To the Graduate Council:

I am submitting herewith a dissertation written by Fred W. DePiero entitled "ADAPTIVE RANGE IMAGE SEGMENTATION AND DIRECT GRAPH MATCHING FOR 3-D OBJECT RECOGNITION: ALGORITHMS, INTEGRATED FRAMEWORK AND IMPLEMENTATIONS ." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy with a major in Electrical Engineering.

Mohan M. Trivedi

Mohan M. Trivedi, Major Professor

We have read this dissertation
and recommend its acceptance:

Steven Mark Serbin

David W. Bouldin

Marshall Pace

Reed Kress

Accepted for the Council:

Associate Vice Chancellor and
Dean of The Graduate School

ADAPTIVE RANGE IMAGE SEGMENTATION AND DIRECT GRAPH MATCHING FOR 3-D OBJECT RECOGNITION: ALGORITHMS, INTEGRATED FRAMEWORK AND IMPLEMENTATIONS

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Fred W. DePiero
May 1996

Dedicated to

my parents, for their guidance and to

my wife, Connie, for her love and support.

Acknowledgments

Many people have contributed to the process that led to the completion of this dissertation. I would particularly like to thank:

- My major professor, Prof. Mohan Trivedi.
- My committee members, Professors Donald Bouldin, Marshall Pace, Steve Serbin and Reid Kress for their invaluable suggestions and guidance.
- The Japan Railways Company and the U.S. Department of Energy for providing financial support for this research.

Abstract

There are three main points of focus in this dissertation, all of which address important aspects of the object recognition problem. These are: **adaptive range segmentation, graph matching** and **calibrated 3-D ranging**. Computationally efficient, robust and accurate means to accomplish these types of problems are critical to improving the state of the art in object recognition.

In addition to the above three main areas, a fourth objective was also set. It was desired to create components using a systemic perspective, allowing them to be put into an integrated framework. Their ability to be integrated into a system is considered to be an important aspect of their real-world utility. Integrating the three main contributions into a complete system permitted validation of the performance of individual components in a system context.

The novelty of the research in 3-D ranging has produced new approaches to calibration and to acquisition. These techniques make for a system that acquires data at frame rate, is accurate and is easy to calibrate. The calibration technique that has been developed permits modeling of all elements of the sensor package in a single step via a least norm solution. This is simpler and can provide greater accuracy than more traditional methods that model the camera and laser separately. The range sensor has an accuracy of better than 0.005 inches.

An adaptive approach for robust, accurate and computationally efficient segmentation of range images has also been developed. It is well matched to the range acquisition process. Adaptivity is a new trend in range image processing. Adaptive schemes are not bound by rigid geometries when accessing data, for

example. This improves performance by better handling outliers. The new approach uses a combination of multiple adaptive mechanisms. Tests demonstrated that segmented regions can be computed in less than 3 seconds, under a wide variety of test conditions. Stable operation was achieved above 90% of the time in the majority of testing trials.

A new approach to graph matching has also been developed as a part of this effort. It provides a tremendous computational advantage over established methods that are either based on search, or on iterative relaxation techniques, or on alternate exhaustive formulations. The new method requires a computational effort that is of polynomial order. It is able to maintain a performance level for subgraph isomorphisms at or above 95% under a wide variety of conditions and with varying levels of noise. The performance on the full size comparisons associated with graph isomorphisms has been found to be 100/100, also under a variety of conditions.

Careful thought went into test conditions for individual components and for integrated tests. Types of surface curvature and discontinuities were important considerations, as were the complexity of objects and the number of database models. A suite of discontinuities and surface types were used in testing. Testing was performed using both synthetic scenes (4000+) and real scenes (500+). Results are reported for individual processing modules and for the integrated system performance.

Preface

While working as an Engineer I had considerable experience with range acquisition systems. Despite successes in the workplace, I was unsatisfied. I had felt limits on my creative freedom. The subjects that really inspired me involved higher levels of scene interpretation. At first I thought that I could expand my skills into these areas while remaining on the job. I now know that this was not very realistic. It was while taking Dr. Trivedi's class in Pattern Recognition that I became intrigued with opportunities at the University of Tennessee. Since being at the University, I have thrived in the very creative research environment that Dr. Trivedi provided. His only requirement was that my dissertation come from my heart.

TABLE OF CONTENTS

CHAPTER		PAGE
1.	**Research Overview**	**1**
1.1	Brief Problem Statement	1
1.2	Design Considerations and Style of an integrated Framework	2
1.2.1	Challenges of an Integrated System	4
1.2.2	Benefits of the New Framework	6
1.3	The RDM Framework	7
1.4	An Application Domain in Future Manufacturing	12
1.5	Keywords of the Problem Domain	15
1.5.1	Active Sensing	15
1.5.2	Adaptive Range Segmentation	15
1.5.3	Model-Based Recognition	16
1.5.4	Exact, Object-Centered, 3-D Representation	16
1.5.5	Graph Matching	16
1.5.6	Integrated System Perspective	17
1.6	Concluding Remarks	17
2.	**Literature Review**	**18**
2.1	Literature Review of Range Segmentation Techniques	18
2.1.1	A Taxonomy for Range Segmentation	18
2.1.2	1-D Nonadaptive Kernels	20
2.1.3	Large 2-D Nonadaptive Kernels	21
2.1.4	Small 2-D Nonadaptive Kernels	22
2.1.5	Adaptive Kernels	23
2.1.6	Qualitative Evaluation of Segmentation Techniques	27

2.1.7 Quantitative Metric for Segmentation Algorithms 30
2.2 Literature Review of Graph Matching Techniques from the Machine Vision Community 33
2.2.1 The Importance of Graph Matching in Machine Vision . . . 33
2.2.2 Taxonomy for Graph Matching 35
2.2.3 Maximal Clique-Based Approaches 36
2.2.4 Relaxation Labeling Techniques 37
2.2.5 Tree Search Techniques 38
2.3 Literature Review of Graph Matching Techniques from Mathematics Community 40
2.4 Literature Review of Structured Light Sensing 44
2.4.1 3-D Ranging 44
2.4.2 Approaches to Backbone Geometry 46
2.4.3 Approaches to SL Correspondence Problem 47
2.4.4 Methods of Pixel-Level Analysis 48
2.4.5 Alternate Illumination Structures for Ranging 49

3. Efficient and Accurate Acquisition of Range Profiles By Structured Light: Design, Calibration and Implementation Issues † **51**
3.1 Introduction to Structured Light Ranging 51
3.2 Structured Light Acquisition 56
3.2.1 Optics and Imaging 57
3.2.2 Image Processing Operations 60
3.2.3 Ranging Computations 63
3.3 Calibration of Structured Light Sensors 64
3.3.1 Calibration Process and Formulation of Calibration Model . 64
3.3.2 Numerical Method of Solution for Calibration Models 69
3.3.3 Evaluation of Calibration Model 71

3.4 Sensitivity Analysis of Structured Light Sensors 77
3.5 Performance Benchmarks and Ranging Experiments 79
3.6 Concluding Remarks 82

4. Range Profile Tracking: A Profile-Based Approach for Adaptive Range Image Segmentation 83
4.1 Importance and Challenge of Range Segmentation 83
4.2 Range Profile Tracking (RPT) Approach 84
4.2.1 Tracking Profiles for an Initial Segmentation 85
4.2.2 Merging Profiles to Form Regions 94
4.2.3 Region Refinement Process 97
4.3 RPT Formulation 100
4.3.1 State Space Model of the Measurement Process 101
4.3.2 The Steady State Kalman Filter 103
4.3.3 Initialization of the Kalman Filter State 105
4.4 Experimental Validation of RPT 106
4.4.1 RPT Testing Trials 107
4.4.2 RPT Performance Summary 108
4.5 RPT Testing on Real Data 109
4.6 Concluding Remarks 113

5. A Direct Method for Graph Matching: Algorithm, Theoretical Analysis, and Testing † 114
5.1 Introduction 114
5.2 The Challenge of Determining Graph Isomorphisms 115
5.3 The DCA Approach to Graph Matching 117
5.3.1 Goals for DCA Algorithm Development 118
5.3.2 An Example Object Representation 119
5.4 Comparing Graph Topology Using Connectivity Signatures 119

5.5 Analysis of Connectivity Signatures 124
5.5.1 Notation and Definitions 125
5.5.2 Assumptions 127
5.5.3 Verifying Isomorphisms 127
5.5.4 Invariance of Connectivity Signatures 128
5.5.5 Strategy to Find Degree of Uniqueness of Connectivity Signatures 131
5.5.6 Statistical Description of a Connectivity Signature 131
5.5.7 Probability of Matching Connectivity Signatures 136
5.5.8 Number of Connectivity Signatures in All Possible Random Graphs 137
5.5.9 Expected Number of Matching Connectivity Signatures . . . 138
5.6 Factors Effecting A Practical Algorithm 139
5.7 DCA Algorithm 140
5.8 Performance Metrics for DCA 144
5.9 Computational and Memory Requirements for DCA 149
5.10 DCA Testing Methodology 149
5.10.1 Relation of Test Conditions to an Object Recognition Scenario 149
5.10.2 Testing Procedure 151
5.11 Experimental Verification: Results and Implications 153
5.11.1 Tests Using DCA to Find Full-Sized Isomorphisms 156
5.11.2 Tests Using DCA to Find Subisomorphisms 157
5.12 Concluding Remarks 161
5.13 Acknowledgements 163

6. Integrated Framework and System Implementation 164
6.1 Motivation for a Systemic Perspective and Implementation 164
6.2 Integrated Framework 165

CHAPTER PAGE

6.2.1 Geometrical Modeling and Graph Construction 167
6.2.2 The Iterative Nature of Geometrical Modeling 172
6.2.3 GEOGRAM Representation 173
6.2.4 GEOGRAM Approach 174
6.2.5 Finding Neighbors 180
6.3 Computing Platforms and System Implementation 180
6.4 Testing Overview 182
6.5 PRIME Testing 186
6.6 RPT Testing 186
6.7 GEOGRAM Testing 192
6.8 Overall System Testing 193
6.9 Concluding Remarks 196

7. Summary, Conclusions and Future Research Directions 198
7.1 Novel Contributions 198
7.2 Further Reseach Areas 200
7.2.1 Multidirectional Optics 200
7.2.2 Opposing Directions for Profile Segmentation 200
7.2.3 Alternate Graphical Representations 201
7.2.4 Merging Adjacent Models 201
7.2.5 Combine Range Data Acquired from Different Views 201
7.2.6 Uncertainties in Node and Edge Properties 202
7.2.7 Fuzzy Graph 202
7.2.8 Normalizing Factors 202
7.2.9 Search for Additional Matches in Scene 203

BIBLIOGRAPHY 204

APPENDICES 218

CHAPTER PAGE

A. ADDITIONAL TEST RESULTS FOR RPT WITH SYNTHETIC SCENES 219

B. ADDITIONAL FIGURES OF OBJECTS USED IN INTEGRATED TESTING 226

C. ADDITIONAL FIGURES OF OBJECTS USED FOR TESTING RPT . 241

VITA **250**

LIST OF TABLES

TABLE **PAGE**

2.1 Qualitative comparison of range segmentation techniques. Accuracy tends to improve with increase kernel area and with the use of iterative methods. Robustness is typically challenged unless an adaptive approach can be used to avoid outliers. The computational efficiency is related to kernel size and placement. 28

2.2 Examples of Quantitative Metric for kernel evaluation. All examples had $A = 100, S = 5$. 32

3.1 Demands on numerical precision for normalized and unnormalized calibration models. 70

3.2 Metrics for Z calibration model for PRIME. A total of 280 calibration triplets were used. A value of 0.005 inches was used for the accuracy of z_i. 73

3.3 Metrics for X calibration model for PRIME. A total of 98 calibration triplets were used. A value of 0.010 inches was used for the accuracy of x_i. 73

3.4 Form of calibration models. 73

3.5 PRIME performance measures as a function of the type of calibration model. Performance of the normalized vs. unnormalized forms was essentially identical, normalized models were used herein. 81

4.1 The following equivalences of X-Strips have been found by traversing the given Y-Strips seen in the example. 96

4.2 Summary of test results on real scenes. Processing parameters for all scenes were set using σ_m = 0.015 in. 112

5.1 Summary of test results when matching full-size graphs. The graphs in these tests had no node or edge properties, no property noise and no topological noise. All graphs were strongly-regular with degree ≈ 4. Performance was maintained at nearly 100%, provided the graph size was above ≈ 10. 156

5.2 Summary of test results examining effect of subgraph size. These results indicate that at least half of the graph should be retained given these size and noise conditions. 157

5.3 Summary of test results examining effect of graph size. Perfomance degrades as the graph size decreases due to the reduced information present in the Connectivity Signature. 158

5.4 Summary of test results examining effect of the number of database elements. Most performance measures remain fairly constant except for the percentage of correct database selections. 158

5.5 Summary of test results examining effect of V_{max}. Values for the V_{max} parameter are given in the second column. A value of 1 is trivial in the sense that no additional information is included in the Connectivity Signatures beyond that of the original Adjacency matrix. The need for the Connectivity Signature can be seen by the jump in correct database selections that accompanys nontrivial V_{max} extents. In all other testing $V_{max} = ceil(N_s/4.0)$. 159

5.6 Summary of results examining effect of the number of subgraphs present in scene. The percentage of correct name tags suffered a significant drop off here. These incorrectly matched nodes would have to be identified and removed during the pose determination process. 160

5.7 Summary of test results examining dynamic range of node and edge properties. Results indicate the importance of the dynamic range of edges. 161

5.8 Summary of test results examining noise in node and edge properties. Results are fairly consistent across the span of noise intensities, except in the extreme cases. 162

5.9 Summary of tests examining addition of topological noise. The topological disturbances associated with multiple scene graphs is more severe than the noise in these tests. 162

6.1 Node properties used in GEOGRAM. Node property vectors contain two elements, an integer describing the type of shape and a real value describing an associated parameter. 174

6.2 Edge properties used in GEOGRAM. As with nodes, edge property vectors contain two elements, an integer describing shape types and a real value describing an associated parameter. 175

6.3 Accuracy of PRIME sensor as measured by the mean absolute residual error. The entry for the machined plate below is taken from the results of the calibration procedure. Sensor noise appears to have an upper bound of < 0.005 inches. 187

6.4 Test results for RPT on real scenes containing a planar surface bounded by jump discontinuities. The volumetric error and surface fits indicate a high accuracy, for this type of simple surface, but with somewhat limited robustness, as indicated by the region count. Trials with an incorrect count typically had 2 or 3 regions present. . 188

6.5 Test results for RPT on real scenes containing a 2nd order surface bounded by jump discontinuities. This surface was a $1/4$ cylinder that can be closely approximated by a $z = f(x, y)$ polynomial. This shows an improved performance over the planar case. 188

6.6 Test results for RPT on real scenes containing two planar surfaces separated by a crease discontinuity. Accuracy measures are reported for both of the two surfaces. Note the improved robustness indicated by the correct region count. 189

6.7 Test results for RPT on real scenes containing a 2nd order surface bounded on two sides by a curvature discontinuity at a join with planar surfaces. Accuracy measures are reported for each of the 3 surfaces. Accurate segmentation has not been achieved by RPT with these curvature discontinuities. 189

6.8 Types of shapes modeled by GEOGRAM and associated parameters. The method of comparing each parameter is indicated. The angular errors have been computed via the absolute value of a dot product. 193

6.9 Results from GEOGRAM tallied for instances of the planar surfaces on scans of Object #1. Data from 33 planes is given. These were located in various locations on the object, as it appeared in various positions and orientations. 194

6.10 Test results for GEOGRAM tallied for instances of the cylidrical surfaces on scans of Object #1. Data from 18 cylinders is given. These were located in various locations on the object, as it appeared in various positions and orientations. Compute durations are the same as in the planar case because these statistics were accumulated for the same set of object scans. 194

6.11 System performance for scans of each of the 8 objects. Results are given as means. A total of 200 scans were made, 25 of each object. . 196

TABLE PAGE

A.1 Results from Monte Carlo analysis of RPT when segmenting a scene containing a single object having a first order surface bounded by jump discontinuities. 100 Trials were run at each noise level (1000 trials in all). The SNR for this set of tests varied from 7.1 to 7.7 db. 221

A.2 Results from Monte Carlo analysis of RPT when segmenting a scene containing a single object having a second order surface bounded by jump discontinuities. 100 Trials were run at each noise level (1000 trials in all). The SNR for this set of tests varied from 5.2 to 5.5 db. 222

A.3 Results from Monte Carlo analysis of RPT when segmenting a scene containing a crease discontinuity separating two planar surfaces. 100 Trials were run at each noise level (1000 trials in all). The SNR for this set of tests varied from 13.2 to 13.3 db. 223

A.4 Results from Monte Carlo analysis of RPT when segmenting a scene containing curvature discontinuities. 100 Trials were run at each noise level (1000 trials in all). The SNR for this set of tests varied from 15.1 to 15.7 db. 224

LIST OF FIGURES

FIGURE PAGE

1.1 Components of the RDM framework for object recognition. 8

1.2 Designing in the Virtual World permits visualization and assembly of prototype elements. Visualizations can be augmented with attributes of components that are otherwise difficult to perceive. A virtual assembly can be accomplished with parts from distributed locations. 13

1.3 Designing in the Real World permits in-hand modification of parts. Scanning modified parts permits design efforts to switch back to the Virtual World. Changes in the geographic location of design teams can accompany transitions between the Real and Virtual Worlds. . 14

2.1 Cartoon scene illustrating the fundamental problem associated with nonadaptive kernels. 32

3.1 Structured Light systems use triangulation to acquire range measurements. In PRIME, the ranging geometry is formed by a laser emission, the reflected light observed by the camera and by a rigid backbone. Optical measurements capture range data in a plane. A conveyer produces the necessary motion for 3-D range data. 52

3.2 The purpose of sensor calibration is to find a mapping between image and world coordinates. In PRIME, this relationship is fixed because of the rigid backbone between the laser and camera. 53

3.3 Image of laser profile seen by camera. The camera's optical filter was removed for this image to better reveal objects in the scene. . . 55

3.4 Laser (left) and camera (right) used in PRIME sensor. The laser generates a plane of light using a cylindrical lens. A bandpass optical filter on the camera yields distinct imagery of the laser plane as it intersects objects in the scene. 57
3.5 PRIME sensor with typical objects. 58
3.6 Block diagram of the processing steps in the PRIME sensor. 60
3.7 Recovering image coordinates at the center of the laser line. 62
3.8 A series of horizontal plates are used to generate calibration triplets. The triplets are used to form an overdetermined set of equations from which calibration models may be determined. This illustrates the calibration procedure for the $z = f(r, c)$ model. 66
3.9 A series of vertical plates are used to generate the data needed for the $y = f(r, c)$ calibration model in PRIME. 67
3.10 An image used for range measurement calibration. The central horizontal line is produced by the intersection of the laser plane with a horizontal metal plate. 68
3.11 Residual error of a 1st order (type 1) Z calibration model. Mean = 0.071 in. Gray levels of each block are scaled to span the extremes of the residual errors in this model (maximum = 0.137 in). 74
3.12 Residual error of a 3rd order (type 4) Z calibration model. Mean = 0.002 in. Gray levels of each block are scaled to span the extremes of the residual errors in this model (maximum = 0.013 in). 75
3.13 Residual error of a 1st order (type 1) X calibration model. Mean = 0.093 in. Gray levels of each block are scaled to span the extremes of the residual errors in this model (maximum = 0.240 in). 75
3.14 Residual error of a 3rd order (type 4) X calibration model. Mean = 0.004 in. Gray levels of each block are scaled to span the extremes of the residual errors in this model (maximum = 0.009 in). 76

3.15 The sensitivity dz/dw of a Fixed-Plane SL system is determined by the angle, θ, between the laser emission and the camera sighting. This describes the effect of the change in image coordinates with respect to changes in height. 78

3.16 Gray scale images captured from the video camera during the ranging process. The optical filter normally used during acquisition was removed for these images, to better illustrate the interaction between the laser plane and object surfaces. The laser appears as a bright line near the bottom of each image. 80

3.17 Measurements captured by PRIME for objects with a variety of surface textures. 80

4.1 RPT has three fundamental steps. "Track Shape Profiles" forms 1-D strips of range data having uniform second order curvature. "Merge Shape Profiles" forms seed regions having 2-D extent. These initial regions are then refined using a seed growing method to improve the accuracy of region boundaries. 86

4.2 Many narrow strips of range data are formed using RPT. At each iteration new measurements are added to each strip and the Kalman filter's states are updated. 1 of 2 orthogonal sets of strips are shown. These "X-Strips" grow in the X direction. 87

4.3 The discontinuity threshold helps to determine the lengthwise extent of a strip. A Kalman filter is associated with each strip and is used to both accumulate range measurements and to provide a basis for the threshold. 89

4.4 Using the discontinuity threshold alone for Strip Formation is insufficient - as seen in this example. This criteria is somewhat "open-loop" in nature. A double check is needed during Strip Formation. Here, the object's shape within the strip shown is not second order in nature, therefore it should be segmented into two components. . 90

4.5 To double check the validity of a KF model, the residual error of the first R contributors to the strip is compared to a threshold. The earliest contributors have the largest moment arm for their residual errors. 91

4.6 Results of Strip Formation in the X direction. The original synthetic object was contaminated with Gaussian noise having $\sigma_m = 0.010$ in. 93

4.7 The orthogonal strip-merging process. Smoothness in the Y direction is used as a criteria to merge adjacent X-Strips. Only two columns of Y-Strips are highlighted here, for clearity. 95

4.8 Results of fitting a second order surface over a synthetic object using RPT. 98

4.9 RPT algorithm. Comments appear in bold. 99

4.10 Gray scale images captured from the video camera during the ranging process. The optical filter normally used during acquisition was removed for these images, to better illustrate the interaction between the laser plane and object surfaces. The laser appears as a bright line near the bottom of each image. 109

4.11 Measurements and processing results from a SL scan of an blocks scene. The Measurements, X-Strips, Y-Strips and final Fitted Surfaces are shown. 110

4.12 Measurements and processing results from a SL scan of a cluttered scene containing planar and second order surfaces. The Measurements, X-Strips, Y-Strips and final Fitted Surface are shown. 110

4.13 Measurements and processing results from a SL scan of marble goddess. The Measurements, X-Strips, Y-Strips and final Fitted Surfaces are shown. 111

4.14 Measurements and processing results from a SL scan of a fabric bear. The Measurements, X-Strips, Y-Strips and final Fitted Surfaces are shown. 111

5.1 Example of the 2 1/2-D relational surface patch representation. Scene on the left contains a cube with 3 visible surfaces, s_i. Node properties on the right are used to describe these surfaces. Edge properties describe the relative orientations of the surface normals using a dot product. 120

5.2 Graph data used when forming the local characterization of nodes. 121

5.3 Connectivity Signature (CS) vectors are formed using powers of the adjacency matrix, A^i. 124

5.4 Steps involved in node classification. Unlabeled arrows describe data flows of attendance ratings. Circles indicate operations where attendance ratings are combined. 143

5.5 Dual histogram of attendance ratings recorded in a subisomorphism problem. Entries for present nodes are given by the lighter (right) bars. Entries associated with absent nodes are indicated with the darker (left) bars. Present and absent nodes are segmented using the attendance rating. Dichotimization is sufficient given such a distribution, provided the global verification step is performed to complete DCA processing. 145

5.6 Dual histogram of the attendance ratings in an isomorphism problem. Note the dichotimization is significantly improved over the subisomorphism case. These data were recorded from a test with node and edge properties having a **zero** dynamic range and with strongly-regular graphs. 146

5.7 Histogram of subgraph size. This test involved database graphs with 30 nodes and a single scene graph having 15 nodes. This subisomorphism problem had a mean $\pm$ standard deviation of 10.7 ± 2.3 nodes. 147

5.8 Fraction of correct labels. This test involved database graphs with 30 nodes and a single scene graph having 15 nodes. The mean $\pm$ standard deviation of correct labels was $98 \pm 5\%$. 148

5.9 An example of a randomly generated database entry. This graph is strongly-regular, each node has degree ≈ 4. 150

5.10 DCA testing procedure. Typically 100 trials were run for each set of test conditions. Each trial involved the random generation of databases and scenes. Both topological and property noise were then added. Three performance metrics were used for benchmarking. 152

5.11 A scene graph generated from randomly selected database contributors. This scene is composed of a single subgraph with some added topological noise that can be seen in the 4 edges that connect the central nodes to the 4 peripheral nodes. 153

5.12 A scene graph formed from 2 database contributors. Excess nodes and edges are present. 154

6.1 The centroid of a segmented patch is highly dependent on measurement geometry. This can result in a poor coincidence between patch and shape centroids. 169

6.2 Objects that are not approximated well by polynomial surface patches tend to have unstable region boundaries. 170
6.3 The cost function for finding a cylinder sums the distances from each range point to the cylinder surface. 179
6.4 Components of the RDM framework for object recognition. 181
6.5 Object #1. 184
6.6 Measurements captured by PRIME of Object #1. 184
6.7 Surfaces fitted by RPT for Object #1. 185
6.8 Graph formed by GEOGRAM of Object #1. 185
6.9 Typical scene for testing RPT for a planar surface bounded by jump discontinuities 190
6.10 Typical scene for testing RPT with a 2nd order surface bounded by jump discontinuities. 190
6.11 Typical scene for testing RPT with two planar regions separated by a crease discontinuity. 191
6.12 Typical scene for testing RPT with curvature discontinuities. 191

A.1 Typical measurements and processing results generated for a planar surface bounded by jump discontinuities. The measurements, X-Strips, Y-Strips and final fitted surface are shown in (a)-(d), respectively. The original measurements contained zero-mean Gaussian noise with σ_m = 0.010 in. and spike noise occurring with a probability of 5 percent. This synthetic scene had a SNR of 7db. . . 219
A.2 Typical measurements and processing results generated for a second order surface bounded by jump discontinuities. The measurements, intermediate results and final fitted surface are shown in (a)-(d), respectively. This synthetic scene had a SNR of 5db. 220

A.3 Typical measurements and processing results generated for a two first order surfaces joined with a crease discontinuity. The measurements, intermediate results and fitted surfaces are shown in (a)-(d), respectively. This synthetic scene had a SNR of 13db. 220

A.4 Typical measurements and processing results generated for an object containing curvature discontinuities. The measurements, intermediate results and fitted surfaces are shown in (a)-(d), respectively. This synthetic scene had a SNR of 15db. 225

B.1 Object #2. 227

B.2 Measurements captured by PRIME of Object #2. 227

B.3 Surfaces fitted by RPT for Object #2. 228

B.4 Graph formed by GEOGRAM of Object #2. 228

B.5 Object #3. 229

B.6 Measurements captured by PRIME of Object #3. 229

B.7 Surfaces fitted by RPT for Object #3. 230

B.8 Graph formed by GEOGRAM of Object #3. 230

B.9 Object #4. 231

B.10 Measurements captured by PRIME of Object #4. 231

B.11 Surfaces fitted by RPT for Object #4. 232

B.12 Graph formed by GEOGRAM of Object #4. 232

B.13 Object #5. 233

B.14 Measurements captured by PRIME of Object #5. 233

B.15 Surfaces fitted by RPT for Object #5. 234

B.16 Graph formed by GEOGRAM of Object #5. 234

B.17 Object #6. 235

B.18 Measurements captured by PRIME of Object #6. 235

B.19 Surfaces fitted by RPT for Object #6. 236

B.20 Graph formed by GEOGRAM of Object #6. 236

B.21 Object #7. 237
B.22 Measurements captured by PRIME of Object #7. 237
B.23 Surfaces fitted by RPT for Object #7. 238
B.24 Graph formed by GEOGRAM of Object #7. 238
B.25 Object #8. 239
B.26 Measurements captured by PRIME of Object #8. 239
B.27 Surfaces fitted by RPT for Object #8. 240
B.28 Graph formed by GEOGRAM of Object #8. 240

C.1 Typical scene used to test RPT in the presence of jump discontinuities bordering a planar surface. 242
C.2 Measurements collected by PRIME of a planar surface. 242
C.3 X-Strips formed by RPT across a planar surface. 243
C.4 Surface fitted by RPT to a planar surface. 243
C.5 Typical scene used to test RPT in the presence of jump discontinuities bordering a 2nd order surface. 244
C.6 Measurements collected by PRIME of a 2nd order surface. 244
C.7 X-Strips formed by RPT across a 2nd order surface. 245
C.8 Surface fitted by RPT to a 2nd order surface. 245
C.9 Typical scene used to test RPT in the presence of a crease discontinuity separating two planar surfaces. 246
C.10 Measurements collected by PRIME for tests with crease discontinuities. 246
C.11 X-Strips formed by RPT across a crease discontinuity. 247
C.12 Surface fitted by RPT bordering a crease discontinuity. 247
C.13 Typical scene used to test RPT in the presence of a curvature discontinuity. 248
C.14 Measurements collected by PRIME of surfaces separated by curvature discontinuities. 248

C.15 X-Strips formed by RPT across a curvature discontinuity. 249
C.16 Surfaces fitted by RPT to regions separated by curvature discontinuities. 249

CHAPTER 1

Research Overview

1.1 Brief Problem Statement

There are three main points of focus in this dissertation, all of which address important aspects of the object recognition problem. These are: **adaptive range segmentation, graph matching** and **calibrated 3-D ranging**. Computationally efficient, robust and accurate means to accomplish these types of problems are critical to improving the state of the art in object recognition. Novel contributions have been made in each of the above three areas in the course of this research. These contributions take the form of algorithms, implementations and a significant level of testing.

In this work, a "computationally efficient" process is considered to be one requiring a worst-case amount of effort that is deterministic and is bounded by a polynomial function of the problem size. Herein, the terms "robustness" and "stability" will be used interchangeably to describe the consistency of results. This refers to the amount of variation in output data that is caused by relatively minor changes in input noise or input conditions - such as the pose of an object.

In addition to the above three main areas, a fourth objective was also set. It was desired to create components using a systemic perspective, allowing them to be put into an integrated framework. Their ability to be integrated into a system is considered to be an important aspect of their real-world utility. The three main contributions were integrated into a complete system. This testbed is not meant to represent a final turn-key implementation. Rather, its implementation permits

validation of individual components in a system context.

Because of the importance of the systemic perspective in this effort, the integrated framework itself has been given the name **RDM**, which stands for Recognition by Direct Matching. This name emphasizes the unique contribution in the graph matching area which permits the matching of two graphs to be performed in a direct and computationally efficient manner. This aspect of the framework tends to provide the greatest distance between an RDM style of system and other kinds of approaches. For example, RDM does not use an approach based on search [1–3], nor on iterative relaxation techniques [4,5], nor on alternate exhaustive formulations [6,7], as is typical.

The following system-level goals were set for the RDM framework:

- Able to recognize 3-D objects from different views.
- Able to handle partial views.
- Tolerant of occlusions.
- Able to recognize a relatively large number of objects.
- Able to recognize objects with complex geometries.
- As rapid, reliable and accurate as possible.

These goals served as the systemic guidelines during the creation of each of the three main contributions in this effort.

1.2 Design Considerations and Style of an integrated Framework

A great amount of research has been done in the area of object recognition for over 30 years [8] [9]. The above goals were used as a guideline to focus on an appropriate style of approach. RDM is model-based. It uses a graph

representation to model objects. These graphs contain descriptions of geometrical models and their relational postures. A discussion of each of these choices follows.

The problem domain associated with recognition in a manufacturing environment is more structured than that of an outdoor environment, for example, and typically a model-based approach is used in this arena [10]. This type of approach is appropriate because there generally are a limited number of objects that are to be recognized and often these objects may be precisely specified in terms of dimensions and fixed tolerances.

A important choice made in the RDM framework is the selection of the *type* of representation. Representations are often grouped into 2-D, 2 1/2-D, and 3-D categories [11] [10]. The 2 1/2-D style is typically used for describing 3-D objects that have been observed from a single viewpoint [12]. The 3-D type usually contains volumetric models [13] or surface descriptions [2]. 2-D approaches are more appropriate for intensity image processing [14] and were not considered for RDM.

View-centered and object-centered are two other common categories for representations [15]. Object-centered descriptions are independent of viewing direction. The preferred method of recognizing 3-D objects from arbitrary views is to use a 3-D representation with object-centered description [16]. This type of approach was selected for RDM. RDM constructs graphs that contain volumetric models and include a relational description of the relative pose of adjacent volumes. The type of representation used in RDM is also termed "exact" in nature [10]. It uses a single model of an object that describes all views.

In the RDM framework importance is placed on the type of object representation: 3-D, object-centered, exact and graphical. Less emphasis is placed on specific features within the representation. From the RDM perspective, the specific assignments made to nodes, edges, and their properties should be left somewhat up to a system designer and are likely to vary from application to application.

Because of the goal to recognize 3-D objects, it was chosen to use an active sensing technique. An active sensing mechanism [17] was preferred for increased tolerance to ambient lighting. This type of method also provides direct range measurements that can be acquired faster and with higher data densities than passive ranging methods.

1.2.1 Challenges of an Integrated System

Various types of noise and errors are introduced during the recognition process. The ranging sensor will often produce noise of multiple types, both lower intensity Gaussian-like corrupting every measurement as well as a higher intensity spike noise that will typically occur with a lower frequency [18]. The effect of measurement noise propagates up into each stage of the recognition process.

Segmentation and object modeling are intermediate stages in the recognition process. These involve the formation and description of discrete groups of points. Measurement noise and outlier effects produce the greatest challenges to segmentation and modeling algorithms [11]. Outliers are data points that are improperly grouped into a segmented region. These are due to sensor noise and to improper boundaries of regions. The net effect of noise and outliers is a loss of robustness, resulting in unstable scene descriptions. Here, "instability" refers to conditions where small changes in measurements cause large changes in a high level model. An object recognition technique will generally need to tolerate these effects to a certain degree.

The use of object models that do not match the true object shape can also lead to instabilities. For example, if polygons are used to describe a cylinder then slight changes in pose or in measurement noise may cause large shifts in the boundaries of each polygon.

Because ranging sensors are line of sight, scene models formed from a single observation point will generally only describe part of an object. Occlusions

of scene elements may be produced by the object itself and/or by scene clutter. This leads to two possible avenues: either deal with the occlusion directly, by identifying partial matches between scene and database models [19], or eliminate the occlusion problem by combining scene information from multiple views until a (near-)complete model is created [20]. The partial matching avenue has been addressed in this effort. This approach is challenging because of the high combinatorics associated with graph matching [21].

To lessen this computational burden approaches use a variety of methods to restrict possible matches (see Chapter 5). For example, the compatibility of node and edge properties must be maintained [22], constraints describing the permissible types and the local structures of neighboring nodes are also used [23, 24] as well as likelihoods of the occurrence of a node [7,25], or local groups of nodes [26]. These are a few examples of different classes of techniques, which, like RDM, focus on graph matching as a separate problem from pose. What is revealing, is to consider the *highly localized* nature of these matching strategies. The compatibilities, constraints and likelihoods involve either the nodes themselves or relationships to their immediate neighbors. These strategies typically operate with a very short horizon, often just 1 node away.

As a result of this limited horizon, techniques will invariably suffer as graphs become larger. As more nodes are available, more candidate matches are available, hence the dynamic range or the dimensionality of property vectors must increase to provide sufficient distinction to guide the matching process. Researchers have reported difficulties extending techniques to conditions with more complex objects [3].

The method of graph matching used in RDM computes a vector describing the topological relationship of nodes and edges to the rest of the graph. This vector reflects the graph's structure within a region of selectable extent. Hence the horizon of this vector is tunable - to include the entire graph, if desired.

1.2.2 Benefits of the New Framework

A key difference in the RDM framework is the separation of recognition and pose efforts. This has a number of advantages. First of all, it eliminates the step-wise refinement of pose calculations (as in tree search approaches [2,27]). Rather, in RDM, pose can be calculated once and in an optimal (least squares [28]) fashion.

Separation of recognition and pose also opens a door of opportunity to solve another significant problem in range processing - the merging of scene data from differing views. In the RDM framework scene descriptions contain only relative data. This provides an opportunity to use the graph matcher developed for RDM to find the mapping between adjacent (overlapping) scene graphs. These overlapping scene descriptions could then be merged. This operation would take place at a high level, providing a computational benefit.

Using a graphical representation with purely relational data also has a future benefit in the area of recognizing articulated objects. An articulated object, such as a prismatic joint of a robot, maintains a known topological relationship amongst its components, although relative positions may change. The connectivity of such an object remains constant. Hence recognition strategies that employ comparisons of graph topology, as in RDM, will be able to accommodate articulated shapes.

The use of relational features for graph properties has an advantage unto itself. Absolute calibration factors can be eliminated from the acquisition process of relative measurements. This provides an opportunity to use RDM in a weakly calibrated ranging system.

An important difference between RDM and other methods of recognition is the computational efficiency gained by its method of graph matching and an ability to find subgraph matches. Studies [29] demonstrated that recognition rates above 95% could be maintained when only half of the object was in view. This has an important impact on the type of object representation that can be used - permitting the use of an exact one, as discussed above. This has a significant

computational advantage. This style of representation is more compact than non-exact, or multiview representations. Multiview schemes [30] store a separate description for each possible view of an object. This results in both a large number of models and introduces increased computational requirements, as each view-dependent model must be examined separately during recognition.

1.3 The RDM Framework

The overall approach to RDM-based recognition is summarized in this section. Details on each component are given in Chapters 3 to 5. Reviews of related techniques are presented in Chapter 2.

RDM system components are illustrated in Figure 1.1. This is a data flow diagram that shows the individual processing steps and the data exchanged between each module. The data flows between modules are indicated with arrows, beginning with the "scene" at the top and an object description at the bottom. The bulletized items to the right provide descriptions of the processing modules.

The RDM framework includes the following components:

- "PRIME:" To Acquire Range Points
- "RPT:" To Segment Points into Regions
- "GEOGRAM:" To Model Each Region
- "DCA:" To Match Scene and Database Models
- "Horn:" To Compute Pose of Scene Object
- "Verify:" To Check the Validity of a Match

Each of these components is introduced below.

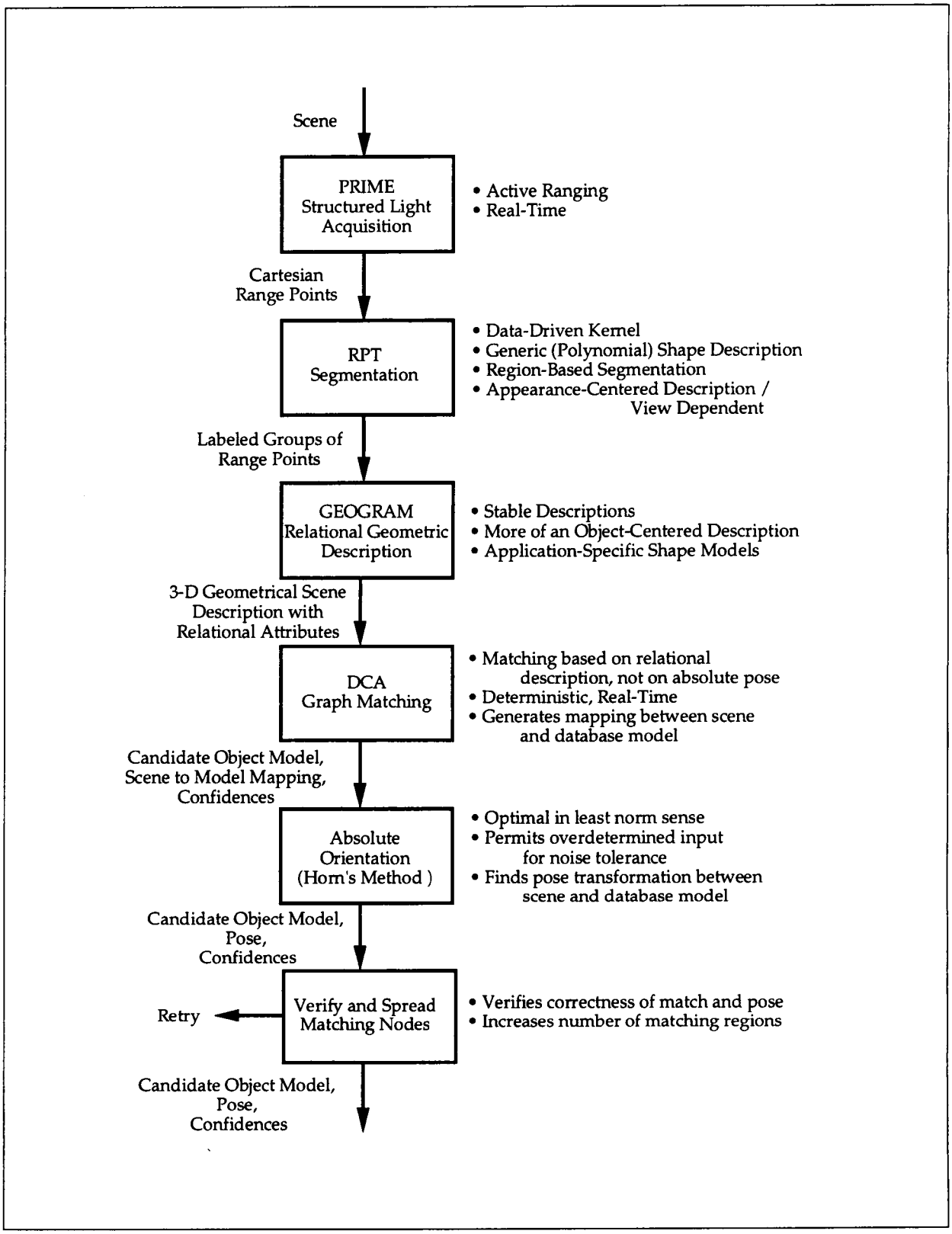


Figure 1.1: Components of the RDM framework for object recognition.

The **PR**ofile **I**maging **M**odul**E** (PRIME) is a Structured Light ranging sensor. PRIME generates profiles, or slices, of Cartesian range data. The novelty of PRIME stems from new approaches to calibration and to pixel-level operations. These techniques make for a system that is acquires data at frame rate, is accurate and easy to calibrate. See Chapter 3. The need for real-time acquisition has inspired some efforts involving custom VLSI implementations [31]. Here, real-time acquisition has been achieved without custom hardware, but rather, by mapping an algorithm onto a pipelined processor. The calibration technique that has been developed permits both the camera optics and laser geometry to be modeled in a single step via a least norm solution. This is simpler and can provide greater accuracy than more traditional methods that model the camera and laser separately [32]. PRIME has an accuracy of better than 0.005 inches.

The raw range data acquired by PRIME lacks any type of higher level interpretation, it is simply a collection of individual points. The next step in modeling and recognition is to form isolated groups of the raw data which can then be annotated with somewhat higher-level descriptions. This initial process of forming groups is referred to as "segmentation". **R**ange **P**rofile **T**racking (RPT) is an adaptive approach for robust, accurate and computationally efficient segmentation of range images that is based on Kalman filtering. It is well matched to the PRIME Structured Light acquisition process. See Chapter 4.

The spectrum of reported range segmentation techniques can be described in terms of the "kernels" that are used to access range data at the pixel level. The kernel has an associated size, shape, placement and feature. A new trend in range image segmentation is the use of adaptive kernels. Adaptivity improves the robustness of segmentation algorithms by better handling outliers. Adaptive schemes vary in approach, some change kernel shape [33], some use prudent kernel prudent positioning [34], and others vary processing methods or parameters [35].

The uniqueness of RPT arises from a 3-way combination of adaptive kernel mechanisms. The size and shape of the kernel vary dynamically. This provides dense regions of varying size that are robustly formed. Kernel boundaries occur when outliers disrupt surface smoothness, at object boundaries or in the event of spike noise, for example. In this type of approach the kernel boundaries are not located by a fixed geometry. This helps to avoid the inclusion of outliers within a segmented region. The state of the RPT processing kernel varies with the measurements encountered. It provides a prediction of future curvature in order to better eliminate outliers from each region. Tests demonstrated that segmented regions can be computed in less than 3 seconds, under a wide variety of test conditions. Stable operation was achieved above 90% of the time in the majority of testing trials.

The purpose of GEOGRAM, **GEO**metrical **GRA**phical **M**odeling, is to generate graphical models from the segmented regions produced by RPT. In general, these models have node properties that describe a geometrical shape, and edge properties that describe the relative orientation of the shapes that are located on adjacent nodes. The challenge of GEOGRAM is to produce components of object graphs that are more stable than the descriptions formed by RPT. An approach to the geometrical modeling problem has been implemented that reduces the number of unknowns in the models in order to improve efficiency. See Chapter 6.

GEOGRAM will produce scene descriptions in graph form. DCA is a new approach to finding subgraph isomorphisms. DCA performs a **D**irect **C**lassification of node **A**ttendance to describe the likelihood of a node being present as an isomorphic member of another graph. Based on this classification a node to node assignment is made between a scene and database graph.

DCA processing represents a tremendous computational advantage on established methods that are either based on search [1–3], or on iterative relaxation techniques [4, 5], or on alternate exhaustive formulations [6, 7]. DCA requires

a computational effort that is of polynomial order. It is approximate in nature. However, it is able to maintain a performance level for subgraph isomorphisms at or above 95% under a wide variety of conditions and with varying levels of noise. The performance on the full size comparisons associated with graph isomorphisms has been found to be 100/100, also under a variety of conditions. See Chapter 5.

The block "Horn's Method" in Figure 6.4 is a published technique for finding pose [28]. The method allows for an overdetermined specification of the transformation and finds a least norm solution. This technique meets the requirements of the system and does so in an optimal fashion. An iterative version of this technique has been implemented. It performs a set number of iterations (3) with reweighting to remove outliers. The residual fitting error of the position of each matched geometrical model is used as a final verification of the recognition and pose processing at the "Verify" stage.

Careful thought went into test conditions for individual components and for integrated tests. Types of surface curvature and discontinuities were important considerations, as were the complexity of objects and the number of database models. In more recent articles on range image processing [11,33,35,36] three types of surface discontinuities have been used: jump, crease and curvature. These have discontinuities in height, slope and slope derivative, respectively. Both first and second order surface curvature was studied herein also. Tests were performed with simpler objects having one type of surface and one type of discontinuity, other tests used more complex objects having mixed types. Testing was performed using both synthetic and real data. The tests with synthetic scenes were necessary because they provided ground truth. The sensor noise level was also ramped up in these studies to examine the stability and accuracy of system components. The objects used in the integrated tests were relatively complex, compared to those used in other range-based end-to-end systems that have been

reported [3,37,38].

1.4 An Application Domain in Future Manufacturing

The application domain for this research is advanced manufacturing. It is envisioned that the future of manufacturing will move away from large scale, mass production. Growing demands for more flexibility and more customized manufacturing runs will change the nature of heavy industry. Production facilities that can achieve greater flexibility will outpace traditional competitors in this new market place by offering greater varieties of goods that better meet customer needs.

Flexibility in the manufacturing process will be enhanced by allowing more diverse elements to contribute in a single production run. This can be achieved more cost effectively by integrating contributors that are geographically distributed, rather than attempting to pull together an ever-diversifying suite under one roof. Better manufacturing solutions will involve distributed components that can link into a process run for a short period of time, contribute, and then break off into new efforts with new collaborators. This style of process is termed "**A**dvanced-**D**istributed **M**anufacturing" (ADM).

In order to better describe the potentials of ADM, two design scenarios are presented in Figures 1.2 and 1.3. These scenarios describe parallel design efforts in the real and virtual worlds. It is envisioned that the ability to design in each of these worlds will provide a synergistic benefit, particularly when a graceful transition between these domains is possible. This will allow iterative prototyping efforts to take place while capitalizing on the advantages each domain provides. For example, components from geographically distributed locations can come together in one virtual environment. A virtual environment can also combine prototype and existing parts in one assembly.

1) Gather solid models in Virtual World:
   a) By Scanning a prototype form
   b) By creating a model manually
   c) By recall existing models

2) Assemble models in Virtual World

3) Use visualization and simulation capabilities of Virtual World:
   a) For concept verification
   b) For clearance checking in dynamic assemblies
   c) In magnified / pseudocolor views
   d) For observing kinematics at reduced speed
   e) For observing component properties that can not be directly perceived, such as the cone of acceptance for an optic fiber
   f) For observing parts under vibration
   g) For observing parts undergoing temperature changes

4) Modify virtual environment in dynamic simulations:
   a) Alter temperatues
   b) Alter stiffness of prototype parts
   c) Alter viscosity of fluids

5) Modify design in Virtual World

6) Use Rapid Molding, switch to Real World

Figure 1.2: Designing in the Virtual World permits visualization and assembly of prototype elements. Visualizations can be augmented with attributes of components that are otherwise difficult to perceive. A virtual assembly can be accomplished with parts from distributed locations.

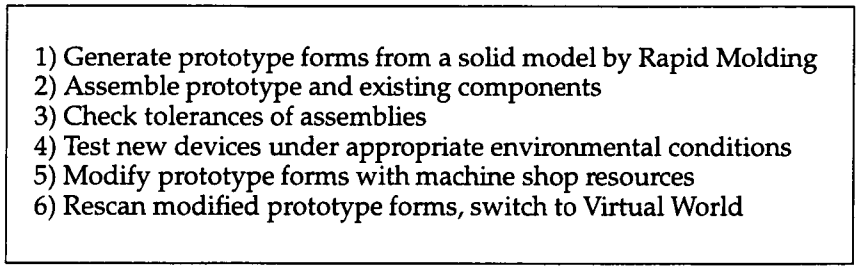


Figure 1.3: Designing in the Real World permits in-hand modification of parts. Scanning modified parts permits design efforts to switch back to the Virtual World. Changes in the geographic location of design teams can accompany transitions between the Real and Virtual Worlds.

