELTAC(I) = DELTAC(I) / 2.
  CI(I) = CI1(I) + DELTAC(I)
  CALL SUMSOR (CI, PHI2, KX, NSMSO, NPOINT, NPAR)
  IF (KX) 6, 6, 5

4 CONTINUE
  GO TO 39

5 IF (PHI1 - PHI2) 9, 9, 7
6 IF (PHI1 - PHI2) 16, 8, 8

8 PHI2 = PHI1

11 N = N + 1
12 IF (N .GT. 20) GO TO 39
  GO TO 3

9 DO 10 I = 1, NPAR
  DELTAC(I) = DELTAC(I) / 2.
  CI(I) = CI1(I) + DELTAC(I)
  CALL SUMSOR (CI, PHI10, KX, NSMSO, NPOINT, NPAR)
  IF (KX) 11, 11, 5
13 IF (PHI1 - PHI10) 13, 13, 12
14 PHI1 = PHI10
  GO TO 9

15 IF (PHI1 - PHI10) 14, 14, 12
16 DO 15 I = 1, NPAR
  DELTAC(I) = DELTAC(I) / 2.
  CI(I) = CI1(I) + DELTAC(I)
  PHI1 = PHI12
  GO TO 32
19 DO 20 I = 1, NPAR
  DELTAC(I) = DELTAC(I) / 2.
  CI(I) = CI1(I) + DELTAC(I)
21 N = N - 1
22 PHI1 = PHI1
  PHI1 = PHI12
  CALL SUMSOR (CI, PHI12, KX, NSMSO, NPOINT, NPAR)
  IF (KX) 23, 23, 22
24 CONTINUE
  GO TO 39
23 IF(PH11-PH12) 30,24,24
24 IF(N) 25,25,28
25 PH1=PH12
26 DO 27 I=1,NPAR
27 C1(I)=C1(I)+DELTAC(I)
28 DO 29 I=1,NPAR
29 DELTAC(I)=2.*DELTAC(I)
30 CONTINUE
31 DELTAC(I)=DELTAC(I)/2.*
N=N+1
32 IF(N.GT.20) GO TO 39
33 QUADN=(3./4.)*(PH12-5.*PH11+4.*PHI0)/(PH12-3.*PH11+2.*PHI0)
33 DO 33 I=1,NPAR
34 C1(I)=C1(I)+QUADN*DELTAC(I)
35 CALL SUMSOR (C,PHI1M,KX,NSISO,NPOINT,NPAR)
36 IF(KX) 34,34,5
37 IF(PHI1M-PHI1) 35,37,37
38 DO 36 I=1,NPAR
39 C1(I)=C1(I)+DELTAC(I)
40 PHI=PHI1M
41 DO 38 I=1,NPAR
42 C1(I)=C1(I)+DELTAC(I)
43 CONTINUE
44 RETURN
45 END

SUBROUTINE GRPREX (C,DELTAC,HIPAR,MXPAR,GRADP,KEXIT,EDRY,NPAR)
DIMENSION C(51),DELTAC(51),GRADP(51),CINDEX(51)
REAL KO,KSTEP,HIPAR(51),MXPAR(51)
INTEGER COMND,CINDEX
CINDEX=0
K=0.
KEXIT=0.
DO 5 I=1,NPAR
COMND(I)=0
IF(C(I)-HIPAR(I)-EDRY) 1,1,3
1 IF(GrandP(I)) 4,2,2
2 GRADP(I)=0.
CINDEX=CINDEX+1
COMND(I)=1
GO TO 5
3 IF(C(I)-MXPAR(I)+EDRY) 5,6,4
4 IF(GrandP(I)) 7,7,5
5 CONTINUE
6 IF(CINDEX-NPAR) 8,6,6
7 CONTINUE
KEXIT=1
DO 7 I=1,NPAR
8 DELTAC(I)=0.
GO TO 17
9 DO 15 I=1,NPAR
10 IF(COMND(I)) 9,9,15
11 IF(GrandP(I)) 10,15,11
12 KSTEP=(C(I)-MXPAR(I)+GRADP(I))
11 KSTEP = (C(1) - MINPAR(1)) / GRADP(1)
12 IF(KO) 14, 14, 13
13 IF(KSTEP - KO) 14, 15, 15
14 KO = KSTEP
15 CONTINUE
DO 16 I = 1, NPAR
16 DELTAC(I) = -KO * GRADP(1)
17 CONTINUE
RETURN
END

SUBROUTINE RANSER (C1, MINPAR, MAXPAR, MRAND, IY, NPOINT, NPAR)
DIMENSION C1(5), C(5), D(5)
REAL MINPAR(5), MAXPAR(5)
WRITE (6, 5001)
5001 FORMAT ('ENTER RANSER SUBROUTINE.')
NSMSQ = 0
DO 1 J = 1, NPAR
1 D(1) = MAXPAR(1) - MINPAR(1)
IX = IY
2 I = 1
CONTINUE
CALL RANDU (IX, IY, YFL)
IX = IY
C(I) = MINPAR(I) + D(I) * YFL
I = I + 1
IF (I - NPAR) 3, 3, 4
3 CALL SMSOR (C1, PHI, KX, NSMSQ, NPOINT, NPAR)
IF (KX) 5, 5, 2
WRITE (6, 5002) NSMSQ
5002 FORMAT (' RANDOM SEARCHING HAS ESTABLISHED A STARTING VALUE FOR PH
21. NSMSQ = '13/1
DO 11 I = 2, MRAND
11 J = 1
CONTINUE
CALL RANDU (IX, IY, YFL)
IX = IY
C(J) = MINPAR(J) + D(J) * YFL
J = J + 1
IF (J - NPAR) 6, 6, 7
CALL SMSOR (C, PHI, KX, NSMSQ, NPOINT, NPAR)
IF (KX) 8, 8, 11
8 IF (PHI - PHIRAN) 11, 11, 9
9 PHI = PHIRAN
DO 10 J = 1, NPAR
10 C(J) = C(J)
11 CONTINUE
WRITE (6, 5003) PHI
5003 FORMAT (' THE SMALLEST VALUE FOUND BY THE RANSER SUBROUTINE FOR T
1HE SUM-SQUARED ERROR IS PHI = 'E15.8\'/')
WRITE (6, 5004)
5004 FORMAT (' MINIMIZING PARAMETER')
1 VALUES'/'' C(1) C(2) C(3)
2 C(4) C(5)/''
WRITE (6, 5005) C(1), I = 1, NPAR
5005 FORMAT (5X, 'F10.5', 4(10X, 'F10.5'))
WRITE (6, 5006)
5006 FORMAT (' EXIT RANSER SUBROUTINE.'
RETURN
END
THE FOLLOWING FOUR SUBROUTINES \ldots \quad \text{TDATA, ESTINT, SUMSIX, RFGRES \ldots}
\ldots \text{ARE USED WHEN AN ELLIPSE TEMPLATE IS TO BE STORED IN THE}
\ldots \text{COMPUTER.}