A theme in the ADM scenarios is parallelism between the real and virtual worlds. The most benefit in these scenarios will be achieved when design and assembly are possible in both of these worlds with graceful transitions. Scanning and rapid molding are key technologies that will permit this crossover between the two worlds. "Scanning" refers to a process of sensing, modeling and recognition, which is the focus of this research. "Rapid molding" is a complementary capability permitting a transition from the virtual environment back to the real world. This involves creating solid forms from solid models. Early versions of this type of system exist. The purpose of these devices is to create forms rapidly and inexpensively, perhaps with somewhat lower tolerances. Current molding devices fabricate plastic components.

1.5 Keywords of the Problem Domain

The purpose of this section is to aide in describing the focus of this effort through the use of keywords. A brief discussion of each keyword is included.

1.5.1 Active Sensing

This type of sensing refers to range measurement techniques that use active sources, such as lasers, to illuminate scenes in predictable manners. The interaction of the source illumination with objects in the scene is observed and used to compute range measurements.

1.5.2 Adaptive Range Segmentation

Segmentation refers to the formation of groups of range points. These groups are typically formed using measures of consistent curvature and provide a basis for higher level descriptions of scenes. Generally speaking, segmentation is achieved by accessing small portions of the range data and grouping consistent regions.

The spatial pattern of access is critical to the success of an approach. Adaptive segmentation is a new trend in range image processing that accesses data in spatial patterns that are not constrained by an artificial, preset shape; these methods allow the spatial patterns to vary dynamically in a data-driven fashion. Processing parameters are also adaptive in these types of approaches.

1.5.3 Model-Based Recognition

A model-based approach to object recognition uses observed data to build a scene model and then compares this against a number of models in a database. This permits portions of scenes to be grouped and identified as instances of known objects and permits object locations to be determined.

1.5.4 Exact, Object-Centered, 3-D Representation

An object representation refers to a specific method of forming numerical descriptions of objects. An exact 3-D representation is one inwhich the object is described using a set of 3-D models, and, there is a one-to-one correspondence between model elements and object elements. This type of representation is also referred to as object-centered, as opposed to view-centered. In a view-centered representation, a separate description is stored for each possible view of an object.

1.5.5 Graph Matching

A graph is an abstract structure describing the interconnectivity of discrete elements. Graphs can also contain application-specific descriptions of each element and of adjacent elements. Graph matching refers to the process of finding a mapping between two graphs that both preserves adjacency relationships and forms compatible assignments in terms of the application-specific descriptions in each graph.

1.5.6 Integrated System Perspective

In order to yield contributions that have the most possible real world utility, an integrated systems perspective was considered essential. This type of perspective brought about the design of a system of components that are "well-matched" in nature.

1.6 Concluding Remarks

There are three main points of focus in this dissertation, all of which address important aspects of the object recognition problem. These are: adaptive range segmentation, graph matching and calibrated 3-D ranging. Computationally efficient, robust and accurate means to accomplish these types of problems are critical to improving the state of the art in object recognition. A systemic perspective is a fourth goal that transcends each of these. This objective will help to improve the real-world utility of the three main efforts by making them better able to be fit into an integrated framework. In addition to the creation, implementation and testing of each of the three main contributions, a testbed was developed that permitted integrated testing. This helped to validate the main contributions.

Novel contributions have been made in each of the above three areas in the course of this research. These contributions take the form of algorithms, implementations and a significant level of testing. The next chapter has a literature review of the three main areas addressed in this effort.

CHAPTER 2

Literature Review

There are three research areas that are relevant to the body of work covered in this dissertation. These are: range image segmentation, graph matching, and calibrated 3-D range acquisition. Each of these subjects is reviewed in this chapter. Taxonomies have been defined that serve to distinguish existing methods with the novel contributions of this research effort.

2.1 Literature Review of Range Segmentation Techniques

This review covers a variety of techniques for segmenting range images. A new trend in this area is the evolution of techniques which are adaptive in their processing. Adaptive methods are not bound by rigid, artificial geometries when processing range points. These techniques use more dynamic processing schemes. This provides a significant advantage by reducing outlier effects during processing.

2.1.1 A Taxonomy for Range Segmentation

Range image segmentation has been an active area of research, for the past 15 years [39] [40]. A number of good survey articles [11] [41] are available. This review effort attempts to gain critical insight into range segmentation by defining a new taxonomy. This taxonomy exposes aspects of a technique which are important to the accuracy, tolerance to outliers and the efficiency of an approach.

This taxonomy has three distinguishing criteria that describe the "kernel"

associated with a given technique. The kernel is a local neighborhood of range points that is processed as a group inorder to measure local curvature, the term "footprint" is analogous. The kernel is described in terms of its area, placement and the processing performed on its data.

1) **Kernel-Real Estate**: This refers to the spatial pattern associated with the pixel level [42] (or low level [43]) access of range data. This determines **how much** data is processed. A standard 2-D convolution kernel is one example [40,44,45]. Other approaches use 1-D convolution [2]. Some techniques permit Kernel-Real Estate to grow dynamically [33]. The size and shape of this spatial region effects the ability of a technique to represent curvature accurately and robustly.

2) **Kernel-Placement**: This specifies **where** and **when** the kernel is applied to range data. Some methods locate a kernel in an orderly raster scan [44]; others operate randomly [34,46,47]. Some techniques are single-pass [48], others are iterative [35]. Placement schemes are also critical for success. Rigid placement strategies suffer from an inability to avoid outlier effects because the rigid schemes inevitably locate kernels in positions that span surface boundaries. This tends to introduce outliers at the lowest level of processing which contaminate the initial region descriptions and must be handled with further processing.

3) **Kernel-Feature**: The Kernel-Feature specifies **what** quantitative means is used to characterize the data within the kernel region. Some techniques extract invariant curvature measures [44], others form surface normals [49], for example. This feature provides the fundamental, local characterization needed for segmentation. This may be in the form of a uniformity measure for region-based techniques [27], or in the form of a discontinuity measure [35], for edge-based methods. Robustness and accuracy are key issues when considering Kernel-Features [36,50,51].

All three of the above criterion are further be qualified by their degree of **adaptivity**. Adaptive Kernel-Real Estate and placement provides a capability to

selectively avoid regions that contain outliers, either by changing shape or by prudent positioning [34]. Adaptive Kernel-Features alter processing parameters or heuristics in a data-dependent fashion. Markov Random Field-based methods operate in a state-based manner that converges during processing [35].

The techniques reviewed below are grouped by the size and shape of their kernel-real estate and by their adaptivity. Algorithms are included from the realms of range segmentation and surface characterization. These areas are both included because of some overlapping objectives that exist in the segmentation domain.

2.1.2 1-D Nonadaptive Kernels

A linear edge-based approach to kernel-features was used in [52]. Edge detection was achieved by locating points where significant changes in surface curvature occur. This was accomplished using four 1-D kernels, oriented 45 degrees apart that were applied to multiscale versions of range data. Critical points were identified by either zero-crossings or extrema in curvature measures and were tracked from coarse to fine images for improved localization. Some edge linking was then performed followed by region growing within the edge boundaries. Finally a biquadratic surface was fit to each region. In this study, Gaussian smoothing is performed prior to curvature estimation. The smoothing kernel varied depending on noise content from ≈ 3 to 18 pixels wide.

The curvature measure used above was based on estimates of the first and second derivative in each direction. The first derivative was approximated using a first-order difference. The second derivative was also found via a simple difference, operating on the first order results. The multiscale processing compensated for the noise inherent with taking simple differences. Limited work has gone into the pursuit of multiple 1-D kernels, which may be due to the necessity for a merging step to combine the 1-D results into regions having 2-D lateral extent.

2.1.3 Large 2-D Nonadaptive Kernels

Differential Geometry provides a Gaussian and mean curvature measures, denoted H and K, that are useful kernel-features for characterizing local surface curvature [44]. This type of technique uses estimates of the first and second order partial derivatives to find the H,K curvature values [53]. These measures are invariant with respect to translation, rotation and scaling. The signs of H,K can be used to classify a point into 1 of 8 surface categories. An image can then be formed based on this classification and used as a basis for region building.

The difficulty in using this type of approach is the sensitivity to noise of the estimates of the partial derivatives. Many approaches have been used to compensate for this problem. In [48] a minimum window size of 9x9 was needed due to the effects of 8-bit quantization noise (this is a 59 db SNR for the input). Noise compensation was achieved using a larger window. The technique uses a weighted combination of 7x7 kernels to yield an effective size of 13x13. Regions are initially established using an H,K-sign image. These estimates were then improved via region growing.

The study in [40] reports on extensive experimentation with the H,K technique. Range data was smoothed in a preliminary step using a Gaussian function with σ = 3. (It is assumed this corresponds to a truncated Gaussian with an 18x18 kernel). A minimum-sized 13x13 kernel for curvature estimates is reported for reliable H,K classifications. The fragility of using these measures as a plane detector is also described. In this case, $H \approx K \approx 0$. Hence thresholds are required to isolate this near-zero condition. A two pass approach was recommended that first isolated planar regions and then segmented the remaining curved surfaces. The approach in [54] uses a similar 2-pass technique.

2.1.4 Small 2-D Nonadaptive Kernels

The large kernel-real estate described above for H,K methods has an associated computational burden which is significant. In [45] a biquadratic surface patch is fit to range data within a smaller 5x5 kernel. Provided the residual error of the patch is below a set threshold, the partials of the fitted surface are computed using the patch coefficients and form both an H,K-sign image and two edge detector images. A hybrid approach is then used to combine these three images into a final segmented result. Problems using the biquadratic fit in this small kernel-real estate are described. Patches near discontinuities were sometimes fitted with a tolerable residual error, but that resulted in poor curvature estimates, because they contained points from different regions of curvature. This is a fundamental problem when a fixed geometry is used for a kernel-real estate.

Another alternative to the large kernels described previously was explored in [50]. Quadratic B-splines and the Discrete Cosine Transform (DCT) were used to estimate local curvature. Superior results are reported using splines. The spline fit occurred in a 9x9 kernel. The partial derivatives found from the analytic surface were then used to find five invariant curvature features. The first feature was the magnitude of the gradient, computed directly from the partials. Other features required finding the direction in which the second directional derivative had its greatest magnitude. This was computed from the two eigenvectors of a 2x2 Hessian, formed using the B-spline's partials. The first and second directional derivatives in the direction of the two eigenvectors completed the set of five kernel-features.

Because of the high sensitivity to noise associated with the estimation of partial derivatives, other kernel-features have been studied for local shape measures such as covariance [51] [55]. The covariance measure described variations in surface normals that occurred across a 7x7 kernel. A method for finding edge features was also incorporated. Range points that were not classified as edges, were processed

using their covariance-based curvature measure. The method produced more uniform region labels, as indicated by an entropy-based consistency measure. However, it did not provide the rich classifications possible with the H,K kernel-features from Differential Geometry.

An intuitively satisfying approach to range image segmentation is to use surface normals for kernel features. In [49] range images were smoothed with a 3x3 neighborhood. Planes are then fit to 5x5 patches to find surface normals. Adjacent patches were clustered together based on a square error criteria. Regions were formed from each cluster and classified based on curvature. Adjacent regions having compatible curvature were merged. In this approach, merging only occurs at the fixed kernel boundaries.

2.1.5 Adaptive Kernels

The technique developed in [35] was a hybrid of the clustering technique in [49] and an edge detection technique based on Markov Random Fields (MRF). The adaptive nature of this approach stems from the temporal state changes in the MRF that effect the Kernel-Feature processing.

The earlier clustering approach [49] nominally tended to oversegment regions. The MRF was used to help select valid region boundaries. The MRF was used to label each range point as either an edge or a non-edge. It contained a priori knowledge quantifying local smoothness and continuity. This involved both conditional probabilities describing the likelihood of edges and non-edges, and a spatial dependence. The spatial weighting was defined within a second order neighborhood. The kernel, defined by the superposition of these patterns, appears to have a 3x3 extent overall. The iterative labeling process began by assigning each range point a NULL label. Iterations continued until all pixels were assigned a label with a sufficient confidence, as determined by a neighborhood consistency measure.

The authors discuss the tradeoff between having smaller MRF neighborhoods and achieving better optimizations. Unfortunately smaller kernels are unable to quantify smoothness as well and this may be a limiting factor in an MRF-based approach. The technique is also described as being typical of edge-based segmentation methods, in that it does not always result in closed boundaries. It should also be mentioned that the MRF processing has a non-deterministic termination condition which makes it less suitable for real-time operations.

Another aspect of kernel adaptivity is the use of data-driven real estate and placement strategies. In this type of scheme, kernel regions are determined by the input data; they are not a fixed, uniform geometry. Here the kernel spreads into the range data without set boundaries. In this scenario, the concepts of a kernel and a segmented region can become much less distinct. A kernel's shape and the shape of a segmented region can be identical, for example. Also, the feature associated with a data-driven kernel can be the region's uniformity criteria itself. This type of adaptive approach allows the kernel region to grow as large as possible, without including outliers. This improves curvature estimation by providing a maximal set of uniform input data.

An adaptive kernel based on a Robust Sequential Estimator (RSE) was developed in [33]. This estimator was an extension to a sequential least squares formulation. New points were incrementally added to the kernel with a weighting determined by the point's residual error and by a non-Gaussian noise model. This noise model aided in the robust handling of spike noise.

In this study, a 7x7 region was used to begin each seed. The locations of these seed regions was fixed. This seed geometry was the only nonadaptive aspect of this technique. The kernel-feature was a patch model of the type $z = f(x, y)$. The form of the model was selected using a criteria that considers both its order and residual error. Using this method, higher order models were penalized because of their natural tendency to fit to the noise. This provided a selection criteria

between planar and biquadratic forms. Once a model type was chosen it was kept as the region grew. All points in the image were considered for inclusion in a given region. Some morphological operations complete the processing, along with a criteria for resolving the membership of points that were classified into more than one region. The growth of the kernel-real estate is considered to be adaptive because the weighted agglomeration of new points is controlled by the state of the kernel-feature. Kernel growth is also not limited by artificial size limits.

The RANSAC method of segmentation was developed in [34]. This technique used a sparse kernel-real estate that was randomly located. A pre-selected surface model was fit to the random locations and the whole data set was then examined for supporting evidence. Although this technique is nondeterministic, and hence not suitable for real-time applications, it is noteworthy because of its strength in tolerating non-Gaussian noise. This robustness is achieved by randomly shifting the kernel real estate until outlier are avoided.

Random kernel-real estate was also used in [47]. Processing was begun by performing a segmentation to locate jump discontinuities. This provided candidate regions which might contain a single type of curvature. Sparse kernel points were selected within the candidate region, choosing 3 for a planar fit, and 9 for a quadratic. The selection process was repeated a large number of times and the residual errors were histogrammed for each random kernel. A heuristic was then applied to each histogram to find the best kernel. Outliers were removed from the region and it was then isolated from the original data set. This process was then repeated with the next candidate region.

In this approach, the points that constitute the kernel-real estate were selected either purely at random or with the use of a Genetic Algorithm (GA). The GA operators modify the selection of kernel points, based on the performance measure obtained from the histogram analysis. GA-based kernel selection follows the work

of [46]. An improvement in speed by a factor of 2-3 is reported when using the GA. This technique is an interesting advancement because the real estate adaptivity is directed by the GA, rather than a purely random modification.

The above 2-D techniques operate on range data which has been acquired on a dense grid. This readily supports the use of functional approximations to surface patches of the form $z = f(x, y)$. These 2-D approaches benefit from a certain simplicity by ignoring the problem of combining data from multiple views. The approaches in [27] [22] address this problem directly. These approaches make no assumptions concerning the arrangement of input range data by using a process of tesselation known as the Delauney triangulation. This representation forms a nearest neighbor-style grouping of range points into triangles. The tesselation procedure works irrespective of the number of views contributing to the set of range data. A similar tesselation of 3-D data is reported in [56].

A Region Adjacency Graph (RAG) is used to group the triangular kernels into regions. Each RAG entry describes the residual error associated with the merger of an adjacent pair of triangles. The RAG is initialized so as to describe all adjacent triangles. The triangles having the best combined fit are joined first, and the merging process continues until either a single region remains, or until the total residual error for all regions exceeds a threshold. Both planar and quadratic fitting is performed. A similar approach involving the merging of adjacent 3-D patches is reported in [12].

The approach is considered to be adaptive because the Kernel-Real Estate and -Placement are determined by the nominal 3-D positions of the range data. Arbitrary boundaries are not imposed when merging is performed. Rather, the boundaries established between RAG nodes are dependent on the initial positions of measurements and on the uniformity criteria.

2.1.6 Qualitative Evaluation of Segmentation Techniques

In this section, the performance measures of speed, accuracy and robustness are qualitatively assessed. A number of fundamental tradeoffs are present in the above approaches, that affect these metrics.

Kernel-Real Estate is one example of an area for design tradeoffs. Larger kernels allow more accurate Kernel-Features to be computed, but require more processing and increase the likelihood of outlier contamination. Smaller ones tend to have an increased sensitivity to noise [52]. Adaptive techniques achieve an optimal balance because they allow size and shape to vary, in order to provide a maximal amount of range data for the computation of a Kernel-Feature without including outliers.

Kernel-Placement also has a significant effect on performance. Random kernel locations, for example [34], avoid outliers simply by retrying until a set of positions is found with good support. This strategy is robust, but nondeterministic and hence is not suited for real-time processing. Techniques using dense, regular-shaped kernels deal with outliers typically by eliminating seed regions with poor residual fits. This approach suffers because a single outlier can be hidden within a set of better-fitting points. This contaminated region might pass a test based on a residual error threshold, allowing the outlier to corrupt a local curvature estimate. Adaptive means are a benefit here also because they shift kernel boundaries to only include uniform regions.

Kernel-Features effect performance by their means to compute accurate curvature measures in the presence of noise sources. Methods involving simple differences to estimate partial derivatives [52] amplify all noise sources, for example. Adaptive means for computing features are beneficial because they can reduce the effects of outliers that may be included in a kernel region. The iterative least squares fitting of [33] permits robust region formation by weighting outliers.

Table 2.1 provides examples of these tradeoffs and their relative impact on

Table 2.1: Qualitative comparison of range segmentation techniques. Accuracy tends to improve with increase kernel area and with the use of iterative methods. Robustness is typically challenged unless an adaptive approach can be used to avoid outliers. The computational efficiency is related to kernel size and placement.

	Real Estate Density	Real Estate Size	Placement Location	Iterative Placement	Feature	Accuracy	Robustness	Computational Efficiency
[44]	Dense	Large	Fixed	No	Set	High	Inherent	Low
[53]	Dense	Large	Fixed	No	Set	High	Challenged	Low
[48]	Dense	Large	Fixed	No	Set	High	Challenged	Low
[40]	Dense	Large	Fixed	No	Set	High	Challenged	Low
[54]	Dense	Large	Fixed	No	Set	High	Challenged	Low
[45]	Dense	Large	Fixed	No	Set	High	Challenged	Low
[52]	Dense	Small	Fixed	No	Set	Low	Challenged	High
[50]	Dense	Small	Fixed	No	Set	Low	Challenged	High
[51]	Dense	Small	Fixed	No	Set	Low	Challenged	High
[55]	Dense	Small	Fixed	No	Set	Low	Challenged	High
[47]	Dense	Small	Fixed	No	Set	Low	Challenged	Low
[46]	Dense	Small	Fixed	No	Set	Low	Challenged	Low
[49]	Dense	Small	Fixed	No	Set	Low	Challenged	High
[27]	Dense	Adaptive	Adaptive	No	Set	High	Inherent	Low
[22]	Dense	Adaptive	Adaptive	No	Set	High	Inherent	Low
[34]	Sparse	Small	Adaptive	Yes	Set	High	Inherent	Low
[35]	Dense	Small	Fixed	Yes	Adaptive	High	Inherent	Low
[33]	Dense	Adaptive	Fixed	No	Adaptive	High	Inherent	High
[36]	Dense	Adaptive	Fixed	No	Adaptive	High	Inherent	High
RPT	Dense	Adaptive	Adaptive	No	Adaptive	High	Inherent	High

performance. The RPT technique, developed as a part of this effort, (last row) is the only method that is adaptive in terms of each of the Kernel-Real Estate, -Placement and -Feature categories.

2.1.7 Quantitative Metric for Segmentation Algorithms

A numerical rating, Q, has been defined for comparing segmentation algorithms. This measure includes the effects of speed, accuracy and robustness. The quality measure mandates good performance in each of these metrics with an equal weighting. Q is assumed to have the form

$$Q \sim (Accuracy)(Robustness)(Computational\ \ Efficieny). \tag{2.1}$$

To make Q as general and as simple as possible, many algorithmic-specific features were not considered. For example, accuracy is assumed to be related only to kernel area using

$$Accuracy \sim \left(1 - \frac{1}{Kernel\ \ Area}\right). \tag{2.2}$$

This represents the potential for a computation to form an accurate result by considering the amount of data available within the kernel. Specific methods of computation are not included in the measure. Note that kernels with adaptive real estate necessitate using a mean kernel area, as measured for a given scene.

The robustness measure considers the amount of outlier-contaminated area falling within kernels vs. the total amount of kernel area that is free of only sensor outliers (spikes). The robustness measure is defined as

$$Robustness \sim \frac{Total\ \ Kernel\ \ Area\ \ Unaffected\ \ by\ \ Outliers}{(Total\ \ Kernel\ \ Area) - (Total\ \ Spike\ \ Area)}. \tag{2.3}$$

Hence this measures the effectiveness of Kernel-Real Estate and -Placement strategies to avoid outlier contamination. As with the accuracy metric, this must be evaluated for a specific data set.

Computational Efficiency is estimated by examining the computational effort on a per pixel basis. This is approximated by comparing the total area of a range image to the total area of all the kernels applied to the image with

$$ComputationalEfficiency \sim \frac{Total\ \ Image\ \ Area\ \ Processed}{Total\ \ Kernel\ \ Area}. \tag{2.4}$$

The above relationships yield

$$Q \equiv \frac{A \sum in_i (\sum a_i - N)}{(\sum a_i)^2 (\sum a_i - S)} \tag{2.5}$$

where, N is the number of kernels, a_i is the area of the i_{th} kernel (in pixels) and A is the total area of the range image, also in pixels. The total area of all pixels contaminated with spike noise is S. Each $a_i = in_i + out_i$ where in_i and out_i are the number of inliers and outliers, respectively, within the i_{th} kernel. The overall quality rating has a range $0 <= Q <= 1$, as do each of the three subcomponents.

A simple, cartoon-like, example is presented to illustrate the rating and to compare RPT to the use of convolution based approaches. The scene constructed for this example is shown in Figure 2.1 and discussed in Table 2.2. This example scene measures 10x10 pixels ($A = 100$) and contains 5 percent spike noise ($S = 5$). Bold gray lines indicate the boundary between two regions. To compute a given value of in_i, outlier-free pixels were counted. Pixels containing spike noise were always considered outliers during this tally. When a kernel crossed the gray boundary line, outliers were defined as being the pixels in the minority region. The transparent boxes in the figure illustrate various instances of kernels. The values for $\sum a_i$ and $\sum in_i$ are given for each box.

The 1x1 case is interesting to consider. It has perfect robustness because a 1x1 kernel is never contaminated by outliers, since it contains only one point. It also has perfect computational efficiency because of its compact size. However, it has zero accuracy because a single point cannot classify local curvature.

Because of the adaptive nature of the RPT kernel, kernel regions are terminated at shape discontinuities and because of spike noise. This reduces the outlier content within RPT kernels, compared to an NxN convolution type of approach that inevitably includes outliers due to a fixed geometry. The least optimal factor with RPT in this case is computational efficiency. This results from the fact that two orthogonal passes are across the data set are required in RPT. The entry "RPT (1-D)" gives the results for one of these two passes.

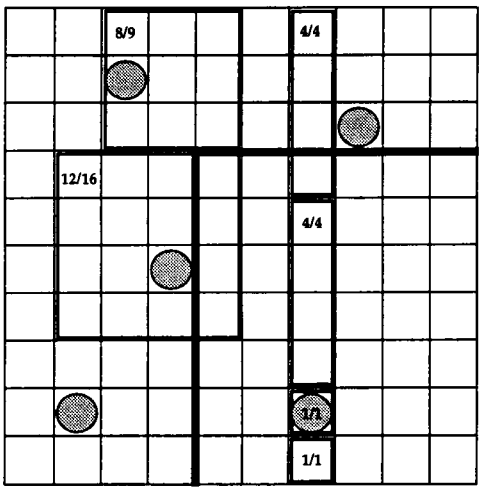


Figure 2.1: Cartoon scene illustrating the fundamental problem associated with nonadaptive kernels.

Table 2.2: Examples of Quantitative Metric for kernel evaluation. All examples had $A = 100$, $S = 5$.

Kernel	N	$\sum a_i$	$\sum in_i$	Accuracy	Robustness	Comp. Eff.	Quality
1x1	100	100	95	0.0	1.0	1.0	0.0
2x2	81	324	284	0.75	0.89	0.31	0.21
3x3	64	576	488	0.89	0.85	0.17	0.13
4x4	49	784	628	0.94	0.81	0.13	0.10
RPT (1-D)	26	100	95	0.74	1.0	1.0	0.74
RPT	51	200	180	0.75	0.92	0.5	0.34

2.2 Literature Review of Graph Matching Techniques from the Machine Vision Community

This review of graph matching has been divided into two sections, one covering work done in the Machine Vision Community, and the other addressing efforts from the field of Discrete Mathematics. Work in the Mathematics Community focuses on the matching (or isomorphism) problem directly. Graph matching is an important problem in Machine Vision, but it is not always addressed directly from a standpoint of topological analysis. Typically application-specific constraints are introduced, for example, that serve to limit the potential number of matches. This allows a technique to be used that is exhaustive in the worst case, but that yields acceptable performance when the number of nodes is limited.

2.2.1 The Importance of Graph Matching in Machine Vision

Object recognition has been studied in the Machine Vision Community of over 30 years [8,9]. At its core object recognition is fundamentally a problem of determining subgraph isomorphisms [22] *i.e.*, of graph matching. Some researchers use global models that describe an object in its entirety [57,58]. This type of approach leads to effective solutions for objects with simple shapes that are not occluded. This kind of capability is very different from the goals set herein, where considerations of occlusion is important and there is an interest in handling objects with complex shapes. The kinds of techniques in the Machine Vision community that address the graph matching problem are generally model-based. These are the types focused on in this review.

The need to match subgraphs arises due to the limitations on view of sensors. Because ranging sensors are line of sight, scene models formed from a single observation point will generally only describe part of an object. Occlusions of scene elements may be produced by the object itself and/or by scene clutter.

This leads to two possible avenues: either deal with the occlusion directly, by identifying partial matches between scene and database models [19], or eliminate the occlusion problem by combining scene information from multiple views until a (near-)complete model is created [20]. The partial matching avenue has been addressed in this research effort. This approach is challenging because of the high combinatorics associated with graph matching [21].

To lessen this computational burden approaches use a variety of methods to restrict possible matches. For example, the compatibility of node and edge properties must be maintained [22], constraints describing the permissible types and the local structures of neighboring nodes are also used [23, 24] as well as likelihoods of the occurrence of a node [7, 25], or local groups of nodes [26]. These are a few examples of different classes of techniques, which, focus on graph matching as a separate problem from pose. What is revealing, is to consider the *highly localized* nature of these matching strategies. The compatibilities, constraints and likelihoods involve either the nodes themselves or relationships to their immediate neighbors.

The extent of these comparison metrics is a revealing means to evaluate different techniques. The term "horizon" is used to describe this extent. This refers to the distance of neighboring graph elements that contribute to the comparison of a candidate node-to-node mapping between two graphs. A shorter horizon involves less computation, but can mislead the matching process because more localized comparisons between the graphs are performed. These strategies described above operate with a very short horizon, on the order of 1-3 nodes. Larger horizons include a greater portion of the graph (in terms of node and edge properties and topology) when considering a candidate mapping.

As a result of this limited horizon, techniques will invariably suffer as graphs become larger. As more nodes are available, more candidate matches are available, hence the dynamic range or the dimensionality of property vectors must in-

crease to provide sufficient distinction to guide the matching process. Researchers have reported difficulties extending techniques to conditions with more complex objects when methods having limited horizons, 3 in this case, are used [3].

2.2.2 Taxonomy for Graph Matching

Because of the importance of graph matching in Machine Vision, many types of techniques have been pursued [10, 11, 59]. Despite the fact that the underlying problem to recognition is subgraph isomorphism, the problem is not typically addressed as such. A more common taxonomy [19, 22] uses three categories: maximal clique-based, relaxation labeling and tree search-based approaches. This breakdown of methods will be used here and augmented using the horizon concept.

Some researchers that have confronted the recognition problem directly from a standpoint of graph isomorphisms include [60, 61]. In the early work of [60] a variety of kinds of subgraph isomorphisms were identified that involved various subsets of nodes either in the scene or database graph. Matching was performed using tree search through attributed graphs. Here a core matching subgraph was identified through exhaustive means and the extended into a larger mapping through a method termed "hierarchical synthesis".

In [61] a graph is formed based on a Hough space analysis. Matching is then performed using an augmented version of a technique established in the Mathematics community [1]. These extensions involve the comparison of node and edge attributes in the graph. The original algorithm also used an exhaustive exploration of a search tree. This was modified to limit the enumeration later in the search in favor of more wandering at the higher levels in the search tree. A significant improvement in performance is described. A verification stage was also added that uses the consistency of neighboring features to eliminate some false matches.

2.2.3 Maximal Clique-Based Approaches

This type of technique builds an association graph which describes all possible compatible mappings between two graphs [6] [62]. Compatibility can be based on node and edge properties and can include a variety of geometrical and topological relationships. The association graph is then searched for a maximal clique [21]. This clique represents the largest possible compatible mapping between the two graphs. The general problem of finding a maximal clique is NP-complete [63]. Hence, this type of approach ends up trading off one NP-Complete problem for another. Exact methods exist for finding maximal cliques that are recursive and use depth-first search [64]. Here a tree that can lead to all possible cliques is searched. Pruning occurs when a branch in the search is found that cannot lead to a clique.

More approximate methods for finding maximal cliques have also been explored, via a Markov Random Field (MRF), for example [7]. In this particular work, the low level acquisition and processing of the sensor were to form likelihoods of observing a given object feature. These likelihoods are used as a compatibility measure in the association graph. The MRF determines memberships in cliques. The MRF operates on each edge on the association graph by either including or excluding it from a clique. Mutual consistency of neighboring clique entries guides the MRF convergence. In this case, the horizon is dictated by the neighborhood arrangement of the MRF. Here these neighbors were limited to two.

Some techniques combine aspects of maximal-clique based approaches and tree search. In [65] an approximation to a maximal clique is found by looking for a high number of occurrences of mutually consistent sets of features that all include a given node. This node is then used as a seed for a tree search operation. In this case the horizon for finding the seed match is determined by the neighborhood of the local feature sets, which is three.

The association graph itself can be troublesome in these approaches. Given the graphs $G^{(1)}$,$G^{(2)}$ with $N^{(1)}$,$N^{(2)}$ nodes, respectively, the association graph A_g contains a node for each compatible pair of nodes in $G^{(1)}$ and $G^{(2)}$. This necessitates that a threshold be applied to some application-specific norm that measures the distance between the node and edge properties of $G^{(1)}$ and $G^{(2)}$. The threshold is a truncation of data, the effects of which cannot be recovered from (other than by repeating with a different threshold). The threshold can eliminate any possibility of including a given pair of nodes in the final clique. Hence it is somewhat of a devastating operation since it is performed first, during the formation of A_g.

2.2.4 Relaxation Labeling Techniques

Considerable effort has been focused on the method of relaxation labeling beginning with [4]. In the earlier versions of this work, discrete relaxation techniques were used. These methods initially assign each scene node to all possible database nodes, a one-to-many mapping. Potential assignments are then iteratively removed until a consistent set remained. Here consistency was based on adjacent node and edge properties, horizon = 1. This is an all-or-nothing type of assignment and is similar to a constraint propagation process [66]. The iterative removal of potential assignments is nondeterministic.

The discrete approach was improved upon [67] by describing the compatibility of a given labeling using continuous values. Compatibility is based on the expected occurrence of a given mapping and on the consistency of neighboring assignments. A cost function is defined that describes the support for a given mapping. The assignments are iteratively optimized to find a local maximum of the mean compatibility of the whole mapping. The local nature of the optimization can be a limitation, as it introduces a strong dependence on a good initial guess. The continuous version of relaxation improves upon the all-or-nothing type of assignments, but the horizon of comparisons is not increased.

This approach can be implemented in a manner that avoids the construction of an association graph. This is an advantage over clique-based approaches. It is approximate, however, in that the final labeling assignments may not be unambiguous and may not correspond to a maximal clique [22].

Efforts in this area have continued by attempting to find ways to improve the means of comparing a scene and database graph during the relaxation process [68–70]. In the most recent example of this type of approach [24] standard backtracking is used to search the permutation tree of all possible mappings. The number of potential mappings is reduced by using discrete relaxation, as in [5]. Search pruning involves the examination of discrepancies in the number of edges incident to a node. This type of comparison has a horizon of 1. Parallel implementations of this algorithm were also studied [71].

2.2.5 Tree Search Techniques

These types of approaches match scene features to database features, starting with the most similar elements [72] [59]. Node and edge properties are typically employed here, as well as local comparisons of adjacent nodes, and constraints on types and locations of adjacent nodes [25,73] (horizon = 1). Tree search methods are incremental processes that improve the estimate of an object's pose with each step down the tree. Kalman filters and other methods can be used to refine the estimate of the object's pose [22,27].

Typically a small set of local features will be matched to the model at each step down the search tree. This local feature set (LFS) is formed such that the locations of its members completely determine the pose between the the model and scene [2,12,65,74]. A complete specification of the object's transform is advantageous because it allows the expected location of the next LFS to be computed without ambiguity. This allows incorrect object models or improper object-to-scene matches to be rejected at relatively high levels in the search tree. However,

this does place a constraint on the selection of features when forming the LFSs, in that the features must be grouped during the matching process. This grouping may not always yield the optimal ordering for the tree search process. The size of the LFS determines the horizon of the comparison at the start of the tree search. The horizon increases in some sense, as the search progresses, because the established matching region determines a transform that is used to eliminate future candidate nodes. The worst case arises when initial guesses are wrong due to the limited horizon, time is then spent exploring an unproductive section of the search tree. Hence the critical aspect of the horizon here, is its initial size at the start of the search. In [2], for example, the initial horizon is limited to 4.

An example of some early work is by [75]. This used a search tree that was automatically created at compile time in order to provide the best possible pruning. Here the horizon was limited to adjacent nodes. This type of strategy has been continued with more recent efforts by [76,77]. In this work the precompiled search tree was further optimized by weighting tree branches by the expected computational cost of associated comparisons.

Two categories of techniques, geometric hashing [78] and Hough transforms [62,79,80] are also considered tree-search in nature. These techniques are similar [26] in the sense that image feature(s) are used to form hypotheses that are then verified via tree search. Verified matches are extended in scope to include more scene features. The horizon associated with the hypothesis generation is the critical aspect for determining efficiency. Methods based on Hough transforms have a horizon of 1. The geometric hashing techniques look at local sets of features, requiring 2 or 3 neighboring nodes to develop a transform - depending on the dimensionality of the scene and objects. Recent efforts in geometric hashing [3,38] include a variety of geometric primitives and color to describe each node.

2.3 Literature Review of Graph Matching Techniques from Mathematics Community

The problem of finding graph isomorphisms has drawn considerable attention from mathematicians and has proven to be elusive. Some mathematicians refer to this study as a "disease" [81]. It has attracted attention because of its practical importance (as in this dissertation) and because of its computational complexity. In fact, the class of computational complexity for the isomorphism problem is not yet known. An efficient algorithm is not yet available that handles all types of graphs with 100% success [21]. Some discussion of the computational aspects of the problem and of particularly challenging types of graphs are appropriate to begin the review.

The computational requirements of finding isomorphisms are of concern because a brute force approach requires effort $O(N!)$ for a graph of N nodes. This is reduced somewhat in cases where automorphisms of a given graph make certain sets of nodes interchangeable without affecting the graph's topology [82]. (Note that in the object recognition scenario this notion of automorphism must be extended to include the similarity of both node and edge properties.)

The subgraph isomorphism problem is proven to be NP-complete [21]. This class of problems require a worst-case amount of effort that is of exponential order $O(a^N)$ and are considered to be intractable for many applications. The number of possible solutions is reduced by any automorphisms that may exist in the subgraphs, as in the case of the full-size comparisons performed during isomorphism testing.

A survey article [81] does a comprehensive job of explaining and comparing the two prevalent approaches to isomorphism algorithms. One approach is to find a feature of a graph, known as a "graph invariant". Early examples of this are the determinant and the characteristic polynomial of the adjacency matrix.

These have been shown to be insufficient [83]. A method based on the immanant of the adjacency matrix [84] was also shown to fail. A graph invariant that is able to always distinguish isomorphisms does not exist to date, but is termed a "complete invariant".

Recall that for the object recognition problem it is necessary to find the actual mapping between the nodes of the two graphs. The comparison of two invariants (consider determinants) might answer the yes/no aspect of the isomorphism question, but would not provide the needed mapping between nodes.

A second strategy for finding an algorithm is to produce some sort of canonical, or standard, labeling of the nodes. This process will take any graph and renumber the nodes in the same sequence, regardless of their original arrangement. Two versions of this approach are common. Some techniques attempt to find the labeling in one fell swoop, others use an iterative refinement of the classification.

The similarity between the coding problem for a graph and the isomorphism problem is also interesting [85]. A code is simply a string of symbols that forms a canonical description of a graph. Two graphs are isomorphic if and only if they have the same codes. This implies the existence of a deterministic decoding algorithm [81] which reproduces the graph from its code. It is interesting to consider invariants such as the determinant or characteristic polynomial in light of the requirement for a decoding algorithm. This suggests that invariants must not be "destructive" to any information in the graph. Certainly more than one graph can have the same determinant. Hence information about the original graph has been lost, and there is no deterministic way to recover the original adjacency matrix.

Coding techniques are used in [86,87]. These methods enumerate all possible subgraphs and place them into a tree structure. A binary number is formed from the adjacency matrix of each subgraph and the tree is ordered in a canonical form based on these numbers. The largest binary value formed in this process is used

as a code for the graph.

Another important consideration when reviewing existing techniques is the generality of the graph structure that a method focuses on. A technique should be chosen that is appropriate for the type(s) of graph structures that are expected to appear in an application. Efficient isomorphism algorithms do exist for graphs with special types of structures. Unfortunately these won't apply well to object recognition problems because of the restricted nature of the graph topologies that are addressed therein.

1) Planar Graphs [88]. Nodes in this type of graph can be arranged on a planar surface such that no edges cross over each other.

2) Planar Triply Connected Graphs [89,90]. Planar graphs, nodes of degree 3.

3) Unrooted Trees [63]. A tree is a graph with no loops.

4) Rooted Trees [85]. A rooted tree has one node identified as a base node.

5) Graphs without k-strongly regular subgraphs [91]. A regular graph has nodes with the same degree. K-strongly regular graphs very nearly have this condition with degree k.

6) Digraphs with no two vertices having the same in-degree or out-degree [92]. In a digraph edges have a direction. This is a very special case. The in and out degrees form a natural classification for each node.

More recent contributions to the study of isomorphisms have come from the random graph community. This branch of study examines a variety of graph properties, as they occur in randomly constructed graphs of varying size [93]. The typical sort of question arising in these studies is, whether or not a given property (connectedness, for example) occurs as graphs become infinitely large.

Two methods are typically used to create graphs, Models A and B. Model A uses a given probability (often 0.5) to determine the existence of an edge in each entry of the adjacency matrix [82]. Model B starts with a given number of edges and places these randomly between nodes.

If polygonal objects are assumed, for example, then neither Model A or B is the best choice. In this case the mean number of sides of the polygonal surfaces will determine the mean number of edges that are incident to each node. This implies that the number of edges per node in a database graph should be roughly constant - it should not be a function of the number of nodes - as is the case with Models A and B. Graphs of this type are known as strongly-regular. This is a significant factor because these types of graphs produce the most challenging inputs to isomorphism routines [91]. Strongly-regular graphs are the most challenging because their consistent degree eliminates a natural vertex classification. This forces an algorithm to go to work solely on the basis of connectivity.

Efficient solutions have been developed in the random graph community [94, 95]. These techniques are able to handle nearly all possible random graphs [82]. The small fraction of graphs which must be rejected tends to zero as the number of nodes increases.

The techniques of [94] [95] use Model A. In [94], for example, it is observed that random graphs of sufficient size will tend to have a subset of nodes that each have a unique degree. These nodes are used to form a "foundation" for their node classification scheme. The remaining nodes are characterized by their connectivity into this foundation. Hence, this style of approach will have limited application to object recognition because of the Model A assumption and the requirement for a subset of nodes to have unique degrees. This technique was extended to the case of strongly-regular graphs [96]. However, in this case, the algorithm requires an exponential amount of effort. Extensions to the subgraph isomorphism case are not addressed in these efforts.

The above techniques focus strictly on full-sized isomorphism problems, not on the subgraph isomorphism case, which is combinatorially much worse. A subisomorphism algorithm presented by [1] is more efficient than a complete enumeration and is noteworthy because it does not take advantage of the ad-

ditional constraints that node and edge properties offer. It operates on graphs based on a connectivity analysis alone using a tree search approach with judicious pruning. An exact amount of computational effort required for the algorithm's subisomorphism version is not given. They abandoned their approach at the level of $(N_s,N_d) \approx (10,15)$ describing this size of problem as being "uncomfortably large".

Another technique in [97] considers both the isomorphism and subgraph isomorphism problem. This is a coding type of approach that decomposes a graph into a series of directed trees. The approach focuses on whether or not two of the codes represent the same digraph. The method uses backtracking in a depth-first search.

In [98] another method for finding subgraphs is investigated. This method searches for maximal cliques in an association graph. This alternate method of solution is proven to be NP-Complete, hence one such problem is exchanged for another [21,63].

2.4 Literature Review of Structured Light Sensing

This review focuses primarily on Structured Light ranging systems that use a plane of laser light. This type of acquisition process is also compared against approaches based on spot lasers. Reviews that cover specific point laser systems as well as other types of laser-based measurements are available in [17,99].

2.4.1 3-D Ranging

The main objective of machine vision is to extract information useful for performing some task from various forms of image inputs [43,100]. Examples of machine vision tasks include, robot guidance [59,101,102], remote sensing [103], medical diagnosis [104,105], various types of inspection [106], document process-

ing [107, 108] and many more. For many applications three dimensional (3-D) descriptors of the scene are required. Conventional cameras capture 2-D images and computational approaches are needed to infer the 3-D descriptors from one or more images. Common approaches for this include the use of 2 or more cameras in binocular and photometric stero [109–115]. Approaches using single images include various structure from "x" (x=shading, texture, shadows,motion) techniques [116]. Recent advances include structure from image streams [117, 118] and frameworks for integrating more than one technique for 3-D information extraction [119].

The above approaches share one very important common feature. They are all "passive" approaches, i.e. they do not need a special source of energy to illuminate the scene. There are obvious advantages of this approach: cost, simplicity of imaging hardware, compatibility with human visual processes, etc. On the other hand these approaches need to also overcome some inherent challenges. These challenges arise from the loss of information associated with the perspective mapping of a 3-D scene onto a 2-D image. This produces a fundamentally ill-posed problem when single images are used to find the 3-D descriptors.

Additional challenges that face machine vision researchers are due to spatial and intensity quantization of images and due to computational requirements. These effects manifest themselves in robustness, speed and accuracy performance metrics. In some sense, the advantages of simpler and low cost acquisition hardware is compensated with the need for sophisticated computational processing and analysis approaches.

"Active" approaches for 3-D vision use specialized illumination sources and detectors. These techniques overcome the fundamental ambiguities associated with passive approaches. Some active techniques use laser radar [17, 99], others use various forms of structured lighting [32, 120–124]. In general these methods are able to eliminate the ill-posed problems associated with passive approaches

by modeling and observing the projected illumination. In the case of laser radar, a spot laser beam is steered across a scene. Range measurements are made by either AM or FM detection schemes. Structured Light techniques model the optical paths associated with emission and detection to compute range data by triangulation.

Structured Light ranging has some particularly attractive features compared to laser radar approaches. SL systems can be designed "from the ground, up" much more so than with a packaged laser radar sensor. This provides significant advantages for customizing the acquisition capabilities of a sensor for a particular application. Generally speaking, SL sensors are also more accurate and can be made more rugged, and less expensively than laser radar devices [18].

2.4.2 Approaches to Backbone Geometry

A taxonomy has been developed that exposes three critical aspects of Structured Light (SL) sensor design. **Backbone Geometry** describes the geometrical relationship between the laser(s) and camera. **Correspondence** and **Pixel-Level Analysis** are image processing issues at the large and small scale, respectively. The correspondence problem deals with establishing an association between a region of an image and the location of the illumination source (in world coordinates). Pixel-level analysis takes over once the correspondence problem has been solved, to find the precise image location of illumination patterns.

This aspect of a SL sensor design has to do with the geometry between the camera and laser plane. "Fixed-Plane" geometries have advantages in calibration, simplicity and ruggedness due to the absence of any moving optical components. This necessitates some other source of motion in order to acquire range data in 3-D. In [125] a fixed-plane is used in conjunction with a rotating table. One of the first industrial applications of SL [126] used a conveyer belt.

"Dynamic-Plane" systems alter backbone geometry during the ranging pro-

cess. These systems are generally able to scan larger areas more quickly, and do not require additional mechanisms to supply motion. Accuracy and calibration can become more challenging, however. Rotating optical components began appearing in the work of [12,127,128]. In [129] the imprecision associated with using gear trains to rotate optics is described. Other approaches have used direct-drive motors, some galvanometer-based, to rotate optics [123,130].

The complexity of modeling the laser reflection can vary significantly, depending on the required accuracy of a system. Simple reflection models [31,131] assume a perfect alignment between the laser line and the axis of rotation. When a front-silvered mirror is rotated about an axis that is offset from the reflecting surface, a displacement of the laser plane is introduced. This displacement is also sometimes ignored. In [123,130] a more complete geometrical model is used.

In [124,132] a half-sphere field of view necessitated the use of a dynamic-plane system with pan and tilt units for both cameras and lasers. Emitter and detector units were also housed separately, requiring an in situ calibration as well as dynamic geometrical models for each unit.

2.4.3 Approaches to SL Correspondence Problem

It is reasonable to consider imaging multiple laser profiles in a single camera image in order to achieve increased acquisition rates. More than one laser profile per image increases the information content, but necessitates a scheme for establishing the correspondence between a laser profile and the associated laser plane geometry. Having no ancillary means to establish correspondence [133] can result in a combinatorically unfavorable problem. In [120] the correspondence problem is addressed using relaxation labeling.

In [129] a color camera was used to observe multicolored profiles. In this approach the position of each colored plane was fixed and all planes were projected simultaneously. This provided images with a clear correspondence relationship

that could be acquired in a single frame time.

In [123] a time-lapse image of closely-spaced profiles was collected. Here the correspondence problem was solved by computing a second registered image that contained position-stamp information for each profile. This was accomplished using simple processing steps, allowing position-stamping to be implemented in real-time on commercially available hardware.

Another useful technique involves collecting a sequence of well registered images while illumination sources are toggled on and off. In this way, each laser profile can be identified by analyzing which images in the sequence it appears [32, 134–136].

If multiple laser profiles are present in an image, or if a sequence of images must be analyzed, then it can be necessary for objects in a scene to remain motionless during the acquisition cycle. These types of approaches introduce limitations in more dynamic environments.

2.4.4 Methods of Pixel-Level Analysis

Once the correspondence between a profile and the actual laser position are established, it is necessary to precisely determine image coordinates at the center of the laser profile. The precision with which these coordinates are located effects the overall accuracy of range measurements. Speed and accuracy tradeoffs exist here.

One approach to speeding up SL acquisition is to perform the "pixel-level" analysis in the analog domain. An early effort in this area [137] made the assumption that stripes are roughly vertical. This approach used dedicated timing hardware to find the illumination on each horizontal scan line. This provided image coordinates at frame rate image. More recent approaches [131] have used VLSI implementations that incorporate analog detection and timing operations in a single chip.

In [129] a peak detection algorithm was used to find image coordinates at the center of the laser profiles. This analysis did not provide results with subpixel accuracy. An analog implementation of this scheme was also proposed.

In [132] the nominal orientation of the laser profile could not be assumed and pixel-level operations had to be performed at different orientations. Here, video images were digitized and pixel-level analysis was performed in different directions, using an adaptive scheme.

2.4.5 Alternate Illumination Structures for Ranging

It is worth considering a variety of structures for illumination when at the early design stage of a SL system. One such approach uses a spot of laser illumination. These are sometimes referred to as "flying dot" systems.

If a video camera is used for this type of system, the approach can suffer in terms of acquisition speed due to the low information density per image [138]. Because of this, many approaches turn to 1-D detectors [27] or to custom optics [139].

Despite the problem of low information density, there is a significant advantage to flying-dot systems that use 2-D video images, or some other type of 2-D detector. When a directional vector (in R^3) is available, from the detector to the flying dot, an on-line confidence measure can be computed for each range point. The ranging process in these systems can be formulated as an intersection calculation of two lines, one along the laser optical axis and the other along the camera sighting of the flying dot. The closest point of intersection between these two lines can be used as a best estimate for a range measurement. The minimum distance between these two lines can then be used as an estimate of the measurement uncertainty. This confidence measure is provided in an on-line, point-by-point manner. This provides great advantages for applications demanding high accuracy and high reliability. SL systems that image a laser line do not possess this

type of inherent accuracy check.

Laser optics are available that project alternate light patterns, circular projections, for example. These may provide advantages for some situations where the structure of the light matches an application-dependent measurement region.

CHAPTER 3

Efficient and Accurate Acquisition of Range Profiles By Structured Light: Design, Calibration and Implementation Issues †

To form a basis for the forthcoming chapters on range segmentation and graph matching, the subject of range acquisition is discussed first, in this chapter. This discussion covers the principle of operation, calibration, and many design details of a common type of laser-based ranging sensor - a Structured Light sensor. Strengths and weaknesses of this kind of sensor are described. A Structured Light sensor was implemented and its performance benchmarked as a part of this research effort. This sensor is refered to as **PRIME**, the **PR**ofile **I**maging **M**odul**E**. PRIME uses novel methods of range acquisition and calibration. It is used to provide an example of the various kinds of processing steps required with this type of sensor. Discussions of other types of ranging sensors may be found in [17,99,141]

3.1 Introduction to Structured Light Ranging

Structured Light (SL) sensing is a well established technique for ranging. A great variety of techniques in this general area have been developed [17,121,126]. The common thread of all these approaches is the underlying use of triangulation. This ranging geometry can be seen in Figure 3.1, which depicts the optical com-

[0†] This chapter was accepted for publication in *Advances in Computers* [140]. It appears herein in near final form.

ponents in PRIME. Here, the laser is projected downward towards objects in the scene. Laser illumination striking an object is observed by the camera to reveal surface profiles which can be converted into Cartesian range data. In PRIME, the ranging triangle is completed by a rigid backbone between the camera and laser.

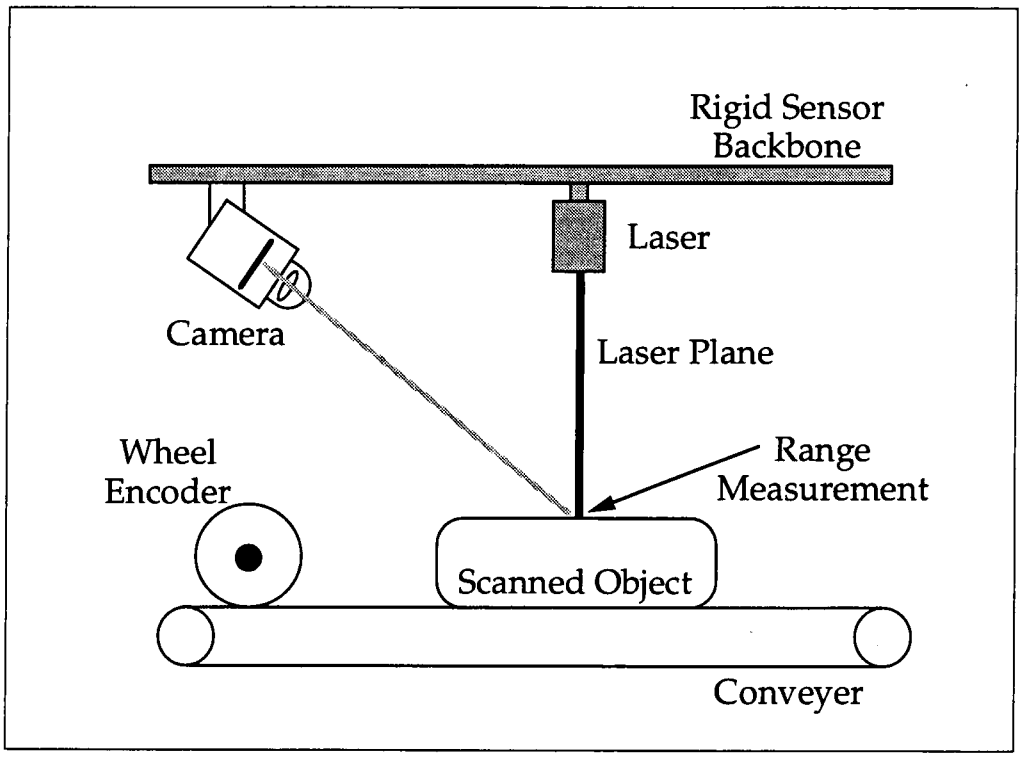


Figure 3.1: Structured Light systems use triangulation to acquire range measurements. In PRIME, the ranging geometry is formed by a laser emission, the reflected light observed by the camera and by a rigid backbone. Optical measurements capture range data in a plane. A conveyer produces the necessary motion for 3-D range data.

The three main challenges in developing a Structured Light system are calibration, accuracy and acquisition speed. Calibration models are needed to relate image to world coordinates. See Figure 3.2. Methods of calibration are critical for success, but are usually not described in sufficient detail in published works. Typically, procedures require examples of image coordinates to be generated by

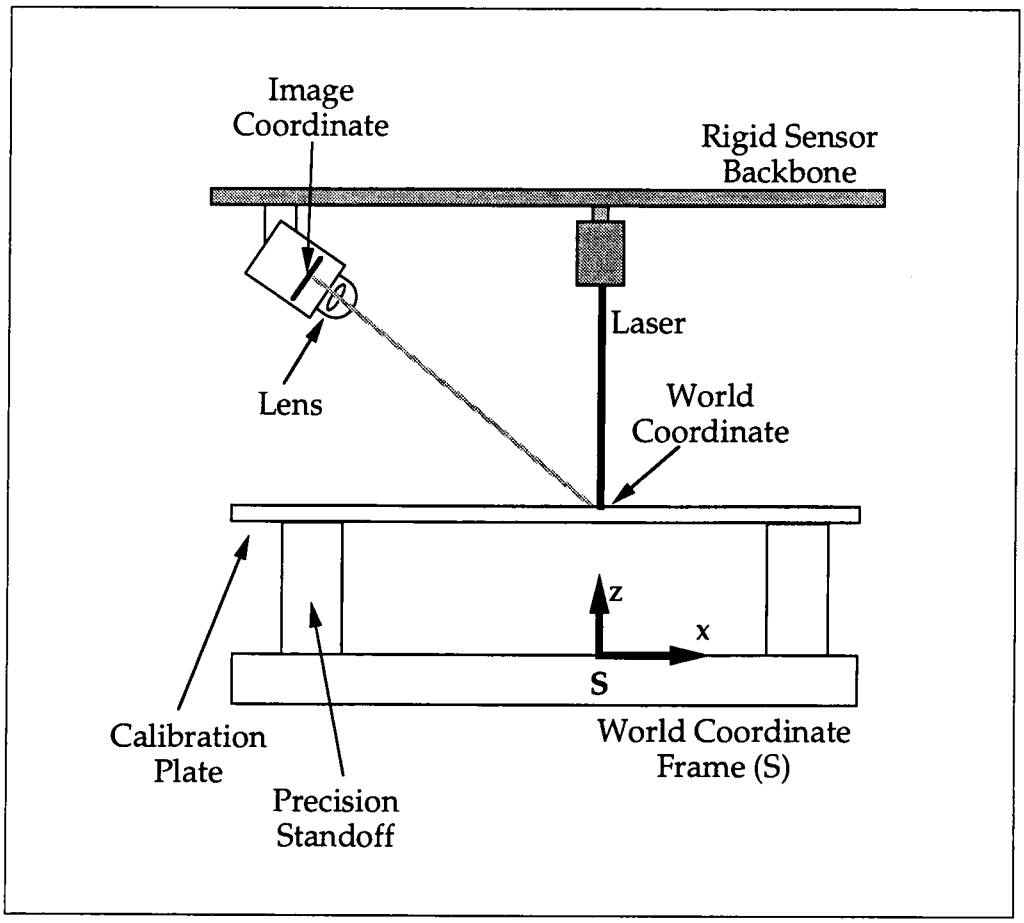


Figure 3.2: The purpose of sensor calibration is to find a mapping between image and world coordinates. In PRIME, this relationship is fixed because of the rigid backbone between the laser and camera.

calibration targets, the positions of which must be known with high accuracy. To ensure accurate ranging in the final system, the calibration data must be generated at distances from targets that span the range of standoffs that will be used during sensing. If the intended standoff is on the order of a few feet or less, then standard optical bench equipment can be of great utility during calibration. Hence, the difficulty of this task can vary with the intended standoff of the sensor.

Challenges in calibration also arise from the complexity of models that are required for sensor kinematics. These kinematics describe the geometrical relationship between the camera and the laser plane. If, for example, these kinematics are fixed then the calibration procedure is much simpler. PRIME is such a system, which is referred to here as having a "fixed-plane geometry", as seen in Figure 3.1. An alternative to having a fixed triangular geometry is to reorient the optical paths while ranging. Using a "dynamic-plane geometry" in this manner [31, 123] can permit larger regions to be scanned more rapidly. These approaches typically reorient the laser beam using low inertia optical components. Beam repositioning in this manner can be achieved with high speed and high precision. While dynamic geometries are attractive from the point of view of acquisition speed, they usually require more complex calibration models [123, 124, 130].

Ranging accuracy is highly dependent on calibration models. However, even the most careful calibration effort can be fruitless if the ranging geometry is unfavorable. This refers to the sensitivity of range measurements to various system parameters, see Section 3.4 for a sensitivity analysis of PRIME.

Acquisition speed is another challenge to SL system design. Range acquisition involves locating the laser profile within camera images. This requires pixel examination and processing in order to precisely locate image coordinates that reside at the center of the laser profile. While the complexity of these algorithms are low, the computational requirements are relatively high. As seen in Figure 3.3, systems such as PRIME must process images with relatively low information

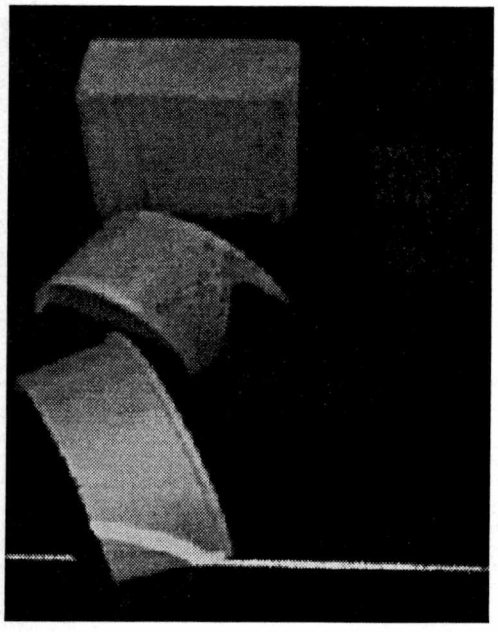


Figure 3.3: Image of laser profile seen by camera. The camera's optical filter was removed for this image to better reveal objects in the scene.

density - this image has only one view of the laser line. Given the size of a standard image and relatively small area of pixels illuminated by the laser line ($\approx 1\%$), a significant amount of pixel-level computational effort must be expended to acquire the range data associated with a single image. Because of this relatively low payoff, many researchers have experimented with ways to pack more laser lines into a single image [32, 123, 129]. Approaches with multiple laser images necessitate using a heuristic to determine the correspondence between image features and laser positions. Several methods for this are reviewed below.

Some discussion is appropriate concerning the fundamental limitations of SL ranging. Surface reflectivity is one such factor. For reliable range data, a scanned object should have surfaces with lambertian reflectivity. Specular surfaces will often reflect too much of the structured illumination away from the camera. This produces voids in range data. Note that the degree of surface reflectivity can be counter-intuitive when dealing with near-InfaRed (IR) laser systems, such as PRIME, since these wavelengths are beyond the human visual range.

Shadowing is also a fundamental problem in SL systems. This occurs when object geometries occlude the laser from the field of view of the camera. Shadowing effects can be reduced when the camera to laser baseline distance is shortened. However, this also increases the sensitivity of the system to measurement noise - see section 3.4.

Ambient lighting is an important design issue. It can be a limitation to SL or other types of optical ranging if ambient sources are unfavorable and cannot be controlled. Monochromatic illumination and matched optical filters for cameras can be used to tackle this problem, as in PRIME, provided the ambient lighting can be setup on a different wavelength. Florescent lighting and near-IR SL ranging make a very nice complementary pair. This combination provides ample room light without contaminating range imagery. Incandescent light is a very poor choice for use with near-IR SL systems.

The novelty of PRIME stems from new approaches to calibration and to pixel-level operations. These techniques make for a system that has real-time acquisition, is accurate, easy to calibrate and made entirely from commercial components. PRIME uses a single plane of laser light to provide ruggedness, accuracy and simple calibration. This dis require effort to achieve real-time acquisition. Acquisition speeds were improved by mapping portions of the pixel-level operations onto dedicated commercial hardware. Because a single laser profile is imaged, and because range acquisition occurs at frame rate, PRIME is able to scan objects that are continuously moving.

3.2 Structured Light Acquisition

A Structured Light sensor includes both optical components and pipelined processing elements. A wide variety of components can be selected for these purposes. In PRIME, for example, the computing components include a Mo-

torola 68040-based single board computer and a Datacube MV20 image processing board. The main processor runs under a vxWorks environment. It is responsible for real time configuration and control of the Datacube hardware, and for applying calibration models.

3.2.1 Optics and Imaging

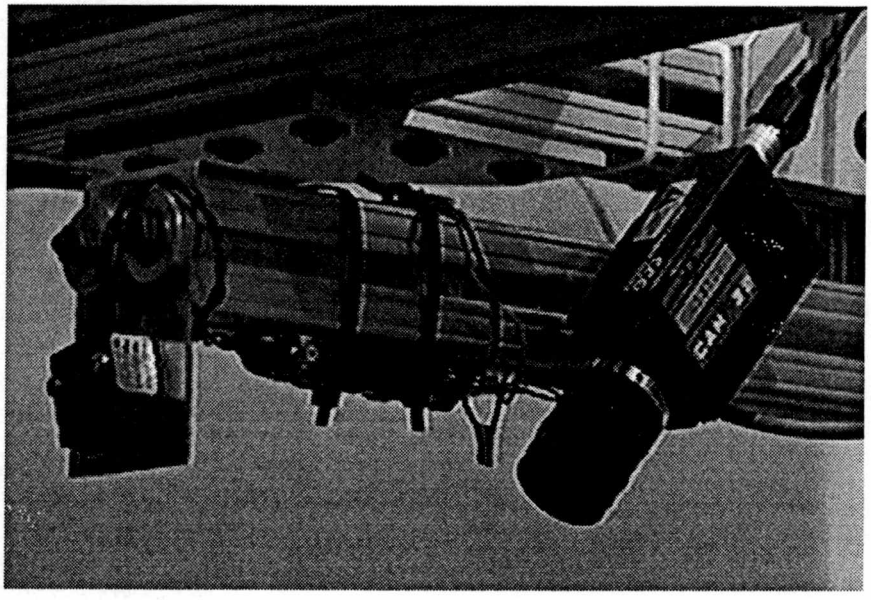


Figure 3.4: Laser (left) and camera (right) used in PRIME sensor. The laser generates a plane of light using a cylindrical lens. A bandpass optical filter on the camera yields distinct imagery of the laser plane as it intersects objects in the scene.

Figure 3.4 shows the optical components that comprise the SL sensor. A near-InfaRed diode laser, seen on the left, illuminates a scene from above. The laser emission is in the form of a plane of light which is generated by a cylindrical lens mounted in the laser housing. The black and white camera, seen to the right, is positioned so as to image the light reflected from the laser plane as it strikes objects in the scene. Figure 3.5 includes typical scanned objects, also.

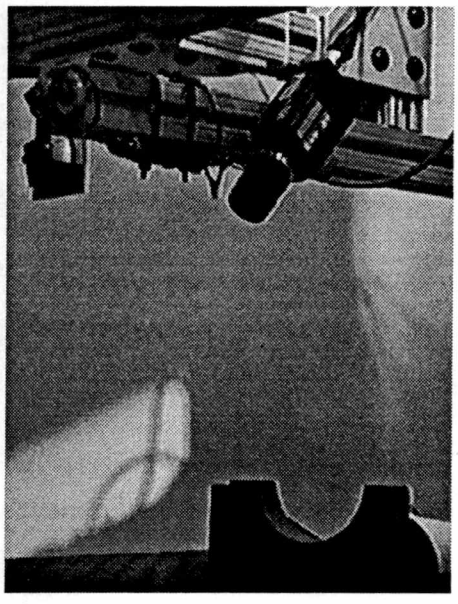


Figure 3.5: PRIME sensor with typical objects.

The camera is outfitted with an optical filter that is matched to the laser optical frequency. These matched optics produce very distinct imagery of the laser profile. The image in Figure 3.3 was taken with the optical filter removed, for presentation purposes. Because CCD cameras are quite sensitive to near-IR, manufacturers typically install an IR cut filter. Such a filter was removed from the camera used in PRIME.

Another aspect of the imaging process has to do with the use of an electronic shutter. In applications with objects that move continuously past the sensor, a certain degree of blurring would nominally occur in each camera image. Blurring increases the uncertainty with which image coordinates at the center of the laser profile can be recovered, and hence, must be limited. The camera's electronic shutter reduces the temporal integration period for each pixel. In this way the frame rate of images is unchanged, but the exposure time is reduced. This yields video streams which are subsampled in time, this means that shape information will be missing between sequential laser profiles. This introduces an upper limit

on the spatial frequency content of shape descriptions and can result in aliasing.

A standard video frame is composed of two interlaced fields. Each field is transmitted sequentially. When using an electronic shutter, each field is exposed individually. Typically the shutter interval for one field will occur immediately before that field is due to be output on the composite video signal. Hence, a single camera image contains data acquired from two separate time instances, 1/60 second apart. This necessitates that the temporally-skewed fields be processed separately when the laser profiles in each are converted to range data.

In PRIME, the physical thickness of the plane of laser light is ≈ 0.040 inches thick. The sensor has been designed for scanning velocities of ≈ 1 in/sec. An electronic shutter interval of $< 1/250$ sec is well suited for this situation because the blurring of the beam is limited to $< 10\%$ of the thickness of the laser plane. In practice this fractional increase in the apparent thickness of the beam has proved to have little effect on range measurements. However, it is true that unfortunate scene geometries can generate greater degrees of blurring. For example, A sloped block will tend to cause the laser line to blur vertically if it is viewed while approaching the camera. Steeper block faces will tend to produce greater degrees of blurring. These factors are very much scene- and application-dependent.

A fundamental tradeoff exists between the ability to localize the position of the laser in an image and the apparent brightness of the beam. The mechanism for adjusting this degree of freedom is the electronic shutter interval. The limiting factor in the range of this adjustment is the amount of optical power in the laser and the sensitivity of the camera. In PRIME, the Automatic Gain Control (AGC) on the camera has been engaged. Since the majority of the camera image is black, the AGC sets analog gains relatively high. This causes the laser illumination to be amplified significantly. For most of the experiments performed with PRIME, the intensity of pixels at the center of the laser line were saturated (256/256).

3.2.2 Image Processing Operations

The images used during range acquisition have an almost totally black background, with gray levels $\approx$ 10-20 out of 256. Given the saturated pixels on the laser profile, this represents a SNR $\approx$ 25 db for the image data. These conditions provide considerable latitude in selecting a gray scale threshold used for producing binary images. These binary images are used in an intermediate step to roughly locate the laser line within the image.

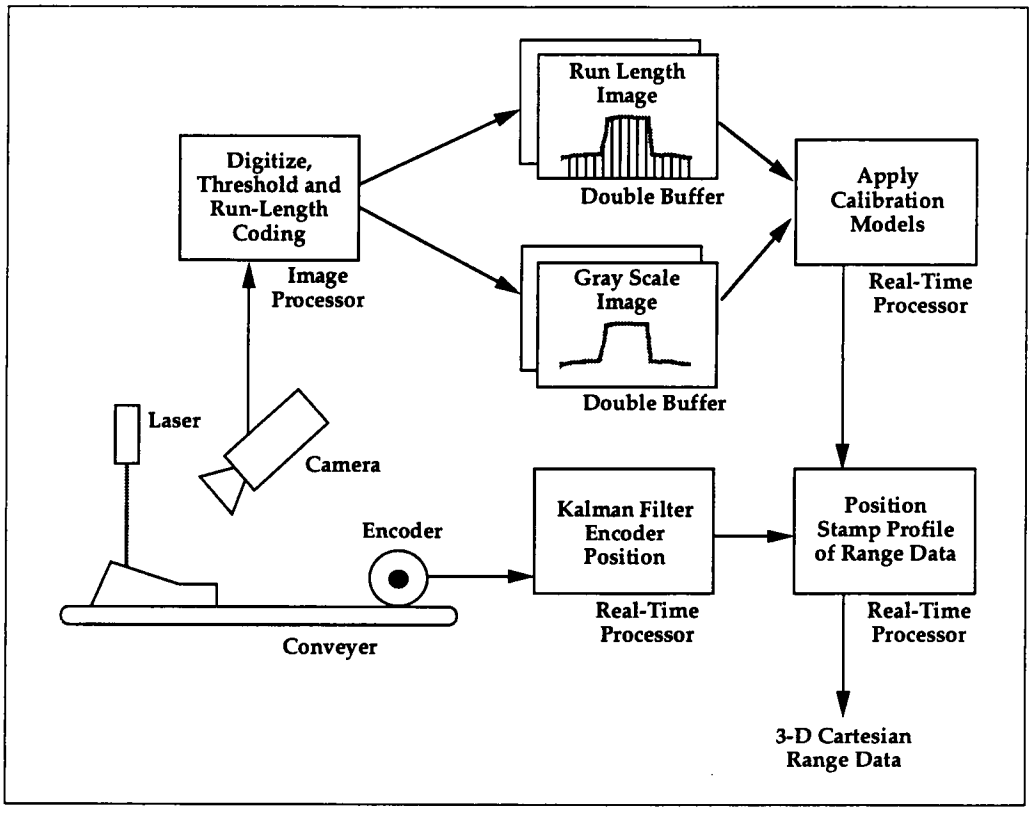


Figure 3.6: Block diagram of the processing steps in the PRIME sensor.

Image processing operations were split between the Datacube board and the main processor for improved pipeline throughput. In order to permit both devices to simultaneously process images, double buffering was used between these two pipeline elements. See Figure 3.6. The Datacube board provided image I/O and storage, displayed the real-time status of the ranging sensor, and was used to

locate the rough position of the laser line within each image.

The rough position estimates were achieved by thresholding and then run length coding [142]. Run length coding (RLC) generates a 1-D array having indecies with a one-to-one correspondence with the columns of the binary image. Each RLC array element contains the height along a column from the bottom of an image up to the first illuminated pixel. The RLC data is depicted in the image buffers of Figure 3.6. Hence the RLC array provides a succinct description of the rough location of the laser line. This information greatly improves the speed of range computations because the Datacube board can provide the RLC data at frame rate. This eliminates an otherwise burdensome effort by the main processor of searching the entire image for a relatively small illuminated region - which could not be done in real-time. Using the run length coded array, the main processor can directly access the required portions of images.

To precisely localize the center of the laser line, pixel values are examined on a cross section of the laser illumination and a weighted centroid calculation is performed. Figure 3.7 illustrates typical pixel intensities, as sampled along a path roughly orthogonal to the laser line. In PRIME, the intensity profiles are typically 5-7 pixels wide with a roughly Gaussian shape. The main processor examines gray scale pixel values in a 20x1 window, W, centered at locations given by the RLC array. A mapping function, $P(g_i)$, is used to describe the likelihood that a pixel having intensity g_i is a member of the laser line. The mean, $\bar{r}$, and variance, σ_r^2, of the row at the center of the laser line are found with

$$\bar{r} = \frac{\sum P(g_i) r_i}{S}, \qquad \sigma_r^2 = \frac{\sum P(g_i) r_i^2}{S} - \bar{r}^2 \tag{3.1}$$

where g_i is the gray level of the pixel in row r_i, and $S = \sum P(g_i)$. Summations are taken within the window, W, and all include pixels above the binary threshold. The image coordinate recovered,$(\bar{r}, c)$, is the mean row together with the column under examination.

It was desired to make the mapping function, $P(g_i)$, a smooth curve varying from 0.0 to 1.0, corresponding to pixels in the background and on the laser profile, respectively. A truncated error function (erf) was chosen for the mapping. Values for $P(g_i)$ were found by integrating a Gaussian and then normalizing so that the area under the erf curve was unity. The shape of the original Gaussian was chosen so that 3σ below the mean were gray levels clearly belonging to the background and 3σ above the mean were levels at the center of the laser profile. The gray levels at these 3σ points were chosen manually by examining a histogram of typical images. Previous experiments in camera calibration [130] have shown as much as a 13% change in centroid location when weighted centroid calculations are done, versus binary methods.

The variance, σ_r^2, of each $(\bar{r}, c)$ coordinate was checked against a threshold as a means to eliminate the use of blurry portions of a laser profile. Blurry imagery can result from excess relative motion or uneven reflectivity, for example. In these situations the reliability with which the center of the laser profile can be recovered is compromised and the generation of range points should be avoided in order to maintain accurate results.

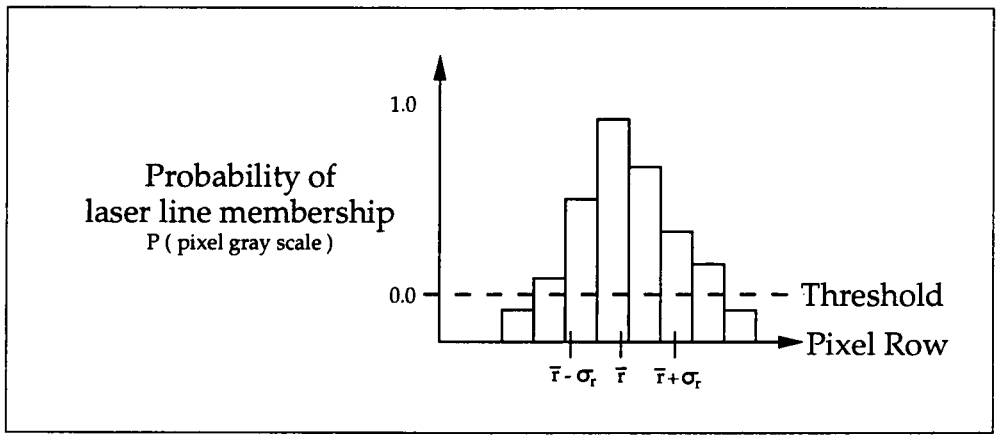


Figure 3.7: Recovering image coordinates at the center of the laser line.

3.2.3 Ranging Computations

A calibration model, M, is used to convert image coordinates, (r, c), to world coordinates, (y, z). The model is applied to an augmented vector of image coordinates. For example, with a second order model

$$\begin{bmatrix} r^2 & c^2 & rc & r & c & 1 \end{bmatrix} M = z \tag{3.2}$$

where z is the Z coordinate of a range point. A similar model is used to generate Y coordinates, see section 3.3.

In PRIME, image data provides the Y and Z components of range points. The X component is found by monitoring the motion of a conveyer belt using a wheel encoder. See Figure 3.1. The processing of encoder readings is interrupt driven, occurring at a rate four times higher than the video field rate (60 Hz). This provides accurate position estimates (in X) for each video field, allowing the range profiles in each to be field to be properly position-stamped. The main processor was responsible for orchestrating the image acquisition process and stamping each acquired profile with the appropriate positions along the conveyer. Having camera image data synchronized with conveyer position stamps allows 3-D Cartesian range data to be computed. The encoder on the PRIME testbed has a resolution of 4000 counts per revolution, corresponding to 0.003 inches of travel per encoder tick.

The main processor uses Kalman filtering [143] to process the raw encoder readings, for improved position estimates. Because of the relatively constant velocity of the PRIME conveyer, the state transitions of the Kalman filter are modeled as having a constant acceleration [144]. The process noise matrix has been setup accordingly [145]. The process noise parameter has been determined experimentally, $q = 50$. The measurement uncertainty σ_m has been set using the encoder spacing and the standard deviation associated with a uniform distribution [146], $\sigma_m = 0.003/\sqrt{12}$ inches.

Because of the sensor geometry, a simple X-Y grid was used to store range data. This provided advantages in terms of complexity and the speed of data storage. It eliminated the need for an octree data structure, for example. The laser plane has a near-vertical orientation. Because of this, multiple range points do not tend to occur that have the same lateral (X-Y) position above the grid.

3.3 Calibration of Structured Light Sensors

Calibration models are required in order to provide the relationship between image and world coordinates (see figure 3.2). A distinction is made here between the problems of calibration versus registration of the sensor. Herein, "calibration" refers to a process in which the location of the world frame (S in the figure) is defined locally to the sensor. Its position is established with the convenience of sensor calibration in mind. "Registration" refers to the process of relating the sensor frame, S, to some other frame that is pertinent to the application - such as a manipulator frame.

In most cases, the design of two aspects of calibration are tightly coupled. These are (1) the process by which calibration data is collected and (2) the formulation of the calibration model. The solution to these two problems typically must be found in a joint manner. The approach taken for PRIME is described below.

3.3.1 Calibration Process and Formulation of Calibration Model

Structured Light ranging is fundamentally a process of triangulation. Calibration is sometimes approached as a process of isolating explicit geometrical parameters of this ranging triangle. In [17] range calculations are described using the law of sines together with a pin hole model of the camera. Note that this would necessitate two separate calibration procedures (each of which would contribute errors).

A one step calibration procedure has been developed for PRIME. This process is very similar to the Two Planes method of camera calibration [147]. In general, any Fixed-Plane SL system can be calibrated in a one step procedure because of the rigid mapping between image and world coordinates. One step procedures have advantages in terms of accuracy and simplicity. Accuracy is improved because models can be found via a single least norm solution.

A model $z = f(r, c)$ relating height, z, to image coordinates has been found using empirical calibration data. This relationship has been determined by analyzing images to find many examples of the triplet

$$\left| \begin{array}{ccc} z_i & r_i & c_i \end{array} \right|. \tag{3.3}$$

Figures 3.8 and 3.9 illustrate the PRIME calibration jig. The jig allows a series of horizontal plates to be located at known heights, z_i, and imaged by the camera. Figure 3.10 shows such an image containing many examples of where the laser plane is imaged at a given vertical height. The pixel-level operations described in section 3.2.2 are used to generate individual triplets. A number of calibration plates were located across the depth of field of the sensor. Precision standoffs were used to locate these in the Z direction. When calibrating for X, a linear table was used to automate positioning.

For increased image fidelity during calibration, a number of images of a given calibration plate were averaged. This reduces the random noise in pixel values that nominally accompany the imaging process [142].

To find the calibration model, an overdetermined set of equations is formed by augmenting the image coordinates (r_i, c_i) of each triplet. A variety of forms of calibration models have been studied. In the case of a second order model, for example, a calibration n-tuple

$$\left| \begin{array}{cccccc} z_i & r_i^2 & c_i^2 & r_i c_i & r_i & c_i & 1 \end{array} \right|. \tag{3.4}$$

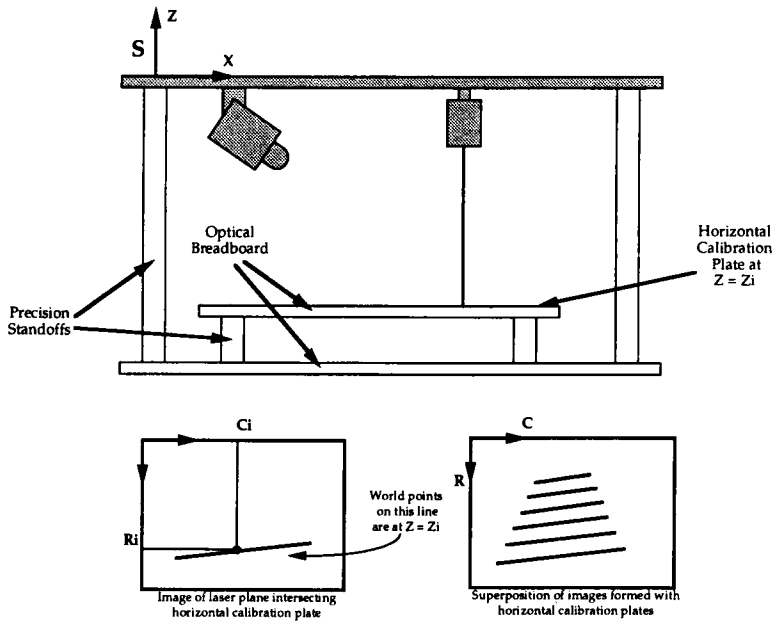


Figure 3.8: A series of horizontal plates are used to generate calibration triplets. The triplets are used to form an overdetermined set of equations from which calibration models may be determined. This illustrates the calibration procedure for the $z = f(r, c)$ model.

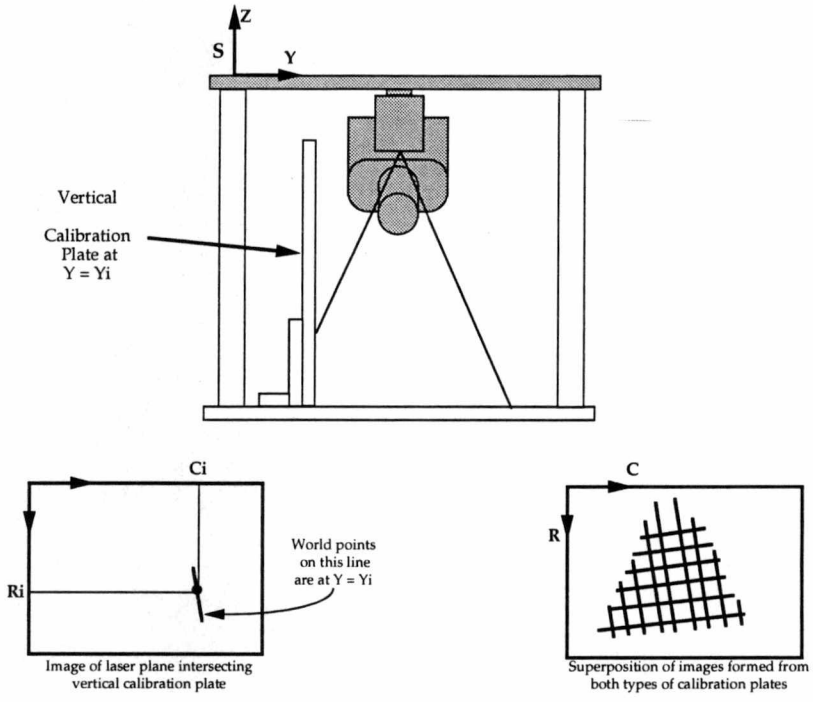


Figure 3.9: A series of vertical plates are used to generate the data needed for the $y = f(r, c)$ calibration model in PRIME.

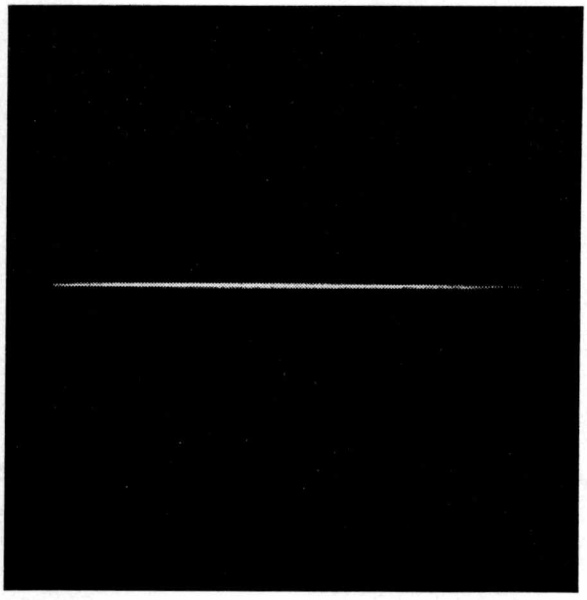


Figure 3.10: An image used for range measurement calibration. The central horizontal line is produced by the intersection of the laser plane with a horizontal metal plate.

is used. These n-tuples are arranged to form

$$\begin{vmatrix} z_1 \\ . \\ z_i \\ . \\ z_n \end{vmatrix} = \begin{vmatrix} r_1^2 & c_1^2 & r_1c_1 & r_1 & c_1 & 1 \\ \cdots & & & & & \\ r_i^2 & c_i^2 & r_ic_i & r_i & c_i & 1 \\ \cdots & & & & & \\ r_n^2 & c_n^2 & r_nc_n & r_n & c_n & 1 \end{vmatrix} \begin{vmatrix} m_1 \\ \cdots \\ m_p \end{vmatrix} \tag{3.5}$$

or

$$Z = FM \tag{3.6}$$

Where

Z (nx1) :contains world coordinates,

F (nxp) :contains augmented pixel coordinates, and

M (px1) :contains model parameters.

In this case $p = 6$. A series of vertical plates are used to generate the calibration triplets needed for the Y model. This process is depicted in Figure 3.9.

3.3.2 Numerical Method of Solution for Calibration Models

Many methods of solution are possible for finding the calibration model, M, in a least norm sense [148]. However, prudent selection of a solution technique can have side benefits in the calibration process. Forming the normal equations [149], for example, allows the solution to be found, but does not provide any additional information. Techniques that reveal the numerical sensitivity of the solution are much more desirable. This provides an indication of how much the solution will change due to slight perturbations in the calibration triplets. The calibration triplets will contain measurement noise, hence the need for an overdetermined specification of the solution and for as low a numerical sensitivity as possible. There is no guarantee that the existence of ill-conditioned calibration data will be apparent in the residual error [150] of the solution. The greatest care in the imaging and pixel-level operations will be fruitless if the solution for a calibration model suffers numerically. The consequence for ranging is that a calibration model may not yield accurate results when new data is applied.

A common reason for the matrix F of Eq. 3.6 to become ill-conditioned occurs due to a poor distribution of the input data points. As seen in Eq. 3.5, the right hand column of entries in F are all 1. The columns of F must be linearly independent [149]. Hence if all calibration triplets possess the same row or column then F will become ill-conditioned - unless SVD is used to prevent this occurrence, after the fact.

One technique for improving numerical performance when finding M involves scaling the image coordinates of the calibration triplets. By normalizing the image coordinates to the range $(0, 1)$, the span of entries in F can be reduced.

Table 3.1: Demands on numerical precision for normalized and unnormalized calibration models.

Order	Span of Entries in F Unnormalized	Span of Entries in F Normalized	Cost Ratio of Spans	Integer Precision of Cost (bits)
1st	$485^1/1$	$1/0.05^1$	$2.4 \quad 10^1$	5
2nd	$485^2/1$	$1/0.05^2$	$5.8 \quad 10^2$	10
3rd	$485^3/1$	$1/0.05^3$	$1.4 \quad 10^4$	14
4th	$485^4/1$	$1/0.05^4$	$3.5 \quad 10^5$	19

This improves the condition of F [148]. If it is assumed that coordinates vary from 5 to 90% of the image then the normalized and unnormalized coordinates cover the ranges $(0.05, 0.95)$ and $(25, 485)$, respectively. The span of largest to smallest entries in F is used as a measure of demand on numerical precision. See Table 3.1. The ratio of the span of entries in F for the normalized vs. unnormalized cases is given as "Cost Ratio of Spans". This gives a measure for the reduction in precision that is achieved by normalizing. This reduction is then expressed as a number of bits.

The table shows that a significantly larger number of bits can be required to represent the ratios for the unnormalized entries. This unnecessary cost in precision is reflected in poorer conditions for F, reaching 6 orders of magnitude! PRIME has been calibrated to an accuracy of 1 part in 1500 (see Section 3.3.3). This necessitates $\approx$ 11 bits of precision for ranging, nominally. The burden of using non-normalized formulations is additive. Note that these costs in precision have been expressed as a number of bits in integer format. The actual numerical effects are more complicated, as these involve floating point operations.

A Singular Value Decomposition (SVD) [148] exposes the singular values of F, which can be used to compute the condition number [149]. This provides an excellent measure of the numerical sensitivity of the solution. The SVD approach

decomposes F

$$F = UDV^T \tag{3.7}$$

where U and V are orthonormal and D is diagonal, containing the singular values of F. The technique also permits "repairs" to be made to D, ensuring its invertibility [150]. This allows the calibration model

$$M = VD^{-1}U^TF \tag{3.8}$$

and the condition number

$$\kappa = d_{max}/d_{min} \tag{3.9}$$

to be easily computed. Here, d_{max} and d_{min} are the largest and smallest diagonal elements of D. F is considered to be ill-conditioned if $1/\kappa$ approaches the precision of floating point computations, for example, no less than 10^{-12} for double precision [150].

3.3.3 Evaluation of Calibration Model

Selecting an appropriate form for a calibration model effects important tradeoffs in acquisition speed and ranging accuracy. This selection process is closely related to that associated with the Two-Planes method of camera calibration [147]. These problems are quite similar because each involve a mapping from image coordinates to a plane of world coordinates.

Recommendations for appropriate forms of camera models [151] serve as a guide, as well as several metrics for model evaluation [123,152] which have been incorporated and extended for PRIME. The metrics used for PRIME involve the computational burden during on-line evaluation and various measures of the quality of the model solution. These are:

1) Compute burden

2) Condition number

3) Span of residual errors

4) Autocorrelation of residual errors

5) Goodness-of-fit, based on chi-squared.

The compute burden is a direct result of the number of terms in the model. The condition number is found from Eq. 3.9. The residual error of the solution is computed by comparing the fitted height of the ith triplet, $z_i' = f(r,c)_i$ to the original height. Ideally $z_i - z_i' \rightarrow 0$, but in practice random measurement noise and higher order lens aberrations keep these from vanishing. Although low residual errors do not guarantee an accurate model, these values are still useful to examine, as a low magnitude is a necessary condition for accuracy. The span of the residual errors is described by computing the maximum, mean and standard deviation of the absolute residuals.

The purpose of calibration is to characterize the distortions in a mapping between two planes. Hence, a proper mapping and proper calibration procedure should result in a set of residual errors with a very low spatial dependence. That is, any uniform pattern or trend in the spatial arrangement of residuals should be very minimal. The residuals should appear as random entries. This random character can be described using the autocorrelation of the residuals.

As a final metric, a goodness-of-fit measure has been computed involving chi-squared and the degrees of freedom in the system of equations [150]. This involves the use of an estimate of the accuracy of z_i, which is noted in the tables. The fit quality, Q, should ideally be $Q \geq 0.1$ to consider a mapping as valid. In some cases values in the range $Q \geq 0.001$ are also deemed acceptable [150].

Tables 3.2 and 3.3 summarize the metrics found when calibrating PRIME. Table 3.4 gives the form of each model that was considered. When finding the Z model, 280 calibration triplets were used and the accuracy of the true heights, z_i, was estimated at 0.005 inches. For Y, 98 triplets and an accuracy of 0.010 inches for x_i were used.

Table 3.2: Metrics for Z calibration model for PRIME. A total of 280 calibration triplets were used. A value of 0.005 inches was used for the accuracy of z_i.

Type	Normalized	Mean Residual (in.)	Standard Deviation of Residual (in.)	Maximum Residual (in.)	Fit Quality	Autocorrelation (1 shift)	Condition Number
1	yes	0.071	0.038	0.137	0.00	0.9	10^1
2	yes	0.007	0.004	0.022	0.00	0.9	10^2
3	yes	0.003	0.003	0.015	0.99	0.7	10^3
4	yes	0.002	0.002	0.013	1.00	0.5	10^3
1	no	0.071	0.038	0.137	0.00	0.9	10^3
2	no	0.007	0.004	0.022	0.00	0.9	10^6
3	no	0.003	0.003	0.015	0.99	0.7	10^9
4	no	0.002	0.002	0.013	1.00	0.5	10^9

Table 3.3: Metrics for X calibration model for PRIME. A total of 98 calibration triplets were used. A value of 0.010 inches was used for the accuracy of x_i.

Type	Normalized	Mean Residual (in.)	Standard Deviation of Residual (in.)	Maximum Residual (in.)	Fit Quality	Autocorrelation (1 shift)	Condition Number
1	yes	0.093	0.059	0.240	0.00	0.8	10^1
2	yes	0.010	0.008	0.033	0.00	0.8	10^2
3	yes	0.009	0.007	0.031	0.03	0.8	10^3
4	yes	0.004	0.002	0.009	1.00	0.8	10^3
1	no	0.093	0.059	0.240	0.00	0.8	10^3
2	no	0.010	0.008	0.033	0.00	0.8	10^6
3	no	0.009	0.007	0.031	0.03	0.8	10^9
4	no	0.004	0.002	0.009	1.00	0.8	10^9

Table 3.4: Form of calibration models.

Type	Coefficients
1	$1\ r\ c$
2	$1\ r\ c\ r^2\ c^2\ rc$
3	$1\ r\ c\ r^2\ c^2\ rc\ r^3\ c^3$
4	$1\ r\ c\ r^2\ c^2\ rc\ r^2c\ rc^2\ r^3\ c^3$

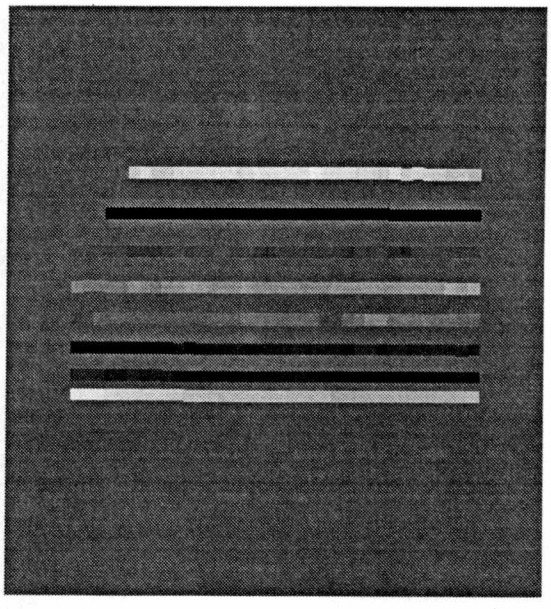


Figure 3.11: Residual error of a 1st order (type 1) Z calibration model. Mean = 0.071 in. Gray levels of each block are scaled to span the extremes of the residual errors in this model (maximum = 0.137 in).

In addition to the tabular summaries, images have also been generated that depict the residual errors. These provide a visualization of any spatial correlation in the errors and of the location of all the calibration triplets, in a single image. Figures 3.11 through 3.14 show images that depict residual errors for types 1 and 4 of the Z and X models, respectively. The images contain a series of small gray blocks, each of which coincides to the location of a calibration triplet and the corresponding residual error. The gray level of the block is scaled so that the data point possessing a minimum residual error is the darkest and the one possessing the largest error is the lightest. Note that a different gray scale mapping is used for each image.

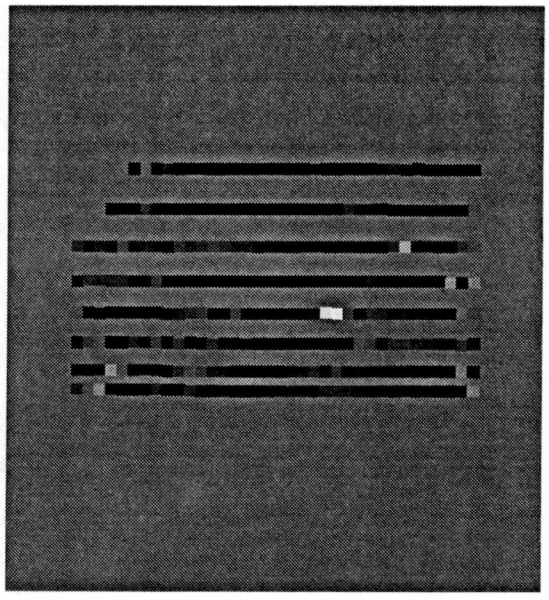


Figure 3.12: Residual error of a 3rd order (type 4) Z calibration model. Mean = 0.002 in. Gray levels of each block are scaled to span the extremes of the residual errors in this model (maximum = 0.013 in).

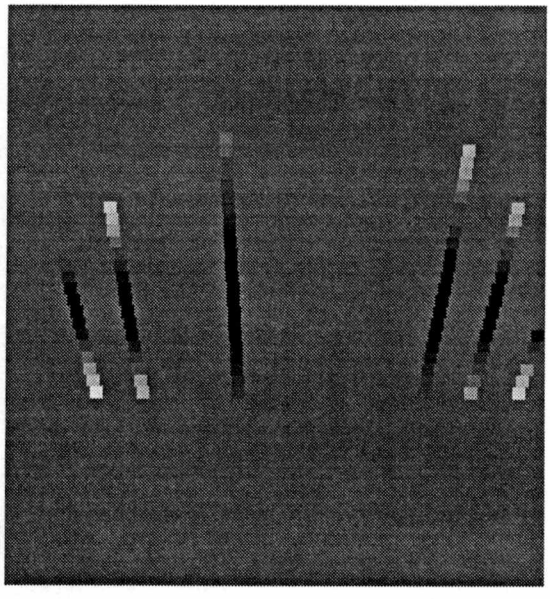


Figure 3.13: Residual error of a 1st order (type 1) X calibration model. Mean = 0.093 in. Gray levels of each block are scaled to span the extremes of the residual errors in this model (maximum = 0.240 in).

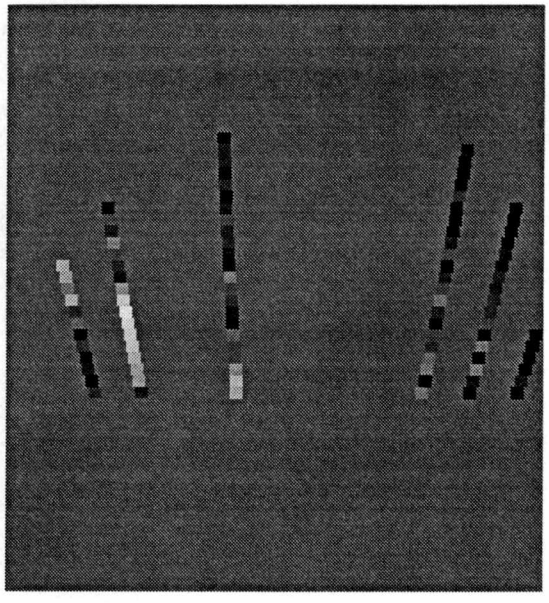


Figure 3.14: Residual error of a 3rd order (type 4) X calibration model. Mean = 0.004 in. Gray levels of each block are scaled to span the extremes of the residual errors in this model (maximum = 0.009 in).