SUBROUTINE DATA (C, SIGMA, AVE, NPOINT)
DIMENSION XD(100), YD(100), X(100), YD(100), C(5)
COMMON/COM2/XDATA(100), YDATA(100)
COMMON/COM3/NGAUS
DATA IZS/1 /
N= NPOINT-1
RX= C(1)
RY= C(2)
A= C(3)
B= C(4)
CTheta=C(5)
CTheta=C(5)
STheta=SIN(THETA)
XINT=2*RX
YD(I)= RX
YO(I)=0
I=2
1 XNO(I)= RX+1*XINT/NPOINT
XO(I)+1=XO(I)
YU(I)= RY*SORT(1.0-(XD(I)+0.2/RX)*2))
Y0(I)+1= YO(I)
I=I+2
IF (I .LT. NPOINT) GO TO 1
XO(I)=RX
YO(I)=0
DO 2 I=1,NPOINT
X(I)=XO(I)+STH*YD(I)*CTH+A
2 YD(I)= XD(I)+STH+YO(I)*STH+B
DO 3 I=1,NPOINT
CALL GAUSS (NGAUS, SIGMA, AVE, V)
XDATA(I)=XD(I)+V
CALL GAUSS (NGAUS, SIGMA, AVE, V)
3 YDATA(I)= YD(I)+V
WRITE (6,5001) NPOINT, RX, RY, A, B, THETA, AVE, SIGMA
5001 FORMAT (I1, THE FOLLOWING, I4, \ldots \text{THE POINTS LIE ON THE BOUNDARY OF AN ELLIPSE WHICH IS CHARACTERIZED BY THE PARAMETERS \ldots /5X, RX = }, F5.2, 22, 3X, RY = }, F5.2, 3X, 'A = ', F5.2, 3X, 'B = ', F5.2, 3X, 'THETA = ', F5.2, 3X, 'SOME NOISE IS CHARACTERIZED BY }\ldots /5X, 'MEAN = ', F5.2, 3X, 'SIGMA = ', F5.2, 3X, 'V
WRITE (6,5002) (IZS, XDATA(I), YDATA(I), I=1,NPOINT)
5002 FORMAT (4(A1, '!E12.5', '!E12.5', '!A1))
RETURN
END

SUBROUTINE ESTINT(C,NPOINT)
REAL IXX, IYY, IXY, I1, I2
DIMENSION C(5, U(100), V(100), W(100), Z(100)
COMMON/COM2/XDATA(100), YDATA(100)
P1=3.1415926536
PIHALF=P1/2.0
XMEAN=0.0
YMEAN=0.0
DO 1 I=1,NPOINT
XMEAN = XMEAN + XDATA(I) / NPOINT
YMEAN = YMEAN + YDATA(I) / NPOINT
XMEAN = XMEAN / NPOINT
YMEAN = YMEAN / NPOINT
DO 2 I = 1, NPOINT
U(I) = XDATA(I) - XMEAN
V(I) = YDATA(I) - YMEAN
IXX = 0.0
IYX = 0.0
IIY = 0.0
DO 3 I = 1, NPOINT
IXX = IXX + U(I)**2
IYX = IYX + V(I)**2
IXY = IXX / NPOINT
IYX = IYX / NPOINT
II = SQRT((IXX - IYX)**2 + 4*IXY*II)
I2 = (IXX + IYX - II)**2 / 2
RATIO = I2 / II
THETA = ATAN2(II - IXX, IXY)
IF (THETA < PI) THEN THETA = THETA + PI
IF (THETA > PI/2) GO TO 4
THETA = PI - THETA
RATIO = RATIO / R
ITEMP = I1
I2 = ITEMP
4 IF (I1 .LT. 0.0335) THEN RX = 0.2
IF (I1 .GE. 0.0335 .AND. I1 .LT. 0.0868) THEN RX = 0.4
IF (I1 .GE. 0.0868 .AND. I1 .LT. 0.167) THEN RX = 0.6
IF (I1 .GE. 0.167 .AND. I1 .LT. 0.274) THEN RX = 0.8
IF (I1 .GE. 0.274 .AND. I1 .LT. 0.408) THEN RX = 1.0
IF (I1 .GE. 0.408 .AND. I1 .LT. 0.569) THEN RX = 1.2
IF (I1 .GE. 0.569 .AND. I1 .LT. 0.756) THEN RX = 1.4
IF (I1 .GE. 0.756 .AND. I1 .LT. 0.970) THEN RX = 1.6
IF (I1 .GE. 0.970 .AND. I1 .LT. 1.222) THEN RX = 1.8
IF (I1 .GE. 1.222 .AND. I1 .LT. 1.490) THEN RX = 2.0
IF (I1 .GE. 1.490 .AND. I1 .LT. 1.773) THEN RX = 2.2
IF (I1 .GE. 1.773 .AND. I1 .LT. 2.094) THEN RX = 2.4
IF (I1 .GE. 2.094 .AND. I1 .LT. 2.443) THEN RX = 2.6
IF (I1 .GE. 2.443 .AND. I1 .LT. 2.816) THEN RX = 2.8
IF (I1 .GE. 2.816 .AND. I1 .LT. 3.217) THEN RX = 3.0
IF (I1 .GE. 3.217 .AND. I1 .LT. 3.665) THEN RX = 3.2
IF (I1 .GE. 3.665 .AND. I1 .LT. 4.110) THEN RX = 3.4
IF (I1 .GE. 4.110 .AND. I1 .LT. 4.582) THEN RX = 3.6
IF (I1 .GE. 4.582 .AND. I1 .LT. 5.090) THEN RX = 3.8
IF (I1 .GE. 5.090) THEN RX = 4.0
IF (RATIO .LT. 0.0.0491 .AND. RX = 0.1*RX
IF (RATIO .GE. 0.0.0491 .AND. RATIO .LT. 0.129) THEN RX = 0.2*RX
IF (RATIO .GE. 0.129 .AND. RATIO .LT. 0.249) THEN RX = 0.3*RX
IF (RATIO .GE. 0.249 .AND. RATIO .LT. 0.408) THEN RX = 0.4*RX
IF (RATIO .GE. 0.408 .AND. RATIO .LT. 0.608) THEN RX = 0.5*RX
IF (RATIO .GE. 0.608 .AND. RATIO .LT. 0.796) THEN RX = 0.6*RX
IF (RATIO .GE. 0.796 .AND. RATIO .LT. 1.125) THEN RX = 0.7*RX
IF (RATIO .GE. 1.125 .AND. RATIO .LT. 1.443) THEN RX = 0.8*RX
IF (RATIO .GE. 1.443 .AND. RATIO .LT. 1.801) THEN RX = 0.9*RX
IF (RATIO .GE. 1.801) THEN RX = RX
C(1) = RX
C(2) = RY
C(3) = XMEAN
C(4) = YMEAN
SUBROUTINE SUMSOR(C,PHI,KX,NSMSO,POINT,NPAR)

DIMENSION C(5),E(200),XMODEL(100),YMODEL(100)

COMMON/CWH/ESTOP
COMMON/COW2/XDATA(100),YDATA(100)

PI=3.1415926536

PIHALF=PI/2.

PI32=3.0/PI/2.