Because of acquisition time did not vary significantly between model types, model selection was based primarily on accuracy and condition number. The normalized form of model type 4 was chosen for both X and Z.

3.4 Sensitivity Analysis of Structured Light Sensors

The question of sensitivity is concerned with the relationship between the accuracy of range measurements to sources of error in the acquisition process. The geometry of the optical paths involved in ranging plays an important role in determining how errors during acquisition are amplified in range measurements.

A Monte Carlo analysis [153] can be used to perform a thorough study of measurement errors. This would involve simulations of the measurement process with representative levels of noise being introduced at each stage. This type of approach is particularly desirable when analyzing a SL sensor with dynamic geometry, as it can be used to study accuracies across the entire measurement space. For PRIME, a simpler approach has been taken.

Referring to Figure 3.15, variations in image coordinates, $d\phi$, produce a displacement of dw at the standoff of the sensor. Because the laser profile is assumed to be roughly horizontal in each camera image and each column is analyzed individually during pixel-level processing, only the sensitivity of height variations to vertical image displacements is considered. (Actually in the center of the image $dy = dw$ for lateral image displacements.) The angle between the laser emission and the camera sighting determines the sensitivity of height errors. As seen in the figure, the amplifying factor is

$$\frac{dz}{dw} = \frac{1}{\sin\theta} \tag{3.10}$$

where θ is the angle between the laser emission and camera sighting. This is related to changes in height by $\sin\theta$.

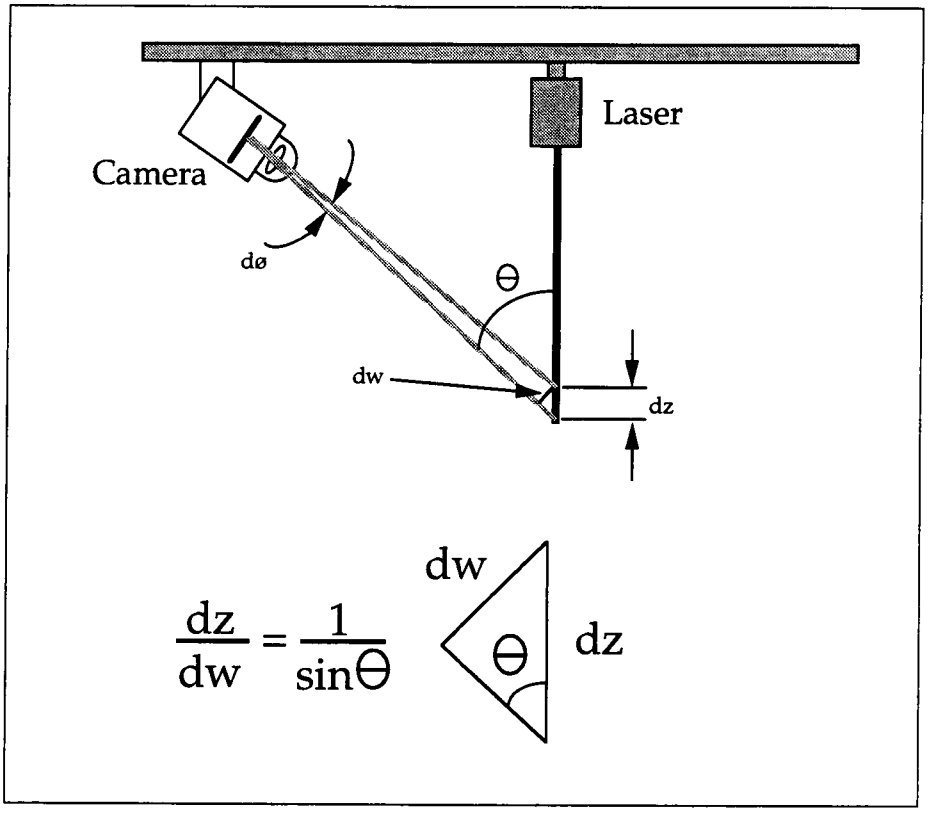


Figure 3.15: The sensitivity dz/dw of a Fixed-Plane SL system is determined by the angle, θ, between the laser emission and the camera sighting. This describes the effect of the change in image coordinates with respect to changes in height.

In Fixed-Plane SL sensors this sensitivity factor varies in a continuous fashion across the laser plane because θ varies somewhat across the laser profile (with motion in and out of the paper in Figure 3.15). For PRIME, the sensitivity varies $1.5 <= \frac{dz}{dw} <= 1.6$. Being near 1, these values are relatively low. As sensitivity improves the degree of shadowing increases. The ranging geometry for PRIME was chosen to somewhat favor sensitivity versus shadowing. Freedom to increase shadowing was deemed acceptable because mostly convex objects with relatively slow rates of curvature were targeted for use with PRIME.

3.5 Performance Benchmarks and Ranging Experiments

Performance benchmarks are very important during sensor research. These give measures of speed and accuracy that are vital for both automated and manual interpretation of data. Example range data and scenes are presented in Figures 3.17 and 3.16, respectively. Images of the scenes were captured by the PRIME video camera with the IR bandpass filter removed. The laser appears as a bright line near the bottom of each image.

A performance metric for active range sensors has been introduced by [17].

$$M = \frac{1}{T^{1/2}} \frac{(D_x D_y D_z)^{1/3}}{(\delta_x \delta_y \delta_z)^{1/3}} \tag{3.11}$$

where T is the point dwell time (sec/point), D_l is the depth of field in the l direction, and δ_l is the measurement uncertainty, also in the l direction. M is the rating. In [17] the dimensionality of contributors, (x, y, z), to this expression is reduced for sensors that generate indexed arrays of range data. This is the case for PRIME in the X direction. Also with PRIME, the depth of field in the direction along the conveyer is somewhat arbitrary, as the conveyer is often run at varying speeds depending on application needs. As in [17], the rating has been reduced

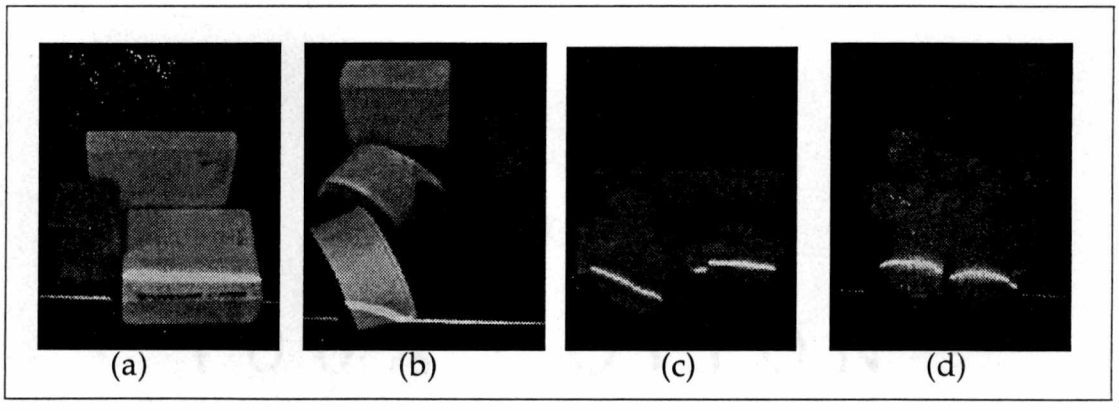


Figure 3.16: Gray scale images captured from the video camera during the ranging process. The optical filter normally used during acquisition was removed for these images, to better illustrate the interaction between the laser plane and object surfaces. The laser appears as a bright line near the bottom of each image.

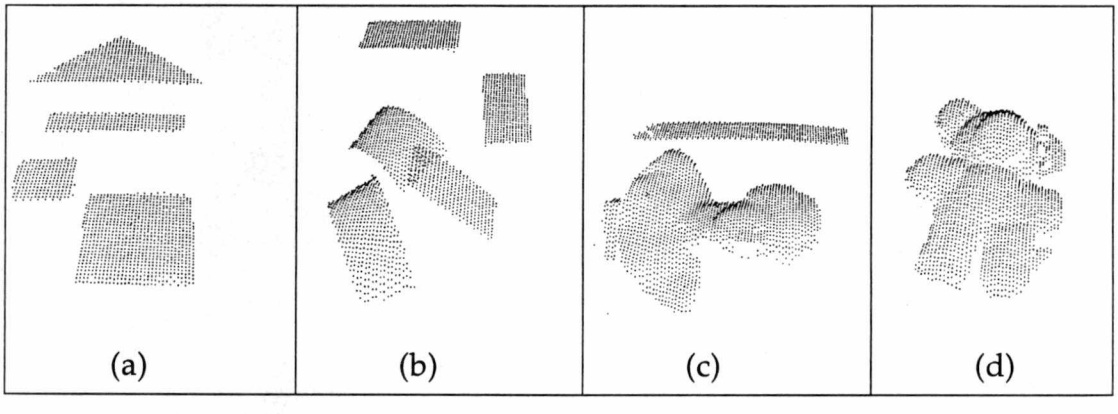


Figure 3.17: Measurements captured by PRIME for objects with a variety of surface textures.

Table 3.5: PRIME performance measures as a function of the type of calibration model. Performance of the normalized vs. unnormalized forms was essentially identical, normalized models were used herein.

Type	δ_z	T	M
1	0.005	0.24	51,500
2	0.004	0.25	70,200
3	0.004	0.25	70,200
4	0.003	0.25	83,900

to

$$M = \frac{D_z}{\delta_z \sqrt{T}}. \tag{3.12}$$

To determine the accuracy of height measurements, a horizontal plate was scanned at a height other than those used during calibration. This plate and the ones used during calibration have a quoted flatness of 0.002 inch variation per foot. At most 6 inches of such a plate are used during calibration and testing. The mean absolute variation of height measurements taken from this type of plate is given by δ_z in Table 3.5.

To determine the point dwell time, a number of scans were made and the mean number of points per video frame was computed. Each type of calibration model was tested. The size of processed imagery was adjusted to maximize the amount of range data, while maintaining frame-rate throughput for each calibration model. The results of these performance benchmarks is given in Table 3.5. These benchmarks were also used in the model selection process. The T values in the table are somewhat conservative and are subject to several percent error, due to empirical nature with which they were determined.

These ratings compare well with those given in [17]. It should be noted that the speed benchmarks quoted here include the application of calibration models. Also, PRIME has been built from commercially available components. This makes

for a system that is generally less expensive and more easily maintained than a custom implementation.

3.6 Concluding Remarks

Despite the longevity of research in Structured Light sensing, a limited amount of published works focus on fundamental design and calibration issues. This chapter has included an introduction to the ranging process, discussions of design tradeoffs, calibration methods, and performance benchmarks. Structured Light ranging has some particularly interesting advantages that allow sensors to be customized for the specific requirements of an application. These sensors can be built "from the ground, up" to yield rugged and inexpensive ranging systems.

PRIME is a Structured Light sensor that has been designed to scan continuously moving objects. PRIME uses a plane of laser light that is mounted in a fixed geometry. The laser illuminates scenes, allowing vertical profiles of range data to be acquired in real time. This approach to 3-D ranging has novel methods of calibration and acquisition providing high accuracy and frame rate data collection.

This completes the discussion of the design and implementation of of calibrated 3-D ranging. This and the following chapters are arranged in order that data is processed in the RDM framework. A discussion of the algorithm and the implementation of a new method for efficiently segmenting range data follows in the next chapter.

CHAPTER 4

Range Profile Tracking: A Profile-Based Approach for Adaptive Range Image Segmentation

Of the three points of focus in this dissertation, range segmentation, graph matching, and calibrated range acquisition, this chapter covers segmentation. The algorithm and implementation of a novel technique are discussed in detail, along with a significant amount of testing.

4.1 Importance and Challenge of Range Segmentation

Segmentation is a fundamental process affecting the overall quality and utility of machine vision systems. For range image analysis to be of practical value and utility, segmentation algorithms should provide results which satisfy often conflicting requirements of speed, accuracy and robustness. The segmentation process is a pixel-level technique for partitioning data into meaningful parts, based on some grouping criteria. The range image segmentation addressed herein assumes the availability of a dense grid of range data, such as that produced by a laser range finder or structured light sensor [99]. In general the grouping criteria will vary depending on the nature of the sensor and on the application. Grouping criteria are typically based on either edge or region features. Edge-based techniques exploit pixel dissimilarities where region-based approaches compute uniformity measures [154]. The groups produced by a segmentation algorithm can be used for object recognition, surface reconstruction or other purposes.

Measurement noise makes segmentation challenging because it corrupts the

uniformity of range data. Two types of noise are common with ranging sensors. Range measurements are often corrupted by lower intensity Gaussian noise and by higher intensity noise spikes. The Gaussian noise typically corrupts every point. Spike noise usually occurs with a low likelihood, erupting in a binary fashion. The effects of these noise sources are manifested in segmentation results by errors in surface fits and surface boundaries. The goal of a segmentation algorithm is to maintain correct partitioning in the presence of noise sources. The term robustness is used to describe the ability of an algorithm to maintain correct results in the presence of outlier contamination, such as spike noise. This definition is similar to that used in the area of robust statistics [155] and is becoming more common in Computer Vision literature [154] [33] [36]. In object recognition applications it is also important that a segmentation technique yield a stable decomposition of range points into these meaningful components [11] [33]. Stability here implys that small local changes in a surface do not radically alter the description of the component parts. Robustness, stability and accuracy are all important performance metrics for range segmentation algorithms.

4.2 Range Profile Tracking (RPT) Approach

In this chapter, a new approach to range image segmentation is presented which balances accuracy, stability and speed tradeoffs. Alternative techniques have been reviewed in Chapter 2, where a taxonomy based in processing "kernels" was introduced. The uniqueness of this new approach is derived from the combination of its Kernel-Real Estate, -Placement and -Features, all of which use adaptive methodologies.

1) The Kernel-Real Estate adapts to the effects of outliers in order to provide dense regions of varying size that are robustly formed.

2) The Kernel-Placement strategy adaptively locates boundaries of kernel re-

gions to avoid outlier contamination. Here, the kernel boundaries are not located by a fixed geometry. Also, the Kernel-Placement is a non-iterative scheme, providing more deterministic processing.

3) Kernel-Features are found in a computationally-efficient manner that permits second order curvature to be estimated. The Kernel-Feature uses a Gaussian noise model of measurements. This provides a basis for outlier rejection and is used to help determine kernel boundaries.

Adaptive kernel processing allows a range segmentation algorithm to avoid the effects of outliers. In RPT, kernel processing is approached as a real-time process of shape-profile tracking. Two orthogonal sets of profiles are tracked across a range image using a Kalman filter-based technique. The Kalman trackers are setup with the capability to follow a second order height variation, such as $z = f(x) = ax^2 + bx + c$. Portions of a vertical profile that maintain a consistent second order variation in curvature are segmented as one complete "strip". The RPT Kernel-Feature is the Kalman filter (KF) state vector. The Kernel-Real Estate and -Placement are determined adaptively using both a Kalman-based predictor and a residual error criteria.

The entire RPT process is described in Fig. 4.1 which shows a breakdown into three components. The first step involves the actual profile tracking. This includes all of the adaptive kernel processing and results in two orthogonal sets of 1-D strips. These strips identify portions of the range data having consistent second order curvature. The second processing step merges these orthogonal strips into 2-D seed regions. A polynomial surface is then fit to each region and used as a consistency criteria for seed growing operations.

4.2.1 Tracking Profiles for an Initial Segmentation

Range profiles are tracked in order to provide an initial 1-D segmentation. RPT forms two sets of orthogonal 1-D strips. The Strip Formation process is accom-

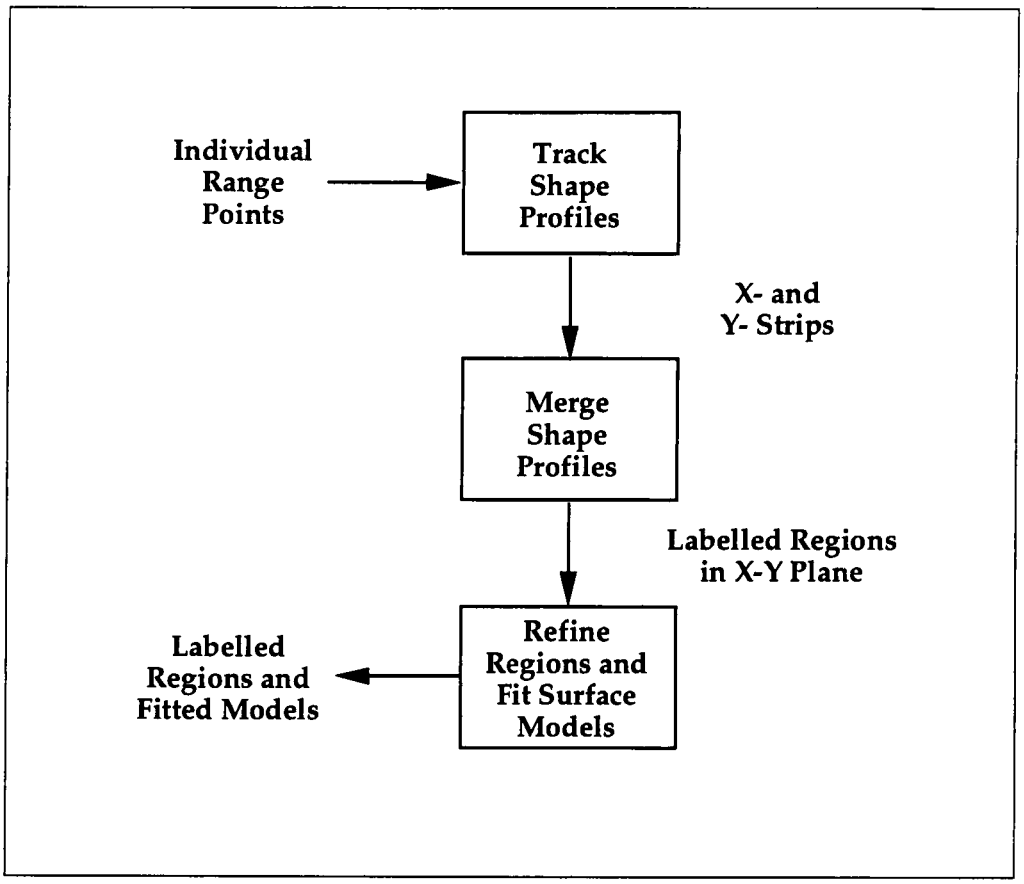


Figure 4.1: RPT has three fundamental steps. "Track Shape Profiles" forms 1-D strips of range data having uniform second order curvature. "Merge Shape Profiles" forms seed regions having 2-D extent. These initial regions are then refined using a seed growing method to improve the accuracy of region boundaries.

plished robustly and in real-time. Real-time here refers to the fact that strips are formed as the profiles arrive from the acqusition system, at video rates. Figure 4.2 depicts one of the orthogonal sets of strips from a top view. These "X-Strips" grow in the X direction. As seen in the figure, a single profile of range data from the sensor provides a column of new points extending in the Y direction. As each profile of range data is acquired, it is accumulated into both the X- and Y-Strips. The X-Strips run orthogonal to the sensor's profile of range data. Each X-Strip is provided with a single point from a sensor profile and advances in length one step. A series of Y-Strips are formed end-to-end along the entire length of each sensor profile.

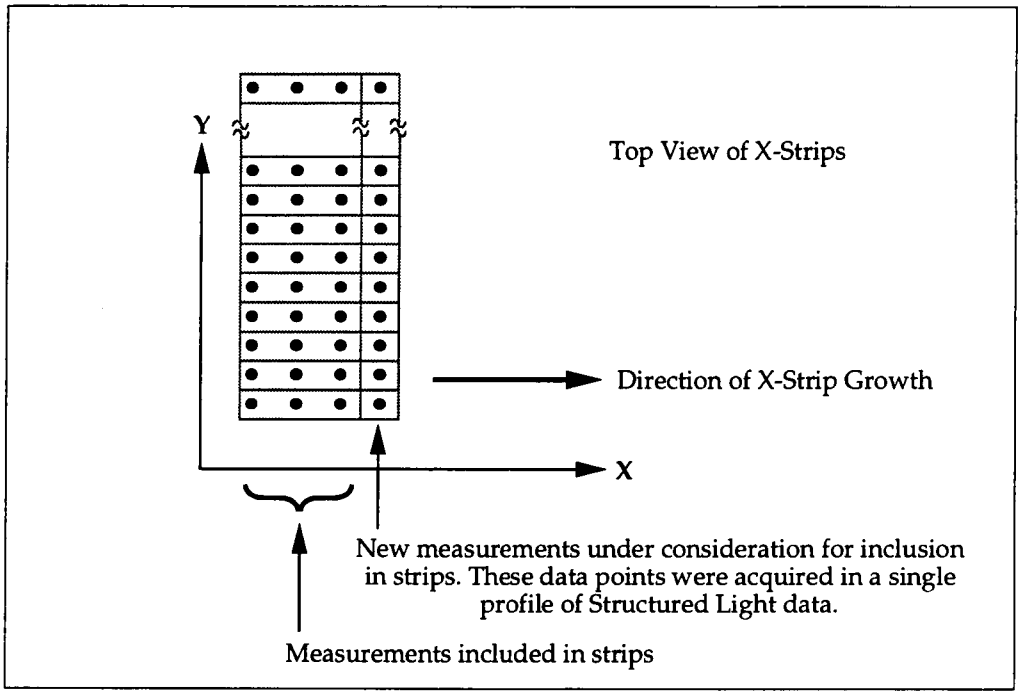


Figure 4.2: Many narrow strips of range data are formed using RPT. At each iteration new measurements are added to each strip and the Kalman filter's states are updated. 1 of 2 orthogonal sets of strips are shown. These "X-Strips" grow in the X direction.

As each sensor profile arrives, measurements are accumulated into various

KFs and decisions are made to possibly terminate a given strip. New strips are positioned end-to-end with their predecessors. Two tests are used to determine strip extent.

The first check occurs prior to the accumulation of a new measurement into a strip, the second occurs after its inclusion. The first test checks for surface discontinuities and outliers. The distance between a range point and its expected location is weighted and compared to a threshold. The weighted distance calculation is performed using the Mahalanobis distance [156], in a manner similar to that used in many predictive tracking applications [22,145].

The KF process noise parameter determines a covariance matrix that describes the expected variation of measurements away from the filter state. Figure 4.3 depicts a side view of a strip, the measurement currently under scrutiny is encircled and has been found to exceed the discontinuity threshold. This event determines the end of a strip. The anticipated position of the measurement is indicated by the gray dot and the discontinuity threshold by the gray I-bar.

Unfortunately the discontinuity threshold is insufficient as a sole criteria for determining strip boundaries. An accompanying on-line residual error check has proven worthwhile to complete the Strip Formation process. Figure 4.4 illustrates the shortcoming of using the discontinuity threshold alone. The gradual curvature of the object seen in this case will tend to steer the KF state upward, without triggering the discontinuity criteria. This is consistent with the fact that no height discontinuity exists in this region. However the given object's surface is somewhat exponential in shape - it is not second order. Hence a segmentation of this surface is desired.

Figure 4.5 describes the on-line residual error check [150]. The process of finding the mean residual error of the first R contributors to the strip is illustrated. This mean is compared to a threshold to determine strip termination. In the figure the state of the KF has begun to "curve upward" due to the gradual transition

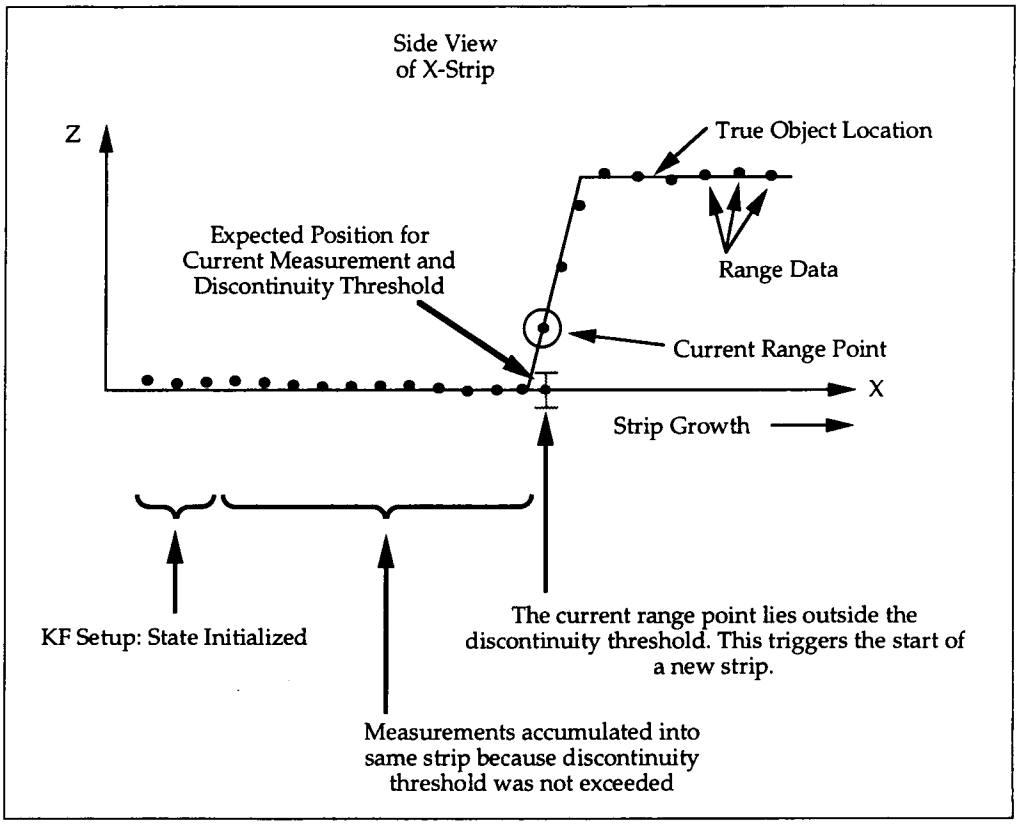


Figure 4.3: The discontinuity threshold helps to determine the lengthwise extent of a strip. A Kalman filter is associated with each strip and is used to both accumulate range measurements and to provide a basis for the threshold.

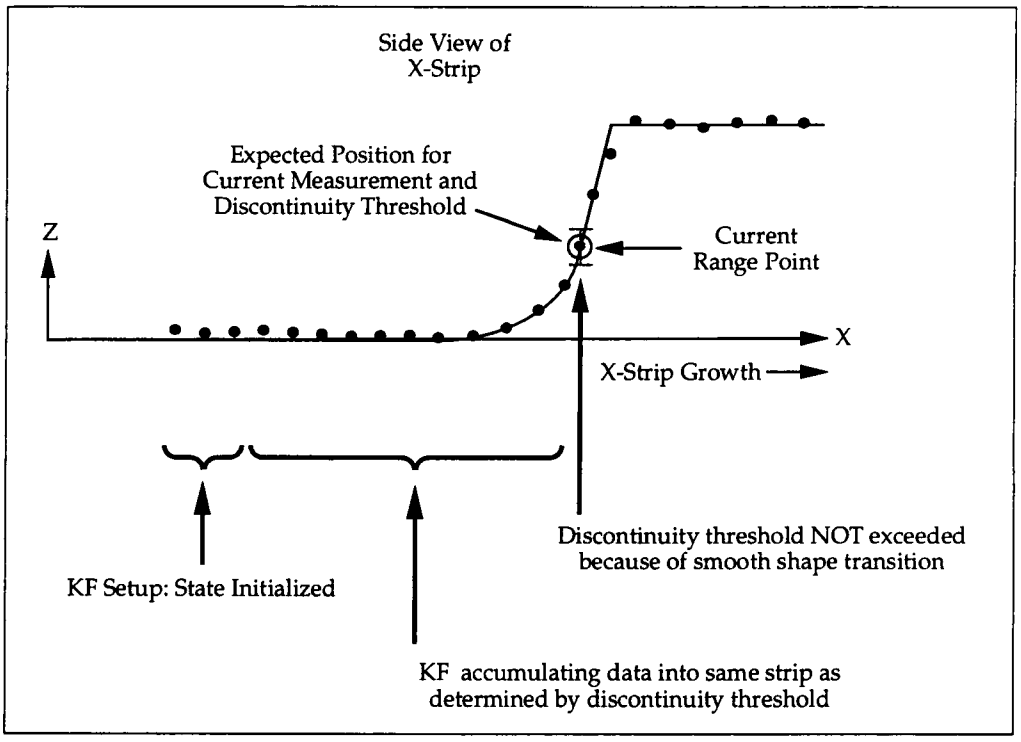


Figure 4.4: Using the discontinuity threshold alone for Strip Formation is insufficient - as seen in this example. This criteria is somewhat "open-loop" in nature. A double check is needed during Strip Formation. Here, the object's shape within the strip shown is not second order in nature, therefore it should be segmented into two components.

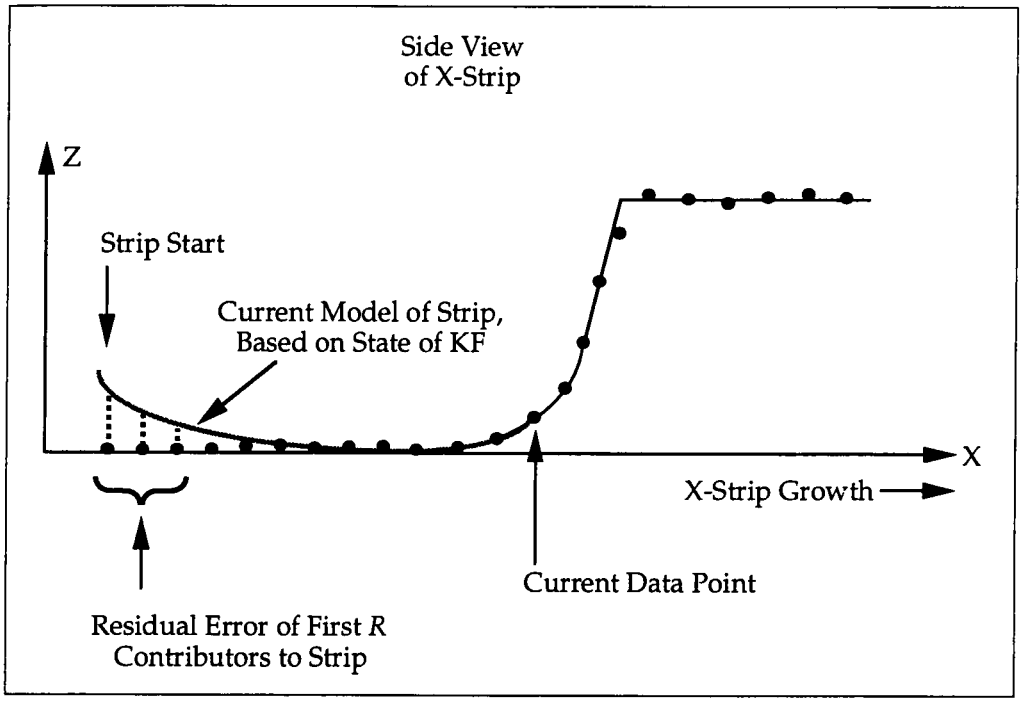


Figure 4.5: To double check the validity of a KF model, the residual error of the first R contributors to the strip is compared to a threshold. The earliest contributors have the largest moment arm for their residual errors.

of the object's shape. The need to terminate the strip is seen by the increased residual error of the strip. Both the residual error and the discontinuity criteria are used to determine strip termination.

The residual error check is found using the KF state $\begin{vmatrix} z & \frac{dz}{dx} & \frac{d^2z}{dx^2} \end{vmatrix}^T$ and summing over the first R members of the strip

$$r_{error} = \sum_{i=1}^{R} |z_0 + \frac{dz}{dx}(i\delta x) + \frac{1}{2}\frac{d^2z}{dx^2}(i\delta x)^2 - z_i^m| \tag{4.1}$$

where $|.|$ denotes absolute value, δx is the grid increment and z_0 is the **modeled** height of the first entry in the strip. For a strip of length L, $z_0 = z - \frac{dz}{dx}(L\delta x) - \frac{1}{2}\frac{d^2z}{dx^2}(L\delta x)^2$. The first R contributors to the strip are used because these will tend to have a greater moment arm to amplify the residual error, as compared to strip contributors that are near the end of the strip. A fixed value of R also places a maximum limit on the amount of kernel processing that occurs on each iteration.

The discontinuity check is somewhat "open loop" in nature because it doesn't include any verification of the appropriateness of the KF state's characterization of a strip. The residual error check tends to "close the loop" in the Kalman accumulation process by testing strip contributors.

Figure 4.6 shows the results of the Strip Formation process operating on a synthetic scene having Gaussian noise with $\sigma_m = 0.010$ in. To make the presentation clearer, the strips that were formed along the "floor" of the scene are absent.

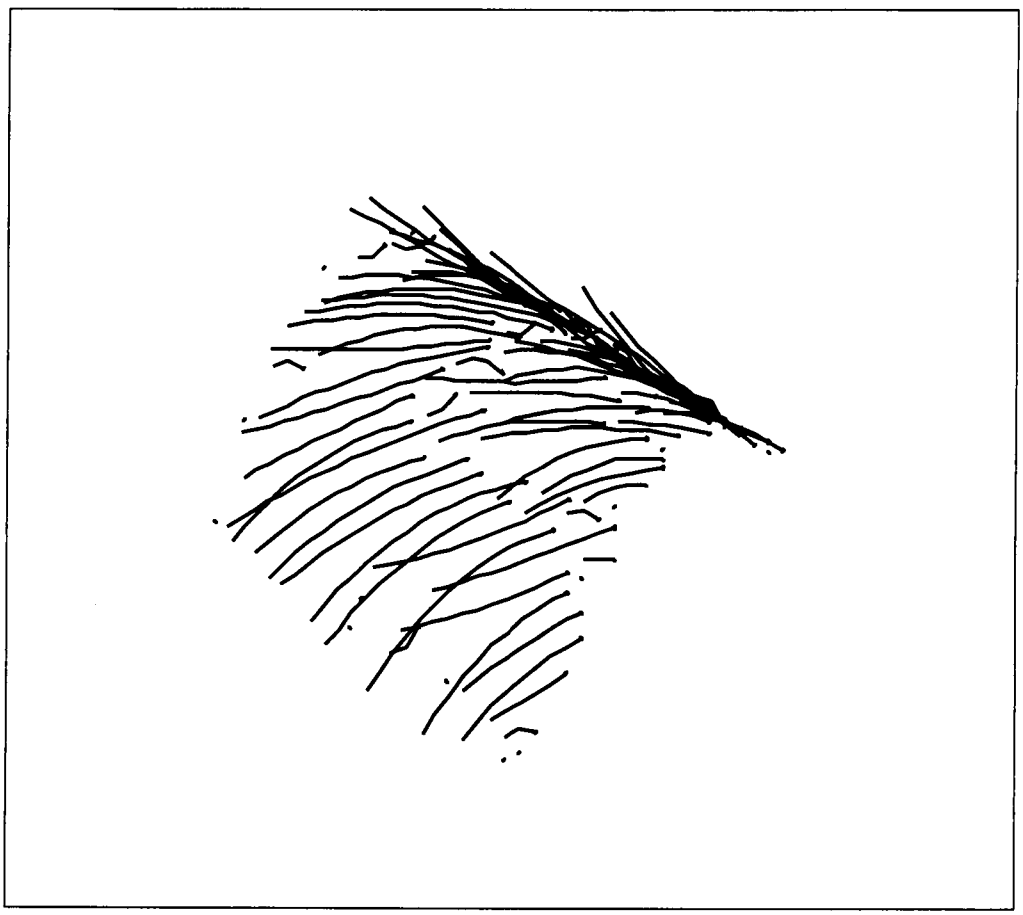


Figure 4.6: Results of Strip Formation in the X direction. The original synthetic object was contaminated with Gaussian noise having $\sigma_m = 0.010$ in.

4.2.2 Merging Profiles to Form Regions

X- and Y-Strip Formation is a process using adaptive kernels to build two orthogonal sets of 1-D regions. This profile-based approach has advantages of speed, accuracy and robustness. However, the Strip Formation process does necessitate a merging step to form regions with 2-D extent. Strip Formation occurs on-line, while PRIME is capturing profiles of range data, where Strip Merging is a post processing step. This merging process is noniterative and yields labeled regions.

Strips quantify smoothness in a given direction. Adjacent strips that are parallel and side-by-side are merged based on smoothness in the orthogonal direction. Initially each X-Strip is given a unique region label and the Y-Strips are then traversed, noting the X region labels that are crossed within the interior of a Y-Strip.

Evidence of equivalent X region labels is accumulated in an equivalence table, similar to that used in a connected component analysis [142]. Standard equivalence tables are nominally setup as a lower triangular NxN matrix with nonzero entries for any equivalent labels. Equivalence data stored in the table can be used to establish a set of labels having the fewest possible members and having the lowest possible label values. In the RPT formulation, the equivalence table does not contain binary entries, but rather, the number of occurences that a pair of region labels have been observed as being equivalent. During the propagation of equivalent labels, only the maximum entry in each row was used. This technique benefitted the overall performance of RPT because it limited the growth of 2-D seed regions somewhat. This was advantageous because it permitted the use of a true 2-D uniformity criteria to be applied prior to the establishment of final segmented regions.

In the current version of RPT, X-Strips are merged based on smoothness in the Y direction. This forms regions denoted by $R_i^{x/y}$. Experiments have been performed in which the roles of X and Y were swapped and in which both two

types of regions $R^{x/y}$ and $R^{y/x}$, were found and merged. Significant improvements were not achieved using these variations. For simplicity only the $R_i^{x/y}$ regions are currently used.

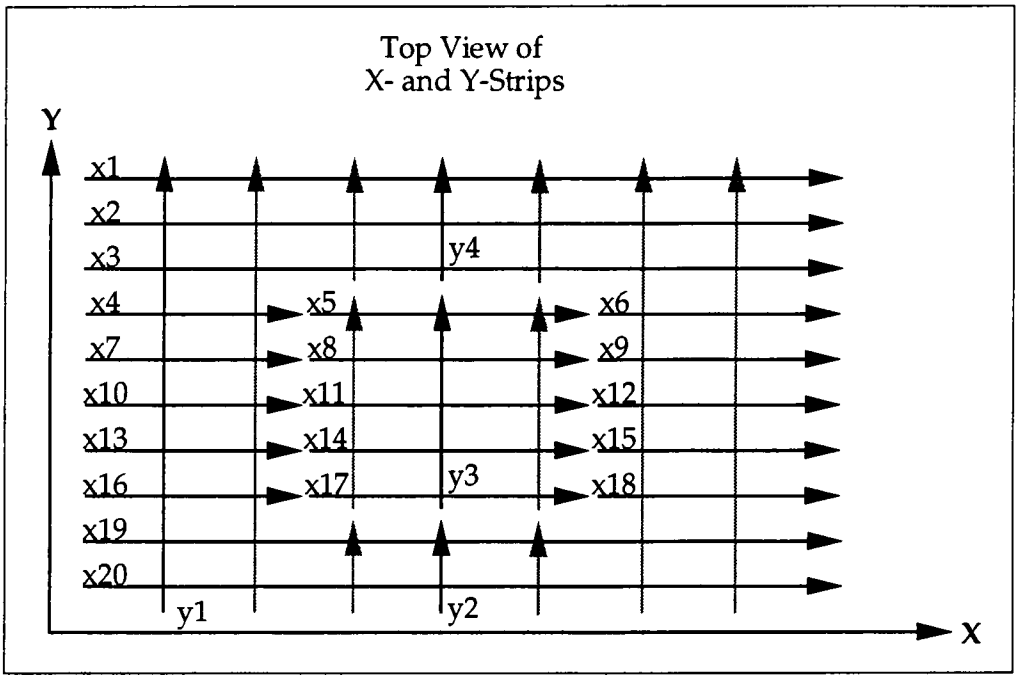


Figure 4.7: The orthogonal strip-merging process. Smoothness in the Y direction is used as a criteria to merge adjacent X-Strips. Only two columns of Y-Strips are highlighted here, for clearity.

In a simple example, Figure 4.7 illustrates a top view of some strips. Each X strip has been labeled with an x_i designation. Only two columns of Y-Strips are examined in the diagram, for clearity (strips y_1 to y_4). The smoothness in the Y direction indicated by strip y_2 provides evidence that strips x_{19} and x_{20} should be merged. Table 4.1 shows all the merging data provided by these two columns of Y-Strips.

Table 4.1: The following equivalences of X-Strips have been found by traversing the given Y-Strips seen in the example.

Y-Strip	Mergeable X-Strips
y_1	x_1 x_2 x_3 x_4 x_7 x_{10} x_{13} x_{16} x_{19} x_{20}
y_2	x_{19} x_{20}
y_3	x_5 x_8 x_{11} x_{14} x_{17}
y_4	x_1 x_2 x_3

4.2.3 Region Refinement Process

The first two steps in RPT produce relatively large and uniform seed regions. However, it is not until this region refinement stage that a true 2-D uniformity criteria is applied. Surface models of the form $z = f(x, y)$ are fit to each region and are used as the basis for the uniformity criteria.

The membership of points in each region requires refinement in order to produce proper segmentation boundaries and surface approximations. This type of refinement is necessary particularly when curvature discontinuities are present (see [35] and section 4.4). When this type of discontinuity is present, local uniformity measures are unable to determine the appropriate boundaries for regions. This is due to the lack of any kind of local discontinuity. These conditions can produce seed regions that are inappropriate for the final segmentation. Hence both seed merging and some region modification are required during the region refinement process. Figure 4.9 shows the processing steps in the RPT algorithm. Figure 4.8 shows the results of final surface fitting for the object seen in Figure 4.6.

The region refinement process begins by fitting a second order surface to each region via least squares [157]. The absolute residual error of each point in the region is then computed and the median of these errors (MdAR) is found. The median provides a robust measure of the residual error associated with inliers of a region [36]. This provides a more correct gauge to weight outliers than a mean absolute residual, for example [150]. The median is computed using a technique requiring $N \log N$ effort [150].

Once polynomial surface models are available for each region, these provide a means to determine consistency of new points that are considered for inclusion into a region. Regions are grown at their periphery in an iterative fashion. Regions are grown one at a time, in order of their increasing MdAR. Each unclaimed pixel in the range image that is at the boundary of the current region is considered for inclusion. The consistency of the point is measured by normalizing the absolute

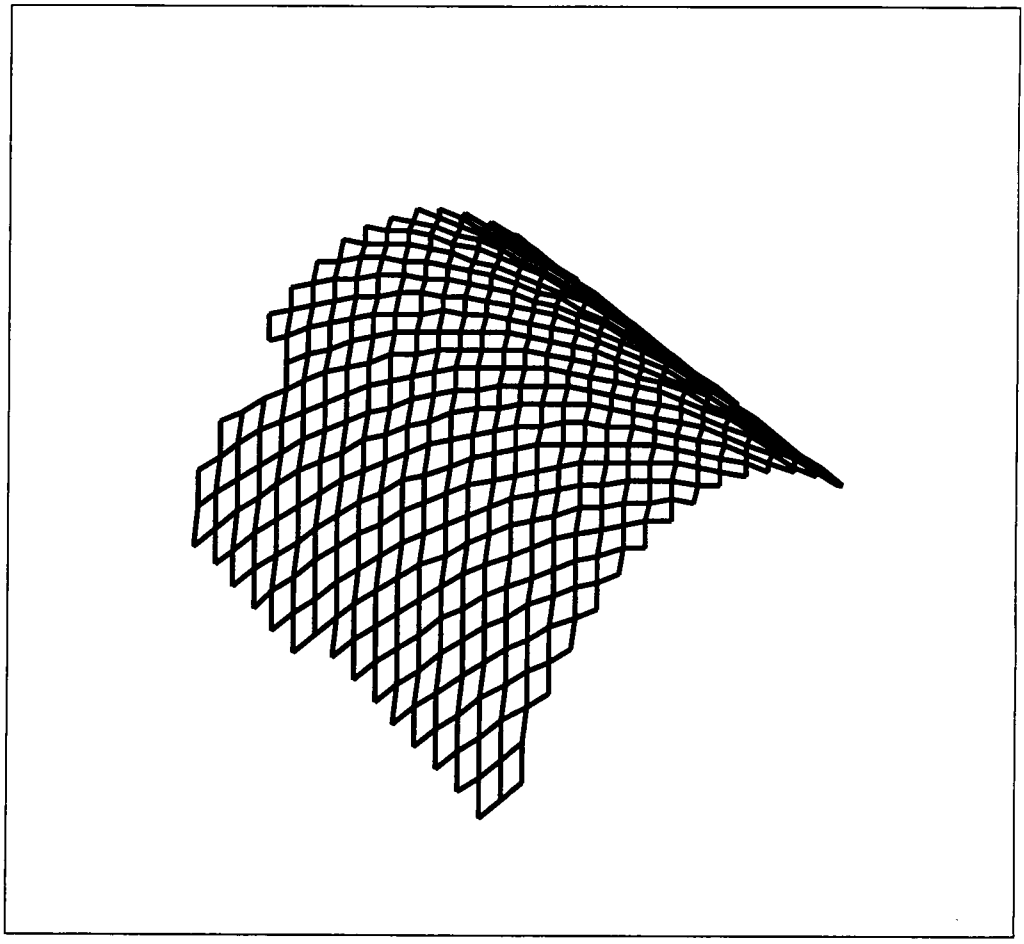


Figure 4.8: Results of fitting a second order surface over a synthetic object using RPT.

Initialization
Drive all KFs to steady state.
Initialize all X- and Y-Strips.

Profile Tracking
Acquire SL Profile.
Accumulate Profile into Each Filter.
    Check discontinuity threshold for measurement,
      initialize new Strip if needed.
    Accumulate measurement into filter.
    Check residual error of Strip,
      initialize new Strip if needed.

Profile Merging
Label interior of each X- and Y-Strip as distinct region.
Tally occurances of mergable X regions based on Y smoothness.
Propagate maximally-equivalent X region labels.
Assign equivalent X region labels, forming initial 2-D seed regions.

Region Refinement
Fit second order surface to each region via least squares.
Find MdAR for each region, split regions with poor residuals.
Drop tiny regions.
Loop twice
  Index regions by increasing MdAR.
  Grow each region at its periphery into all unclaimed pixels that
    maintain consistent curvature. Grow regions with lower MdAR first.
  Fit second order surface to each region via least squares.
  Find MdAR for each region.
  Single iteration of reweighted least squares.
  Split regions with poor residuals and refit surface.
  Drop tiny regions.

Figure 4.9: RPT algorithm. Comments appear in bold.

fitting error of the point with the MdAR. To be included in a region the normalized fit error must not exceed a threshold of 5 *i.e.* a new point cannot vary from the anticipated height of the region (as predicted by the surface model) by more than 5 times the MdAR.

Once each region has been given a chance to grow to its fullest extent, new surface models are computed. This is accomplished using least squares and the new MdARs are found, as before. A single iteration of reweighted least squares is also performed. The Tukey Biweight function is used to reweight each point (normalized by the MdAR) prior to the least squares fitting [158].

Checks are made on the size of each region during the refinement process so that very small ones can be eliminated. The MdAR is also checked in order to judge the overall fit of each region. Regions with poor fits are assumed to span curvature discontinuites, as discussed above. These regions are split using a 3x3 grid that is sized to the region, resulting in 9 subregions. The new subregions are fit with polynomial models and the region growing process begins again. This splitting operation is repeated once.

4.3 RPT Formulation

The form of models used for profile tracking, $z = f(x), z = f(y)$ and for surface approximations $z = f(x, y)$ have been parameterized in a consistent style. A synergy is present because of this consistency. The mutual benefit arises from the creation of profile segments which are consistent with the form of the surface patch model. The formulation of profile and surface models is critical to the speed, accuracy, stability and robustness of RPT.

To track a profile of the form $z = f(x)$, Kalman filters (KF) have been setup to follow the second order height variations. This involves using a state vector of the form $\left| \begin{array}{ccc} z & \frac{dz}{dx} & \frac{d^2z}{dx^2} \end{array} \right|^T$. This formulation is very similar to the tracking of a

time-domain signal that is modeled with a constant temporal acceleration, but for the exchange of the independent variable. More details on the Kalman filter implementation are provided below.

4.3.1 State Space Model of the Measurement Process

A state space system is used to describe the ranging process including noise sources. The state of this system describes the true object surface. State estimation reveals the surface curvature in the presence of noise and is achieved using a Kalman filter. The Kalman state vector is the Kernel-Feature of RPT.

The KF state describes the curvature of an X- or Y-Strip from the perspective of its endpoint. The state includes the current height and slope of the tangent as well as the strip's vertical acceleration - which is a constant. This corresponds to a state vector $\mathbf{a} = \left| z \quad \frac{dz}{dx} \quad \frac{d^2z}{dx^2} \right|_i^T$. A continuous model of the system with zero noise would have $\frac{d^3z(x)}{dx^3} = 0$. In practice, changes in acceleration are due to process noise $\frac{d^3z(x)}{dx^3} = \tilde{\nu}(x)$ which is assumed to have a zero mean and a Gaussian density [145]. The "Process Noise" associated with shape estimation is simply the variation of surfaces beyond second order curvature. The measurement system, expressed in terms of a continuous position variable (x), is given by

$$\frac{d\mathbf{a}(x)}{dx} = \mathbf{A}\mathbf{a}(x) + \begin{vmatrix} 0 \\ 0 \\ 1 \end{vmatrix} \tilde{\nu}(x) \tag{4.2}$$

$$z_m = \mathbf{H}\mathbf{a}(x) + \begin{vmatrix} 0 \\ 0 \\ 1 \end{vmatrix} \tilde{w}(x) \tag{4.3}$$

where

$$\mathbf{A} = \begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{vmatrix}, \tag{4.4}$$

$\mathbf{H}$ is the Kalman filter's measurement matrix

$$\mathbf{H} = \begin{vmatrix} 1 & 0 & 0 \end{vmatrix} \tag{4.5}$$

and z_m is a height measurement. The measurement noise $\tilde{w}(x)$ is assumed to be Gaussian with zero mean. This is a standard formulation, but for the spatial derivatives.

Kalman filters are used to estimate the state of this system. To design the filters, a discrete form of the above matrices is required along with covariance matrices for the process noise and measurement uncertainty [144]. The state transition matrix associated with a sampling period of δx is

$$\mathbf{\Phi} = e^{\mathbf{A}\delta x} = \begin{vmatrix} 1 & \delta x & \frac{1}{2}\delta x^2 \\ 0 & 1 & \delta x \\ 0 & 0 & 1 \end{vmatrix}. \tag{4.6}$$

The discrete form of the measurement matrix, $\mathbf{H}$, contains the same numerical values for both the continuous and discrete domains. It is assumed that the measurement uncertainty remains constant, and because measurements are scalar heights, the measurement covariance matrix reduces to a scalar variance σ_m^2. The intensity of the process noise is also assumed to remain constant. This yields a covariance matrix for the process noise of

$$\mathbf{Q} = \begin{vmatrix} \delta x^5/20 & \delta x^4/8 & \delta x^3/6 \\ \delta x^4/8 & \delta x^3/3 & \delta x^2/2 \\ \delta x^3/6 & \delta x^2/2 & \delta x \end{vmatrix} q. \tag{4.7}$$

where q is the process noise parameter [145].

There are interesting tradeoffs between the measurement uncertainty σ_m^2 and the process noise q [144]. Relatively large q values will encourage RPT to track surfaces closely, but will also permit measurement noise to corrupt strip curvature, tending to shorten strips due to the residual error limits. Relatively large σ_m discourage the KF from trusting new measurements. This worsens the tracking ability of the strips, but does reject more noise.

A basis for selecting σ_m is available using data found during the sensor calibration process. As a validity check during calibration, the residual error between world coordinates and modeled approximations are computed. The mean residual error of the modeling errors is used to approximate σ_m. For tests on synthetic scenes, σ_m was varied and is reported along with the test results. For tests on real data $\sigma_m = 0.015$ in. The process noise parameter was determined experimentally. In all tests described herein $q = 100$. Because of the low sensor noise associated with SL and a desire for high accuracy, q was chosen proportionally much higher than σ_m.

4.3.2 The Steady State Kalman Filter

The nominal Kalman filter formulation allows for a number of time-varying quantities, such as the measurement matrix, process noise, measurement uncertainty and state transisition matrix. In RPT these quantities are time invariant. Given this and conditions on observability and controllability of the state, the state covariance and predicted state covariance matrices will converge [145]. The Kalman Gain matrix will also converge a steady state value. This has a significant computational benefit which is exploited in RPT. Time invariant conditions are also beneficial because they imply that the measurement uncertainty remains constant, and this assures that the discontinuity threshold used during Strip Formation can remain a simple constant.

The observability requirement ensures that sufficient state information will be available to prevent the state uncertainty from becoming unbounded. This is determined by the measurement and state transition matrices, $\mathbf{H}$ and $\mathbf{\Phi}$. The controllability requirement guarantees that some process noise will enter each component of the state vector. This prevents the covariance of any component of the state estimate from converging to zero [145]. Near-zero estimates of state covariance limit state changes and reduce accuracies in surface estimates.

Intuitively, it can be appreciated that the observability and controllability conditions in this measurement system are satisfied. Process variations alter the state's slope-derivative component and are integrated into each of the other state components over time (controllability). Also because of the integral relationships between height, slope and slope-derivative, variations in each state component are observed through the system's estimated height values (observability). In [145] the steady state conditions are verified for this type of constant acceleration filter.

In RPT, the nominal formulation for Kalman filter equations are used during the filter initialization process [145]. A zero measurement is applied to the filter on successive iterations until the various KF matrices converge. Convergence is monitored by subtracting subsequent Kalman Gain matrices and applying the infinity norm [149]. After convergence, $\mathbf{\Lambda}^{-1}$ and $\mathbf{K_i}$ are copied for use with the steady state version of the KF equations. These "stead-state" calculations are given in Eq. 4.9 through 4.11

Before accumulating a new measurement z_i^m into the filter, a discontinuity threshold is applied to test the validity of including z_i^m in the current strip. The discontinuity threshold is expressed in terms of the Mahalanobis distance, d_i, that describes the distance of the measurement away from the estimate of the current state [22,145].

$$d_i = (z_i^m - \mathbf{H}\tilde{\mathbf{a}}_\mathbf{i})^T \mathbf{\Lambda}^{-1} (z_i^m - \mathbf{H}\tilde{\mathbf{a}}_\mathbf{i}) \tag{4.8}$$

where $\mathbf{\Lambda}$ is the measurement covariance matrix. For RPT, in steady state, this

reduces to

$$d_i = (z_i^m - \tilde{z}_i)^2/\sigma_m^2 \tag{4.9}$$

where $\tilde{z}_i$ is the current height estimate, which is available in the vector containing the predicted state. The accumulation of a new measurement into the filter requires only 2 steps in steady state. First, the state must be advanced with

$$\mathbf{a_i} = \tilde{\mathbf{a}}_\mathbf{i} + \mathbf{K_i}(z_i^m - \mathbf{H}\tilde{\mathbf{a}}_\mathbf{i}) \tag{4.10}$$

where, $\mathbf{K_i}$ is the Kalman Gain matrix. Second, the state estimate for the next iteration is found with

$$\tilde{\mathbf{a}}_{\mathbf{i+1}} = \mathbf{\Phi a_i}. \tag{4.11}$$

4.3.3 Initialization of the Kalman Filter State

To start a new strip at the i_{th} position, three measurements $(z_i^m, z_{i+1}^m, z_{i+2}^m)$ are required. A second order curve is traced through these points to determine the filter's initial state $\left| \; z \;\; \frac{dz}{dx} \;\; \frac{d^2z}{dx^2} \; \right|_i^T$. Assuming a constant acceleration and using the incremental relationship

$$z_{i+1} = z_i + \frac{dz}{dx}_i \delta x + 0.5\frac{d^2z}{dx^2}\delta x^2 \tag{4.12}$$

yields an expression for the initial state

$$\left| \; z \;\; \frac{dz}{dx} \;\; \frac{d^2z}{dx^2} \; \right|_i^T = \left| \; z_i^m \;\; \frac{-3z_i^m + 4z_{i+1}^m - z_{i+2}^m}{2\delta x} \;\; \frac{z_i^m - 2z_{i+1}^m + z_{i+2}^m}{\delta x^2} \; \right|^T \tag{4.13}$$

Because of the assumption of a constant acceleration, $\frac{d^2z}{dx^2}_i = \frac{d^2z}{dx^2}_j = \frac{d^2z}{dx^2}$ for all i, j within a strip. Hence the subscripts on the height acceleration terms have been dropped. A similar procedure is used for the Kalman filters that process data in the Y direction.

Because three points are needed to initialize the KF state, a profile delay buffer has been introduced in between the range acquisition and the Strip Formation stages. This allows the points (z_{i+1}^m, z_{i+2}^m) to be readily accessed.

4.4 Experimental Validation of RPT

Experimental validation of RPT was necessary in order to quantify RPT performance. These metrics include speed, accuracy, stability and robustness. A controlled environment was needed that allowed ground truth to be compared with processed results. Synthetic scenes provided this type of conditions. Testing with synthetic data provided a basis for comparison against the analysis of scenes scanned by a novel Structured Light (SL) testbed that has been designed, built and calibrated as a part of this study. In order to achieve statistically valid estimates of performance measures, Monte Carlo analyses [153] were run in large numbers with randomly generated scenes.

In the tests reported below a 100x100 grid of range data was generated. Scene styles varied. Range data was corrupted with both zero-mean Gaussian and spike noise. The standard deviation (σ_m) of the noise Gaussian was selected to vary from below that of the SL sensor on the RPT Testbed to well above it. Tests ramped the Gaussian noise level from $\sigma_m = 0.010$ to 0.100 inches. The SL sensor used with RPT does not typically generate spike noise. Nevertheless, noise spikes were added to the synthetic data sets to demonstrate the robustness and versatility of RPT. The probability of spikes occurring in each data set was 5 percent. One hundred randomly-generated scenes were processed at each Gaussian noise level.

A Signal-to-Noise (SNR) has been used to describe the noise. This was calculated by examining the N contributors to a shape

$$SNR = 10 \log \sum_{i=1}^{N} (z_i^t)^2 / \sum_{i=1}^{N} (z_i^t - z_i^m)^2 \quad db \tag{4.14}$$

where z_i^t is the true shape height and z_i^m is the corrupted value [159]. The SNR values were dominated by the measurement deviations introduced by spike noise. The low level Gaussian noise barely effected the SNR values. Hence the SNR is quoted only once for each group of tests below.

A single object within a scene was used for performance assessment on each trial run. Because test runs can have multiple output regions in general, it was necessary to have an automated means of selecting the proper output region for comparison with ground truth. This selection was based on centroid location. The output region with the closest matching centroid was compared to the true object to assess performance on a given trial. Three measures have been used to quantify RPT performance:

(1) The percentage of trials with correct number of segmented regions,

(2) The percent error in volume, and

(3) The percent error in centroid location.

The percentage of trials with the correct number of final regions is recorded under "Percent Trials OK", in the tables. Tables A.1 through A.4. Only on trials with the correct final number of regions were the other performance measures tallied.

4.4.1 RPT Testing Trials

Tests of RPT have been performed on scenes containing jump, crease and curvature discontinuities. Performance with each type of discontinuity has been examined separately. A jump discontinuity refers to an edge within the range data having a step change in height. In the first series of tests (see Table A.1) a planar surface bounded by jump discontinuities was randomly generated in each scene. The height, width, orientation, location and slope (0.0 to 1.0) were all randomly selected for the object. Figure A.1 illustrates a typical surface in these tests. The original measurements including spike noise are seen in (a). X- and Y-Strips are shown in (c) and (d), respectively and the final fitted surface appears in (d). Note that measurements, strips and surfaces on the "floor" are absent from these plots.

The second series of tests examined the performance of RPT with a second

order surface bounded by jump discontinuities. See Fig. A.2 and Table A.2. The height, width, orientation, and location of these objects were all randomly selected. Typical measurements, the intermediate and final results are all shown in the figure.

A crease discontinuity refers to an edge within the range data having a step change in the orientation of surface normals, but no discontinuity in height. To study the performance of RPT under these conditions, a V-shaped object was generated having two planar surfaces that met in a central trough. The height, width, orientation, location and slope (0.8 to 1.0) were all randomly selected. One of the two planar surfaces was chosen as a basis for computing the performance measures for each trial. See Fig. A.3 and Table A.3.

The last style of testing examined performance when segmenting surfaces bounded by curvature discontinuities [35]. These boundaries have continuous height and slope with an abrupt change in slope derivative. The objects generated for these scenes had a curved midsection that smoothly joined two planar sections, as seen in Fig. A.4. This style of shape is similar to that used in [33]. The object was randomly positioned in the scene on each trial. The curved midsection was used to the compute performance metrics, see Table A.4.

4.4.2 RPT Performance Summary

Test results indicate a high degree of robustness, stability and accuracy can be attained by RPT at high speeds. A total of 4000 trials were run in all. The "Percent Trials OK" indicates the percentage of 100 trials in which the number of final segmented surfaces was correct in each style of test. This reflects the degree of stability of RPT. The low percent error in the centroid and volume calculations indicate that both accurate and robust results are achievable in the presence of Gaussian and spike noise. The duration of on-line processing is within frame rate and has a very low standard deviation, indicating very promising utility for real-

time operation. Offline processing for these scenes typically required a couple seconds or less. Note that the statistics on the various performance metrics were only collected for trials having the correct number of final segmented surfaces.

The Structured Light sensor developed for use with RPT has a measurement noise of $\sigma_m \approx 0.010$ inches. Consistent performance below a few percent error is achieved for each type of test, at these noise levels. Performance drops off in a reasonably linear fashion as the Gaussian noise level increases. This is evidenced in a gradual increase in the mean and standard deviation of the centroid and volume errors and in the reduction of the percentage of trials with the correct number of segmented surfaces

4.5 RPT Testing on Real Data

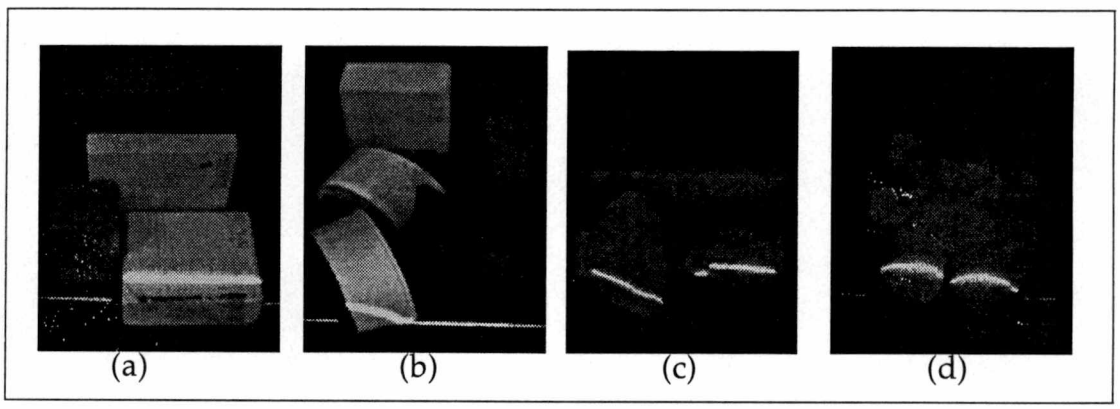


Figure 4.10: Gray scale images captured from the video camera during the ranging process. The optical filter normally used during acquisition was removed for these images, to better illustrate the interaction between the laser plane and object surfaces. The laser appears as a bright line near the lower part of each image.

Black and white images of each scene are given in Figure 4.10 and processing results are presented in Figs. 4.11 through 4.14. A variety of scenes have been

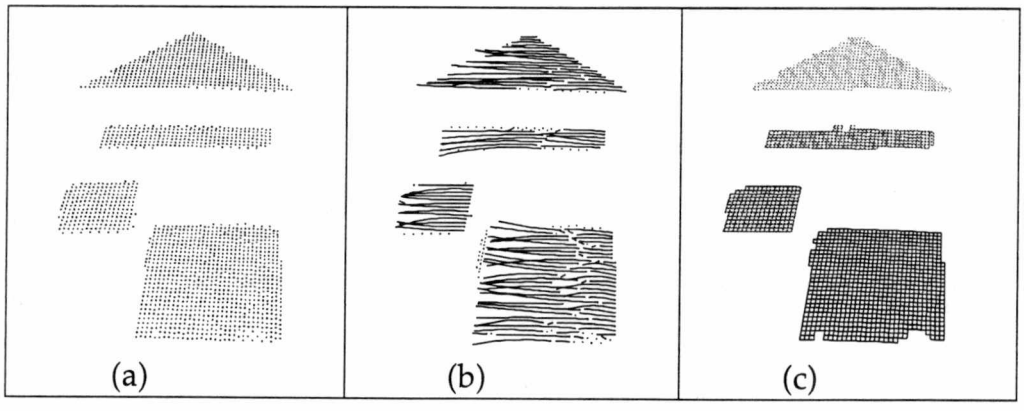


Figure 4.11: Measurements and processing results from a SL scan of an blocks scene. The Measurements, X-Strips and final Fitted Surfaces are shown.

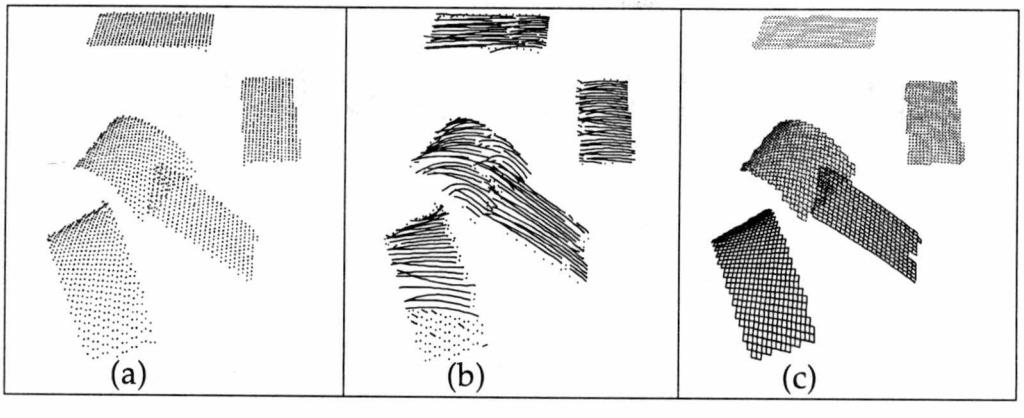


Figure 4.12: Measurements and processing results from a SL scan of a cluttered scene containing planar and second order surfaces. The Measurements, X-Strips and final Fitted Surface are shown.

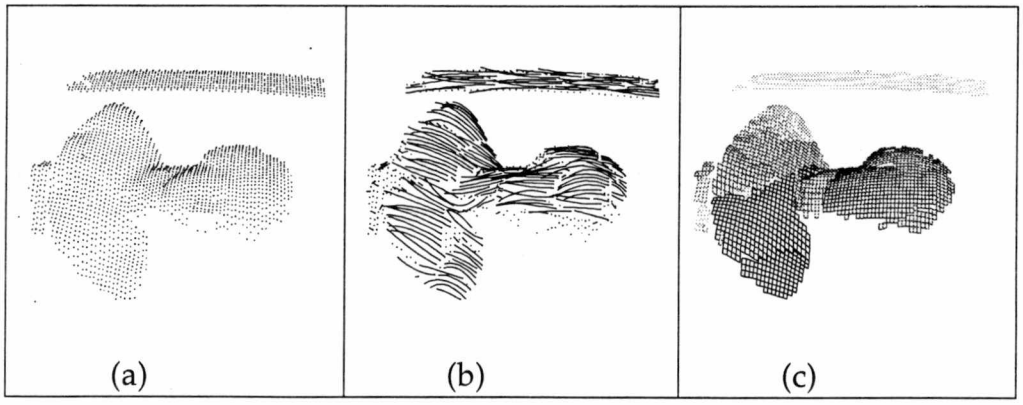


Figure 4.13: Measurements and processing results from a SL scan of marble goddess. The Measurements, X-Strips and final Fitted Surfaces are shown.

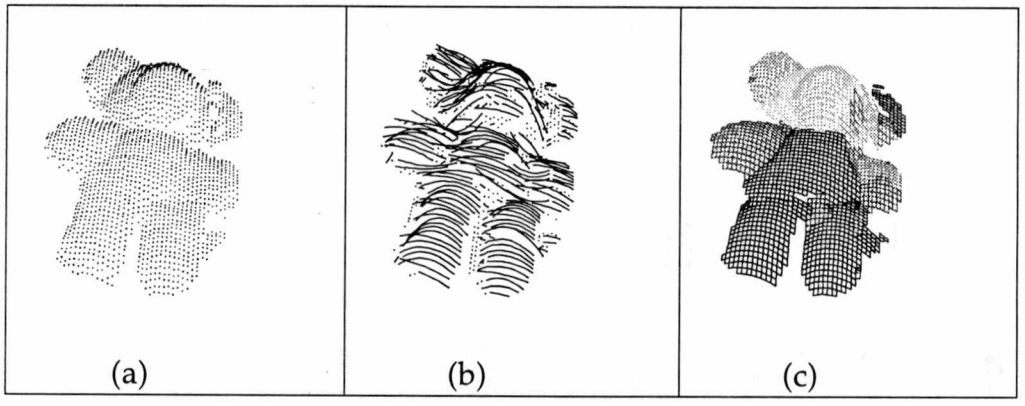


Figure 4.14: Measurements and processing results from a SL scan of a fabric bear. The Measurements, X-Strips and final Fitted Surfaces are shown.

Table 4.2: Summary of test results on real scenes. Processing parameters for all scenes were set using σ_m = 0.015 in.

Scene	On Line Processing (sec)	Off-Line Processing (sec)	Number of Final Surfaces	Residual Errors of Surfaces Min,Max (inches)
Blocks	0.007	3.1	4	0.003, 0.007
Clutter	0.007	3.8	5	0.005, 0.019
Goddess	0.007	9.0	10	0.009, 0.043
Bear	0.007	6.2	11	0.013, 0.046

chosen to illustrate segmentation of both geometric and non-geometric shapes. First, second and higher order surfaces have been included. Note the absence of noise spikes in the Structured Light data of Figs. 4.11-a through 4.14-a. This is typical for the PRIME sensor. Processing parameters for RPT were constant for all scenes and were set using σ_m = 0.015 in. as a basis. In Figs. 4.11 through 4.14, X- and Y-Strips are shown in (b) and (c), respectively. Final surfaces appear in (d). Each segmented surface is plotted in a different gray level.

Test results for the four scenes are given in Table 4.2. On-line processing speeds remained well within video frame rate. The off-line processing speeds are increased somewhat from the synthetic scenes due to a larger number of surfaces. The residual error of fitted surfaces is also presented in the Table. To find these entries, the mean residual error for each surface was computed and the minimum and maximum of these means is reported. No actual surface roughness data are known for the objects. However, the variations in texture are relatively large, comparing the smooth wooden blocks to the fabric bear, for example. These variations can be seen in the residual fitting errors, to be approximately one order of magnitude. These tests indicate that RPT works well on both synthetic and

real data.

4.6 Concluding Remarks

The objective of this research was to develop a robust, stable and accurate approach to range image segmentation, that executes as rapidly as possible. Available techniques demonstrate very good shape characterization abilities, but tend to be somewhat slow relative to RPT. Extensive testing has shown that RPT executes quickly. Tests demonstrated that segmented regions can be computed in less than 3 seconds, under a wide variety of test conditions. Stable operation was achieved above 90% of the time in the majority of testing trials.

RPT achieves this performance using an adaptive kernel that is based on Kalman filtering. This approach allows the RPT processing kernel to become arbitrarily large, provided consistency of the underlying surface curvature estimate is maintained. The large kernel size and nonrigid placement strategy help to improve the accuracy of curvature estimation. The use of Kalman filtering provides a computationally efficient means of modeling second order curvature.

RPT is a novel contribution to the field of range image processing. It meets the objective in this research for an improved means of adaptive range segmentation. The objective for improving the computational efficiency of graph matching is addressed in the next chapter.

CHAPTER 5

A Direct Method for Graph Matching: Algorithm, Theoretical Analysis, and Testing †

Of the three points of focus in this dissertation, range segmentation, graph matching, and calibrated range acquisition, this chapter covers graph matching. This effort is considered to be important in that it provides a significant advantage in computational efficiency compared to other methods. The algorithm and implementation are discussed in detail. A theoretical performance measure is also analyzed. A significant amount of testing was been done to study the performance of the algorithm under a variety conditions. These tests examined problem size, the signal to noise ratio of graph properties, various levels of topological noise, and scene graph complexity.

5.1 Introduction

Object recognition is fundamentally a problem of finding subgraph isomorphisms in that a (complete) model describes objects in their entirety; in contrast to an observed object where all features are not typically seen in a single view [22]. Because of its importance to Machine Vision [160], a novel approach to finding subgraph isomorphisms has been pursued as part of this research effort.

Reported approaches are reviewed in Chapter 2. The spectrum of approaches

† This chapter was accepted for publication in *Pattern Recognition Journal* [29]. It appears herein in near final form.

falls short on options that are able to process subgraphs efficiently. Efforts in the Discrete Mathematics community have provided some efficient matching algorithms, but none for strongly-regular graphs and none for the subgraph case. Efforts in the Machine Vision community focus on subgraphs, but rely on rich node and edge properties inorder to mitigate the effect of using a low horizon when analyzing candidate mappings (see Ch. 2).

A novel technique has been developed in this research that is computationally efficient. It also handles the subgraph case with strongly-regular graphs. Attention has been focused on testing the new technique under challenging, realistic conditions. Extensive experiments have been run on artificial data sets that were generated with a relatively low distinction in the local character of each node. The simulation tool developed to test the technique serves as a means to carefully investigate performance under varying conditions. Tests indicate an ability to match subgraphs with a 95% success rate. A graceful degradation of performance has also been documented as topological noise increases, up to the point of half of the original nodes.

5.2 The Challenge of Determining Graph Isomorphisms

A graph G: (N, N_p, E, E_p) is defined to have the typical nodes (N) and edges (E) as well as both node properties (N_p) and edge properties (E_p). Let $G^{(1)}$ and $G^{(2)}$ be two graphs with $N^{(1)}$ and $N^{(2)}$ nodes, respectively. $G^{(1)}$ and $G^{(2)}$ are said to be isomorphic if there exists a mapping which is both one-to-one and onto that associates the nodes of $G^{(1)}$ to $G^{(2)}$. The mapping also must maintain all adjacencies so that any pair of adjacent nodes in $G^{(1)}$ are also adjacent in $G^{(2)}$ under the mapping [21]. For object recognition it is also required that both node and edge properties be maintained by the mapping. The similarity of these properties must be determined by norms or other measures that are application dependent.

In object recognition applications, a match between a given graph (a "scene") and a set of known graphs (a "database") must be found. The reported match is found using the above norms and a measure of topological similarity.

Subgraph isomorphism is a condition of isomorphism that exists between two subgraphs. Object recognition is fundamentally a problem of subgraph isomorphism [22] in that a database describes its objects in their entirety; in contrast to an observed object where all features are not typically seen in a single view. Real scenes also include clutter resulting in a scene graph that contains extra nodes and edges, beyond a simple subset of database graphs.

It is the amount of effort associated with finding isomorphisms that is of concern in applications. The class of difficulty for the isomorphism problem is actually not known [21]. However, a brute force approach requires effort O(N!) for a graph of N nodes. This is reduced somewhat in cases where automorphisms of a given graph make certain sets of nodes interchangeable while still maintaining the isomorphism condition [82].

Because ranging sensors are light of sight devices, typically only one side of an object may be seen in a given view. This necessitates a partial graph match, *i.e.*, the determination of a subgraph isomorphism. The subgraph isomorphism problem is proven to be NP-complete [21]. This categorization states that a candidate solution to a problem can be *verified* with an amount of effort that is a polynomial of N. However, no deterministic bounds are given by this categorization for the amount of effort required to find the correct candidate solution. This class of problems require a worst-case amount of effort that is of exponential order $O(a^N)$. A factor that increases the number of subgraph matches is the size range of possible matches under consideration. As the size of matching subgraphs approaches 1, for example, the number of possible matches (for a single node) goes to N.

The amount of effort associated with isomorphism and subisomorphisms

would often be considered intractable for many applications. It is because of this intractibility that this effort provides a significant advancement.

A factor increasing the difficulty of the isomorphism problem is the variation in degree that is present in a graph. Strongly regular graphs have no variation in degree and are the most challenging type for finding isomorphisms [81]. The degree of a node forms a natural classification for nodes. Hence candidate node-to-node assignments could be pruned out based on this feature. Techniques have been developed using this kind of approach [94] [95] and are reviewed in Chapter 2.

No existing methods are available that are both efficient and that operate on regular graphs. Reported techniques also lack any possible extensions for finding subgraph isomorphisms.

5.3 The DCA Approach to Graph Matching

The novel approach to graph matching is to evaluate evidence describing the likelihood of a node's predicted attendance in another graph. The evidence is based on measures that are local to each node. A global verification step completes the process. In this way, the algorithm performs a **D**irect **C**lassification of node **A**ttendance (**DCA**). The presence of a node in the other graph is viewed in an isomorphic sense, i.e., there is some node-to-node mapping under which the matched nodes appear as identical members of their graphs. A node's attendance in the other graph is rated on a scale 0.0 to 1.0. The evidence that describes each node characterizes its local structural properties and any node and edge properties that a graph may possess. After the best matching pairs of nodes are identified, the nodes of one graph are reordered to allow a side by side verification of the graphs' similarity.

DCA is superior to other approaches to graph matching because of the "ex-

tended horizon" concept, introduced in Chapter 2. This refers to the nature of the comparisons that are performed between two graphs during the matching process. Existing techniques evaluate potential node-to-node mappings by comparing properties in a highly localized fashion, only then properties of the immediate nodes, their edges, and adjacent nodes (in some cases), are used. The horizon of these comparisons, is limited to neighboring nodes. In DCA, node-to-node comparisons use a structural description that describes the local graph topology up to N/2 nodes away from a node under consideration. Hence in DCA, the horizon associated with node-to-node comparisons is significantly larger than that used in existing techniques.

5.3.1 Goals for DCA Algorithm Development

The goals for finding subgraph isomorphisms were driven by the object recognition application and by an interest in active vision research [161]. First and foremost, it was desired to have an efficient algorithm to facilitate integration with active vision systems. Results which are approximate - either in terms of accuracy or the size of extracted matches - were acceptable, rather than more lengthy and potentially more complete analyses. This is consistent with trends in active vision [117,162,163]. We prefer a rapid examination of a scene followed by active exploration. Exploration can yield more information about a scene and can result in the availability of new viewpoints for observations. Achieving a rapid analysis of a scene is commensurate with this goal of active exploration.

In object recognition it is also very helpful to provide more than just a simple yes or no answer to the isomorphism question. A node-to-node mapping between the scene and database graphs is a required final result. The mapping allows the adjacency matrices and other properties of the two graphs to be compared directly.

Some object recognition algorithms depend on node and edge properties having a very "information rich" character. Requiring close matches based on these

"rich" properties greatly reduces the number of mappings that must be considered. Too great a reliance on such characterizations can create problems due to occlusions and other noise sources. Our goal is to find an algorithm that relies on the dynamic range of these properties as little as possible. This will provide an increased noise tolerance and broader applicability.

5.3.2 An Example Object Representation

It is believed that the DCA approach to finding graph subisomorphisms can have applicability beyond object recognition as in [164], but here, the algorithm will be presented in this context. To aide this presentation, a specific style of object representation will be used in the discussion. The representation assumed herein will be a 2 1/2-D relational surface patch model [12]. In general these graphs have surface shape parameters associated with their nodes. Edges are used to describe the relative orientation between adjacent surfaces. See Fig. 5.1 for an example of this type of representation.

In general there are tradeoffs between the overall size of an object's graph and the fineness of the representation. An example of a finer representation than above is a winged-edge graph [165]. These include nodes for each surface, surface-edge and corner. Edges in the graph would have to be redefined accordingly. This has the advantage of providing a more redundant description of an object and hence, improving noise tolerance. However this type of representation will also produce larger graphs which could be computationally prohibitive. These types of tradeoffs are being examined using randomly generated test cases.

5.4 Comparing Graph Topology Using Connectivity Signatures

The novelty and practical utility of the direct-classification of node attendance (DCA) approach is derived from an integration of both application-specific data

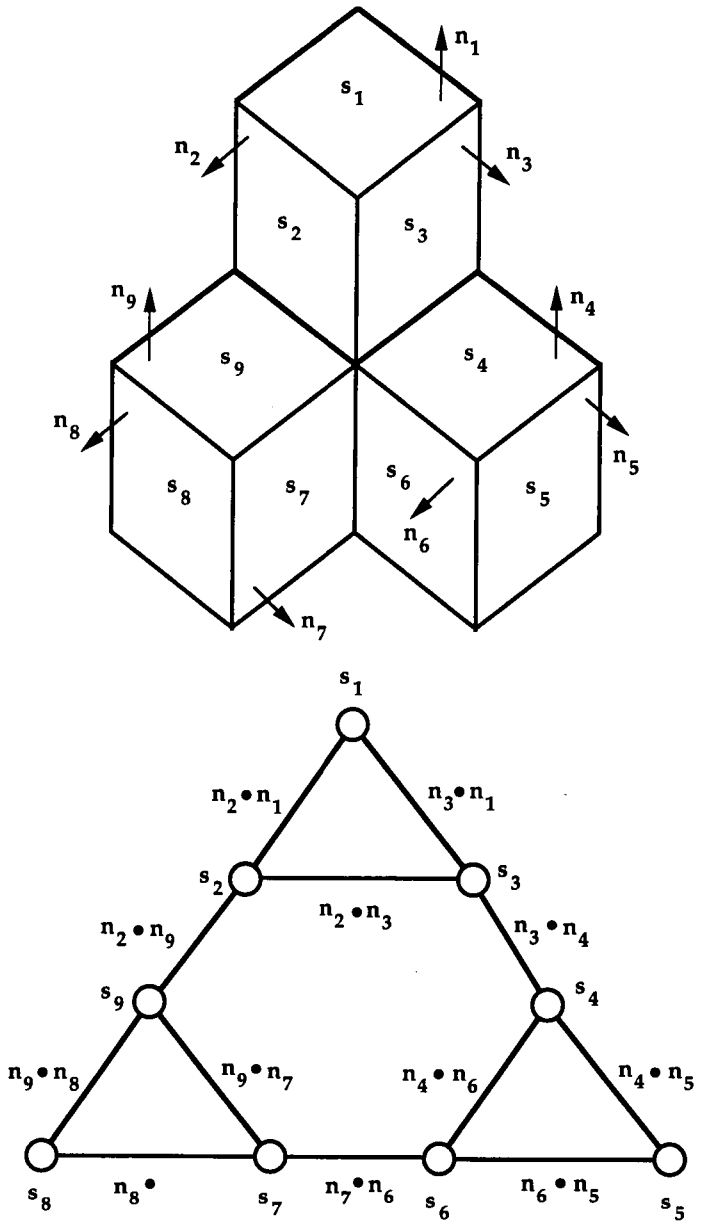


Figure 5.1: Example of the 2 1/2-D relational surface patch representation. Scene on the left contains a cube with 3 visible surfaces, s_i. Node properties on the right are used to describe these surfaces. Edge properties describe the relative orientations of the surface normals using a dot product.

and a topological description as a means for node consistency checks. If the graphs G^1 and G^2 are under scrutiny and node n_i^1 of G^1 and n_k^2 of G^2 are being compared, then the application-specific data includes an examination of the node properties of n_i^1 and n_k^2, the properties of those edges incident to n_i^1 and n_k^2, as well as the similarity of node properties for all nodes that are adjacent to n_i^1 and n_k^2. See Fig. 5.2.

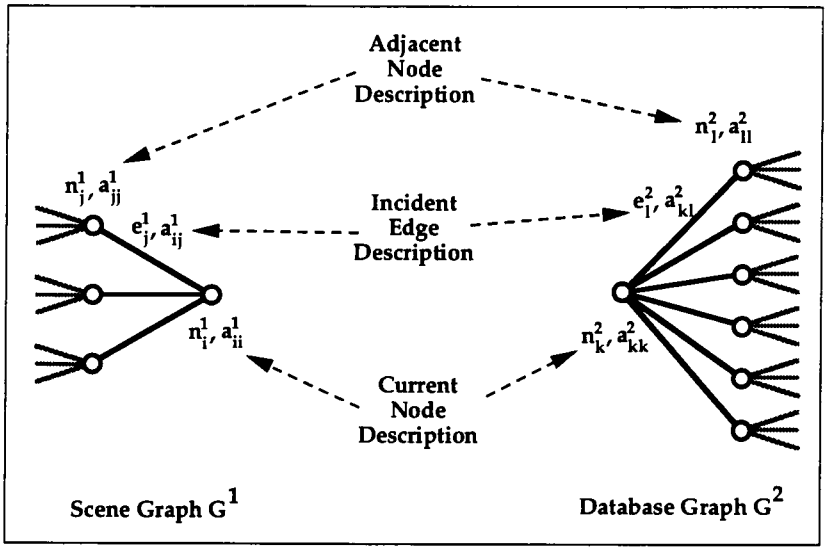


Figure 5.2: Graph data used when forming the local characterization of nodes.

A vector quantity has been defined to describe local graph topology It is referred to as the "Connectivity Signature" (CS). Note that this new approach to graph matching does not attempt to challenge the NP-Complete status of the subgraph isomorphism problem. Rather, due to the high degree of distinction that the CS vectors provide, a candidate mapping can be directly identified. However, the candidate solutions may contain errors in a few percent of experimental trials. As discussed in the definition of an NP-Complete algorithm, a verification step (requiring a polynomial amount of effort) is performed.

The extent of the CS vector is variable and DCA's performance has been studied as function of this parameter. If too restricted, not enough Connectivity information is included for any benefit. If too broad, then problems occur because

of mismatches in the Connectivity Signatures of the scene and database graphs that are due to the absence of unobserved nodes in the scene graph.

In general, the extent of the Connectivity Signature will need to be tuned for an application. It is largely determined by the expected observability of objects that is associated with a given sensor. The Connectivity Signature is computed by forming A^m for $0 < m < V_{max}$, where A is the adjacency matrix. An element ${a_{ij}}^m$ of A^m, describes the number of paths from node i to node j of length m [21]. V_{max} determines the extent of the signature. These powers of A are used to form a data structure called the "A^m cube" which consists of layers of the matrices A^m. A^1 forms the lowest layer, followed by A^2, A^3, and so on, up to $A^{V_{max}}$. After each A^m layer is formed, its values are normalized by dividing through by the smallest nonzero element in the layer. In this way an element ${a_{ij}}^m$ describes the relative degree of connectivity from node i to j, when compared against all other paths of length m.

The Connectivity Signature for n_i^1 and n_j^1, $i \neq j$, is a vertical column in the A^m cube above each element of the original adjacency matrix (${a_{ij}}^1$), denoted by

$$a_{ij} = \left| \begin{array}{c} {a_{ij}}^{V_{max}} \\ \vdots \\ {a_{ij}}^2 \\ {a_{ij}}^1 \end{array} \right| . \tag{5.1}$$

In this case n_i^1 and n_j^1 were distinct. A Connectivity Signature has also been defined for a single node. This describes the closed paths of various lengths that include n_i^1. These are denoted by a_{ii}.

The general approach of DCA is to compare nodes by combining local comparisons of node and edge properties with the a_{ij} Signatures. All node pairs n_i^1 and n_j^2 are compared in this fashion to form a total of N^1N^2 attendance ratings. Peak attendance values are identified. Currently this is done by a simple approach

that passes over the attendance ratings to find a peak, records this node-to-node mapping, and repeats until a mapping is found for all node pairs. Matching subgraphs of the scene and database entry are formed, each of size $K \leq min(N^1, N^2)$. The nodes of the scene subgraph are reordered using the attendance peaks so that n_1^1 and n_1^2 are associated with a peak, as are n_2^1 and n_2^2 and so on.

At this stage all the properties of n_i^1 and n_i^2 can be directly compared and a verification step is performed. Thresholds are applied to the difference in node and edge properties and to differences in the adjacency matrix. All subgraph nodes are examined in this manner and the worst matching node is removed. This process is repeated (at most K times) until either a null graph exists or until all properties match suitably well. Any remaining nodes with zero degree are then removed. Note that if this is compared to a hypothesis-and-test approach to object recognition, then here only a single hypothesis and a single test are performed for each database entry.

Good performance of the algorithm was achieved with $V_{max} \approx N\rho$, where $N = min(N^1, N^2)$, $\rho = 0.25$. Values of V_{max} in this range are necessary due to the limited observability of an object that is possible from a given viewpoint. V_{max} is related to $N\rho$ because path lengths used in the Connectivity Signature need to be limited in order to provide the best possible match between the database and expected scene conditions. Note that when forming Connectivity Signatures for the database, V_{max} could be set to N. However, if this same value of V_{max} were used for the scene, then the resulting signatures would be dissimilar because the subgraph in the scene won't contain all the paths in the database entry. For this reason the value of V_{max} must be limited when the database is analyzed so as to match the expected scene conditions as best as possible. When DCA is applied to an isomorphism problem, the V_{max} parameter can vary over a broader range with little effect on performance.

5.5 Analysis of Connectivity Signatures

CS vectors provide a description of graph topology from the point of view of a single node. See Figure 5.3. DCA uses the local description provided by CS vectors to find a node-to-node mapping between two graphs. This section addresses the viability of using Connectivity Signatures for determining graph isomorphisms.

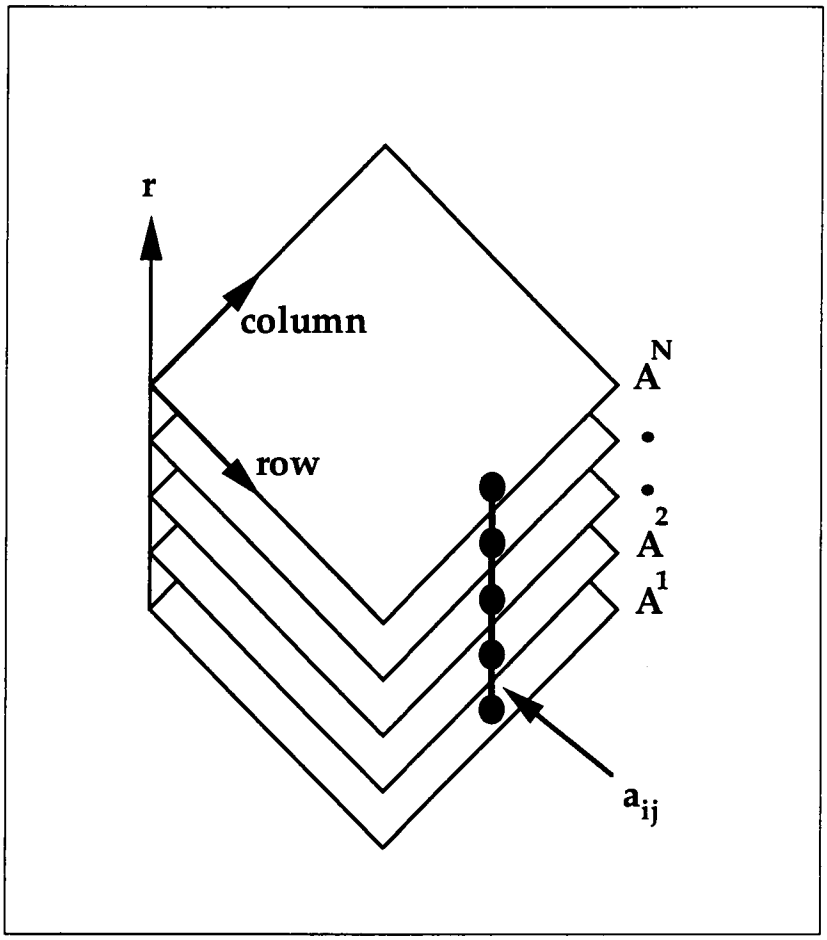


Figure 5.3: Connectivity Signature (CS) vectors are formed using powers of the adjacency matrix, A^i.

It will be shown experimentally below, in Section 5.11 that the CS vectors have sufficient descriptive capability to be useful in graph isomorphism algorithms.

Here, the degree of uniqueness of CS vectors is examined in a more analytical approach. Using random graphs it will be shown that the expected number of matching CS vectors that occurs across the space of *all random graphs* remains less than one, as the number of nodes becomes infinitly large. From this result it is concluded that the CS vectors form a useful basis for isomorphism algorithms in large graphs. Notation and various definitions follow, along with a list of assumptions. In these analyses, node and edge properties are not considered. Also, the analysis will be restricted to the case of comparing graphs of the same size.

The DCA algorithm looks for node mappings $\boldsymbol{n}_i^{(1)} \leftrightarrow \boldsymbol{n}_k^{(2)}$ that exhibit exact matches. The matches are computed using CS vectors, $\boldsymbol{a}_{ij}^{(1)} - \boldsymbol{a}_{kl}^{(2)} = \mathbf{o}$, for example. Valid mappings must have matching CS vectors for the nodes under consideration, their incident edges, and their adjacent nodes. (See Figure 5.2). Hence the uniqueness of the CS vectors is critical to the success of DCA finding a candidate mapping.

5.5.1 Notation and Definitions

Random graphs are typically generated using one of three methods, refered to as Model A,B and C [82]. Model A will be used herein. It uses a fixed probability $\boldsymbol{p}$ to determine the occurance of edges in a graph. If $\boldsymbol{p} = \mathbf{0.5}$ then the random graph formation process can be envisioned as the flipping of a coin to determine the existence of each possible edge. Models B and C are used to form graphs with a set number of edges. Restrictions of this type are not included here.

- Graphs are labelled, *i.e.*, nodes have a fixed identity. The convention used herein assigns label $\boldsymbol{i}$ to node $\boldsymbol{n}_i$. Labelled graphs are the more intuitively appropriate type to be associated with isomorphism problems.

- A Graph $G^{(1)}$ has $N^{(1)}$ nodes. Superscripts in the style (i) are used to indicate a particular graph, *i.e.*, $G^{(1)}$ is a distinct graph from $G^{(2)}$.

- A Graph G has an N x N adjacency matrix, A. The adjacency matrix provides a complete description of graph structure. The connectivity of node n_i is described by the i_{th} row of A.

- $A^1 = A$, has elements a_{ij}^1. $a_{ij}^1 = 1$ implies that an edge exists between nodes i and j, $a_{ij}^1 = 0$ implies that no edge exists.

- $A^2 = AA$ has elements a_{ij}^2. The meaning of $a_{ij}^r, i \neq j$ is the number of paths from node i to node j of length r [21]. The element a_{ii}^r gives the number of closed loop paths of length r that include node i.

- The Connectivity Signature, a_{ij}, is an N x 1 vector defined as $\left| \begin{array}{ccc} a_{ij}^1 & \ldots & a_{ij}^N \end{array} \right|^T$.

- To simplify notation when dealing with two graphs, $G^{(1)}$ and $G^{(2)}$ are considered to have Connectivity Signatures a_{ij} and b_{ij}, respectively. In this convention, $b_{ij} = \left| \begin{array}{ccc} b_{ij}^1 & \ldots & b_{ij}^N \end{array} \right|^T$. d_{ij} is also used below to designate the difference $a_{ij} - b_{ij}$.

- A "match" between two Connectivity Signatures, a_{ij} and b_{ij}, is denoted $a_{ij} = b_{ij}$, and occurs if $a_{ij}^r = b_{ij}^r, \forall r$.

- p is the probability of an edge occuring in Model A and $q = 1 - p$.

- The degree of a node is equal to the number of edges incident to it.

- A regular graph contains nodes all with the same degree.

- An isomorphic mapping between node i and j is denoted $n_i \leftrightarrow n_j$.

5.5.2 Assumptions

The following assumptions are used in the derivation below:

- No directed graphs. This implies $A = A^T$.
- No self loops, this implies, $a_{ii}^1 = 0$, $\forall i$.
- No multigraphs, this implies, $a_{ij}^1 = 0$ or 1, $\forall i \neq j$.
- Graphs are large.
- Concerning Model A, p is constant and positive $\forall a_{ij}, i \neq j$.
- The occurance of edges in Model A are assumed to be independent and uncorrelated events.
- Model A is used to approximate a regular graph having N nodes each with degree r, $p = r\ N$.
- A variation in degree is not assumed to exist in a graph. In fact, graphs under consideration in the derivation below may be strongly regular.
- No application-specific colorings for nodes or edges are assumed to be present.
- An infinite bit width is assumed for the (integer) calculations associated with CS vectors. The practical implications of this are examined in section 5.6.

5.5.3 Verifying Isomorphisms

When using the definition of A given in Section 5.5.1, there is a one-to-one correspondance between an adjacency matrix and a labelled graph [21]. This permits an isomorphism to be verified by comparing adjacency matricies element by element.

Of course, for this type of verification to be useful and valid, the two graphs must not only be isomorphic, but must also have the same node ordering. Considered from this perspective, determining an isomorphism can be viewed as a process of reordering the nodes of one graph to match another. This brings the brute force $N!$ level of effort for finding isomorphisms to light, in that there are N possible rows in which the first node can reside in A, $N - 1$ for the second node, and so forth.

This method is used for verification in DCA. This verification is "global" in the sense that all edges in each graph are compared. It is necessary that a global verification be performed in DCA because the candidate mapping is identified using a *best-case, local,* means of comparison. The CS vector differences are local in nature because they involve the *number* of paths that begin and end on certain nodes. Because only the number of paths is considered, these actual routes provided by the other nodes in the graph is not explicity described within a CS vector. Furthermore, DCA looks for a best-case correspondance between the edges of two nodes under consideration. This correspondance isn't necessarily valid. It is a favorable indication that a given node-to-node mapping might be correct. Hence, a global verification is required to eliminate the favorable mappings generated for candidate solutions.

In experimental trials, the favorable nature of the candidate solutions has not proven to be tolerable provided certain minimum bounds are maintained on graph size. In section 5.11 it is shown that excellent results can be obtained (in 100/100 cases) when comparing graphs of the same size and without any application colorings, provided the graphs have at least 10 nodes.

5.5.4 Invariance of Connectivity Signatures

If CS vectors are to be used in graph isomorphism algorithms then it is necessary that the vectors be invariant with respect to the ordering, or labeling, of nodes.

This means that the numerical value of a CS vector can not change when nodes are reordered. Any useful feature for determining isomorphisms should be invariant in this manner [81].

Intuitively, it is expected that this would be true. Consider that a_{ij}^r gives the number of paths of length r from node i to j. The ordering of nodes in the graph should have no effect on this, since the number paths is a purely topological property.

To examine the invariance of the CS vectors, the fundamental operation of exchanging two labels is considered first. A "label swap" is defined as the exchange of two labels. In the convention herein, the i_{th} node of a graph has label i and appears in the i_{th} row (and column) of the Adjacency matrix. A label swap alters the order of appearence of two rows and two columns of A. A series of label swaps will modify A to $\overline{A}$. In this rearrangement, the (i,j) element of A is moved to $(\bar{i},\bar{j})$. The property of invariance dictates that two isomorphic graphs (such as A and $\overline{A}$) have the same a_{ij} entries, but in a different arrangement. This must be true for each component of a_{ij}.