PI2=2.0/PI

NSMSO=NSMSQ+1

KX=0

PHI=0.0

RX=C(1)

RY=C(2)

A=C(3)

B=C(4)

THETA=C(5)

CTH=COS(THETA)

STH=SIN(THETA)

DO 4 I=1,POINT

XD=XDATA(I)

YD=YDATA(I)

WD=(XD-A)*CTH+(YD-B)*STH

ZD=(-(XD-A)*STH+(YD-B)*CTH

IF ((WD.*EQ.0.0.,AND. ZD.*EQ.0.0)) GO TO 1

IF ((WD.*EQ.0.0.,AND. ZD.*LT.0.0)) GO TO 2

TPSI=ZD/WD

WM=RX*RY/SQRT(RY**2+(RX*TPS1)**2)

PSI=ATAN2(IZD,WM)

IF (PSI.GT.PIHALF.) AND. PSI.LT.PI32) WM=-WM

ZM=WM*TPS1

GO TO 3

1 WM=0.0

ZM=RY

GO TO 3

2 WM=0.0

ZM=-KY

3 XMODEL(I)=WM*CTH-ZM*STH+A

YMODEL(I)=WM*STH+ZM*CTH+B

E(I)=XDATA(I)-XMODEL(I)

4 E1=POINT=YDATA(I)-YMODEL(I)

NPT2=2*POINT

DO 5 I=1,NPT2

IF (ABS(E(I)).LT.ESTOP) GO TO 5

KX=1

WRITE (6,5002) (C(J),J=1,NPAR)

5002 FORMAT ('THE SYSTEM IS UNSTABLE FOR THESE PARAMETER VALUES. THE SUM-

SQUARED ERROR WILL NOT BE EVALUATED. THE PARAMETERS ARE'/'

1 = 'E15.'8,' C(2) = 'E15.'8,' C(3) = 'E15.'8,' C(4) = 'E15.'8,' C(5) =

2 'E15.'8)

GO TO 6

5 PHI=PHI+E(I).*2

PHI=PHI/NPOINT

6 CONTINUE

RETURN

END
SUBROUTINE REGRES(C,GRADP,RETA,PHI,NPOINT,NPAR)
DIMENSION C(10),GRADP(10),RETA(10),E(200),Z(200,5),S(5,5),L(5),M(5)
MMODEL(100),YMODEL(100),DERX4(100,5),DERY4(100,5)
COMMON/CIM2/XDATA(100),YDATA(100)
PI=3.1415926536
PIHALF=PI/2.
P132=3.*PI/2.
P12=2.*PI
RX=C(1)
RY=C(2)
A=C(3)
B=C(4)
THETA=C(5)
CTH=COS(THETA)
STH=SIN(THETA)
DO 6 I=NPOINT
X0=XDATA(I)
Y0=YDATA(I)
HD=(XD-A)*CTH+(YD-B)*STH
ZD=-(XD-A)*STH+(YD-B)*CTH
IF (HD .GE. 0.0 .AND. ZD .GE. 0.0) GO TO 2
IF (HD .GE. 0.0 .AND. ZD .LT. 0.0) GO TO 3
PSI=ATAN2(ZD,HD)
IF (PSI .LT. 0.0) PSI=PI+PSI
TPSI=ZD/HD
WM=RX*RY/SORT(RY*^2+(RX*TPSI)**2)
F1=SORT(RY*^2+(RX*TPSI)**2)
F2=1.0/F1
F3=(ND*STH+ZD*CTH)/(WD*2)
F4=(-WD*CTH+ZD*STH)/(WD*2)
F5=(WD*2+ZD*2)/(WD*2)
DNMDRX=RX/F1-1.0*(RX*TPSI)**2/F2
DNMDRY=RY/F1-1.0*(RY*TPSI)**2/F2
DNY=1.0/F1
DNMDRX=DNMDRX
DNMDRY=DNMDRY
DNY=DNY
DNMDTH=DNMDTH
1
ZM=WM*TPSI
NZMDRX=DNMDRX*TPSI
OZMDRY=DNMDRY*TPSI
NZMDA=DNMDA*TPSI+WM*F3
OZMDN=DNMDN*TPSI+WM*F4
OZMDTH=DNMDTH*TPSI-4.*F5
GO TO 5
7
WM=0.0
ZM=RY
NZMDRY=1.0
GO TO 4
3
WM=0.0
ZM=-RY
NZMDRY=-1.0
4
DNMDRX=0.0
DNMDRY=0.0
DNMDA=0.0
DNMDN=0.0
THE FOLLOWING FOUR SUBROUTINES .... DATA, ESTIMT, SUMSOR, REGRES ....
ARE USED WHEN A RECTANGLE TEMPLATE IS TO BE STORED IN THE COMPUTER.

SUBROUTINE DATA(C, SIGMA, AVE, NPOINT)
DIMENSION C(5), X(100), Y(100), X(100), Y(T100), X(100), Y(100), X(100), Y(100)
COMMON/C102/XDATA(100), YDATA(100)
COMMON/C103/MGAHS
DATA IZS/1/
XMAX=C(1)
YMAX=C(2)
RX=XMAX
RY=YMAX
A=C(3)
r = c(4)
theta = c(5)
cth = cos(theta)
sinth = sin(theta)
ymin = -yax
xmin = -xax
np = npoint/4
xint = xax - xmin
yint = yax - ymin
n = np + 1

1  y1(i) = ymin + (i-1) * yint/np
2  y2(i) = yax
3  y3(i) = yax - (i-1) * xint/np
4  y4(i) = ymin

5  yo(i) = y(i)
n = np + 2
nn = 2 * np + 1

6  xo(i) = x0(i) = x2(i) - np

7  xo(i) = x3(i) - 2 * np

8  xo(i) = x4(i) - 3 * np

9  yo(i) = xo(i) * cth - yo(i) * sth + a
10  yo(i) = xo(i) * sth + yo(i) * cth + b
do 10 i = 1, npoint

call gauss(ngaus, sigma, ave, v)
ydata(i) = ydata(i) + v

call gauss(ngaus, sigma, ave, v)

write (6, 5001) npoint, rx, ry, a, b, theta, ave, sigma
5001 format ("the following", i4, " points lie on the boundary of a rectangle which is characterized by the parameters", 5f5.2, " rx =", f5.2, "ry =", f5.2, "theta =", f5.2, "a =", f5.2, "b =", f5.2, "sigma =", f5.2)
write (6, 5002) (izs, xdata(i) , ydata(i) , i = 1, ns), izs, xdata(npoint), ydata(npoint)
5002 format (4(a1, " (", f12.5, ",", f12.5, ")")
end
SUBROUTINE ESTINT(C,NPOINT)

REAL IXX, IYY, IXY, XI, X2
DIMENSION C(5), H(100), V(100), Y(100), Z(100)
COMMON/C wx2/XDATA(100), YDATA(100)
PI=3.1415926536
PIHALF=PI/2.0
PI2=3.0*pi/2.0
PI2=2.0*pi
XMEAN=0.0
YMEAN=0.0
DO 1 I=1,NPOINT
XMEAN=XMEAN+XDATA(I)
YMEAN=YMEAN+YDATA(I)
1 XMEAN=XMEAN/NPOINT
YMEAN=YMEAN/NPOINT
DO 2 I=1,NPOINT
2 U(I)=XDATA(I)-XMEAN
V(I)=YDATA(I)-YMEAN
IXX=0.0
IXY=0.0
IYY=0.0
DO 3 I=1,NPOINT
IXX=IXX+U(I)**2
IXY=IXY+U(I)*V(I)
IYY=IYY+V(I)**2
3 IX=IXX+IYY
12=IXX+IYY
RATIO(12)=12/I
THETA=ATAN2((12-IXX,IXY)
IF (THETA.*LT.0.0) THETA=THETA+PI2
IF (THETA.*GT. PIHALF .AND. THETA.*LE. PI) THETA=THETA-PIHALF
IF (THETA.*GT. PI .AND. THETA.*LE. PI32) THETA=THETA-PI2
IF (THETA.*GT. PI32) THETA=THETA-PI_32
IF (THETA.*GT. PI32) THETA=THETA-PI_32
IF (THETA.*LT.0.0669) RX=0.2
IF (THETA.*GE. 0.0669 .AND. I1 .LT. 0.174) RX=0.4
IF (THETA.*GE. 0.174 .AND. I1 .LT. 0.335) RX=0.6
IF (THETA.*GE. 0.335 .AND. I1 .LT. 0.549) RX=0.8
IF (THETA.*GE. 0.549 .AND. I1 .LT. 0.846) RX=1.0
IF (THETA.*GE. 0.816 .AND. I1 .LT. 1.137) RX=1.2
IF (THETA.*GE. 1.137 .AND. I1 .LT. 1.51) RX=1.4
IF (THETA.*GE. 1.51 .AND. I1 .LT. 1.94) RX=1.6
IF (THETA.*GE. 1.94 .AND. I1 .LT. 2.43) RX=1.8
IF (THETA.*GE. 2.43 .AND. I1 .LT. 2.96) RX=2.0
IF (THETA.*GE. 2.96 .AND. I1 .LT. 3.55) RX=2.2
IF (THETA.*GE. 3.55 .AND. I1 .LT. 4.19) RX=2.4
IF (THETA.*GE. 4.19 .AND. I1 .LT. 4.88) RX=2.6
IF (THETA.*GE. 4.88 .AND. I1 .LT. 5.63) RX=2.8
IF (THETA.*GE. 5.63 .AND. I1 .LT. 6.44) RX=3.0
IF (THETA.*GE. 6.44 .AND. I1 .LT. 7.29) RX=3.2
IF (THETA.*GE. 7.29 .AND. I1 .LT. 8.20) RX=3.4
IF (THETA.*GE. 8.20 .AND. I1 .LT. 9.17) RX=3.6
IF (THETA.*GE. 9.17 .AND. I1 .LT. 10.18) RX=3.8
IF (THETA.*GE. 10.18) RX=4.0
IF (RATIO.*LT.0.0225) RY=0.1*RX
IF (RATIO.*GE. 0.0225 .AND. RATIO.*LT. 0.0625) RY=0.2*RX
IF (RATIO.*GE. 0.0625 .AND. RATIO.*LT. 0.1225) RY=0.3*RX
IF (RATIO.*GE. 0.1225 .AND. RATIO.*LT. 0.2025) RY=0.4*RX
IF (RATIO.*GE. 0.2025 .AND. RATIO.*LT. 0.3025) RY=0.5*RX
IF (RATIO.*GE. 0.3025 .AND. RATIO.*LT. 0.4225) RY=0.6*RX

IF (RATIO . GE . 0.4225 . AND . RATIO . LT . 0.5625) RY = 0.7*RX
IF (RATIO . GE . 0.5625 . AND . RATIO . LT . 0.7225) RY = 0.8*RX
IF (RATIO . GE . 0.7225 . AND . RATIO . LT . 0.9025) RY = 0.9*RX
IF (RATIO . GE . 0.9025) RY = RX

C(1) = RX
C(2) = RY
C(3) = XMEAN
C(4) = YMEAN
C(5) = THETA
RETURN
END

SUBROUTINE SUMSOR(C, PHI, RX, NSHSO, NPOINT, NPAR)
DIMENSION C(5), F(200), XMODEL(100), YMODEL(100)
COMMON/COM1/ESTOP
COMMON/COM2/XDATA(100), YDATA(100)
NSHSO = NSHSO + 1
RX = C(1)
RY = C(2)
A = C(3)
B = C(4)
THETA = C(5)
CTH = COS(THETA)
STH = SIN(THETA)

ANGLE1 = ATAN2(-RY, RX) + PI2
ANGLE2 = ATAN2(RY, RX)
ANGLE3 = ATAN2(RY, -RX)
ANGLE4 = ATAN2(-RY, -RX) + PI2

DO 10 I = 1, NPOINT
XD = XDATA(I)
YD = YDATA(I)
WD = (XD - A) * CTH + (YD - A) * STH
ZD = -(XD - A) * STH + (YD - A) * CTH
IF (ZD . LT. 0.0 . AND . WD . GE . 0.0) GO TO 1
IF (WD . LT. 0.0 . AND . ZD . GE . 0.0) GO TO 3
IF (ZD . LT. 0.0 . AND . WD . LE . 0.0) GO TO 5
IF (WD . LT. 0.0 . AND . ZD . LE . 0.0) GO TO 7

TPS1 = ZD/WD
CPS1 = 1.0/TPS1
PSI = ATAN2(ZD, WD)
IF (PSI . LT. 0.0) PSI = PSI + PI2
IF (PSI . LE. ANGLE2 . OR . PSI . GE. ANGLE1) GO TO 2
IF (PSI . GT. ANGLE2 . AND . PSI . LT. ANGLE3) GO TO 4
IF (PSI . GE. ANGLE3 . AND . PSI . LE. ANGLE4) GO TO 6
IF (PSI . GT. ANGLE4 . AND . PSI . LT. ANGLE1) GO TO 8
1
TPS1 = 0.0
XMODEL(I) = RX*(CTH - TPS1*STH) + A
YMODEL(I) = RX*(STH + TPS1*CTH) + B
GO TO 9

2
CPS1 = 0.0
XMODEL(I) = RY*(CPSI*CTH - STH) + A
YMODEL(I) = RY*(CPSI*STH + CTH) + B
GO TO 9

3
CPS1 = 0.0
XMODEL(I) = RY*(CPSI*CTH - STH) + A
YMODEL(I) = RY*(CPSI*STH + CTH) + B
GO TO 9

5
TPS1 = 0.0
XMODEL(I) = RX*(CTH - TPS1*STH) + A
YMODEL(I) = RX*(STH + TPS1*CTH) + B
GO TO 9

6
CPS1 = 0.0
XMODEL(I) = RX*(CTH - TPS1*STH) + A
YMODEL(I) = RX*(STH + TPS1*CTH) + B
GO TO 9

7
CPS1 = 0.0
SUBROUTINE REGRFS(C,GRADP,BETA,PHI,NPOINT,NPAR)

DIMENSION C(5),GRADP(5),BETA(5),E(200),Z(200,5),S(5,5),L(5),N(5),

XMODEL(100),YMODEL(100),DEXY(100,5),DERXY(100,5)

COMMON/CUM2/XDATA(100),YDATA(100)

PI=6.20319

RX=C(1)

RY=C(2)

A=C(3)

B=C(4)

THETA=C(5)

CTH=COS(THETA)

STH=SIN(THETA)

ANGLE1=ATAN2(-RY,RX)+PI2

ANGLE2=ATAN2(RY,RX)

ANGLE3=ATAN2(RY,-RX)

ANGLE4=ATAN2(-RY,-RX)+PI2

DO 10 I=1,NPOINT

X=XMODEL(I)

Y=YDATA(I)

WD=(X-A)*CTH+(Y-B)*STH

ZD=(X-A)*STH+(Y-B)*CTH

IF (ZD .EQ. 0.0 .AND. WD .GE. 0.0) GO TO 1

IF (WD .EQ. 0.0 .AND. ZD .GT. 0.0) GO TO 3

IF (ZD .EQ. 0.0 .AND. WD .LT. 0.0) GO TO 5

IF (WD .EQ. 0.0 .AND. ZD .LT. 0.0) GO TO 7

TPS1=ZD/WD

CPS1=1.0/TPS1

DTPDA=(WD*STH+ZD*CTH)/(WD**2)