The label swap operation can be expressed in terms of pre- and post-multiplication by an interchange permutation matrix S [148]. As a simple example, consider post multiplication with an instance of S that exchanges the first two columns of A.

$$\begin{bmatrix} \sharp & \flat & . & . & . \\ \sharp & \flat & . & . & . \\ \sharp & \flat & . & . & . \\ \sharp & \flat & . & . & . \\ \sharp & \flat & . & . & . \end{bmatrix} \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} = \begin{bmatrix} \flat & \sharp & . & . & . \\ \flat & \sharp & . & . & . \\ \flat & \sharp & . & . & . \\ \flat & \sharp & . & . & . \\ \flat & \sharp & . & . & . \end{bmatrix} \tag{5.2}$$

Similarly, the first two rows of A can be exchanged via premultiplying with the

same matrix

$$\begin{bmatrix} 0 & 1 & & & & \\ 1 & 0 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{bmatrix} \begin{bmatrix} \sharp & \sharp & \sharp & \sharp & \sharp \\ \flat & \flat & \flat & \flat & \flat \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \end{bmatrix} = \begin{bmatrix} \flat & \flat & \flat & \flat & \flat \\ \sharp & \sharp & \sharp & \sharp & \sharp \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \end{bmatrix}. \tag{5.3}$$

A matrix S that is used to swap the labels k and l can be formed using the rule

$$s_{ij} = \begin{cases} 1, & i = j \text{ and } i \neq k \text{ and } i \neq l \\ 1, & i = k \text{ and } j = l \\ 1, & i = l \text{ and } j = k \\ 0, & \text{else}. \end{cases} \tag{5.4}$$

This is equivalent to exchanging rows k and l in the identity matrix [148]. Note that if S' premultiplies S'', and $S' = S''$ then S' exchanges rows in S'' yielding the identity. Hence $SS = I$. Using two of the S matricies, a single label swap can be performed by

$$SAS = \overline{A} \tag{5.5}$$

The question of invariance may now be posed as whether or not $\overline{A^r}$ is strictly a rearrangement of the rows and columns of A^r, *i.e.*, that no changes to the values in the a^r_{ij} entries have occurred, only their (i, j) row and column arrangement. This is easily seen to be true for any power of A. For example,

$$\overline{A^2} = \overline{AA} = SAS\,SAS = SAAS = SA^2S = \overline{A^2} \tag{5.6}$$

$$\overline{A^3} = \overline{AAA} = SAS\,SAS\,SAS = SAAAS = SA^3S = \overline{A^3}. \tag{5.7}$$

Hence the a^r_{ij} entries are unaffected by label swaps.

The implication of an automorphic mapping can be appreciated when considering the label swapping process. An automorphism is an isomorphic mapping of a graph onto itself. This means that some non-trivial sequence of label swaps results in an adjacency matrix that is identical to some initial $\boldsymbol{A}$, *i.e.*,

$$\overline{\boldsymbol{A}} = \boldsymbol{S}' \cdots \boldsymbol{S}'' \boldsymbol{A} \boldsymbol{S}'' \cdots \boldsymbol{S}' = \boldsymbol{A}. \tag{5.8}$$

Palmer describes the implications of automorphisms on the combinatorics of the matching process in terms of group theory [82]. Here, it is only cited that their existence introduces multiple solutions to the mapping problem. These multiple solutions are manifested in permutations of $\boldsymbol{A}$ that are identical.

5.5.5 Strategy to Find Degree of Uniqueness of Connectivity Signatures

The viability of DCA hinges on the degree of uniqueness in the Connectivity Signatures (CS). This uniqueness is examined in a probabalistic sense using random graphs [93]. This is accomplished by deriving the expected number of matching CS vectors, given all possible random graphs with N nodes. This expected number of matches is denoted by $\boldsymbol{\Delta}_N$ and is shown to be less than 1. Hence, the conclusion is made that CS vectors form an adequate characterization of local node topology, and, as such, are useful in finding graph isomorphisms.

5.5.6 Statistical Description of a Connectivity Signature

Model A is assumed for the $\boldsymbol{a}_{ij}^1$ entries. A class of binary random variables $\boldsymbol{x}_1$ is defined as a means to generate the $\boldsymbol{a}_{ij}^1$ entries. An instance of one of these random variables is denoted by $\boldsymbol{x}_1'$.

$$\boldsymbol{x}_1' = \begin{cases} 1, & \text{edge present at } \boldsymbol{a}_{ij}^1 \\ 0, & \text{no edge present} \end{cases} \tag{5.9}$$

The probability density function of x_1 is

$$f(x_1) = q\delta(0) + p\delta(1). \tag{5.10}$$

Similar classes of random variables x_r are defined for all a_{ij}^r entries. Expressions for x_r are developed next, followed by their probability density functions, means and variances.

The entries in A^2 are the sums of products of A^1 entries

$$a_{ij}^2 = \sum_{k=1}^{N} a_{ik}^1 a_{kj}^1 \tag{5.11}$$

Hence, a given instance of x_2 is a sum of products of the x_1 entries,

$$x_2 = \sum_{k=1}^{S_2} (x_1' x_1'')_k = \sum^{S_2} \prod^{P_2} x_1^{(')}. \tag{5.12}$$

Here, x_1' and x_1'' are instances of the x_1 class of random variables, $P_2 = 2$ and $S_2 \approx N$. $x_1^{(')}$ denotes a particular instance of an x_1 variable. Actually only $N-1$ pairs of the x_1 variables are present in the summation above because $a_{ii}^1 = 0, \forall i$. This approximation is used to simplify the expressions below, and is reasonable because of the assumption that N is large. The entries in A^3 can also be written as the sums of products of A^1 entries

$$a_{ij}^3 = \sum_{k=1}^{N} a_{ik}^2 a_{kj}^2 = \sum_{k=1}^{N} \left(\sum_{l=1}^{N} a_{il}^1 a_{lk}^1 \right) \left(\sum_{m=1}^{N} a_{km}^1 a_{mj}^1 \right) \tag{5.13}$$

Thus, the random variable x_3 can be written

$$x_3 = \sum^{N} \left(\sum x_1' x_1'' \right) \left(\sum x_1''' x_1'''' \right) = \sum^{S_3} \prod^{P_3} x_1^{(')}. \tag{5.14}$$

A similar approximation for the number of terms in the summation is used here, giving $S_3 \approx N^3$ and $P_3 = 4$. Considering the a_{ij}^4 entries

$$x_4 = \sum^{N} \left(\sum^{N^3} \prod^{4} x_1' \right) \left(\sum^{N^3} \prod^{4} x_1'' \right) = \sum^{S_4} \prod^{P_4} x_1^{(')} \tag{5.15}$$

where $S_4 \approx N^7$ and $P_4 = 8$. Examining the trend for the number of terms in the products, P_r, and in the sums, S_r

$$P_r = 2^{r-1} \tag{5.16}$$

$$S_r = N^{2^{r-1}-1} = N^{P_r-1}. \tag{5.17}$$

The mean, μ_1, and variance, σ_1^2, for a_{ij}^1 can be computed directly

$$\mu_1 = E\{x_1\} = \int_{-\infty}^{\infty} x_1 f(x_1)\, dx_1 = 0q + 1p = p \tag{5.18}$$

$$\sigma_1^2 = E\{(x_1 - \mu_1)^2\} = E\{x_1^2\} - E\{x_1\}^2 \tag{5.19}$$

and

$$E\{x_1^2\} = \int_{-\infty}^{\infty} x_1^2 f(x_1)\, dx_1 = 0q + 1^2 p = p \tag{5.20}$$

so

$$\sigma_1^2 = p - p^2 = pq. \tag{5.21}$$

Note that $\mu_1 > 0$ because of the $p > 0$ assumption.

Assuming that N is large, the *central limit theorem* [146] can be used to find the probability density function of x_2. This theorem states that the probability density function of a sum of a large number of random variables approaches a Gaussian density. Furthermore, if the N contributing random variables are identically distributed, having a mean and variance of (μ, σ^2), then the density function of the sum will have $(N\mu, N\sigma^2)$. Hence, to find μ_2 of x_2, the contributors, $y_1^{(\prime)}$, to the x_2 summation

$$x_2 = \sum^{S_2} \prod^{P_2} x_1^{(\prime)} = \sum^{S_2} y_2^{(\prime)} \tag{5.22}$$

must be examined. These have a mean of

$$E\{y_2\} = E\{x_1' x_1''\} = E\{x_1'\} E\{x_1''\} \tag{5.23}$$

because of the independence of the a_{ij}^1 entries [146]. Because all instances of x_1 have identical densities,

$$E\{y_2\} = E\left\{\prod^{P_2} x_1^{(\prime)}\right\} = \prod^{P_2} E\left\{x_1^{(\prime)}\right\} = p^{P_2} = p^2 \tag{5.24}$$

and by the central limit theorem

$$\mu_2 = Np^{P_2}. \tag{5.25}$$

Similarly, for higher powers of A, the contributors $y_r^{(\prime)}$ have an expected value

$$E\{y_r\} = E\left\{\prod^{P_r} x_1^{(\prime)}\right\} = \prod^{P_r} E\left\{x_1^{(\prime)}\right\} = p^{P_r} \tag{5.26}$$

again, because of the assumption of independence in the a_{ij}^1 entries and because p is constant $\forall i, j$. This yields a mean of

$$\mu_r = S_r E\{y_r\} = S_r p^{P_r} = N^{P_r - 1} p^{P_r} \tag{5.27}$$

for elements of the CS vector.

The variance of x_2 can also be found using the central limit theorem. The contributors, y_2, to the x_2 summation have a variance

$$\sigma_{y_2}^2 = E\{(y_2)^2\} - (E\{y_2\})^2 = E\{(x_1' x_1'')^2\} - (p^2)^2 \tag{5.28}$$

The probability density function of $y_2 = x_1' x_1''$ is needed to evaluate this expectation integral. Because x_1' and x_1'' are independent and uncorrelated [166]

$$f(y_2) = f(x_1', x_1'') = f(x_1') f(x_1''). \tag{5.29}$$

This gives

$$f(x_1', x_1'') = q^2\,\delta(0,0) + pq\,\delta(0,1) + pq\,\delta(1,0) + p^2\,\delta(1,1). \tag{5.30}$$

The second moment of y_2 can now be found

$$E\{(y_2)^2\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x_1')^2 (x_1'')^2 f(x_1', x_1'')\, dx_1'\, dx_1'' = 1^2\, 1^2\, p^2 = p^2 \tag{5.31}$$

because only the $\delta(1,1)$ term survives in this integral. The probability density functions for y_r are formed in a similar manner, having P_r independent (and identical) functions, $f(x_1')$, multiplied together

$$f(y_r) = f(x_1') \cdots f(x_1''). \tag{5.32}$$

This is in the form

$$f(y_r) = q^{P_r}\,\delta(0,\ldots,0) + pq^{P_r-1}\,\delta(0,\ldots,0,1) + \cdots + p^{P_r}\,\delta(1,\ldots,1). \quad (5.33)$$

Note that only the last term has a delta function in the form $\delta(1,\ldots,1)$. This permits the second moment of y_r to be found

$$E\{(y_r)^2\} = E\left\{\left(\prod^{P_r} x_1^{(\prime)}\right)^2\right\} \quad (5.34)$$

and because x_1 variables only take on binary values

$$E\left\{\left(\prod^{P_r} x_1^{(\prime)}\right)^2\right\} = E\left\{\prod^{P_r}\left(x_1^{(\prime)}\right)^2\right\} \quad (5.35)$$

which gives

$$E\{(y_r)^2\} = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(x_1')^2\cdots(x_1'')^2\,f(x_1')\cdots f(x_1'')\,dx_1'\cdots dx_1'' = p^{P_r} \quad (5.36)$$

again, because only the $\delta(1,\ldots,1)$ term survives the integration. This gives a variance for y_2 of

$$\sigma_{y_2}^2 = E\{(y_2)^2\} - E\{y_2\}^2 = p^2 - (p^2)^2 = p^2(1-p^2) \quad (5.37)$$

and similarly, a variance for y_r of

$$\sigma_{y_r}^2 = E\{(y_r)^2\} - E\{y_r\}^2 = p^{P_r} - (p^{P_r})^2 = p^{P_r}(1-p^{P_r}). \quad (5.38)$$

By the central limit theorem the variance σ_r^2 of x_r is

$$\sigma_r^2 = \sum^{S_r}\sigma_{y_r}^2 = S_r p^{P_r}(1-p^{P_r}) = N^{P_r-1}p^{P_r}(1-p^{P_r}) \quad (5.39)$$

Thus the components of a Connectivity Signature in a random graph have a Gaussian probability density function with a mean and variance of

$$\begin{cases} \mu_r = (Np)^{P_r}/N \\ \sigma_r^2 = N^{P_r-1}p^{P_r}(1-p^{P_r}) \end{cases} \quad (5.40)$$

where $P_r = 2^{r-1}$.

5.5.7 Probability of Matching Connectivity Signatures

When two CS vectors match exactly the difference

$$\boldsymbol{d}_{ij} = \boldsymbol{a}_{ij} - \boldsymbol{b}_{ij} \tag{5.41}$$

must have $\boldsymbol{d}_{ij}^{r} = \mathbf{0}, \forall \boldsymbol{r}$. The probability of this occurance must now be computed. Note that the difference of two Gaussian random variables also has a Gaussian density function. Also, the expected value of the difference of two Gaussian random variables is the difference of the expected values [166]. Hence the expected value of the $\boldsymbol{d}_{ij}$ vector is zero. The variance of the difference of two Gaussian random variables is the sum of the variances [146]. So, the variance of $\boldsymbol{d}_{ij}$ is twice that of $\boldsymbol{a}_{ij}$.

The probability $\boldsymbol{\lambda}_r$ that the $\boldsymbol{r}_{th}$ component of $\boldsymbol{d}_{ij}$ is zero is found by integrating the Gaussian density from $-\mathbf{1}$ to $\mathbf{1}$.

$$\boldsymbol{\lambda}_r = \frac{\mathbf{1}}{\mathbf{2}\sqrt{\mathbf{2}\boldsymbol{\pi}}\boldsymbol{\sigma}_r} \int_{-1}^{1} \boldsymbol{e}^{\frac{-x^2}{2(2\sigma_r^2)}}\, \boldsymbol{dx}_r. \tag{5.42}$$

Because of the relatively high variance, this integral can be closely approximated by a single ("narrow") rectangle. This yields an upper bound of

$$\boldsymbol{\lambda}_r < \frac{\mathbf{1}}{\sqrt{\mathbf{2}\boldsymbol{\pi}}\boldsymbol{\sigma}_r}. \tag{5.43}$$

There is actually a weak correlation between the components of $\boldsymbol{d}_{ij}^{r}$. This is due to the fact that each component is a sum of products of the same set of random variables ($\boldsymbol{x}_1$). To avoid introducing these correlations, the probability, $\boldsymbol{\Lambda}$, that all $\boldsymbol{d}_{ij}^{r}$ components are zero is approximated by a less restrictive case. Here, the probability that just the $\boldsymbol{N}_{th}$ component is zero, $\boldsymbol{d}_{ij}^{N} = \mathbf{0}$ is used. This gives an upper bound for $\boldsymbol{\Lambda}$ of

$$\boldsymbol{\Lambda} < \frac{\mathbf{1}}{\sqrt{\mathbf{2}\boldsymbol{\pi}}\boldsymbol{\sigma}_N} = \frac{\boldsymbol{N}}{\sqrt{\mathbf{2}\boldsymbol{\pi}}(\boldsymbol{N}\boldsymbol{p})^{P_N}(\mathbf{1} - \boldsymbol{p}^{P_N})} \tag{5.44}$$

For large N, $p^{P_N} \to 0$. Simplifying the denominator and maintaining the upper bound, Λ becomes

$$\Lambda < \frac{N}{\sqrt{2\pi}(Np)^{P_N}} < \frac{N}{(Np)^{P_N}} = \frac{N}{(Np)^{2^{N-1}}}. \tag{5.45}$$

This gives an upper bound on the probability that two random CS vectors have identical components.

5.5.8 Number of Connectivity Signatures in All Possible Random Graphs

The number of random graphs (based on Model A) having N nodes is denoted G_N. It can be found by considering the number of possible adjacency matricies. The adjacency matrix is symmetric

$$\begin{bmatrix} 0 & * & * & \cdots & * \\ & 0 & * & \cdots & * \\ & & \ddots & & \vdots \\ & & & 0 & * \\ & & & & 0 \end{bmatrix} \tag{5.46}$$

with a zero diagonal, because there are no self loops. The number of entries in each row above the diagonal increases sequentially from 1 (in the $N-1$ row) up to $N-1$ in the first row. Hence there are a total number of distinct entries, E_N, is

$$E_N = \sum_{i=1}^{N-1} i \tag{5.47}$$

A closed form version of this summation is available [167], giving

$$E_N = \frac{N(N-1)}{2} \tag{5.48}$$

which can be written

$$E_N = \frac{N!}{2!(N-2)!} = \binom{N}{2}. \tag{5.49}$$

Because multigraphs are not considered, a_{ij}^1 can only take on the values 0 and 1. Hence, each of the E_N entries can take on 2 possible values, giving

$$G_N = 2^{E_N} = 2^{\binom{N}{2}} \tag{5.50}$$

For each of the G_N graphs there are $\binom{N}{2}$ CS vectors. This gives the total number of Connectivity Signature vectors, C_N

$$C_N = \binom{N}{2} G_N = \binom{N}{2} 2^{\binom{N}{2}}. \tag{5.51}$$

5.5.9 Expected Number of Matching Connectivity Signatures

Using eqs. (5.45) and (5.51), the expected number of CS vectors that match exactly is bounded by

$$\Delta_N < \Lambda C_N = \frac{N \binom{N}{2} 2^{\binom{N}{2}}}{(Np)^{2^{N-1}}}. \tag{5.52}$$

It is of interest to see if the value of Δ_N remains less than 1 as N increases. If $\Delta_N < 1$ then

$$N \binom{N}{2} 2^{\binom{N}{2}} < (Np)^{2^{N-1}} \tag{5.53}$$

or

$$\frac{1}{2} N^2 (N-1) 2^{N(N-1)/2} < (Np)^{2^{N-1}}. \tag{5.54}$$

Each side of the inequality is positive. This permits ln() of both sides to be taken.

$$\ln \frac{1}{2} + 2 \ln N + \ln(N-1) + N(N-1) \ln \sqrt{2} < 2^{N-1} \ln(Np). \tag{5.55}$$

Taking the limit as $N \to \infty$ permits constant terms to be dropped. This gives

$$2 \ln N + \ln(N-1) + N^2 \ln \sqrt{2} - N \ln \sqrt{2} < 2^{N-1} \ln(Np). \tag{5.56}$$

On the left, the $\ln(N)$ terms are dominated by the N^2 term, simplifying to

$$N^2 \ln \sqrt{2} < 2^{N-1} \ln(Np). \tag{5.57}$$

This is expressed in the form

$$gh < g'h' \tag{5.58}$$

where $g = N^2$, $h = \ln\sqrt{2}$ and $g' = 2^{N-1}$, $h' = \ln(Np)$. Note that g, h, g' and h' are all positive and hence the inequality of Eq. 5.57 is true because $g < g'$ and $h < h'$.

This shows that the expected number of matching CS vectors is less than 1 for large N. This implies that CS vectors have a relatively high degree of uniqueness, in a probabalistic sense. Hence, it is concluded that CS vectors are a practical means for identifying candidate solutions to the isomorphism problem.

5.6 Factors Effecting A Practical Algorithm

The above derivation assumed that the CS vectors could be computed and compared using exact arithmetic. The actual calculations can be accomplished using integer arithmetic, which would nominally support exact calculations, but for bit width limitations. To examine this limitation, the worst case type of graph is considered - this is a fully connected graph, with $a^1_{ij} = 1, \forall i \neq j$. So, for $i \neq j$,

$$a^2_{ij} = \sum_{k=1}^{N} a^1_{ik} a^1_{kj} = N - 1 \approx N \tag{5.59}$$

and

$$a^3_{ij} = \sum_{k=1}^{N} a^2_{ik} a^2_{kj} = N(N-1)^2 \approx N^3 \tag{5.60}$$

$$a^4_{ij} = \sum_{k=1}^{N} a^3_{ik} a^3_{kj} = N[N(N-1)^2]^2 \approx N^7 \tag{5.61}$$

hence the recursive relationship

$$\begin{cases} a^{r+1}_{ij} = N(a^r_{ij})^2 & , r \geq 3 \\ a^2_{ij} = N - 1 \end{cases} \tag{5.62}$$

This result is quite similar to the mean value found for CS vector elements given in Eq. 5.40, but here $p = 1$. These entries grow quite rapidly, when compared to the integer precision of standard machines. For example,if $N = 5$, the worst-case a_{ij}^N entry is $\approx 2^{34}$ which is beyond the bit width of typical machines. Even having $p = 0.5$ doesn't help, the standard bit width is still exhausted before $N = 6$.

Because integer arithmetic cannot be used for the CS vectors, the powers of A must be computed using floating point operations. The limited precision of these calculations can be modeled as a noise source [168] which will add to the variance of the CS vectors. This necessitates the use of a norm to measure the magnitude of the difference d_{ij}.

There are serveral considerations that must be taken into account when defining this norm. The components d_{ij}^r having larger values of r tend to have much larger numerical values, as seen by the trend in Eq. 5.62. This will tend to cause the entries associated with larger values of r to "overwhelm" the norm. Here the a_{ij}^r entries are normalized for each value of r.

Another factor arises when node and edge properties are introduced into the isomorphism problem. In this case, norms describing the similarity of these properties must be combined with the CS vector norms. These norms were setup to produce values between 0 and 1. This allowed a variety of mechanisms to be studied for combining the similarity measures.

5.7 DCA Algorithm

Steps in the DCA algorithm are presented below in pseudocode.

Subgraph Formation Stage
Compare each pair of nodes in each graph
1)For each node of n_i^1 of G^1:
  2)For each node of n_k^2 of G^2:
    Compare edges incident to current nodes
    3)For each edge e_j^1 incident to n_i^1:
      4)For each edge e_l^2 incident to n_k^2:
        4a)Compare connectivity of a_{ij}^1 with a_{kl}^2
        4b)Compare adjacent nodes' connectivity of a_{jj}^1 with a_{ll}^2
        4c)Compare edge properties of e_j^1 with e_l^2
        4d)Compare adjacent nodes' properties of n_j^1 with n_l^2
        4e)Combine results of (4a) to (4d) via IOP
    3a)Save comparison of best matching edges and adjacent nodes
  Compare current nodes
  2a)Compare connectivity of a_{ii}^1 with a_{kk}^2
  2b)Compare node properties of n_i^1 with n_k^2
  Find similarity of current nodes
  2c)Combine (3a) (2a) and (2b) to form attendance rating of n_i^1 to n_k^2

Form matching subgraphs
5) Find peaks of attendance ratings to define mapping from all n_i^1 to n_k^2
6) Form subgraphs g^1 and g^2 using node pairs n_i^1 and n_k^2 with attendance ratings above threshold T
7) Use node-to-node mapping to reorder nodes in g^1 such that n_1^1 maps to n_1^2, n_2^1 maps to n_2^2, etc

Subgraph Verification Stage
8) Eliminate nodes with poorly matching Np
9) Eliminate nodes with poorly matching Ep
10) Eliminate nodes with poorly matching A^1

In step (6) the subgraphs $\boldsymbol{g}^1$ and $\boldsymbol{g}^2$ of $\boldsymbol{G}^1$ and $\boldsymbol{G}^2$ respectively, are formed by applying the threshold T to the attendance ratings in order to find "significant" node pairs. A value of 0.5 was used for T in all the tests reported herein.

In steps (4c), (4d) and (2b) node and edge properties are compared. These methods of comparison are application-dependent. In these tests, node and edge properties were all scalars. A pair of scalar properties (a,b) were compared using

the relationship

$$1.0/(1.0 + |a - b|) \tag{5.63}$$

where || denotes absolute value. Connectivity Signatures are compared in (4a), (4b) and (2a). Two signatures (c,d) are compared in a similar means using

$$\frac{1.0}{V_{max}} \sum_{v=1}^{V_{max}} 1.0/(1.0 + |c^v - d^v|). \tag{5.64}$$

The selection of the methods used to combine evidence in (4e) and (2c) is critical to achieving good performance with DCA. See Fig. 5.4. No optimal means for combining evidence is claimed to have been found. Experiments have been performed using various methods, such as the Independent Opinion Pole (IOP), Linear Opinion Pole, Harmonic Opinion Pole [169] and Theory of Evidence (TE) [170]. Combinations of these techniques have also been attempted. Bayesian classifiers [156] were used to combine the results of multiple techniques, however this did not appreciably impact final results.

When analyzing subgraphs, the method that appeared to work best used TE

$$s_{ds} = 1.0 - (1.0 - s_1)(1.0 - s_2) \tag{5.65}$$

where s_1 and s_2 are attendance ratings and s_{ds} is a combined value. When analyzing full-sized graphs in an isomorphism problem, IOP

$$s_{iop} = s_1 s_2 \tag{5.66}$$

worked well. This mode switch was required at step (2c) when combining the evidence associated with neighboring nodes with the current nodes. The two modes of operation are indicated in Fig. 5.4 by the split circle. The unlabeled arrows in the figure represent data flows of attendance ratings. The circles indicate an operation taking place that combines attendance values.

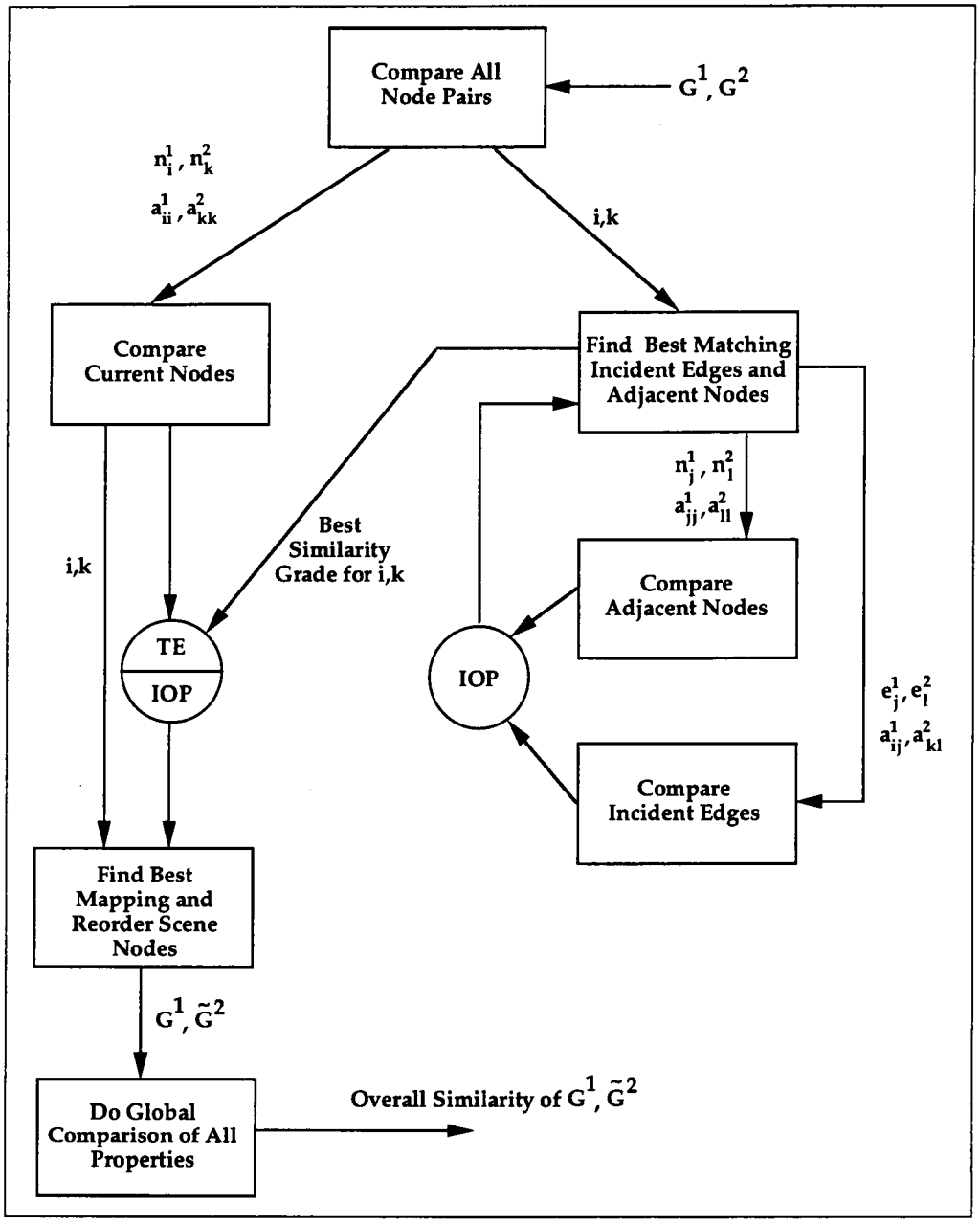


Figure 5.4: Steps involved in node classification. Unlabeled arrows describe data flows of attendance ratings. Circles indicate operations where attendance ratings are combined.

5.8 Performance Metrics for DCA

Several performance measures have been established for DCA that allow test conditions to be evaluated and parameters to be adjusted. Fig. 5.5 shows a histogram of attendance ratings for nodes that were members of present and absent subgraphs. Dual data sets are plotted with pairs of light and dark bars on the same scale. This "dual histogram" has left (darker) entries that describe the attendance of nodes belonging to database graphs that were absent from the scene. The right (lighter) entries describe the attendance of nodes that belonged to database graphs that were present in a scene. Nodes of these two categories are referred to as "present" and "absent" nodes.

As seen in the histogram, absent nodes tend to rank lower and present nodes rank higher on the scale. The segmentation of nodes is not perfect in the case of subisomorphisms. Note the subset of absent nodes that rank high on the scale. It is believed that this is somewhat unavoidable because these entries represent single node subisomorphisms, which seem to occur relatively often. The verification step that completes DCA processing eliminates these nodes from the final subgraph by comparing the reordered adjacency matricies. This enforces a global connectivity check which is above and beyond the requirement of matching local properties.

Examining the dual histograms is not always a certain indication of DCA's ability to recognize objects. It can give a rough indication. In full-size isomorphism problems, the segmentation is very good - see Fig. 5.6 - and recognition success rates have been benchmarked at 100 out of 100 cases. Also note that this test case had a dynamic range of *zero* for the node and edge properties, and was run on strongly-regular graphs. The matching in these tests was achieved solely via the Connectivity Signatures.

Another benchmark of DCA's performance is the size of matched subgraphs.

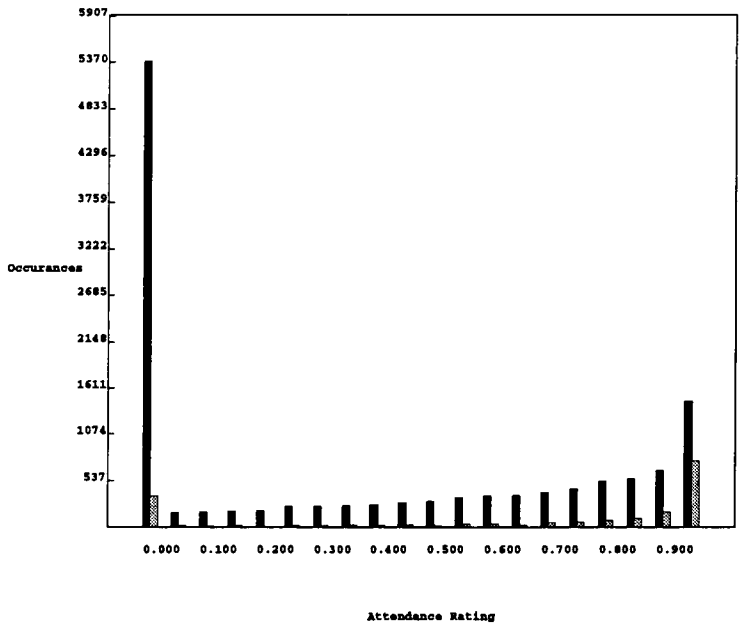


Figure 5.5: Dual histogram of attendance ratings recorded in a subisomorphism problem. Entries for present nodes are given by the lighter (right) bars. Entries associated with absent nodes are indicated with the darker (left) bars. Present and absent nodes are segmented using the attendance rating. Dichotimization is sufficient given such a distribution, provided the global verification step is performed to complete DCA processing.

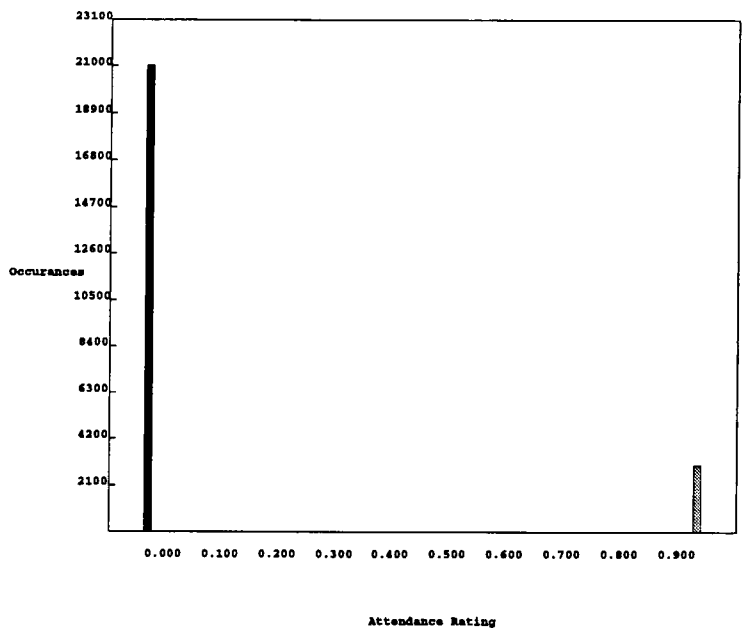


Figure 5.6: Dual histogram of the attendance ratings in an isomorphism problem. Note the dichotimization is significantly improved over the subisomorphism case. These data were recorded from a test with node and edge properties having a **zero** dynamic range and with strongly-regular graphs.

Fig. 5.7 shows a histogram of these sizes on tests involving database graphs with 30 nodes and a single scene graph having 15 nodes. These 100 trials had a mean subgraph size of 10.7 and standard deviation of 2.3 nodes. A metric is needed to determine the best match between the scene contents and database entries. Subgraph size has been weighted by a norm measuring the closeness of node and edge properties for this final quality measure.

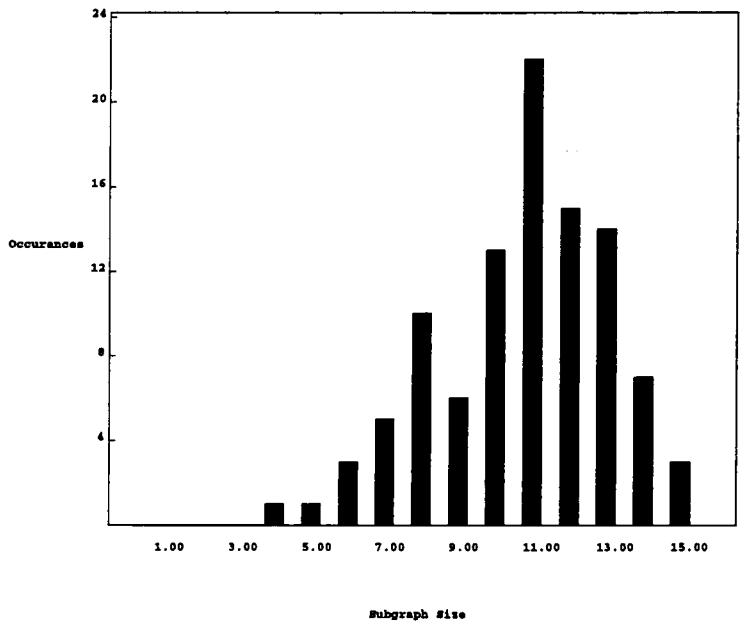


Figure 5.7: Histogram of subgraph size. This test involved database graphs with 30 nodes and a single scene graph having 15 nodes. This subisomorphism problem had a mean $\pm$ standard deviation of $\mathbf{10.7 \pm 2.3}$ nodes.

During testing each node of each graph in the database was assigned a unique name tag for use as a final check of correctness. The percentage of correct name tags was also tallied as a performance measure. Unfortunately random subgraph automorphisms do exist in this type of testing, and the name tags flag automorphisms as an error, despite perfectly matching topologies and all other properties. Fig. 5.8 shows a histogram of the fraction of correct labels in the case described

above. The mean $\pm$ standard deviation of this metric in the test case cited above was $98 \pm 5\%$.

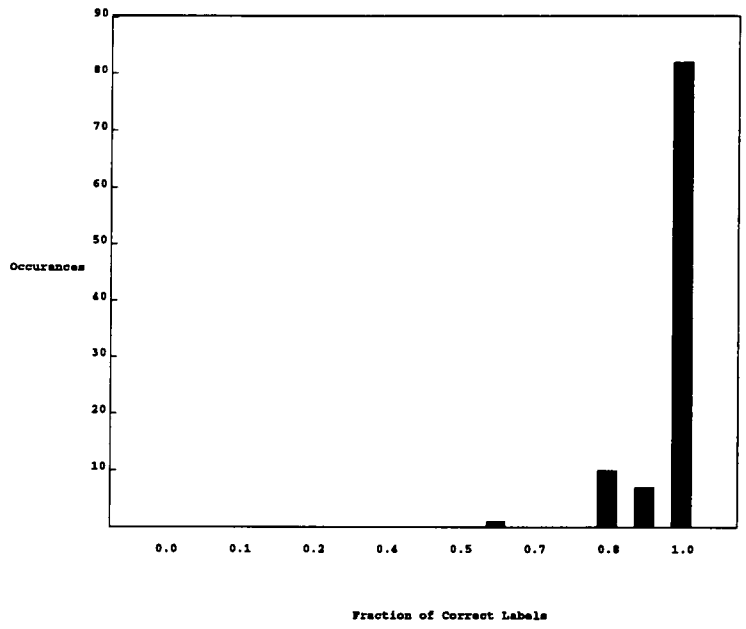


Figure 5.8: Fraction of correct labels. This test involved database graphs with 30 nodes and a single scene graph having 15 nodes. The mean $\pm$ standard deviation of correct labels was $98 \pm 5\%$.

5.9 Computational and Memory Requirements for DCA

The effort required for each database examination is O(N^3) for graphs with N nodes. Loops (1) and (2) in the algorithm have an upper limit of N_s and N_d. Loops (3) and (4) have a limit determined by the shape complexity of objects, which is not a function of N_d because of the strongly-regular graphs. Hence, the limit on loops (3) and (4) is determined by the mean degree of the nodes. Comparing the Connectivity Signatures requires effort O(V_{max}). Overall this yields an O(N_s N_d V_{max}) which is O(N^3).

The A^m Cube must also be generated for each scene. This is an O(N^4) operation, but is only done once, prior to the database examinations. DCA requires O(N^3) memory for the A^m Cubes.

5.10 DCA Testing Methodology

Test parameters have been designed to reflect conditions in an object recognition scenario as closely as possible, while still permitting the use of randomly generated cases that could be run in large numbers. This section discusses how the test conditions were derived and presents the testing procedure in a step-by-step manner.

5.10.1 Relation of Test Conditions to an Object Recognition Scenario

One test parameter is the fraction of a database graph that is observable in a scene. Consider a rectangular solid for example, most viewpoints reveal 1/2 of its surfaces. Certain views can restrict the fraction down to 1/6. Hence most tests were performed with randomly generated subgraphs that were half the size of their original database versions.

The amount of interconnectivity in a graph was an important test parameter.

As discussed above, strongly-regular graphs are the most appropriate for object representation. All tests were performed with this type of graph. Note that a graph which has precisely the same number of edges incident to each node can not exist in all arbitrary cases [21]. Hence an algorithm was written to generate graphs that are strongly-regular (or almost regular) by simply adding edges randomly between any nodes that are below a given degree. This generated graphs rapidly and maintained the near regular condition. The graph in Fig. 5.9 was generated using this method.

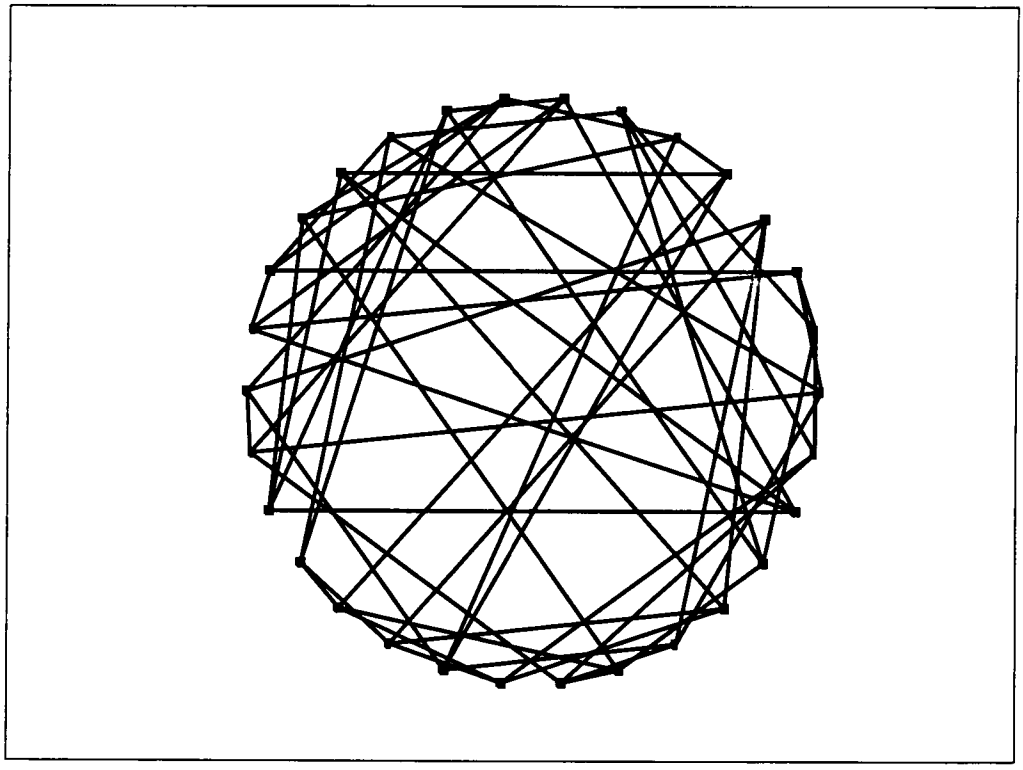


Figure 5.9: An example of a randomly generated database entry. This graph is strongly-regular, each node has degree ≈ 4.

Another important parameter that affects the applicability of the test results is the dynamic range of node and edge properties. In some applications node properties can be made quite rich involving various shape parameters, texture or color, and so forth [59]. This sort of node property would likely be a vector

quantity. Only scalar integer quantities were considered here. This was in an attempt to keep the tests challenging. In most tests a dynamic range of 4 was used for both node and edge properties. This can be thought of as a set of objects having surfaces with 4 distinct shapes, for example. This includes all possible colorations, textures, shapes and sizes. The dynamic range of edge properties can be envisioned as 4 discrete relative orientations between adjacent surfaces. In a 2-D application involving the matching of features between two images [113], edge properties could be used to describe the North, South, East or West relative locations. These values were selected with the assumption that most machine vision applications would have properties with at least this much dynamic range.

The noise level added to each node and edge property during the simulated sensing process was also varied. Flat noise with zero mean was added to each node and edge property. A noise peak of 0.25 was typically used. This corresponded to a +/-12.5% peak variation away from the ideal values stored in the database.

5.10.2 Testing Procedure

The individual steps in the testing process are described in Fig. 5.10. Each trial consists of 100 cases, beginning with the generation of a database of objects. A number of database entries were randomly selected to be scene contributors. A subgraph was formed for each of these by randomly dropping nodes. The node ordering was then randomly perturbed and a scene constructed using all contributors. Topological noise was added by placing random interconnecting edges between the scene contributors and by adding spurious nodes. DCA then compared the scene against each database entry. The best matching database entry was retained as the final selection.

Fig. 5.11 shows a scene constructed from a single object's subgraph. Four additional noise nodes were added to the scene to represent the effect of background

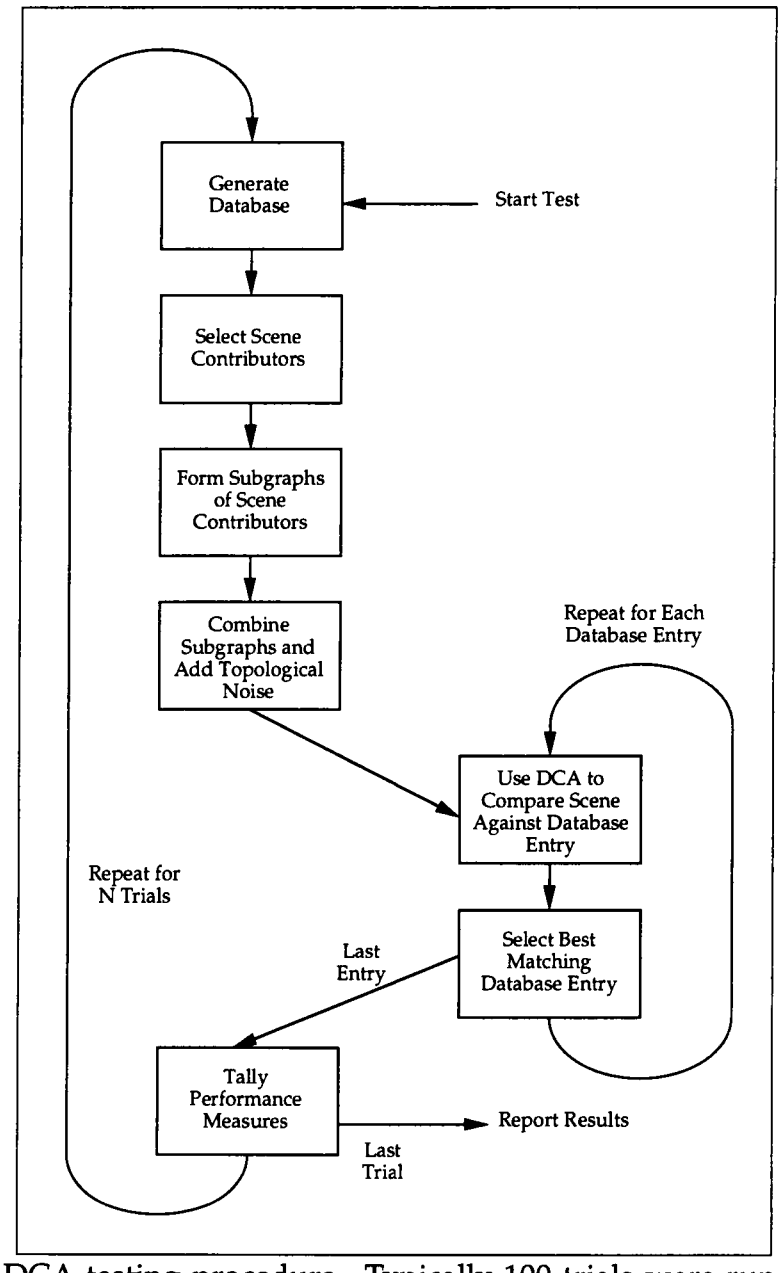


Figure 5.10: DCA testing procedure. Typically 100 trials were run for each set of test conditions. Each trial involved the random generation of databases and scenes. Both topological and property noise were then added. Three performance metrics were used for benchmarking.

clutter. The topological noise can be seen in the 4 edges that connect the central nodes to the 4 peripheral (noise) nodes. Fig. 5.12 has two contributing subgraphs, 8 outlying nodes, and 4 interconnecting edges between the database contributors. In both of these scenes, subgraphs had 1/2 the number of nodes of their original database versions.

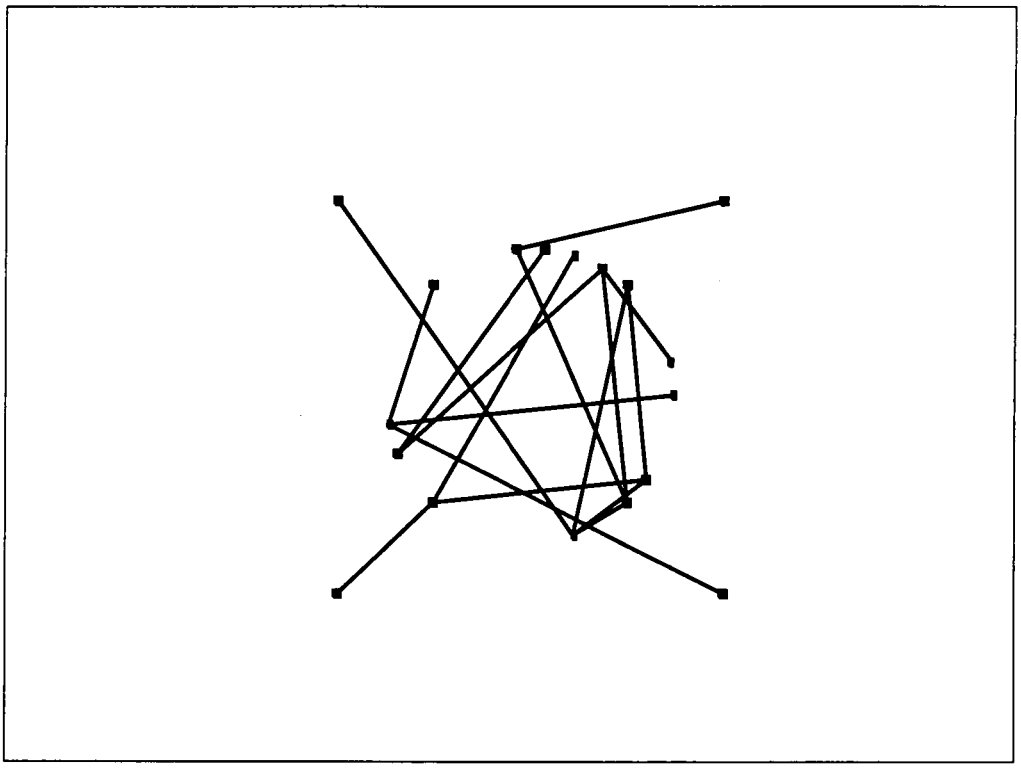


Figure 5.11: A scene graph generated from randomly selected database contributors. This scene is composed of a single subgraph with some added topological noise that can be seen in the 4 edges that connect the central nodes to the 4 peripheral nodes.