DTPDB=(MD*CTH+ZD*STH)/(WD**2)

DTPTH=-(WD**2+ZD**2)/(WD**2)

DCPDA=(ZD*CTH+WD*STH)/(ZD**2)

DCPDB=(ZD*STH+WD*CTH)/(ZD**2)

DCPTH=-(WD**2+ZD**2)/(ZD**2)

PSI=ATAN2(ZD,WD)

IF (PSI .LT. 0.0) PSI=PSI+PI2

IF (PSI .LT. ANGLE2 .OR. PSI .GE. ANGLE1) GO TO 2

IF (PSI .LT. ANGLE2 .AND. PSI .LT. ANGLE3) GO TO 4

IF (PSI .LT. ANGLE4 .AND. PSI .LT. ANGLE1) GO TO 6

IF (PSI .LT. ANGLE4 .AND. PSI .LT. ANGLE3) GO TO 8

TPS1=0.0

DTPDA=0.0
XMODEL(I) = RX*(CTH+TPSI*STH) + A
YMODEL(I) = RX*(STH+TPSI*CTH) + B
DERXM(I,1) = CTH-TPSI*STH
DERXM(I,2) = 0.0
DERXM(I,3) = -RX*STH*DTPDA+1.0
DERXM(I,4) = -RX*STH*DTPDH
DERXM(I,5) = -RX*((DTPDTH+1.0)*STH+TPSI*CTH)
DERYN(I,1) = STH+TPSI*CTH
DERYN(I,2) = 0.0
DERYN(I,3) = RX*CTH*DTPDA
DERYN(I,4) = RX*CTH*DTPDH+1.0
DERYN(I,5) = RX*((DTPDTH+1.0)*CTH-TPSI*STH)

GO TO 9

CPSI = 0.0
DCPDA = 0.0
DCPDB = 0.0

DCPOTH = 0.0

XMODEL(I) = RX*(CPSI*CTH-STH) + A
YMODEL(I) = RX*(CPSI*STH*CTH) + B
DERXM(I,1) = 0.0
DERXM(I,2) = CPSI*CTH-STH
DERXM(I,3) = CY*CTH*DCPDA+1.0
DERXM(I,4) = CY*CTH*DCPDB
DERXM(I,5) = CY*((DCPOTH-1.0)*CTH-CPSI*STH)
DERYN(I,1) = 0.0
DERYN(I,2) = CPSI*STH+CTH
DERYN(I,3) = CY*STH*DCPDA
DERYN(I,4) = CY*STH*DCPDB+1.0
DERYN(I,5) = CY*((DCPOTH-1.0)*STH+CPSI*CTH)

GO TO 9

TPSI = 0.0
DTPDA = 0.0
DTPDH = 0.0

DTPDTH = 0.0

XMODEL(I) = -RX*(CTH-TPSI*STH) + A
YMODEL(I) = -RX*(STH-TPSI*CTH) + B
DERXM(I,1) = -CTH+TPSI*STH
DERXM(I,2) = 0.0
DERXM(I,3) = RX*STH+DTPDA+1.0
DERXM(I,4) = RX*STH+DTPDH
DERXM(I,5) = RX*((DTPDTH+1.0)*STH+TPSI*CTH)
DERYN(I,1) = -STH-TPSI*CTH
DERYN(I,2) = 0.0
DERYN(I,3) = -RX*CTH*DTPDA
DERYN(I,4) = -RX*CTH*DTPDH+1.0
DERYN(I,5) = -RX*((DTPDTH+1.0)*CTH-TPSI*STH)

GO TO 9

CPSI = 0.0
DCPDA = 0.0
DCPDB = 0.0

DCPOTH = 0.0

XMODEL(I) = -RY*(CPSI*CTH-STH) + A
YMODEL(I) = -RY*(CPSI*STH+CTH) + B
DERXM(I,1) = 0.0
DERXM(I,2) = CPSI*CTH+STH
DERXM(I,3) = -RY*CTH*DCPDA+1.0
DERXM(I,4) = -RY*CTH*DCPDB
DERXM(I,5) = -RY*((DCPOTH-1.0)*CTH-CPSI*STH)
DERYN(I,1) = 0.0
DERYN(I,2) = -(CPSI*STH+CTH)
DERYM(I,3) = -RY*STH*OCPDA
DERYM(I,4) = -RY*STH*NCPDA*1.0
DERYM(I,5) = -RY*{OCPDA-1.0}*STH+GPS1**CTH

E(I) = XDATA(I) - XMODEL(I)
E(I+NPOINT) = YDATA(I) - YMODEL(I)
DO 10 J = 1, NPAR
Z(I,J) = DERX(I,J)
DO 11 I = 1, NPAR
Z(I,J) = DERY(I,J)
NPT2 = 2*NPOINT
DO 11 I = 1, NPAR
S(I,J) = 0.0
DO 12 J = 1, NPAR
GRADP(J) = S(I,J)
DO 13 J = 1, NPAR
BETA(J) = S(I,J) + Z(K,J)*Z(K,J)
CALL MINV(S, NPAR, D, L, N)
DO 14 J = 1, NPAR
PHI = PHI + S(I,J)*Z(I,J)
DO 15 J = 1, NPAR
PHI = PHI / NPAR
DO 16 J = 1, NPAR
PHI = PHI + S(I,J)*Z(I,J)
PHI = PHI / NPAR
RETURN

THE FOLLOWING TWO SUBROUTINES .... DATA, ESTIMATE .... ARE USED
WHEN EITHER A CONVEX OR CONCAVE CRESCENT IS TO BE STORED IN THE
COMPUTER. THE PARAMETER 'CONVEX' MUST BE SPECIFIED IN MAIN.

SUBROUTINE DATA (C, SIGMA, AVE, NPOINT)
DIMENSION X0(100), Y0(100), XD(100), YD(100), C(5)
COMMON/COM2/XDATA(100), YDATA(100)
COMMON/COM3/NGAUS
COMMON/CONV/CONVEX
INTEGER CONVEX
DATA IZS/1 /
MS = NPOINT-1
RX = C(1)
RYCIR = C(1)
RY = C(2)
A = C(3)
B = C(4)
THETA = C(5)
CTH = COS(THETA)
STH = SIN(THETA)
E1 = 1./RX**2
E2 = 1./RYCIR**2
XINT = 2.*RX
NPT2 = NPOINT/2+1
DO 1 I = 1, NPT2
XD(I) = RX + 2.*Z(I-1)*XINT/NPOINT
YD(I) = SQRT((1./E2)*(1.-E1*XD(I)**2))
NPTL = NPOINT/2-1
RYCIR = C(2)
IF (RYPELP .EQ. 0.0) GO TO 3
E2=1./RYELP*W*2
DO 2 I=1,NPTELP
XO(I+NPOINT/2+1)=RX-2.*XI*XINT/NPOINT
2 YO(I+NPOINT/2+1)=SORT((1./E2)*(1.-E1*YO(I+NPOINT/2+1)*W*2))
GO TO 5
3 DO 4 I=1,NPTELP
XO(I+NPOINT/2+1)=RX-2.*XI*XINT/NPOINT
4 YO(I+NPOINT/2+1)=0.0
5 IF (CONVEX) R,B,6
6 NSTART=NPOINT/2+2
7 YO(I)=YO(I)
8 DO 9 I=1,NPOINT
XO(I)=XO(I)*STH+YD(I)*STH+B
9 YD(I)=XO(I)*STH+YD(I)*STH+B
10 YDATA(I)=YD(I)+V
IF (CONVEX .EQ. 0) GO TO 11
WRITE (6,*5001) NPOINT,RX,RY,A,B,THETA,AVE,SIGMA
11 WRITE (6,*5002) NPOINT,RX,RY,A,B,THETA,AVE,SIGMA
12 WRITE (6,*5003) (I,Z*,XDATA(I),YDATA(I),I=1,NS),IZS,XDATA(NPOINT),YD
RETURN
END

SUBROUTINE ESTIMT (C*,NPOINT)
REAL RX,RX,YM,IZX,IXY
INTEGER CONVEX
COMMON/COM7/CONVEX
DIMENSION C(5),W(100),V(100),U(100),Z(100)
COMMON/COM2/XDATA(100),YDATA(100)
PI=3.1415926536
PIHALF=PI/2.0
XMEAN=0.0
YMEAN=0.0
DO 1 I=1,NPOINT
XMEAN=XMEAN+XDATA(I)
1 YMEAN=YMEAN+YDATA(I)
XMEAN=XMEAN/NPOINT
YMEAN=YMEAN/NPOINT
DO 2 J=1,NPOINT
U(J)=XDATA(J)-XMEAN
2 V(J)=YDATA(J)-YMEAN
IXX=0.0
IXY=0.0
YY=0.0
DO 3 I=1,NPOINT
IXX=IXX+U(I)*S
IYY=IYY+V(I)*S
3
IYY=IYY+V(I)
IXX=IXX/NPOINT
IYY=IYY/NPOINT
RATIO=(IXX+IYY+SORT(IXX-IYY)*S+4.*IXY*S)/2.*
I2=(IXX+IYY+SORT(IXX-IYY)*S+4.*IXY*S)/2.*
RATIO=12/RATIO
THETA=ATAN2(IXX,IXY)
IF (THETA .GE. PIHALF) GO TO 4
THETA=THETA-PIHALF
ITEMP=11
I1=12
I2=ITEMP
RATIO=1.0/RATIO
CONTINUE
CITH=COS(THETA)
STH=SIN(THETA)
DO 5 I=1,NPOINT
W(I)=U(I)*CITH+V(I)*STH
5
Z(I)=-U(I)*STH+V(I)*CITH
MIN=W(I)
MAX=W(I)
J=1
K=1
DO 7 I=1,NPOINT
IF (W(I) .GE. MIN) GO TO 6
MIN=W(I)
7
IF (W(I) .LE. MAX) GO TO 7
MAX=W(I)
K=1
CONTINUE
ZMIN=(Z(J)+Z(K))/2.
H=ZMIN
IF (H .LT. 0.0) H=0.0
A=XMEAN+H*STH
B=YMEAN+H*CITH
IF (11 .LT. 0.0335) RX=0.2
IF (11 .GE. 0.0335 .AND. 11 .LT. 0.0868) RX=0.4
IF (11 .GE. 0.0868 .AND. 11 .LT. 0.167) RX=0.6
IF (11 .GE. 0.167 .AND. 11 .LT. 0.274) RX=0.8
IF (11 .GE. 0.274 .AND. 11 .LT. 0.408) RX=1.0
IF (11 .GE. 0.408 .AND. 11 .LT. 0.569) RX=1.2
IF (11 .GE. 0.569 .AND. 11 .LT. 0.756) RX=1.4
IF (11 .GE. 0.756 .AND. 11 .LT. 0.970) RX=1.6
IF (11 .GE. 0.970 .AND. 11 .LT. 1.222) RX=1.8
IF (11 .GE. 1.222 .AND. 11 .LT. 1.490) RX=2.0
IF (11 .GE. 1.490 .AND. 11 .LT. 1.773) RX=2.2
IF (11 .GE. 1.773 .AND. 11 .LT. 2.094) RX=2.4
IF (11 .GE. 2.094 .AND. 11 .LT. 2.443) RX=2.6
IF (11 .GE. 2.443 .AND. 11 .LT. 2.816) RX=2.8
IF (11 .GE. 2.816 .AND. 11 .LT. 3.217) RX=3.0
IF (11 .GE. 3.217 .AND. 11 .LT. 3.645) RX=3.2
IF (11 .GE. 3.645 .AND. 11 .LT. 4.110) RX=3.4
IF (11 .GE. 4.110 .AND. 11 .LT. 4.582) RX=3.6
IF (11 .GE. 4.582 .AND. 11 .LT. 5.090) RX=3.8
IF (11 .GE. 5.090) RX=4.0
IF (CONVEX .EQ. 0) GO TO 9
THE FOLLOWING TWO SUBROUTINES ... SUMSOR, REGRES ... ARE USED
WHEN A CONVEX CRESCENT TEMPLATE IS TO BE STORED IN THE COMPUTER.

SUBROUTINE SUMSOR(C, PHI, KX, NSMSO, NPOINT, NPAR)

DIMENSION C(5), F(200), XMODEL(100), YMODEL(100)

COMMON/COM1/FSTDP
COMMON/COM2/XDATA(100), YDATA(100)
PI=3.1415926536
PIHALF=PI/2.
PI32=2.*PI

NSMSO=NSMSO+1
KX=0
PHI=0.0
RX=C(1)
RY=C(2)
A=C(3)
B=C(6)

THETA=C(5)

CTH=COS(THETA)

SIN=THETA

DO 6 I=1,NPOINT
X=DATA(I)

YD=YDATA(I)

WD=(X-A)*CTH+(YD-B)*SIN

ZD=(X-A)*SIN+(YD-B)*CTH

IF (WD.LT. 0.0) GO TO 1

1 IF (WD.LE. 0.0) GO TO 2

TPS=ZD/V1

PSI=ATA2(ZD, V1)

2 IF (PSI.LT. 0.0) PSI=PSI+PI2

END
208

IF (PSI .GT. PIHALF .AND. PSI .LT. P32) WM=- WM

GO TO 3

1

WM=0.0

ZM=RX

GO TO 3

2

WM=0.0

ZM=-RY

3

XMODEL(I)=WM*CTH-ZM*STH+A

YMODEL(I)=WM*STH+ZM*CTH+B

RY=C(2)

E(I)=XDATA(I)-XMODEL(I)

NPT2=2*NPOINT

DO 5 I=1,NPT2

9

SUBROUTINE REGRFS (C,GRADP,PHI,NPOINT,YPAR)

DIMENSION C(10),GRADP(10),BETA(T01),E(200),Z(200,5),S(5,5),L(5),N(5),XMODEL(100),YDATA(100)

COMMON/XHALF/XDATA(100),YDATA(100)

PI=3.1415926536

PIHALF=PI/2.

P32=3.0*PI/2.