5.11 Experimental Verification: Results and Implications

Extensive simulations of the recognition process have been completed and results follow. These tests have been performed in an abstract graphical domain.

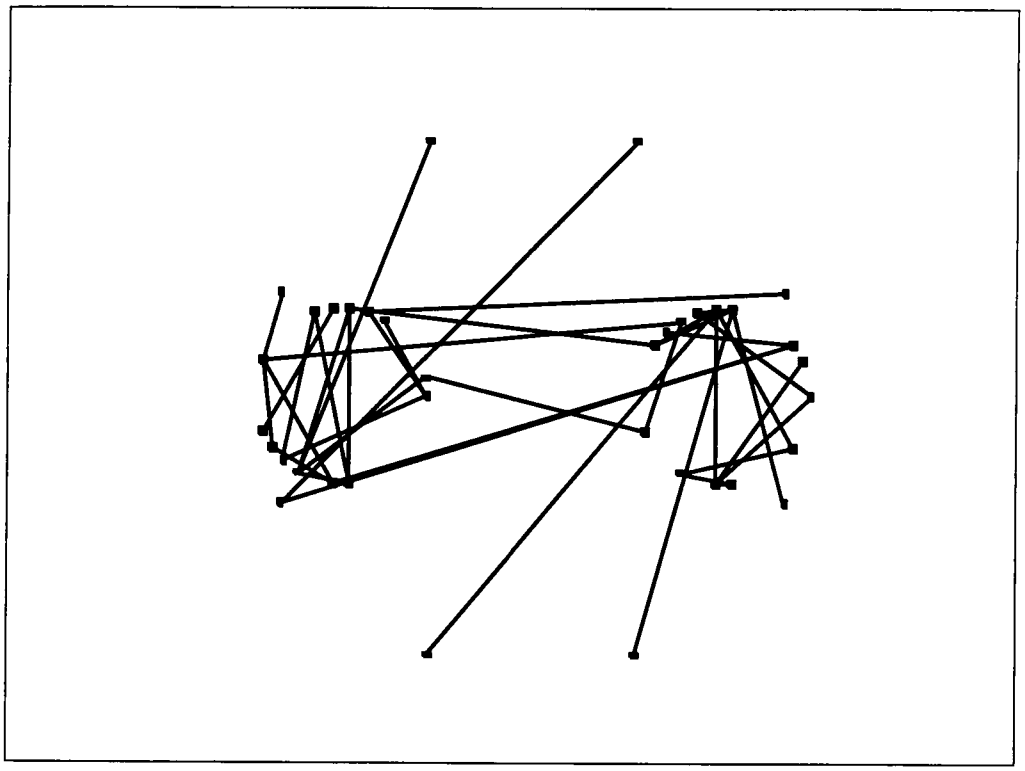


Figure 5.12: A scene graph formed from 2 database contributors. Excess nodes and edges are present.

Hence, the physical location and orientation of object features were not part of the testing process. For this reason tests on pose determination have not been completed. All tests were run on a Silicon Graphics Indy2 workstation that had a 100 MHz R4000 RISC processor with an R4010 floating point unit. Tests were run at normal priorities and in a multi-user environment.

Tables are used to present the test conditions and performance measures below. An explanation of the table headings follows.

- "D/S Number of Graphs" in the first column designates to the number of graphs in the database and scene, respectively.
- "D/S Size of Graphs" gives the size of graphs in the database followed by the size of each subgraph contributor to the scene.
- "Excess N/E in Scene" are the number of extra nodes added to each scene and number of extra interconnecting edges added between each subgraph contributor.
- The "Rate Hz" is the average rate at which each database entry was examined in Hertz. Note that some preliminary processing was necessary for the scene graph. This was only performed once for each scene. For graphs of these sizes this required less than one second and is not included in these figures.
- "% Correct Selection" is the percentage of trials in which correct database contributors were selected. All tests were performed with 100 trials.
- "Subgraph Size" gives the mean $\pm$ standard deviation of the size of the extracted subgraphs. These subgraphs were reduced in size until all properties and topologies of the scene and database matched within specified tolerances.

Table 5.1: Summary of test results when matching full-size graphs. The graphs in these tests had no node or edge properties, no property noise and no topological noise. All graphs were strongly-regular with degree ≈ 4. Performance was maintained at nearly 100%, provided the graph size was above ≈ 10.

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	10/10	0/0	0/0, .0/.0	34.3	97	10.0 ±0.0	75.6 ± 26.1
8/1	16/16	0/0	0/0, .0/.0	9.5	100	16.0 ± 0.0	99.5 ± 3.0
8/1	20/20	0/0	0/0, .0/.0	2.6	100	20.0 ± 0.0	99.8 ± 1.4
8/1	24/24	0/0	0/0, .0/.0	2.7	100	24.0 ± 0.0	99.7 ± 1.6
8/1	30/30	0/0	0/0, .0/.0	1.3	100	30.0 ± 0.0	100.0 ± 0.0
8/1	40/40	0/0	0/0, .0/.0	0.4	100	40.0 ± 0.0	99.9 ± 0.7

- "% Correct Tag" is the percentage of correct name tags present in the matched subgraphs. The name tags uniquely identify each node of each graph in the database. This performance measure is also given in mean ± standard deviation form.

5.11.1 Tests Using DCA to Find Full-Sized Isomorphisms

DCA's yielded the best performance on full-sized isomorphism problems. To emphasize this a series of tests were run having a zero dynamic range for node and edge properties. Here, only the Connectivity Signatures were available as a means to determine node attendance. Table 5.1 show a correct database selection rate of 100/100 and a full sized matching subgraph in all tests. The percentage of correct name tags was not totally perfect. This is due in part to automorphisms which are more prevalent under these test conditions because the lack of node and edge properties makes nodes less distinctive. In all tests $V_{max} = N_s/\mathbf{2.0}$ and IOP was used to combine evidence. The number of nodes ranged from 10 to 40 in these tests.

Table 5.2: Summary of test results examining effect of subgraph size. These results indicate that at least half of the graph should be retained given these size and noise conditions.

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/10	4/4	4/4, .25/.25	11.9	64	4.3 ± 1.9	91.4 ± 18.3
8/1	30/15	4/4	4/4, .25/.25	6.8	99	11.2 ± 2.3	96.9 ± 5.6
8/1	30/20	4/4	4/4, .25/.25	4.6	100	17.7 ± 1.6	98.9 ± 3.1
8/1	30/25	4/4	4/4, .25/.25	2.9	100	22.7 ± 1.4	99.9 ± 1.1

5.11.2 Tests Using DCA to Find Subisomorphisms

DCA's performance on subisomorphisms has been tested under a variety of conditions. These are grouped into two categories for clarity. The first examines size-related parameters and the second looks at effects associated with node and edge properties and with noise.

Problem Size Effects

Graph size and the number of graphs are important factors affecting performance. In general graphs need to be sufficiently large before the Connectivity Signature attains a sufficient character to help distinguish nodes. This can be seen in Tables 5.2 and 5.3, as well as in the tests on full-sized isomorphisms above. The fraction of nodes present in a scene subgraph was varied in Table 5.2. Changes in this fraction correspond to varying occlusion levels or to varying sensor capabilities.

All tests were run with 8 database graphs and 1 subgraph in each scene. The size of graphs in the database was fixed at 30 nodes. In all cases V_{max} was set using the ceiling function $V_{max} = ceil(N_s/4.0)$. TE was used to combine evidence describing the node attendance. These results indicate that at least half

Table 5.3: Summary of test results examining effect of graph size. Perfomance degrades as the graph size decreases due to the reduced information present in the Connectivity Signature.

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	20/10	4/4	4/4, .25/.25	15.0	95	7.2 ± 1.7	95.3 ± 12.0
8/1	24/12	4/4	4/4, .25/.25	11.7	98	9.0 ± 2.0	94.5 ± 10.0
8/1	36/18	4/4	4/4, .25/.25	4.2	99	13.8 ± 2.6	97.7 ± 5.0
8/1	40/20	4/4	4/4, .25/.25	3.3	99	14.7 ± 2.8	96.0 ± 6.5

Table 5.4: Summary of test results examining effect of the number of database elements. Most performance measures remain fairly constant except for the percentage of correct database selections.

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
4/1	30/15	4/4	4/4, .25/.25	7.0	99	10.9 ± 2.3	96.4 ± 6.4
8/1	30/15	4/4	4/4, .25/.25	6.2	99	11.2 ± 2.3	96.9 ± 5.6
16/1	30/15	4/4	4/4, .25/.25	6.8	96	11.4 ± 2.0	95.7 ± 7.2
32/1	30/15	4/4	4/4, .25/.25	7.2	93	11.3 ± 2.1	96.1 ± 7.3
64/1	30/15	4/4	4/4, .25/.25	7.3	90	11.8 ± 2.0	97.5 ± 6.4

of the graph should be retained under these conditions.

The effect of graph size is examined in Table 5.3. This effect is important because of its implications to object representation and on overall computational effort. Given these test conditions, it appears that a winged-edge graph may be appropriate for applications that involve relatively simple objects. In this representation a cube has 26 nodes. V_{max} and TE were used as above.

When a large number of graphs are present in the database the recognition process becomes more challenging because of the additional potential matching subgraphs that are available. Results in Table 5.4 indicate that most performance

Table 5.5: Summary of test results examining effect of V_{max}. Values for the V_{max} parameter are given in the second column. A value of 1 is trivial in the sense that no additional information is included in the Connectivity Signatures beyond that of the original Adjacency matrix. The need for the Connectivity Signature can be seen by the jump in correct database selections that accompanys nontrivial V_{max} extents. In all other testing $V_{max} = ceil(N_s/4.0)$.

D/S Number of Graphs	D/S Size and V_{max}	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/15,1	4/4	4/4, .25/.25	8.9	8	2.0 ± 0.0	56.2 ± 46.4
8/1	30/15,2	4/4	4/4, .25/.25	8.1	96	11.6 ± 2.6	97.6 ± 6.1
8/1	30/15,3	4/4	4/4, .25/.25	7.3	96	11.8 ± 2.4	95.8 ± 8.3
8/1	30/15,4	4/4	4/4, .25/.25	7.0	99	11.2 ± 2.3	96.9 ± 5.6
8/1	30/15,5	4/4	4/4, .25/.25	6.4	100	10.7 ± 2.3	98.0 ± 4.9
8/1	30/15,6	4/4	4/4, .25/.25	6.0	100	9.0 ± 2.6	98.6 ± 6.7
8/1	30/15,7	4/4	4/4, .25/.25	5.6	99	7.0 ± 2.6	99.5 ± 3.0
8/1	30/15,8	4/4	4/4, .25/.25	5.1	89	4.7 ± 2.1	99.8 ± 1.5

measures remain fairly constant except for percentage of correct database selections which dropped slightly to 90% when 64 database entries are present. These tests begin to give a feeling for the discriminatory ability of DCA. An application requiring many dozens of database objects may need to have an increased dynamic range for its node and edge properties, than the ranges used here.

The optimal choice for the V_{max} parameter is related to size, so these tests have been included here. V_{max} parameter determines the extent of the Connectivity Signature. As seen in Table 5.5, performance peaks in the range $4 \leq V_{max} \leq 6$.

The best choice of V_{max} and other parameters will be largely application-dependent. The relative importance of subgraph size vs. correct database selection, vs. correct node mappings will vary between applications. In all other tests reported herein $V_{max} = ceil(N_s/4.0)$.

Table 5.6: Summary of results examining effect of the number of subgraphs present in scene. The percentage of correct name tags suffered a significant drop off here. These incorrectly matched nodes would have to be identified and removed during the pose determination process.

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/15	4/4	4/4, .25/.25	6.8	99	11.2 ± 2.3	96.9 ± 5.6
8/2	30/15	4/4	4/4, .25/.25	3.1	98	10.1 ± 1.9	87.1 ± 12.5
8/3	30/15	4/4	4/4, .25/.25	1.8	94	10.0 ± 2.0	78.2 ± 16.6
8/4	30/15	4/4	4/4, .25/.25	1.2	98	9.9 ± 1.7	74.4 ± 16.8

Another size-related issue is the number of subgraph contributors to each scene. In Table 5.6 performance metrics remain reasonably consistent, except for the percentage of correct name tags. This had a rather serious drop off, down to $\approx 75\%$ for scenes with 4 subgraphs. It should be possible to mitigate some of the effects of this degredation during the pose determination stage. At this point the residual error of each node's physical location can be computed and outliers can then be removed.

Evaluation of Dynamic Range and Noise Effects on DCA Performance

The dynamic range of node and edge properties has a very direct impact on performance because of its ability to improve the local characterization of nodes. Various combinations of node and edge ranges are presented in Table 5.7. Albeit, this is a tiny sampling of the span of all possible cases, it appears that the range of edge properties needs to be ≥ 4 until the range of node properties becomes ≥ 16. Note the poor results when the edge range drops to 0. Here also, the subgraph size increases above the true size of the contributor. This is consistent with the poor percentage of label matches in these cases.

Table 5.7: Summary of test results examining dynamic range of node and edge properties. Results indicate the importance of the dynamic range of edges.

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/2	30/15	4/4	4/2, .25/.25	3.0	78	10.2 ± 1.9	75.6 ± 18.6
8/2	30/15	4/4	4/4, .25/.25	2.5	98	10.1 ± 1.9	87.1 ± 12.5
8/2	30/15	4/4	8/0, .25/.0	3.3	91	18.7 ± 2.4	46.5 ± 14.6
8/2	30/15	4/4	8/2, .25/.25	3.3	97	11.1 ± 1.8	87.2 ± 13.1
8/2	30/15	4/4	8/4, .25/.25	3.3	99	11.5 ± 1.9	95.2 ± 9.0
8/2	30/15	4/4	16/0, .25/.0	3.3	94	15.5 ± 2.2	70.8 ± 12.6
8/2	30/15	4/4	16/2, .25/.25	3.2	100	12.2 ± 1.6	94.6 ± 7.9
8/2	30/15	4/4	16/4, .25/.25	3.2	100	12.2 ± 1.4	98.1 ± 4.5
8/2	30/15	4/4	16/8, .25/.25	3.1	100	12.2 ± 1.3	98.7 ± 4.0

The results of tests adding noise to node and edge properties were interesting. A gradual decline in performance can be seen in Table 5.8 as the noise took on larger values, as expected. The low noise cases were suprising. Note the severe drop off in performance with zero noise. In applications with very low property noise, a weighting scheme may have to be introduced to reduce the effect of property comparisons when forming the attendance ratings.

The effects of topological noise shown in Table 5.9 were not as severe as the cases presented in Table 5.6 where additional scene graphs were added. Increasing the number of scene graphs has a much greater influence on the scene's topology than the noise introduced here.

5.12 Concluding Remarks

The objective of this research effort was to develop a subgraph isomorphism algorithm that will work well in object recognition applications. Current techniques for shape-based recognition are typically quite time consuming and can have undesirable tradeoffs in speed vs. accuracy. DCA can meet the computa-

Table 5.8: Summary of test results examining noise in node and edge properties. Results are fairly consistent across the span of noise intensities, except in the extreme cases.

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/15	4/4	4/4, .0/.0	7.0	16	2.8 ± 1.0	57.9 ± 41.9
8/1	30/15	4/4	4/4, .05/.05	6.8	95	7.3 ± 2.6	97.9 ± 6.4
8/1	30/15	4/4	4/4, .10/.10	6.8	94	11.9 ± 2.3	96.8 ± 5.3
8/1	30/15	4/4	4/4, .15/.15	6.8	95	11.9 ± 2.2	96.4 ± 5.8
8/1	30/15	4/4	4/4, .20/.20	6.9	99	11.6 ± 2.3	96.8 ± 5.6
8/1	30/15	4/4	4/4, .25/.25	7.0	99	11.2 ± 2.3	96.9 ± 5.6
8/1	30/15	4/4	4/4, .30/.30	7.0	100	10.6 ± 2.5	96.9 ± 6.7
8/1	30/15	4/4	4/4, .35/.35	6.9	99	9.9 ± 2.7	97.1 ± 7.1
8/1	30/15	4/4	4/4, .40/.40	6.9	97	9.4 ± 2.8	96.6 ± 8.4
8/1	30/15	4/4	4/4, .45/.45	6.7	97	8.3 ± 2.9	97.7 ± 6.0

Table 5.9: Summary of tests examining addition of topological noise. The topological disturbances associated with multiple scene graphs is more severe than the noise in these tests.

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/15	0/0	4/4, .25/.25	9.0	98	9.5 ± 2.7	97.8 ± 6.9
8/1	30/15	2/2	4/4, .25/.25	8.3	98	10.5 ± 2.3	97.6 ± 6.3
8/1	30/15	4/4	4/4, .25/.25	7.2	99	11.2 ± 2.3	96.9 ± 5.6
8/1	30/15	6/6	4/4, .25/.25	6.7	98	11.3 ± 2.2	95.8 ± 9.5

tional demands associated with an active sensing paradigm.

The DCA approach evaluates evidence describing the likelihood of a node's attendance in another graph. The evidence is based on node and edge properties and on a local Connectivity Signature. Peak attendance values are identified to form a node-to-node mapping and a global verification step completes the process. After the best matching pairs of nodes are identified, the nodes of one graph are reordered to allow a side by side verification of the graphs' similarity.

Attention has been focused on testing DCA under challenging conditions. Test cases included both topological and feature noise. DCA produced good results in over 95% of test cases and under a wide variety of conditions. These results indicate that DCA has potential in a variety of applications, particularly due to the low reliance on the dynamic range of node and edge properties.

This concludes the discussion of the third of three novel contributions made in this dissertation. The next chapter covers the motivation and implementation of an integrated framework. Testing of individual components within the framework is discussed, as well as integrated testing.

5.13 Acknowledgements

The authors would like to thank Scott M. Thayer for his technical inputs and constructive criticism.

CHAPTER 6

Integrated Framework and System Implementation

At this point each of the main areas of research: segmentation, matching and range acquisition, have been described including their algorithms, implementation and individual tests. Beyond the goal of making these individual contributions, each has been integrated into the systemic RDM framework. Their ability to be integrated into a framework is an important aspect of their real-world utility. This instantiation of RDM is not meant to represent a final turn-key implementation. Rather, its implementation permits validation of individual components in a system context.

This chapter covers the system framework and implementation. Tests have been run on a variety of real scenes to check performance of each module as it operates within a system. Test results are provided along with a comparison to the overall functionality and performance of a number of other systems.

6.1 Motivation for a Systemic Perspective and Implementation

A recent workshop sponsored by NSF/ARPA [15] made a critical recommendation calling for further research efforts in the implementation of end-to-end systems for object recognition. This type of integration puts a high demand in terms of requirements and performance on each individual acquisition, segmentation and matching module. The importance of these added demands is founded in a desire to make technical contributions towards **real** applications.

A systemic perspective was used when defining the requirements for each

processing module of this framework. This perspective manifests itself in terms of coverage, by filling all the gaps to form an end-to-end system, and in terms of a well matched character of each module. For example, the range sensor acquires profiles of data which are processed directly by the segmentation algorithm. The framework combines acquisition, segmentation, modeling, matching and pose determination has been defined.

6.2 Integrated Framework

The integrated framework includes all of the acquisition, segmentation, modeling and matching efforts that were part of this research. The most critical element (in terms of benefits to computational efficiency) is the graph matching. Hence the framework has been named **RDM** for **R**ecognition by **D**irect **M**atching. The framework components are illustrated in Figure 6.4. This is a data flow diagram that shows the individual processing steps and the data exchanged between each module. The data flows between modules are indicated with arrows, beginning with the "scene" at the top and an object description at the bottom. The bulletized items to the right provide descriptions of the processing modules. The RDM framework includes the following components:

- "PRIME:" To Acquire Range Points
- "RPT:" To Segment Points into Regions
- "GEOGRAM:" To Model Each Region
- "DCA:" To Match Scene and Database Models
- "Horn:" To Compute Pose of Scene Object

The functionality and purpose of each of these is reviewed below and put into context of the whole framework.

The **PR**ofile **I**maging **M**odul**E** (PRIME) is the Structured Light ranging sensor. PRIME generates profiles, or slices, of Cartesian range data. This raw range data lacks any type of higher level interpretation, it is simply a collection of individual points.

Range **P**rofile **T**racking (RPT) is an adaptive approach for robust, accurate and computationally efficient segmentation of range images that is based on Kalman filtering. It is well matched to the PRIME Structured Light acquisition process in terms of the rate and form of data exchanged. RPT forms regions of range data that each possess uniform curvature. The uniformity criteria is a polynomial function of the form $z = f(x, y)$ that provides a generic segmentation capability for first or second order curved surfaces.

The purpose of GEOGRAM, **GEO**metrical **GRA**phical **M**odeling, is to generate graphical models from the segmented regions produced by RPT. In general, these graphs have node properties that describe a geometrical shape, and edge properties that describe the relative orientation of the shapes that are located on adjacent nodes. The details of GEOGRAM have not been provided yet in previous chapters and will be described below in Section 6.2.1.

DCA is a new approach to finding subgraph isomorphisms. DCA performs a **D**irect **C**lassification of node **A**ttendance to describe the likelihood of a node being present as an isomorphic member of another graph. Based on this classification a node to node assignment is made between a scene and database graph. The graphs generated by GEOGRAM contain strictly relative data that is extracted from the scene - no absolute positions are used during the matching process. This separates the recognition from matching process and is a distinctive character to this framework. Absolute positions are computed for the centroid of each geometrical primitive, for use in determining an object's pose.

The block "Horn's Method" in Figure 6.4 is an established technique for finding pose [28]. The method allows for an overdetermined specification of the

transformation and finds a least norm solution. This technique meets the requirements of the system and does so in an optimal fashion. An iterative version of this technique has been implemented. It performs a set number of iterations (3) with reweighting to remove outliers. The residual fitting error of the position of each matched geometrical model is used as a final verification of the recognition and pose processing at the "Verify" stage.

6.2.1 Geometrical Modeling and Graph Construction

To implement a complete system it was necessary to form models of each segmented region as well as the graph structures describing each scene. A number of representation issues come into focus here and are addressed in the sections below. In general, these models have node properties that describe a geometrical shape, and edge properties that describe the relative orientation of the shapes that are located on adjacent nodes.

A number of factors affect stability and computational efficiency when constructing a graphical model. It will be shown that a representation based on geometrical models yields an improved stability of node and edge properties. The disadvantage of using geometrical models arises from computational factors. The fitting of a geometrical model to noisy data is a nonlinear least-squares problem. Solution methods for this type of problem are iterative in nature [171]. This introduces problems associated with having good initial conditions, and the problem of nondeterministic completion times. Also, the computational burden increases considerably as more unknowns are added.

The approach taken herein provides the benefit of a more stable geometrical model while mitigating some of the computational problems. This is achieved via a two stage processing approach. First RPT is used to segment range points into outlier-free regions. The curvature of each region is then analyzed to determine the appropriate category of geometrical model. This analysis also provides the

principle directions of the curved patch. The directional information is used to reorient the range data prior to the iterative fitting. This technique reduces the number of unknowns associated with certain geometrical models, allowing solutions to be obtained more rapidly.

Challenges associated with the representation and modeling process are described in the following sections along with details of this modeling approach. The technique used for modeling herein is called **GEOGRAM** which stands for **GEO**metrical **GRA**phical **M**odeling.

Stability of Centroids with Geometrical Models

One immediate question arrises when considering the specifics of a graphical representation: Why use geometrical models at all? Why not build a representation having node properties based on the $z = f(x, y)$ polynomial models generated by RPT? The problem with this type of approach arises from the instability of certain features. These instabilities are a consequence of shadowing effects that occur during the range acquisition process.

The location of each part of a graphical model must be described in some fashion in order to support recognition and pose calculations. The location of each segmented region is best described by some type of centroid, because of the reduction in noise that occurs with the process of taking a mean [28].

Consider using the centroid of a geometrical model versus the centroid of a polynomial patch. Each side of Figure 6.1 shows a surface patch observed in a given measurement geometry. The left side shows the observed surface when the cylinder's axis is normal to the laser plane. On the right, the cylinder axis lies in the laser plane. Because of the shadowing effect of the sensor, the observed patch on the right side of the figure is not symmetric about the true cylinder. This causes the patch centroid and the geometrical centroid to have a poor coincidence. Hence, the geometrical models yield an improved degree of centroid stability.

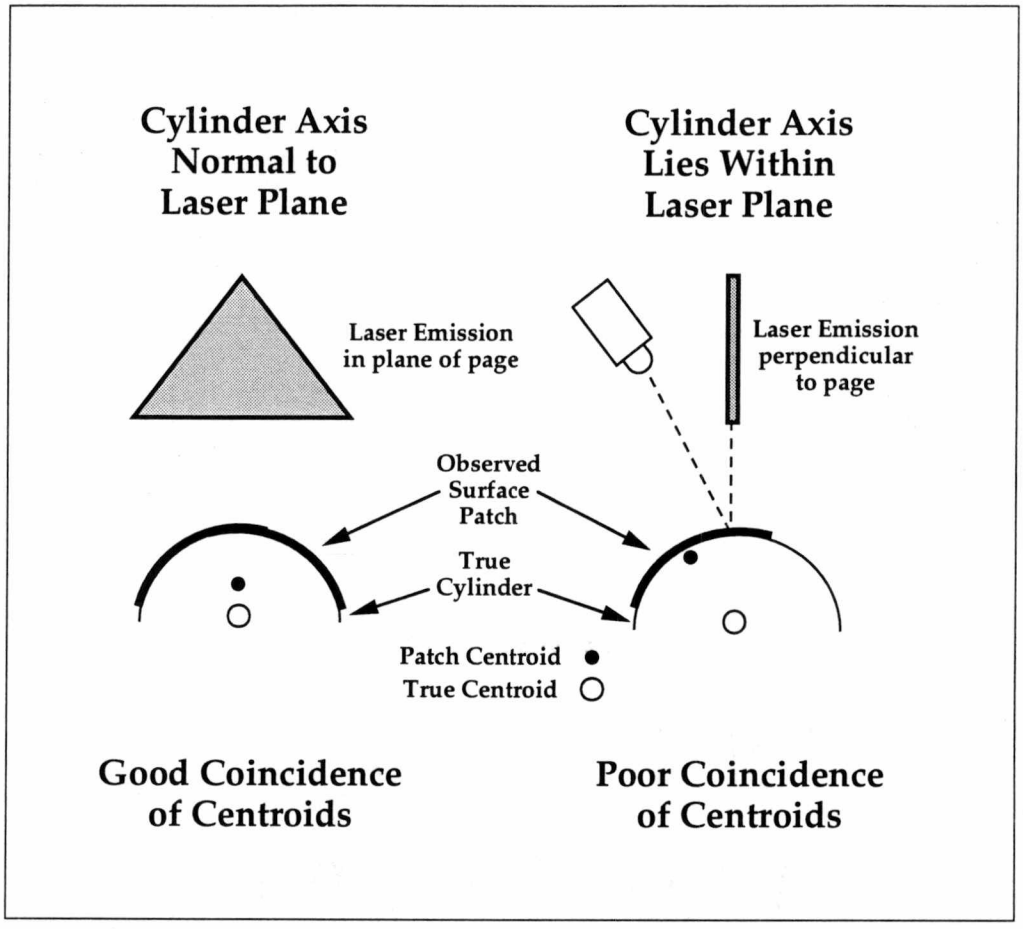


Figure 6.1: The centroid of a segmented patch is highly dependent on measurement geometry. This can result in a poor coincidence between patch and shape centroids.

Stability of Region Boundaries with Matching Types of Models

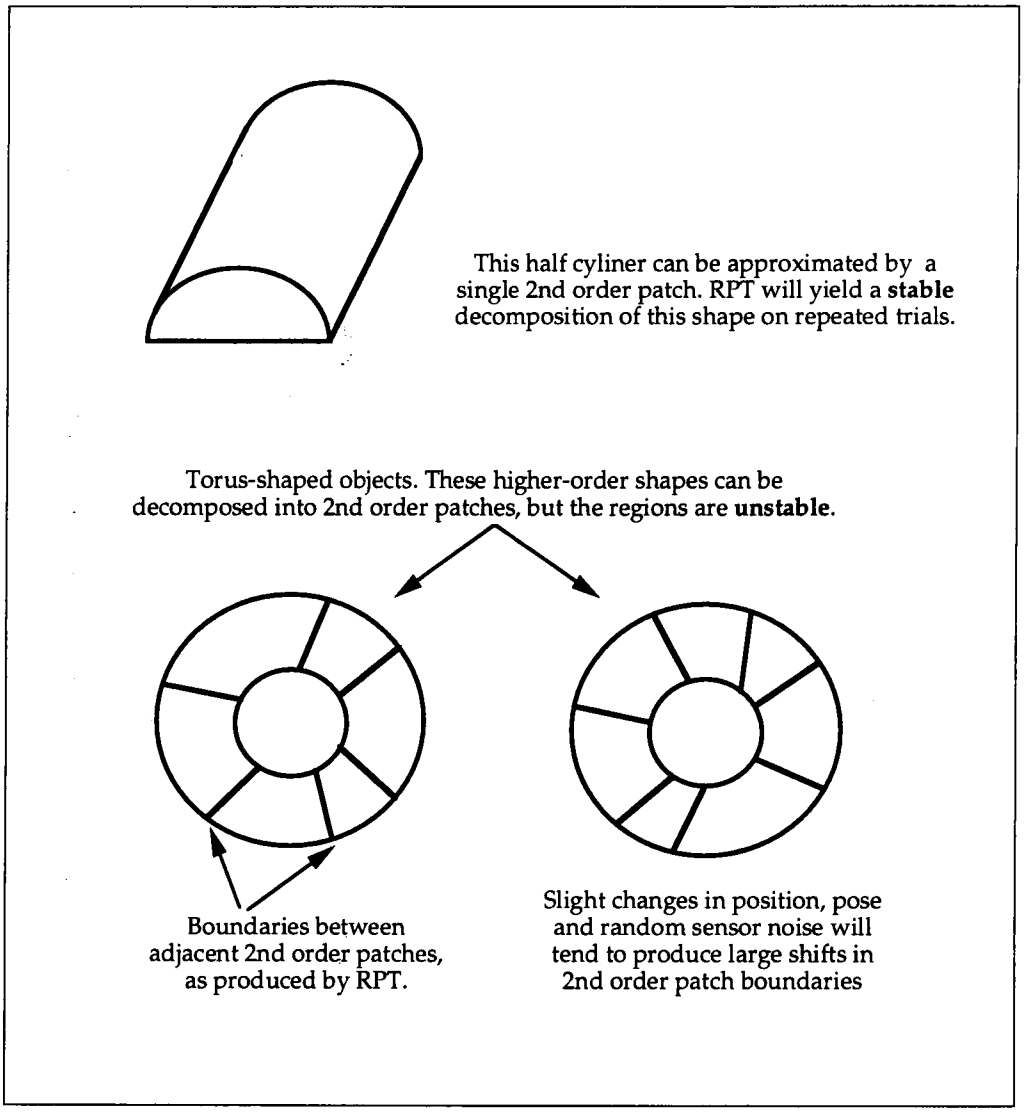


Figure 6.2: Objects that are not approximated well by polynomial surface patches tend to have unstable region boundaries.

This system has been developed with the primary goal of modeling and recognizing objects having components with geometrical forms. It is important that the form of surface model match the true surface. When this is not the case, instabilities in region boundaries can arise. This is illustrated in Figure 6.2. Here, RPT has been used to segment a torus-shaped object. Because the $z = f(x, y)$

surface patch model does not match the true surface in this case, a small shift in pose or in the input noise will tend to produce large shifts in region boundaries.

It has been observed during experiments with RPT that a 1/4 cylinder can be segmented with greater stability than a 1/2 cylinder. This is because the polynomial approximation matches the 1/4 cylinder much more closely. These results provide an example of the shape stability problem.

Clearly it is beneficial to have model forms that match scene conditions. This is actually achieved in a two stage process. RPT runs first, handling various shapes in a somewhat generic fashion, albeit with a limited stability. Then GEOGRAM executes, having more application-specific modeling capabilites. GEOGRAM uses a suite of models that better match the true scene content. This two stage process provides a computational advantage, as will be discussed in Section 6.2.2.

Generally speaking, segmentation algorithms have parameter settings that adjust the aggressiveness of merging operations. Adjustments of this sort of parameter will tend to cause the final number of segmented regions to be slightly higher or lower. In this system it is best to adjust a segmentation algorithm so that it tends to yield **too many regions**. This provides GEOGRAM with an opportunity to merge adjacent regions once the more appropriate forms of models are introduced.

Observed Inaccuracies of Polynomial Models

An observation was made that seemed to diminish the benefits of creating a representation built on the polynomial models. This effect was noticed while performing Monte Carlo simulations [153] as a means to evaluate various representations. It was observed that upwards of a 10% shift in the polynomial coefficents occured when the size of scene elements changed.

It is believed that this is due to the quantization noise that is introduced when range data from relatively small objects is binned into a horizontal grid. In these

cases not only is the number of data points supporting the curve fit reduced, but also these data are corrupted due to the binning process. This parameter shift represented a pretty serious concern, if the polynomial coefficients were to be used as a feature vector for recognition. This observation amounted to yet another reason to not use the polynomial models as a basis for the node and edge properties in the graphical representation.

6.2.2 The Iterative Nature of Geometrical Modeling

The benefits of creating a graphical representation based on geometrical elements are evident in the above discussion. However, the computational difficulties still remain. These computational matters are now considered, beginning with a discussion of the iterative nature of the geometrical modeling process.

The need for iterative solution methods can be seen even in the case of simple geometric forms, such as a circle. To demonstrate the problem of a closed form approach, this formulation will be attempted and its inadequacy revealed.

Points (x_i, y_i) on a circle of radius r obey the relationship

$$(x_i - h)^2 + (y_i - k)^2 = r^2 \tag{6.1}$$

where (h, k) is the center. Closed form techniques that provide a least norm solution [157] may be formulated using an error function, E. A set of simultaneous equations revealing the least norm solution is then setup by taking partials of E with respect to each unknown and equating these to zero. The error $E = E(h, r, k)$ for N points is

$$E = \sum_{i=1}^{N} ((x_i - h)^2 + (y_i - k)^2 - r^2)^2 \tag{6.2}$$

A solution $[h_o k_o r_o]^T$ having minimal error has the property

$$\frac{\partial E}{\partial h} = \frac{\partial E}{\partial k} = \frac{\partial E}{\partial r} = 0. \tag{6.3}$$

The problem with formulating the solution in this manner can now be seen. The partials in Eq. 6.3 are *nonlinear* functions of the unknowns. For example, $\frac{\partial E}{\partial r}$ has the form

$$-4r\left((x_i - h)^2 + (y_i - k)^2 - r^2\right). \tag{6.4}$$

The second order terms prevent the formation of a linear set of equations. This necessitates an iterative means of solution.

The problem of having nonlinear partials will occur in any type of geometrical model having terms in the form $c_1(x - c_2)^2$. Terms of this sort introduce both the unknowns (c_2) and their squares (c_2^2) into the solution vector. While it is possible to setup a least squares formulation that contains both c_2 and c_2^2 as unknowns, for example, the least norm solution does not guarantee the second order constraint between these unknowns.

6.2.3 GEOGRAM Representation

The definition of node and edge properties is given in this section. Nodes correspond to geometrical models and edges exist wherever geometrical models are determined to be adjacent. Node properties describe geometrical shapes. Edge properties describe the relative orientation of the shapes that are located on adjacent nodes.

It was not attempted to make node and edge properties extremely "rich" in nature, in the current implementation of this system. Rich feature vectors having a high dimensionallity can help the recognition process by reducing the potential number of scene-to-database matches. However, in the current implementation simple features were used that provided more challenging tests. Future efforts may expand upon these feature vectors.

Generally speaking, simpler feature vectors can provide an advantage - if, in simplifying the features, an improvement in stability or repeatibility can be achieved.

Table 6.1: Node properties used in GEOGRAM. Node property vectors contain two elements, an integer describing the type of shape and a real value describing an associated parameter.

	Type of Shape	Shape Parameter
1	Plane	Area
2	Cylinder	Radius
3	Sphere	Radius

In the present system three types of geometrical models are used: planes, cylinders and spheres. The features used for each node are given in Table 6.1. Edge properties are given in Table 6.2.

Node and edge property vectors each have two components, one discrete (integer valued) and the other analog (real valued). The types of shapes ("Plane" or "Cylinder-Cylinder") are discrete. Comparisons of discrete properties must match exactly. Shape parameters ("Radius") are analog. These must match to within a set tolerance.

Note that all the parameters used for node and edge properties are relational in nature. This is consistent with one of the basic tenets of this research. Centroids are computed for each node also (which are absolute), but these are used for pose calculations, not during graph matching.

6.2.4 GEOGRAM Approach

Processing in GEOGRAM begins with a curvature analysis of the polynomial surfaces computed by RPT. This allows the style of geometrical model to be determined. After finding the geometrical model, the original range image is analyzed to determine which regions are neighbors. A graph is then created

Table 6.2: Edge properties used in GEOGRAM. As with nodes, edge property vectors contain two elements, an integer describing shape types and a real value describing an associated parameter.

	Types of Adjacent Shapes	Parameter
1	Plane-Plane	Angle Between Normals
2	Plane-Cylinder	Angle Between Normal and Axis
3	Plane-Sphere	Normal Distance Between Plane And Centroid
4	Cylinder-Cylinder	Angle Between Axes
5	Cylinder-Sphere	Normal Distance From Axis To Centroid
6	Sphere-Sphere	Distance Between Centroids

having the appropriate node and edge properties.

The curvature analysis provides an additional benefit for cylindrical models. Nominally a cylindrical model would have 6 degrees of freedom (DOF): 3 for the centroid, 2 for angular orientation, and 1 for the radius. This many DOFs would introduce a rather serious compute burden. To reduce this effort in the case of a cylinder, the principle directions of the surface patch are computed and used to reorient the range data so that the cylinder's axis is parallel one coordinate axis. This eliminates 2 orientation and one displacement DOFs from the iterative processing. This actually reduces the computational burden of the cylinder to below that of the sphere (which has 4 DOF - centroid plus radius). The curvature analysis also provides all the data needed for the planar model.

Curvature Analysis

Analysis of the curvature of the $z = f(x, y)$ patches yields the type of surface and, in the case of a cylinder, yields the principle directions of the surface. This is accomplished via a covariance matrix, λ, of surface normals. The eigenvectors of λ indicate the directions of dominant variations of curvature. Each corresponding eigenvalue indicates the degree of curvature in the given direction. This type of technique is somewhat similar to a method used for range image segmentation [55], but here the patches have already been segmented, are considerably larger, and the usage is for descriptive purposes only.

Generally speaking, λ describes the typical variations of the surface normals and the eigenvectors give the direction inwhich these variations occur. The nature of the covariance matrix can be appreciated by considering the equation

$$x^T \lambda x = 1. \tag{6.5}$$

Provided that λ is positive definite, the vectors x that satisfy 6.5 form an n-dimensional ellipsoid centered at the origin [149]. The eigenvectors of λ coincide with the axes of this ellipsoid, and the eigenvalues provide an indication of the length of each ellipsoidal axis. The positive definiteness of λ is considered below.

The covariance matrix is defined as

$$\lambda = E\{(n_i - m)(n_i - m)^T\} \tag{6.6}$$

where m is the mean of the normals, n_i. For N normal vectors the mean and covariance can be approximated by [172]

$$m = \frac{1}{N} \sum_{i=1}^{N} n_i \tag{6.7}$$

and

$$\lambda = \frac{1}{N} \sum_{i=1}^{N} (n_i - m)(n_i - m)^T. \tag{6.8}$$

In the case of a sphere, variations in the surface normals will be reasonably well distributed in all 3 directions. In this case the 3 eigenvalues will have a magnitude on the order of the radius of the sphere. A threshold was used to distinguish between "dominant" and "non-dominant" eigenvalues. This threshold was set at 1/1000, giving a sphere 3 dominant eigenvalues.

Cylindrical surfaces have variations in their normals in two directions. This is indicated by two dominant eigenvalues. The non-dominant eigenvector points in the direction of minimal curvature which is coincident with the direction of the cylinder's axis. For a planar surface only a single eigenvector is dominant. Variations in surface normals in the other directions are due to noise only.

In [157] it is shown that $\boldsymbol{\lambda}$ is symmetric and positive definite, except for some "unusual" cases inwhich $\boldsymbol{\lambda}$ is positive semi-definite. In these cases an $\boldsymbol{x}$ vector exists such that $\boldsymbol{x}^T\boldsymbol{\lambda}\boldsymbol{x} = 0$. Assuming that there are a relatively large number of surface normals that contribute the computation of $\boldsymbol{\lambda}$, and that some noise is present in each component of the normals then the magnitude of the smallest eigenvalue will be related to the noise intensity.

Surface normals are computed analytically, given the surface patch model. A point $\boldsymbol{p}_i$ on the surface is

$$\boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \\ \boldsymbol{z}(\boldsymbol{x}_i, \boldsymbol{y}_i) \end{bmatrix} \tag{6.9}$$

where $\boldsymbol{z}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \boldsymbol{m}_0\boldsymbol{x}^2 + \boldsymbol{m}_1\boldsymbol{y}^2 + \boldsymbol{m}_2\boldsymbol{x}\boldsymbol{y} + \boldsymbol{m}_3\boldsymbol{x} + \boldsymbol{m}_4\boldsymbol{y} + \boldsymbol{m}_5$. The normal $\boldsymbol{n}_i$ is formed by the cross product of the tangents at the $\boldsymbol{i}_{th}$ point. The tangent in the $\boldsymbol{X}$ direction is

$$\boldsymbol{t}^x = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{x}} = \begin{bmatrix} 1 \\ 0 \\ 2\boldsymbol{m}_0\boldsymbol{x} + \boldsymbol{m}_2\boldsymbol{y} + \boldsymbol{m}_4 \end{bmatrix} \tag{6.10}$$

and the tangent in Y is

$$t^y = \begin{bmatrix} 0 \\ 1 \\ 2m_1 y + m_2 x + m_3 \end{bmatrix}. \tag{6.11}$$

The cross product yields

$$n = \begin{bmatrix} -2m_0 x - m_2 y - m_3 \\ -2m_1 y - m_2 x - m_4 \\ 1 \end{bmatrix}. \tag{6.12}$$

This expression for n can then be evaluated at points within the region of support for the patch, $n(x_i, y_i)$. This method is prefered to using discrete points in the range image because of the smoothing that occurs during the least squares fitting of the surface patch.

The eigenvalues and eigenvectors of λ are found using the Jacobi method [148]. This method is guaranteed to work for all real symmetric matricies [150]. It is also faster than the QR method for this 3x3 covariance matrix. Jacobi works by performing a series of similarity transforms on λ until a near diagonal matrix is achieved. This near diagonal matrix has offdiagonal entries with a magnitude on the order of the machine precision, making them essentially zero. The diagonal entries contain the eigenvalues. The product of the similarity transforms yields a matrix containing the eigenvectors.

Iterative Model Fitting

Parameters of the geometrical models are found using the method of Hooke-Jeeves [173]. This method encorporates both exploratory search steps along the axis of each unknown, as well as a pattern search component. The pattern search moves in a direction set by the trend established on previous iterations. This

type of technique can yield faster convergence in many cases. For example, if the direction of the gradient is not exactly aligned with a coordinate axis, then a technique using only exploratory steps would have to use a step-like approximation to the gradient. In Hooke-Jeeves the pattern step can form a better approximation to the gradient. The exploratory steps are performed at each iteration inorder to track changing directions in the gradient.

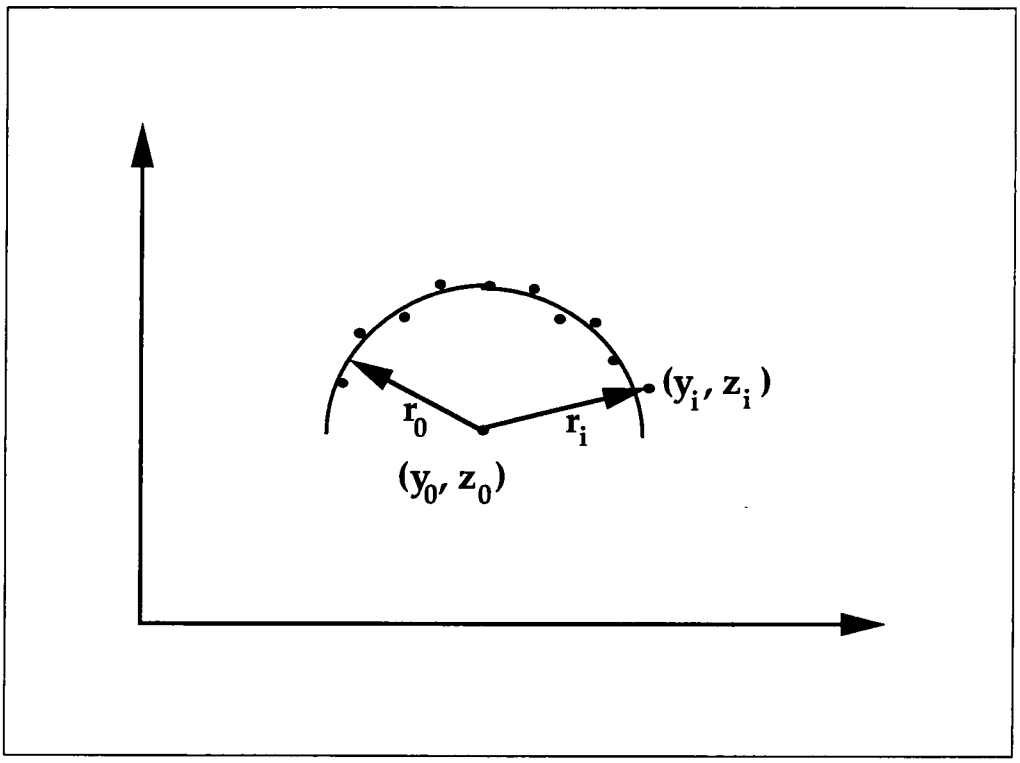


Figure 6.3: The cost function for finding a cylinder sums the distances from each range point to the cylinder surface.

The cost function for finding a cylinder is illustrated in Figure 6.3. The error associated with the unknowns (y_o, z_o, r_o) is found by summing the distance between each range point and the surface of the cylinder. The cost function for a sphere is formed in a similar manner using a 3 DOF centroid plus a radius.

6.2.5 Finding Neighbors

Finding neighboring regions in a range image is a straight forward operation. Because of shadowing effects, neighbors sometimes appear as only being "close" and not touching. To alleviate this problem a dilation is performed [142]. This grows each region to fill in small gaps. This can lead to improper adjacencies too, of course. To help reduce this type of error the length of borders is computed and a threshold is applied. Border lengths are determined by making a single pass over the image and counting the number of occurrences two different regions occupy adjacent pixels.

6.3 Computing Platforms and System Implementation

RDM has been implemented as a networked system having some processing elements that run under a real-time operating system (vxWorks) and others that run under Unix. Figure 6.4 illustrates the RDM system components. The user has a Unix command-line interface to the system. Procedure files are typically used when running large numbers of tests. A previous version had used some socket connections between modules for greater speed. This mode of operation has now been replaced by a method exchanging range data through files. Despite the loss of speed this method is favored because it generates a permanent record of all intermediate results.

In the current configuration, PRIME runs under the real-time operating system. RPT and GEOGRAM execute within one Unix program. DCA and the pose calculation execute in another. The pipelined processing of RPT/GEOGRAM and DCA/pose has been distributed on separate machines. This would permit greater throughput in a real application.