A=1.0

THETA=C(5)

CTH=COS(THETA)

STH=SIN(THETA)

DO R=1,NPOINT

XO=XDATA(I)

YO=YDATA(I)

HO=(XO-A)*CTH+(YO-B)*STH

ZD=(XO-A)*STH+(YO-B)*CTH

IF (HO .LT. 0.0 . .0 . .0) GO TO 4

IF (HO .LT. 0.0 . .0 . .0) GO TO 5

TPS=2*HO

PSI=ATAN2(ZD,H0)

IF (PSI .LT. 0.0) PSI=PSI+PI2

IF (PSI .GE. 0.0 . .0 . .0) PHI=PI

WM=RX*RY/SQRT(RY**2+(RX*TPS)**2)

F1=SQRT(RY**2+(RZ**TPS)**2)

F2=F1+3

F3=(RX*STH+ZD*CTH)/(V**2)

F4=(-RX*STH+ZD*CTH)/(V**2)

F5=(2*CTH+ZD**2)/(V**2)
IF (PSI .GT. PI) GO TO 1
DWNDRX=1.0/SORT(I.0+TIPS1**2)
DWNDRY=0.0
GO TO 2

1 CONTINUE
DWNDRX=RY/F1-RY*(RX*TPSI)**2/F2
DWNDRY=RX/F1-RX*RY**2/F2

2 CONTINUE
DWMRHY=-RX**3*RY*ZM/F3/(U**2+F)
DWNDRY=-RX**3*RY*ZM/F4/(U**2+F)
DWNDRX=-RX**3*RY*ZM/F5/(U**2+F)
IF (PSI .LE. P1HALF .OR. PSI .GE. P132) GO TO 3
WM=.UN
DWNDRX=-DWMRHX
DWNDRY=-DWMRHY
DWNDRX=-DWMRHY
DWNDRX=-DWMRHY
DWMRHY=-DWMRHY
DWMRHY=-DWMRHY
DWMRHY=-DWMRHY
3 ZM=WMTPSI
DWMRHX=DWMRHX*TPSI
DWMRHY=DWMRHY*TPSI
DWMRHY=DWMRHY*TPSI+U**2/F3
DWMRHY=DWMRHY*TPSI+U**2/F4
DWMRHY=DWMRHY*TPSI-U**2/F5
GO TO 7
4 WM=0.0
ZM=RX
DWMRHY=1.0
GO TO 6
5 WM=0.0
ZM=RY
DWMRHY=-1.0
6 DWMRHX=0.0
DWMRHY=0.0
DWNDRX=0.0
DWNDRY=0.0
DWNDRX=0.0
DWNDRY=0.0
DWNDRX=0.0
DWNDRY=0.0
DWNDRX=0.0
7 XHADEL(I)=WM*CTH-ZM*STH+A
YHADEL(I)=WM*STH+ZM*CTH+B
DERXH(I,1)=DWMRHX*CTH-DWMRHX*STH
DERXH(I,2)=DWMRHY*CTH-DWMRHY*STH
DERXH(I,3)=DWMRHY*CTH-DWMRHY*STH+1.0
DERXH(I,4)=DWMRHY*CTH-DZM*RHY*CTH
DERXH(I,5)=(DWMRHY-ZM)*CTH-(DWMRHY+ZM)*STH
DERYH(I,1)=DWMRHY*STH+DWMRHX*CTH
DERYH(I,2)=DWMRHY*STH+DWMRHX*CTH
DERYH(I,3)=DWMRHY*STH+DZM*RHY*CTH
DERYH(I,4)=DWMRHY*STH+DZM*RHY*CTH
DERYH(I,5)=(DWMRHY-ZM)*STH+(DWMRHY+ZM)*CTH
F(I)=XDATA(1)-XHADFL(I)
F(I+1)=YDATA(I)-YHADEL(I)
RY=C(2)
GO 0 , J=1,MPAR
Z(I,J)=DERXH(I,J)
GO 0 , I=1,MPAR
Z(I,J)=DERYH(I,J)
GO 1 , J=1,MPAR
THE FOLLOWING TWO SUBROUTINES SINSOR, REGRES ARE USED WHEN A CONCAVE CRESCENT TEMPLATE IS TO BE STORED IN THE COMPUTER.

SUBROUTINE SINSOR(C, PHI, XX, NSMASO, NPOINT, NPAR)
 DIMENSION C(5), F(200), XMODEL(100), YMODEL(100)
 COMMON/COM1/ESTOP
 COMMON/COM2/XDATAt100), YDATA(100)
 COMMON/COM6/
 PM1 = 3.141597653
 PM1HALF = PM1/2.
 PM12 = 2. * PM1
 NSMASO = NSMASO + 1
 XX = 0
 PHI = 0.0
 NPOINT = 0
 RX = C(1)
 RY = C(2)
 A = C(3)
 B = C(4)
 THETA = C(5)
 CTH = COS(THETA)
 STH = SIN(THETA)
 DO 6 I = 1, NPOINT
 XD = XDATA(I)
 YD = YDATA(I)
 HD = (XD - A) * CTH + (YD - N) * STH
 ZD = -(XD - A) * STH + (YD - N) * CTH
 IF (HD .GE. 0.0 .AND. ZD .GT. 0.0) GO TO 2
 IF (HD .GE. 0.0 .AND. ZD .LE. 0.0) GO TO 5
 PS1 = ATAN2(ZD, HD)
 IF (PS1 .LT. 0.0) PS1 = PSI+PI2
 IF (PS1 .GT. PI) PS1 = PSI-PI2
 TPS1 = ZD/HD
 WMC = RX/SORT(1.0+TPS1**2)
 WME = RX*RY/SORT(RY**2+(RX*TPS1)**2)
 IF (PS1 .GE. 0.0 .AND. PSI .LE. PIHALF) GO TO 1
 WMC = WMC
 WME = WME
 1 ZMC = WMC*TPSI
 ZME = WME*TPSI

SUBROUTINE REGRES(PHI, PP, XX, NSMASO, NPOINT, NPAR)

GO TO 3
2  WNC=0.0
    WME=0.0
    ZNC=RX
    ZME=RY
3  XNC=WNC*CTH-ZNC*CTH+A
    YNC=WNC*CTH+ZNC*CTH+0
    XME=WME*CTH-ZME*CTH+A
    YME=WME*CTH+ZME*CTH+0
    EC1=XD*XNC
    EC2=YN*CYC
    EE1=XD*YME
    EE2=YN*YME
    ECO=EC1*EC2+EE1*EE2
    EES=EE1*EE2+EC1*EC2
    IF (EES EQ LE, ECS0 GO TO 4
    XMODEL(1)=XNC
    YMODEL(1)=YN
    E(1)=EC1
    E(1+NPOINT)=EC2
    GO TO 6
4  XMODEL(1)=XME
    YMODEL(1)=YME
    E(1)=EE1
    E(1+NPOINT)=EE2
    GO TO 6
5  E(1)=0.0
    E(1+NPOINT)=0.0
    NOMIT=0
6  CONTINUE
    NPT=2*NPOINT
    DO 7 I=1,NPT
    IF (ABS(E(I)) LE. ESTOP) GO TO 7
    KX=1
5002 WRITE (6,5002) (C(J),J=1,NPAR)
      5002 FORMAT (' THE SYSTEM IS UNSTABLE FOR THESE PARAMETER VALUES, THE S
               1 UN-SQUARED ERROR WILL NOT BE EVALUATED, THE PARAMETERS ARE:
               C(1) =',E15.8,' C(2) =',E15.8,' C(3) =',E15.8,' C(4) =',E15.8,' C(5) ='
               'E15.8')
    GO TO 8
7  PHI=PHI+E(I)**2
      PHI=PHI/(NPOINT-NOMIT)
8  CONTINUE
RETURN
END

SUBROUTINE REGF, C,GRADP,BETA,PHI,NPOINT,NPAR)
DIMENSION C(10),GRADP(10),BETA(10),E(200),Z(200),S(5,5),L(5,5)
  1,XMODEL(100),YMODEL(100),DERXM(100,5),DERYN(100,5)
COMMUN/COM?,YNATA(100),YNATA(100)
PI=3.141592653
PHALF=PI/2.
P13=3.*PI/2.
P12=2.*PI
NOMIT=0
RX=C(1)
RY=C(2)
A=C(3)
B=C(4)
THETA=C(5)
C=THCOS(THETA)
STH = SIN(THETA)

DO 13 I = 1, NPO1NT
  XD = XDATA(I)
  YD = YDATA(I)

  WD = (XD - A) * CTH + (YD - B) * STH
  ZD = -(XD - A) * STH + (YD - B) * CTH
  IF (IUD * EO * 0.0 * AND. ZD * GT. 0.0) GO TO 2
  IF (YD * EO * 0.0 * AND. ZD * LE. 0.0) GO TO 7
  PSI = ATAN2(ZD, YD)

  IF (PSI * LT. 0.0) PSI = PSI + PI
  IF (PSI * GT. PI) GO TO 7
  TPSI = ZD / HUD

  WMC = RX / SORI(1.0 + TPSI**2)
  WME = RX * RY / SORI(RY**2 + (RX * TPSI)**2)
  IF (PSI * GE. 0.0 * AND. PSI * LE. PIHALF) GO TO 1
  WM = WMC

  WME = WMC * TPSI
  WM = WME * TPSI
  GO TO 3

  WMC = 0.0
  WME = 0.0

  ZHC = RX
  ZCE = RY

  XHC = WMC * CTH - ZHC * STH + A
  YHC = WME * STH + ZHC * CTH + B
  XME = WME * CTH - ZME * STH + A
  YME = WME * STH + ZME * CTH + B

  EC1 = XD - XME
  EC2 = YD - YME
  EE1 = XD - XME
  EE2 = YD - YME

  ECSO = EC1**2 + EC2**2
  EESO = EE1**2 + EE2**2
  IF (EESO * GT. ECSO) RX = RX
  IF (IUD * EO * 0.0 * AND. ZD * GT. 0.0) GO TO 8
  WM = RX * RY / SORI(RY**2 + (RX * TPSI)**2)
  F1 = SORI(RY**2 + (RX * TPSI)**2)
  F2 = F1**2
  F3 = -(IUD * STH * ZD * CTH) / (IUD**2)
  F4 = -(IUD**2 + ZD**2) / (IUD**2)
  IF (EESO * LE. ECSO) GO TO 4
  DWM = 1.0 / SORI(1.0 + TPSI**2)
  DWM = 0.0
  GO TO 5

  DWM = RX / F1 - RX * (RX * TPSI)**2 / F2
  DWM = RX / F1 - RX * RY / F2

  DWM = RX / F1 - RX * ZD / F2
  DWM = RX * RY + ZD / F2
  DWM = RX / F1 - RX * RY**2 / F2
  IF (PSI * LE. PIHALF) GO TO 6
  WM = WM

  DWM = WM
  DWM = WM

  ZHC = ZME + TPSI
  DZMDRX = DWM * RX + TPSI
  DZMDRY = DWM * RY + TPSI
  DZMDA = DWM + TPSI + F3
DZHDI = DYNDA * TPSI + W * E4
DZHDTH = DYNTHI * TPSI - W * E5
GO TO 10

7  DEXXH (1, 1) = 0.0
    DEXXH (1, 2) = 0.0
    DEXXH (1, 3) = 0.0
    DEXXH (1, 4) = 0.0
    DEXXH (1, 5) = 0.0
    DERYH (1, 1) = 0.0
    DERYH (1, 2) = 0.0
    DERYH (1, 3) = 0.0
    DERYH (1, 4) = 0.0
    DERYH (1, 5) = 0.0
E(I) = 0.0
E(1 + NPOINT) = 0.0
NOUT = NOUT + 1
GO TO 12

8  W = 0.0
  ZK = PY
  DZMOR = 1.0
  DZMORX = 0.0
  DZMDRY = 0.0
  DZMORI = 0.0
  DZMORD = 0.0
  DZMDTH = 0.0
  DZMORI = 0.0
  DZMORDI = 0.0

10 XMODEL (1) = W * STH - XYZTH + A
    YMODEL (1) = W * STH + XYZTH + B
    E(I) = XDATA(I) - XMODEL(I)
    E(I + NPOINT) = YDATA(I) - YMODEL(I)

11 DEXXH (1, 1) = DZMRIC * STH - DZMIRX * STH
    DEXXH (1, 2) = DZMDRY * STH - DZMDRY * STH
    DEXXH (1, 3) = DZMDRA * STH - DZMDRA * STH + 1.0
    DEXXH (1, 4) = DZMDRA * STH - DZMDRA * STH
    DEXXH (1, 5) = (DZMOTH - ZK) * STH - (DZMOTH + W) * STH
    DERYH (1, 1) = DZMDRX * STH + DZMDRX * STH
    DERYH (1, 2) = DZMDRY * STH + DZMDRY * STH
    DERYH (1, 3) = DZMDREN * STH + DZMDREN * STH
    DERYH (1, 4) = DZMDREN * STH + DZMDREN * STH + 1.0
    DERYH (1, 5) = (DZMOTH - ZK) * STH + (DZMOTH + W) * STH

12 RY = C(2)
    DO 13 J = 1, NPAR
    Z(I, J) = DEXXH(I, J)
13 Z(I + NPOINT, J) = DERYH(I, J)
    NPT2 = 2 * NPOINT
    DO 14 I = 1, NPAR
    DO 14 J = 1, NPAR
    S(I, J) = 0.0
    DO 14 K = 1, NPT2
    S(I, J) = S(I, J) + Z(K, I) * Z(K, J)
    CALL MINV(S, NPAR, D, L, H)
    DO 15 J = 1, NPAR
    GRANDP(J) = 0.0
    DO 15 I = 1, NPT2
    GRANDP(J) = GRANDP(J) - 2. * Z(I, J) * E(I)
    DO 16 I = 1, NPAR
    BETA(I) = 0.0
    DO 16 J = 1, NPAR
    BETA(I) = BETA(I) - (1 / 2.0) * (S(I, J) * GRANDP(J))
THE FOLLOWING TWO SUBROUTINES **** RANDU, GAUSS **** ARE LIBRARY SUBROUTINES WHICH ARE USED TO GENERATE UNIFORM AND NORMAL RANDOM NUMBERS, RESPECTIVELY. THEY ARE LISTED HERE FOR COMPLETENESS.

SUBROUTINE RANDU(IX, IY, YFL)
  Y = IX*65539
  IF (Y) 5,6,6
  Y = Y+2147483647+1
  YFL = Y
  YFL = YFL*0.04656613E-09
  RETURN
END

SUBROUTINE GAUSS(IX, S, AM, V)
  A = 0.0
  DO 50 1 = 1, 12
    CALL RANDU(IX, IY, Y)
  IX = IY
  A = A + Y
  V = (A - 0.0)*S + AM
  RETURN
END