The emphasis on the implementation herein was not so much on achieving the fastest and cleanest possible version of software, but rather, on having extensive

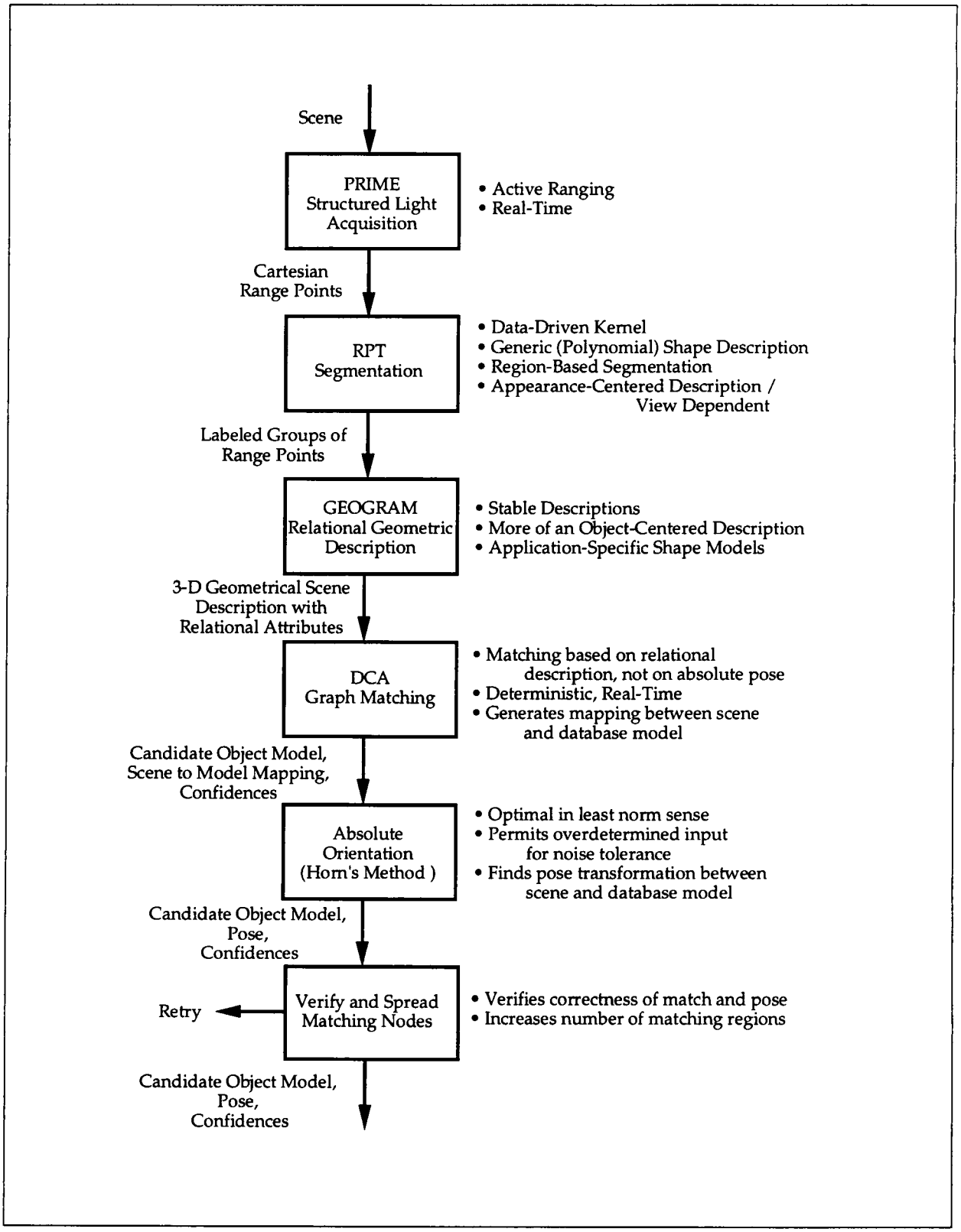


Figure 6.4: Components of the RDM framework for object recognition.

performance measures built in. This goal was achieved by adding data storage fields to the graph data structures. This permitted intermediate results from all stages of processing to be accumulated and made available for analysis and storage.

6.4 Testing Overview

Now that the individual components of this effort have been discussed as well as their integrated framework, end-to-end testing will be covered. These tests provide a validation for the individual contributors to the framework. The tests also demonstrate the fact that the individual contributions were created with a systems perspective in mind.

Careful thought went into the test conditions used for integrated testing. Types of surface curvature and dicontinuities were important considerations, as were the complexity of objects and the number of database models. In more recent articles on range image processing [11,33,35,36] three types of surface discontinuities have been used: jump, crease and curvature. These have discontinuities in height, slope and slope derivative, respectively. Both first and second order surface curvature was studied. Tests were performed with simpler objects having one type of surface and one type of discontinuity, other tests used more complex objects having mixed types. The tests involving the simpler objects paralleled the synthetic trials discussed in Chapter 4. It was also desired to use objects that were relatively complex, compared to those used in other range-based end-to-end systems that have been reported [3,5,37,38,131].

A number of other factors also introduced constraints associated with the creation of the objects used in testing. These included the field of view and shadowing effects of PRIME, the limitations of RPT in handling curvature discontinuities, the suite of available geometrical models in GEOGRAM and the

minimum number of nodes required by DCA.

Another important factor affecting the style of objects that were constructed had to do with the limited viewpoint of the sensor. It was chosen to not explore methods of range data fusion during the course of this research. Hence, objects had to be designed that were essentially one-sided and that could be observed from a top view.

Tests documented in Chapter 5 explored the effect of DCA's performance when recognizing scene subgraphs containing half of the nodes of a database graph. Based on these studies it is felt that DCA can perform well when operating on graphs describing one side of an object. However, this style of testing has not been studied to date with real scenes. The graphs generated from these real scenes did have some topological noise - they were not complete descriptions - hence this type of noise source was present, but not to the degree of the studies in Chapter 5.

The typical sort of object used for system testing is shown in Figure 6.5. To make the recognition problem challenging, each object was given the same number of surface features (10). Also, all cylindrical surfaces were made to have the same radius. The planar regions had similar dimensions also. Furthermore, the relative orientation between surfaces was restricted to a few angles: 0, 45, and 90 degrees. This low dynamic range of object features was used to demonstrate the ability of DCA by **not** providing any highly distinctive features.

Figures 6.5 through 6.8 show an image of Object #1, the measurements acquired by PRIME, The X-Strips formed by RPT followed by the fitted regions, and lastly the graphical representation produced by GEOGRAM. Sets of these types of figures are repeated for each object in Figs. B.1 through B.28 in the Appendix. In the figures depicting graphs (Figure 6.8, for example), nodes are labeled p_i and c_i, indicating planes and cylinders, respectively. There is a correspondence in the physical locations of the nodes with the centroid of each segmented region in Figure 6.7.

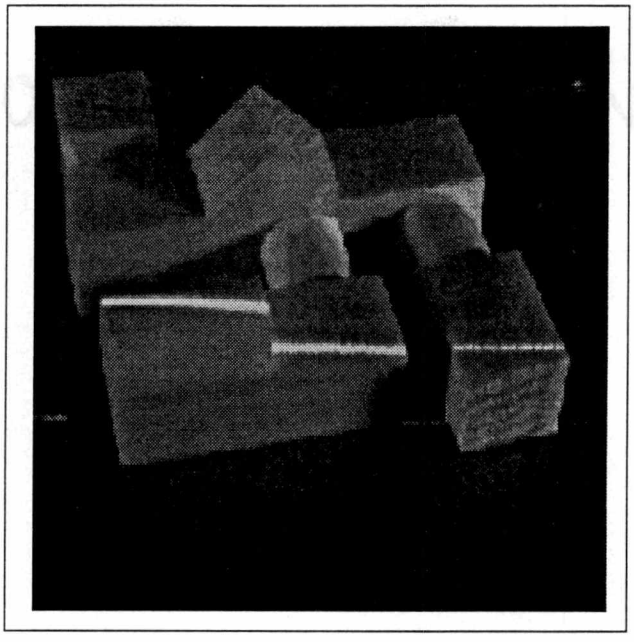


Figure 6.5: Object #1.

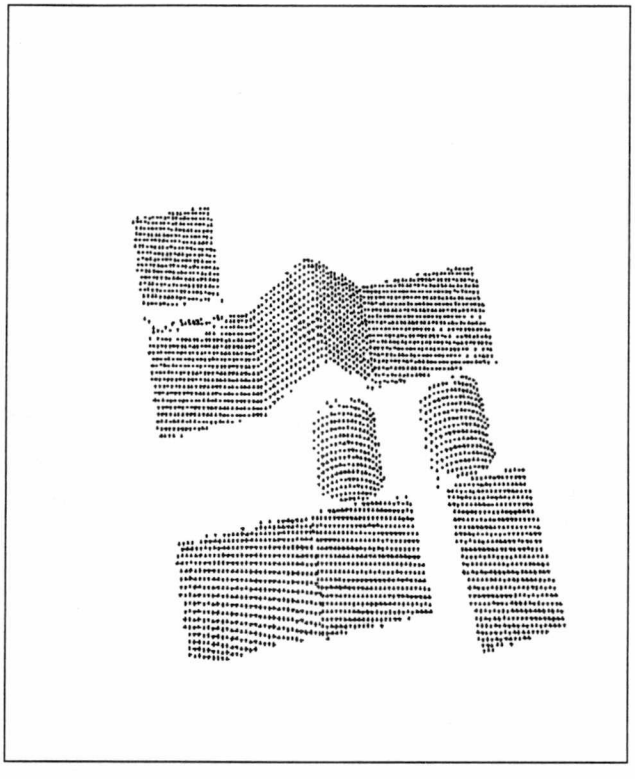


Figure 6.6: Measurements captured by PRIME of Object #1.

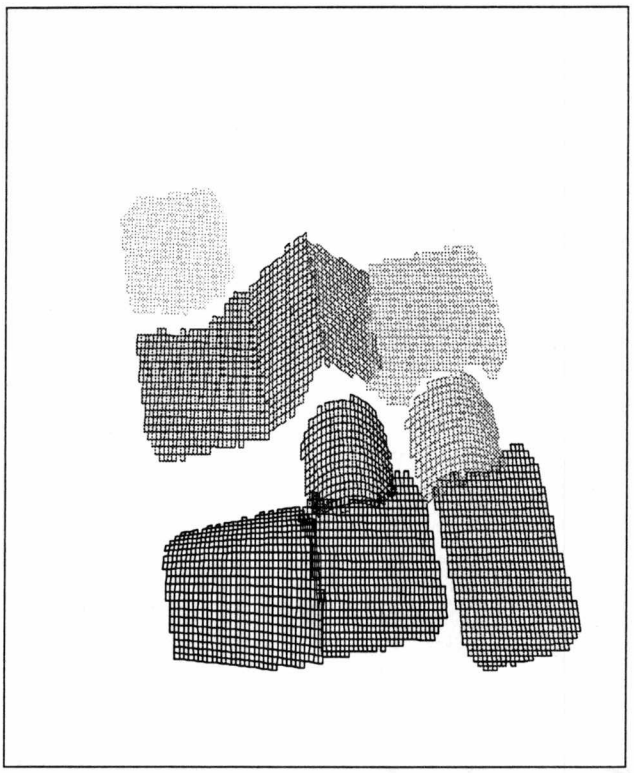


Figure 6.7: Surfaces fitted by RPT for Object #1.

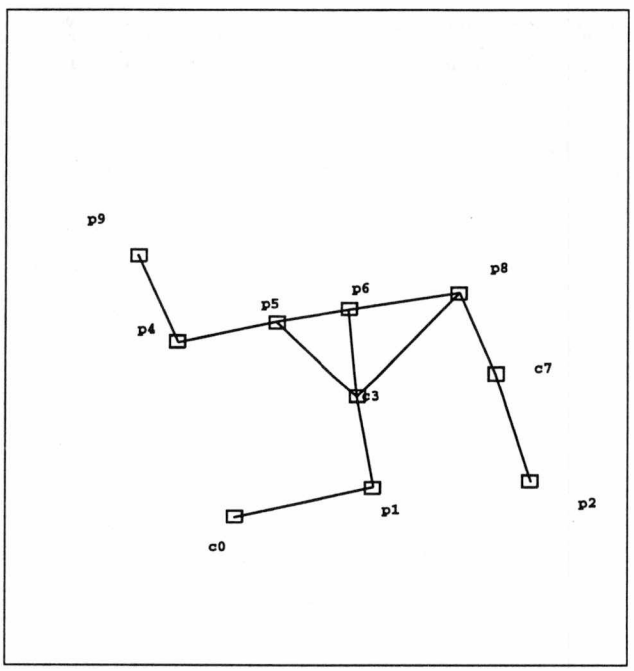


Figure 6.8: Graph formed by GEOGRAM of Object #1.

6.5 PRIME Testing

The important aspects of performance for PRIME were **accurate measurements** and **real-time acquisition**. Acquisition was a deterministic process and clocked by the sensor's video camera. The acquisition process performed reliably, maintaining video rate processing during all scanning operations.

To test measurement accuracy, "smooth" surfaces were scanned and then fitted with polynomial models of the form $z = f(x, y)$. The mean absolute residual error of these fitted models was used as a measure of accuracy.

No measures are available of the true smoothness of the objects used in testing. Wooden blocks were used for most testing because of the flexibility of forms available. Machined metal plates were used during calibration. Table 6.3 gives the residual errors for the various surfaces. The result from the machined plate is favorably biased because of the higher order polynomial model that was used to fit points on the (essentially) flat plate. This higher order model will tend to fit to the noise, slightly reducing the mean absolute error. The case of the 1/4 cylinder may be somewhat deceptive due to this same sort of effect. A value of 0.005 inches is used for a conservative estimate of the accuracy of PRIME.

6.6 RPT Testing

The important aspects of performance for RPT are the **speed** of the process, and the **robustness** and **accuracy** of the segmented regions.

Herein, robustness has been quantified using the number of regions that RPT yields. Ideally these should be constant when observing a scene from a fixed viewpoint. The robustness is expressed as a percentage of the number of trials having the correct number of segmented regions. This definition of robustness may not be typical, but it is a practical, "bottom-line" measure of overall effectiveness.

Table 6.3: Accuracy of PRIME sensor as measured by the mean absolute residual error. The entry for the machined plate below is taken from the results of the calibration procedure. Sensor noise appears to have an upper bound of < 0.005 inches.

Type of Surface	Type of Fit	Accuracy (inches)
Flat Wooden Block	1st Order	0.005
Wooden Quarter Cylinder	2nd Order	0.004
Flat Machined Plate	4th Order	0.002

This definition emphasizes the stability and consistency of the process.

Accuracy for RPT is defined in terms of the consistency of the sizes of segmented regions. For this, statistics on the volume and residual fitting error have been computed. Execution speeds have also been tallied.

Tables 6.4 through 6.7 describe tests with the same types of discontinuities as those used in the synthetic scenes of Chapter 4. These are: jump, crease and curvature discontinuities. Both planar and 2nd order surfaces are used in these tests. Images of the objects used in these tests can be seen in Figures 6.9 through 6.12

Table 6.4: Test results for RPT on real scenes containing a planar surface bounded by jump discontinuities. The volumetric error and surface fits indicate a high accuracy, for this type of simple surface, but with somewhat limited robustness, as indicated by the region count. Trials with an incorrect count typically had 2 or 3 regions present.

Volumetric Error	2.3%
Surface Fitting Error	0.005 in.
Correct Region Count	60%
Mean On-Line Compute Time	0.012 sec.
Mean Off-Line Compute Time	9.1 sec.
Number of Trials	50

Table 6.5: Test results for RPT on real scenes containing a 2nd order surface bounded by jump discontinuities. This surface was a 1/4 cylinder that can be closely approximated by a $z = f(x, y)$ polynomial. This shows an improved performance over the planar case.

Volumetric Error	1.8%
Surface Fitting Error	0.005 in.
Correct Region Count	96%
Mean On-Line Compute Time	0.012 sec.
Mean Off-Line Compute Time	3.9 sec.
Number of Trials	49

Table 6.6: Test results for RPT on real scenes containing two planar surfaces separated by a crease discontinuity. Accuracy measures are reported for both of the two surfaces. Note the improved robustness indicated by the correct region count.

Volumetric Error	6.9, 8.3%
Surface Fitting Error	0.005, 0.005 in.
Correct Region Count	86%
Mean On-Line Compute Time	0.012 sec.
Mean Off-Line Compute Time	5.9 sec.
Number of Trials	50

Table 6.7: Test results for RPT on real scenes containing a 2nd order surface bounded on two sides by a curvature discontinuity at a join with planar surfaces. Accuracy measures are reported for each of the 3 surfaces. Accurate segmentation has not been achieved by RPT with these curvature discontinuities.

Volumetric Error	54, 54, 55%
Surface Fitting Error	0.0.005, 0.006, 0.005 in.
Correct Region Count	76%
Mean On-Line Compute Time	0.013 sec.
Mean Off-Line Compute Time	8.2 sec.
Number of Trials	50

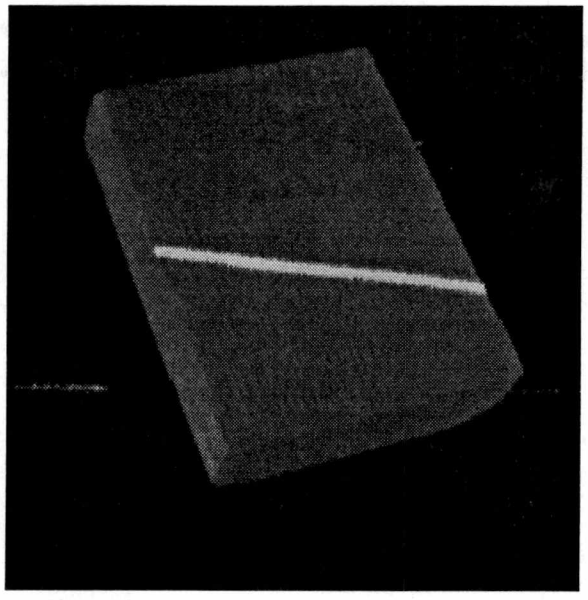


Figure 6.9: Typical scene for testing RPT for a planar surface bounded by jump discontinuities

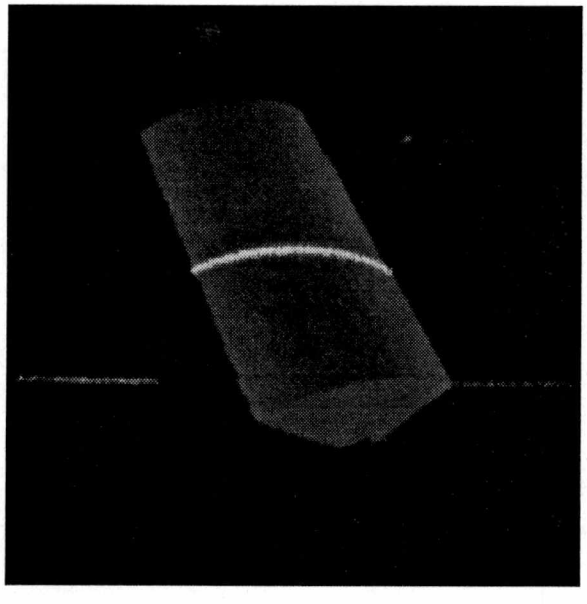


Figure 6.10: Typical scene for testing RPT with a 2nd order surface bounded by jump discontinuities.

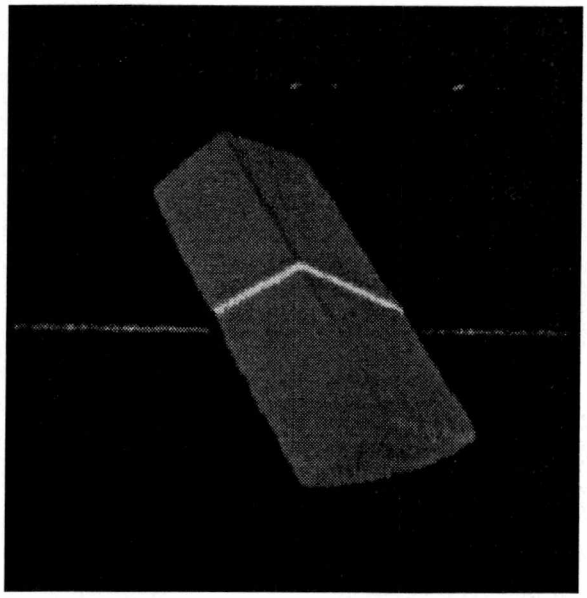


Figure 6.11: Typical scene for testing RPT with two planar regions separated by a crease discontinuity.

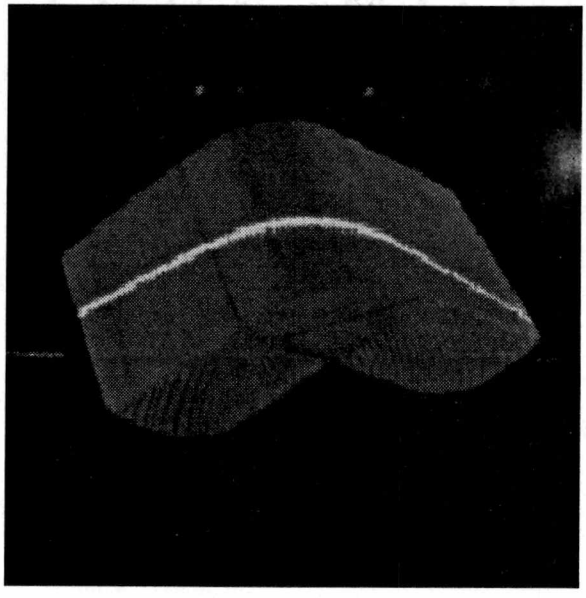


Figure 6.12: Typical scene for testing RPT with curvature discontinuities.

It can be seen in Table 6.7 that curvature discontinuities result in greater inaccuracies than the other, less subtle, types of boundaries. Another interesting trend can be seen that the simplest style test - jump discontinuities surrounding a planar surface - does *not* yield the best results. These cases ranked more poorly in robustness than the more complicated scenes. This was somewhat surprising. It is believed that this is because the residual error becomes quite low for these simple planar surfaces. During region growing, the median absolute residual (MdAR) is used as a basis for a threshold limiting the the addition of new pixels to a region. When the MdAR becomes too small, surfaces tend to remain fractured with an excess number of regions. It is believed that this can be remedied by establishing a lower bound on the accepted MdAR.

6.7 GEOGRAM Testing

The important aspects of performance for GEOGRAM were **speed** and **accuracy** of the geometric model parameters. The definition of these model parameters varies somewhat depending on the type of model, but generally include: 1) centroid location, 2) any orientation data, and 3) any shape parameter associated with the model.

Parameters used in the GEOGRAM models are summarized in Table 6.8. The method used to compare instances of these parameters is also given in the table. In each case a scalar is computed for the comparison metric. Test results for position and orientation errors are given in Tables 6.9 and 6.10, for planes and cylinders, respectively. GEOGRAM was tested under cluttered scene conditions using Objects #1 through #8.

Note that because cluttered scenes were used for these tests it was necessary to first recognize objects before these accuracy measures could be accumulated. This somewhat intertwines the metrics for accuracy with the RDM system's recognition

Table 6.8: Types of shapes modeled by GEOGRAM and associated parameters. The method of comparing each parameter is indicated. The angular errors have been computed via the absolute value of a dot product.

Type of Shape	Model Parameter	Accuracy Measure
Plane	Centroid Location	Euclidean Distance
	Normal Orientation	Angular Error
Cylinder	Centroid Location	Euclidean Distance
	Axial Orientation	Angular Error
	Radius	Absolute Difference

rate. This seems to be somewhat unavoidable for cluttered real scenes.

6.8 Overall System Testing

The important aspects of system performance were **recognition rate, accuracy** and **speed**. Recognition rate is defined as the fraction of trials that the correct database selection was made. Accuracy is a metric of pose and is described using the displacement and angular error between the scene and database graphs.

To compare orientations, the dot product of unit vectors is computed and the absolute value then taken. This is used to find an angular error via inverse cosine. This method does not penalize the comparison of "norm-up" and "norm-down" for a plane with essentially the same orientation. Similarly the axial directions of cylinders are not penalized in this manner.

Accuracy measures of the system are computed after a matching subgraph between the scene and database is found. The best fit coordinate transforma-

Table 6.9: Results from GEOGRAM tallied for instances of the planar surfaces on scans of Object #1. Data from 33 planes is given. These were located in various locations on the object, as it appeared in various positions and orientations.

Centroid Error	Mean	0.090 in.
	Standard Deviation	0.177 in.
Normal Orientation Error	Mean	0.803 deg.
	Standard Deviation	1.528 deg.
Compute Time (10 models fitted)	Mean	0.874 sec.
	Standard Deviation	0.354 sec.

Table 6.10: Test results for GEOGRAM tallied for instances of the cylidrical surfaces on scans of Object #1. Data from 18 cylinders is given. These were located in various locations on the object, as it appeared in various positions and orientations. Compute durations are the same as in the planar case because these statistics were accumulated for the same set of object scans.

Centroid Error	Mean	0.145 in.
	Standard Deviation	0.239 in.
Normal Orientation Error	Mean	2.838 deg.
	Standard Deviation	5.616 deg.
Compute Time (10 models fitted)	Mean	0.874 sec.
	Standard Deviation	0.354 sec.

tion between these models is then computed. The displacement and orientation differences between these coordinate frames is used to describe system accuracy.

The process of computing pose and then accuracy begins with the node-to-node correspondence between scene and database features provided by DCA. Horn's method for finding absolute orientation [28] is then used to find the best fit transformation. This transform is applied to the scene coordinates. Due to noise throughout the system, there is some residual error between the matched positions. This residual error is used for a final verification of the overall operation. It is this residual that must be quantified to describe the accuracy of the system.

To quantify the position and orientation error of the residual, Horn's technique is used again. The resulting transform typically contains some slight displacement and rotation errors. The displacement error can be read directly from the homogeneous coordinate transform [174]. The displacement error is also equal to the distance between the means of the two sets of corresponding points. Unfortunately, the rotational error is not as conveniently obtained.

Homogeneous coordinate transforms express rotations using a 3x3 orthonormal matrix. This is somewhat cumbersome - a scalar is much more desirable for describing the rotational error. The "axis and angle" representation provides an intuitive method for describing rotations. In this representation an orientation of a specified angle is performed in a right-handed sense about a given axis. Horn describes the conversion between coordinate transformations, quaternions, and the axis and angle representations [28].

Test results for the RDM system are given in Table 6.11. A total of 200 scans were made, 25 of each object. In 70% of the scans the correct object was selected from the database. Pose determination was possible in approximately half of these cases, as determined by a check of the residual error of the pose transformation. The mean duration is by far the fastest of any laser-based graph-matching system

Table 6.11: System performance for scans of each of the 8 objects. Results are given as means. A total of 200 scans were made, 25 of each object.

Displacement Error	0.193 in. (6.4%)
Orientation Error	0.846 deg.
Recognition Rate	70%
Duration	11.5 sec.

reported to date. The difference in recognition rate in these tests versus the 95% figures seen in the synthetic scenes of Chapter 5 reflect the importance of end-to-end, integrated testing.

6.9 Concluding Remarks

Previous chapters have discussed testing of individual modules to demonstrate their utility. Here, the fourth objective of this research has been realized - that of creating components using a systemic perspective that allows them to be put into an integrated framework. This objective was validated by the implementation and testing of an RDM testbed.

A relatively low number of end-to-end testbeds like RDM, using Structured Light-based recognition, have been reported - compared to the very large amount of effort on system components. Complete systems can be found in [2, 3, 5, 12, 27, 38, 74, 131]. Even these examples are somewhat of an "apples-to-oranges" comparison, differing from the RDM implementation in various ways. Some of these systems focus on one or two types of objects. The complexity of objects varies also. Unfortunatley, only a subset of the reported efforts give some kind of performance numbers.

In [74], a single part was used. No performance figures are given. In [27] a single part was also used, here a displacement error of 0.120 in. and orientation error of 2.3 degrees is reported.

The culmination of efforts in [3, 5, 38] has lead to a system yielding similar results to RDM. The primary difference being the use of 2 kinds of objects. A 30% failure rate is reported in these studies. No figures are given on pose accuracy. Their system is based on local feature sets and requires that scenes possess visible corners of objects that have convex shape. Concern is expressed regarding the performance of the system with objects having more complex shapes.

In [2] 10 objects are used. Results of matching appear to be quite good, but no statistics are given for a large number of scenes, nor are any figures given on pose accuracy. The system did take over an hour to analyze a scene.

The work of [61] is a close neighbor to RDM in that a graphical representation is used, and the recognition problem is formulated directly using isomorphisms. This work also includes 8 objects. However, the objects used have planar surfaces only and work with synthetic scenes is reported.

In [131] a system was developed to track the motion of an arbitrarily shaped object. This work focusses on high speed acquisition and pose determination. Segmentation, feature extraction, high level matching were not involved and only a single object was tracked. The effort did yield a 0.040 inch position error and 1 degree orientation error.

Overall the RDM testbed compares well to the above systems, albeit each has some differences in its purpose. For real systems to be developed, specific tasks are needed. The performance for the RDM testbed reflect the given task used in testing. Portions of RDM could be optimized for new tasks involving new objects. Tasks involved in the advanced manufacturing scenarios described in Chapter 1 would typically include constraints and knowledge of specific shapes that could be exploited to optimize performance.

CHAPTER 7

Summary, Conclusions and Future Research Directions

There are three main points of focus in this dissertation, all of which address important aspects of the object recognition problem. These are: **adaptive range segmentation, graph matching** and **calibrated 3-D ranging**. Computationally efficient, robust and accurate means to accomplish these types of problems are critical to improving the state of the art in object recognition. Novel contributions have been made in each of the above three areas in the course of this research. These contributions take the form of algorithms, implementations and a significant level of testing.

In addition to the above three main areas, a fourth objective was also set. It was desired to create components using a systemic perspective, allowing them to be put into an integrated framework. Their ability to be integrated into a system is considered to be an important aspect of their real-world utility. The three main contributions were integrated into a complete system. This testbed is not meant to represent a final turn-key implementation. Rather, its implementation permits validation of individual components in a system context.

7.1 Novel Contributions

The **PR**ofile **I**maging **M**odul**E** (PRIME) is a Structured Light ranging sensor. PRIME generates profiles, or slices, of Cartesian range data. The novelty of PRIME stems from new approaches to calibration and to pixel-level operations. These techniques make for a system that is acquires data at frame rate, is accurate and

easy to calibrate. The calibration technique that has been developed permits both the camera optics and laser geometry to be modeled in a single step via a least norm solution. PRIME has an accuracy of better than 0.005 inches.

Range **P**rofile **T**racking (RPT) is an adaptive approach for robust, accurate and computationally efficient segmentation of range images that is based on Kalman filtering. It is well matched to the PRIME Structured Light acquisition process. The uniqueness of RPT arises from a 3-way combination of adaptive kernel mechanisms. The size and shape of the kernel vary dynamically. This provides dense regions of varying size that are robustly formed. Kernel boundaries occur when outliers disrupt surface smoothness, at object boundaries or in the event of spike noise, for example. In this type of approach the kernel boundaries are not located by a fixed geometry. This helps to avoid the inclusion of outliers within a segmented region. Tests demonstrated that segmented regions can be computed in less than 3 seconds, under a wide variety of test conditions. Stable operation was achieved above 90% of the time in the majority of testing trials.

DCA is a new approach to finding subgraph isomorphisms. DCA performs a **D**irect **C**lassification of node **A**ttendance to describe the likelihood of a node being present as an isomorphic member of another graph. Based on this classification a node to node assignment is made between a scene and database graph. DCA requires a computational effort that is of polynomial order. It is approximate in nature. However, it is able to maintain a performance level for subgraph isomorphisms at or above 95% under a wide variety of conditions and with varying levels of noise. The performance on the full size comparisons associated with graph isomorphisms has been found to be 100/100, also under a variety of conditions.

Careful thought went into test conditions for individual components and for integrated tests. Types of surface curvature and discontinuities were important considerations, as were the complexity of objects and the number of database

models. Some tests were performed with simpler objects having one type of surface and one type of discontinuity, other tests used more complex objects having mixed types. Testing was performed using both synthetic scenes (4000+) and real scenes (500+). The tests with synthetic scenes were helpful because they provided ground truth. The sensor noise level was also ramped up in these studies to examine the stability and accuracy of system components.

7.2 Further Reseach Areas

This effort has opened a number of new doors of opportunity for further research. Opportunities exist for follow on work in each of the three main areas: calibrated range acquisition, segmentation, and graph matching.

7.2.1 Multidirectional Optics

Structured lighting suffers from shadowing problems. These could be reduced by using multiple cameras and/or multiple laser planes. For example, having one laser and two cameras would be useful to reduce shadowing. This type of system would still generate range measurements within a single plane (the laser plane) at any given instant. This would yield more complete laser profiles, in general.

7.2.2 Opposing Directions for Profile Segmentation

Profile segmentation in RPT is directional in nature. As a result of this the rising edge and falling edge of segments do not receive exactly the same processing. It would be interesting to process profiles in both directions and then combine the results.

7.2.3 Alternate Graphical Representations

Currently the difference in orientation between adjacent nodes is used to color edges. In the future it would be interesting to add another dimension to edge coloring that describes the distance between the centroids of adjacent geometrical models.

Other researchers have defined graphs with nodes that contain multiresolution descriptions of objects. For example, some nodes might describe the sides of a polygonal surface, these would be connected to a node describing the entire surface, which would be connected (along with other nodes) to a node describing an entire solid. Hence these node descriptions contain data at different levels of coarseness. The type of topological noise present in this type of graph is not clear, and its impact on the performance of DCA would have to be studied.

7.2.4 Merging Adjacent Models

Another useful extension to GEOGRAM would be the possibility of merging adjacent geometrical models. As stated, this merging would be done in the domain of geometrical models, not in the polynomial surface domain of RPT. It will be important that RPT yield "oversegmented", or "too many" regions, as opposed to "too few" in order to support this effort.

7.2.5 Combine Range Data Acquired from Different Views

This represents a new application of DCA in range image processing. DCA could be used to merge scene graphs that were constructed from slightly different views of an object. This would achieve data fusion at a high level, thus providing a computational advantage. This step would be performed prior to recognition.

7.2.6 Uncertainties in Node and Edge Properties

In the current implementation of DCA, the node and edge property vectors contain typical values for a given object. These were generated experimentally by finding the mean of feature vectors over a number of trials. An improvement to this approach would be to also store the standard deviation of the features for use when comparing scene and database properties.

7.2.7 Fuzzy Graph

Since acquisition and segmentation are not near perfect processes, the nodes and edges present in a scene graph vary somewhat on repeated scans. DCA can be used to find the best mapping between these scene examples, the frequency of occurrence of each node and edge could then be recorded. These occurrences could then be used to form a "Fuzzy Graph". A Fuzzy Graph is then able to describe nodes and edges which always occur, never occur, or sometimes occur. This could be accomplished by tagging the nodes and edges of a graph with a probability value describing the likelihood of it being present. These probabilities would have to be incorporated into comparison norms.

7.2.8 Normalizing Factors

In the current version of DCA the CS vector elements a_{ij}^r are normalized using the largest (i, j) entry for a given r. This approach is probably adequate when dealing with the full-size isomorphism problem. However, when attempting to match subgraphs it is likely that the largest a_{ij}^r entry may result from topological noise that is present outside of a core-matching region. Having different normalizing factors will tend to result in poorer CS vector comparisons.

The choice of this normalizing factor may not be easily determined. As seen in Eq. 5.62 the a_{ij}^r entries grow so rapidly, even with $p < 1$, that a poor choice

could result in a loss of precision.

7.2.9 Search for Additional Matches in Scene

After a matching subgraph has been identified that exists between a scene and database, there may still be some remaining nodes that have physical locations which are consistent with the transform established for the core subgraph. A quick search of the remaining scene nodes could be performed, checking each for consistency and combining those within a close tolerance. This operation would tend to increase the size of the matching subgraphs found in each scene. Techniques like this have been used in the past [22, 72] however, these methods suffer from a poor starting condition and become highly iterative in the worst case. By merging this type of tree-search method with DCA, the starting condition for search is improved significantly. Tree searching can then just be used to clean up any stray nodes from a scene.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] J. R. Ullmann, "An algorithm for subgraph isomorphism," *Journal ACM*, vol. 23, no. 1, pp. 31–42, 1976.

[2] T.-J. Fan, G. Medioni, and R. Nevatia, "Recognizing 3-d objects using surface descriptions," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 11, no. 11, pp. 1140–1157, 1989.

[3] A. C. Kak and J. L. Edwards, "Experimental state of the art in 3-d object recognition and localization using range data," in *Proceedings of Workshop on Vision for Robots in the IROS'95 Conference*, (Pittsburgh, PA), IROS, August 3-5 1995.

[4] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operators," *IEEE Trans. on System, Man, and Cybernetics*, no. 6, pp. 420–443, 1976.

[5] W.-Y. Kim and A. C. Kak, "3-d object recognition using bipartite matching embedded in discrete relaxation," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 13, no. 3, pp. 224–251, 1991.

[6] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone, "A versatile computer-controlled assembly system," *IJCAI*, pp. 298–307, 1973.

[7] M. D. Wheeler and K. Ikeuchi, "Sensor modeling, markov random fields and robust localization for recognizing partially occluded objects," in *Proceedings of The Image Understanding Workshop*, (Wash. D. C.,), DARPA, April 1993.

[8] L. G. Roberts, "Machine perception of three-dimensional solids," in *Symposium on Optical and Electro-Optical Information processing Technology*, (Cambridge, MA), MIT Press, 1965.

[9] T. O. Binford, "Visual perception by computer," in *Proceedings of the IEEE Systems Science and Cybernetics Conf.*, (Miami, FL), IEEE, Dec. 1971.

[10] R. T. Chin and C. R. Dyer, "Model-based recognition in robot vision," *ACM Computing Surveys*, vol. 18, no. 1, pp. 67–108, 1986.

[11] P. J. Besl and R. C. Jain, "Three-dimensional object recognition," *ACM Computing Surveys*, vol. 17, no. 1, pp. 75–145, 1985.

[12] M. Oshima and Y. Shirai, "Object recognition using three dimensional information," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 5, no. 4, pp. 353–361, 1983.

[13] P. J. Flynn and A. K. Jain, "Cad-based computer vision: From cad models to relational graphs," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 13, no. 2, pp. 114–132, 1991.

[14] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves abd two-dimensional shapes," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 8, no. 1, pp. 34–43, 1986.

[15] M. Hebert, J. Ponce, T. E. Boult, A. Gross, and D. Forsyth, "Report on the NSF/ARPA workshop on 3-d representation for computer vision," in *Proceedings of the 7th International Joint Conf. on Artificial Intelligence*, (CUNY Graduate Center, New York City), NSF/ARPA, Dec 5-7 1994.

[16] D. Marr, *Computer Vision Systems*, ch. Representing Visual Information, pp. 61–80. Orlando, FL: Academic Press, 1978.

[17] P. J. Besl, "Active, optical range imaging sensors," *Machine Vision and Applications*, vol. 1, pp. 127–152, 1988.

[18] S. Kweon, R. Hoffman, and E. Krotkov, "Experimental characterization of the perceptron laser rangefinder," Tech. Rep. CMU-RI-TR-91-1, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1991.

[19] R. Nevatia, *Machine Perception*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[20] B. R. Abidi, *Automatic Sensor Placement for Volumetric Object Characterization*. PhD dissertation, University of Tennessee, Department of Electrical Engineering, June 1995.

[21] L. R. Foulds, *Graph Theory Applications*. New York: Springer-Verlag, 1992.

[22] O. D. Faugeras, *Three-Dimensional Computer Vision, A Geometric Viewpoint*. London: MIT Press, 1993.

[23] R. M. Haralick and L. G. Shapiro, "The consistent labeling problem: Part 1," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 1, pp. 173–184, 1979.

[24] L. Cinque, D. Yasuda, L. G. Shapiro, S. Tanimoto, and B. Allen, "An improved algorithm for relational distance graph matching," *Pattern Recognition*, vol. 29, no. 2, pp. 349–359, 1996.

[25] R. A. Brooks, "Model-based three-dimensional interpretations of two-dimensional images," *IEEE Trans. on Pattern Anal. Machine Intell.*, pp. 140–150, March 1983.

[26] Y. C. Hecker and R. M. Bolle, "On geometric hashing and the generalized hough transform," *IEEE Trans. on System, Man, and Cybernetics*, vol. 24, no. 9, pp. 1328–1338, 1994.

[27] O. D. Faugeras and M. Hebert, "The representation, recognition and locating of 3d shapes from range data," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 27–52, 1986.

[28] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of Optical Society of America*, vol. 4, no. 4, pp. 629–642, 1987.

[29] F. W. DePiero, M. M. Trivedi, and S. Serbin, "Graph matching using a direct classification of node attendance," *Pattern Recognition*, vol. 29, no. 6, 1996.

[30] J. J. Koenderink and A. J. V. Doorn, "The internal representation of solid shape with respect to vision," *Biol. Cybern.*, vol. 32, pp. 211–216, 1979.

[31] T. Kanade, A. Gruss, and L. R. Carley, "A very fast VLSI rangefinder," in *Proceedings of the 1991 IEEE Intl. Conf. on Robotics and Automation*, (Sacramento, CA), pp. 1322–1329, IEEE, April 1991.

[32] A. M. McIvor and R. J. Valkenburg, "Calibrating a structured light system," Tech. Rep. 362, Industrial Research Limited, Auckland, New Zealand, February 1995.

[33] K. L. Boyer, M. J. Mirza, and G. Ganguly, "The robust sequential estimator: A general approach and its application to surface organization in range data," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 16, no. 10, pp. 987–1001, 1994.

[34] R. C. Bolles and M. A. Fischler, "A ransac-based approach to model fitting and its application to finding cylinders in range data," in *Proceedings of the 7th International Joint Conf. on Artificial Intelligence*, (Vancouver, B.C., Canada), pp. 637–643, IJCAI, Aug 24-28 1981.

[35] A. K. Jain and S. G. Nadabar, *Markov Random Fields Theory and Application*, ch. Range Image Segmentation Using MRF Models, pp. 543–572. Boston: Academic Press, 1993.

[36] Y. Mainguy, J. B. Birch, and L. T. Watson, "A robust variable order facet model for image data," *Machine Vision and Applications*, vol. 8, pp. 141–162, 1995.

[37] O. D. Faugeras, "New steps towards a flexible 3-d vision system for robotics," in *Proc. 7th Int. COnf. on Pattern Recognition*, (Montreal, Canada), pp. 796–805, IEEE, July 30-Aug. 2 1984.

[38] C. H. Chen and A. C. Kak, "A robot vision system for recognizing 3-d objects in low-order polynomial time," *IEEE Trans. on System, Man, and Cybernetics*, vol. 19, no. 6, pp. 1535–1563, 1989.

[39] R. O. Duda, D. Nitzan, and P. Barnett, "Use of range and reflectance data to find planar surface regions," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 1, no. 3, pp. 254–271, 1979.

[40] E. Trucco and R. B. Fisher, "Experiments in curvature-based segmentation of range data," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 17, no. 2, pp. 177–182, 1995.

[41] S. S. Sinha and R. Jain, *Handbook of Pattern Recognition and Image Processing*. New York: Wiley, 1994.

[42] M. A. Fischler and O. Firschein, *Intelligence: the Eye, the Brain and the Computer*. Reading, MA: Addison-Wesley, 1987.

[43] M. M. Trivedi and A. Rosenfeld, "On making computers see," *IEEE Trans. on System, Man, and Cybernetics*, vol. 19, no. 6, pp. 1333–1335, 1989.

[44] P. J. Besl and R. C. Jain, "Invariant surface characteristics for 3-d object recognition in range images," *Computer Vision, Graphics and Image Processing*, vol. 33, pp. 33–80, Jan 1986.

[45] N. Yokoya and M. D. Levine, "Range image segmentation based on differential geometry: a hybrid approach," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 11, no. 6, pp. 643–649, 1989.

[46] G. Roth and M. D. Levine, "A genetic algorithm for primative extraction," in *Proc. 4th Int'l Conf. on Genetic Algorithms*, pp. 487–494, 1991.

[47] X. Yu, T. D. Bui, and A. Krzyzak, "Robust estimation for range image segmentation and reconstruction," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 16, no. 5, pp. 530–538, 1994.

[48] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 10, no. 2, pp. 167–192, 1988.

[49] R. Hoffman and A. K. Jain, "Segmentaion and classification of range images," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 5, no. 9, pp. 608–620, 1987.

[50] L. T. Watson, T. J. Laffey, and R. M. Haralick, "Topographic classification of digital image intensity surfaces using generalized splines and the discrete cosine transform," *Computer Vision, Graphics and Image Processing*, vol. 29, no. 2, pp. 143–167, 1985.

[51] J. Berkmann and T. Caelli, "Computation of surface geometry and segmentation using covariance techniques," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 16, no. 11, pp. 1114–1116, 1994.

[52] T.-J. Fan, G. Medioni, and R. Nevatia, "Segmented descriptions of 3-d surfaces," *IEEE Transactions on Robotics and Automation*, vol. 3, no. 6, pp. 527–538, 1987.

[53] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*. Englewood Cliffs, NJ: Prentice-Hall, 1976.

[54] H. S. Yang, "Range image segmentation and classification via split-and-merge on surface curvatures," in *Proc. Fourth Int'l Conf on Pattern Recognition*, (Cambridge, England), pp. 58–67, British Pattern Recognition Soc., 1988.

[55] P. Liang and J. S. Todhunter, "Representation and recognition of surface shapes in range images," *Computer Vision, Graphics and Image Processing*, vol. 52, no. 10, pp. 78–109, 1990.

[56] T. C. Henderson, "Efficient 3-d object representations for industrial vision systems," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 5, no. 6, pp. 609–617, 1983.

[57] B. K. P. Horn, "Extended gaussian image," *Proceedings of IEEE*, vol. 61, no. 3, pp. 1671–1686, 1984.

[58] K. Ikeuchi, "Recognition of 3-d objects using the extended gaussian image," in *Proc. 7th Int. Joint Conf. AI*, (Vancouver, B.C., Canada), Aug 24-28 1981.

[59] M. M. Trivedi and C. Chen, *Advances in Computers*, vol. 32, ch. Sensor-Driven Intelligent Robotics, pp. 105–148. Boston: Academic Press, 1988.

[60] S. Watanabe, ed., *Frontiers of Pattern Recognition*, ch. Some Techniques for Recognizing Structure in Pictures, pp. 1–29. New York: Academic Press, 1972.

[61] J. R. Engelbrect and F. M. Wahl, "Polyhedral object recognition using hough-space features," *Pattern Recognition*, vol. 21, no. 2, pp. 155–167, 1988.

[62] C. R. Bidlack and M. M. Trivedi, "Geometric model based object recognition and localization robotic manipulation tasks," in *Applications of Artificial Intelligence IX Conf.*, (Orlando), pp. 270–280, SPIE, April 1991.

[63] A. V. Aho, J. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison Wesley, 1974.

[64] J. Bron and H. Kerbosch, "Algorithm 457 - finding all cliques in an undirected graph," *Comm. ACM*, vol. 16, no. 575, pp. 218–231, 1973.

[65] R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible objects: the local feature-focus method," *International Journal of Robotics Research*, vol. 1, no. 3, 1982.

[66] P. H. Winston, *Artificial Intelligence.* Reading, MA: Addison-Wesley, 1984.

[67] R. A. Hummel and S. W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Trans. on Pattern Anal. Machine Intell.*, no. 5, pp. 267–287, 1983.

[68] R. M. Haralick and L. G. Shapiro, "The consistent labeling problem: Part 1," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 1, pp. 173–184, 1979.

[69] L. G. Shapiro and R. M. Haralick, "Structural descriptions and inexact matching," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 3, pp. 504–519, 1981.

[70] L. G. Shapiro and R. M. Haralick, "A metric for computing relational description," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 7, pp. 90–94, 1985.

[71] B. Allen, D. Yasuda, S. Tanimoto, L. G. Shapiro, and L. Cinque, "A parallel algorithm for graph matching and its MasPar implementation," in *Proceedings of Workshop Comput. Architect. Mach. Percept.*, pp. 13–18, IEEE, 1993.

[72] N. Ayache and O. D. Faugeras, "(hyper): A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 8, no. 1, pp. 44–54, 1986.

[73] W. E. L. Grimson, "The combinatorics of object recognition in cluttered environments using constrained search," in *Proceedings of 1988 Int. Conf. Computer Vision*, pp. 218–227, ICCV, 1971.

[74] R. Bolles and P. Horaud, "3DPO: A three-dimensional part orientation system," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 3–26, 1986.

[75] C. A. Goad, "Special purpose automatic programming for three-dimensional model-based vision," in *Proceedings of the Image Understanding Workshop*, (Arlington, VA), pp. 94–104, ARPA, 1982.

[76] K. Ikeuchi and T. Kanade, "Automatic generation of recognition programs," *IEEE Special Issue on Computer Vision*, vol. 76, pp. 1016–1035, 1988.

[77] K. S. Hong, K. Ikeuchi, and K. D. Gremban, "Minimum cost aspect classification: A module of a vision algorithm compiler," Tech. Rep. CMU-CS-90-124, CMU School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1990.

[78] Y. Lamdan and H. J. Wolfson, "Geometric hashing: A general and efficient model-based recognition scheme," in *Proceedings Second Int. Conf. on Computer Vision*, pp. 238–249, IEEE, Dec. 1988.

[79] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[80] G. Stockman, "Object recognition and localization via pose clustering," *Computer Vision, Graphics and Image Processing*, pp. 361–387, June 1987.

[81] R. C. Read and D. G. Corneil, "The graph isomorphism disease," *Journal of Graph Theory*, vol. 1, no. 1, pp. 339–363, 1977.

[82] E. M. Palmer, *Graphical Evolution*. New York: John Wiley and Sons, 1985.

[83] L. E. Druffel, *Graph Related Algorithms: Isomorphism, Automorphism and Containment*. PhD thesis, Systems and Information Science, School of Engineering, Vanderbuilt University, 1975.

[84] J. Turner, "Generalized matrix functions and the graph isomorphism problem," *SIAM J. Appl. Math.*, vol. 16, pp. 520–526, 1968.

[85] R. C. Read, *Graph Theory and Computing*,, ch. The coding of various kinds of unlabeled trees. New York: Academic Press, 1972.

[86] Y. J. Shah, G. T. Davida, and M. K. McCarthy, "Optimum features and graph isomorphism," *IEEE Trans. Systems, Man and Cybernetics*, vol. 4, pp. 313–319, 1974.

[87] F. Stockton, "Linearization and standardization of graphs," Tech. Rep. 2-68-34430, Shell Development Co., 1968.

[88] J. E. Hopcroft and R. E. Tarjan, "A v2 algorithm for determining isomorphism of planar graphs," *Information Processing Letter*, no. 1, 1971.

[89] J. E. Hopcroft, "An n log n algorithm for isomorphism of planar triply connected graphs," Tech. Rep. STAN-CS-71-192, Stanford Computer Science, University of Toronto, Canada, 1971.

[90] L. Weinberg, "A simple and efficient algorithm for determining isomorphism of planar triply-connected graphs," *IEEE Trans. Comm. Technol.*, vol. CT-13, pp. 142–148, 1960.

[91] D. G. Corneil, "Graph isomorphism," Tech. Rep. 18, Dept. of Comp. Sci., University of Toronto, Canada, 1970.

[92] A. N. Kelikhov, L. S. Bershteyn, and P. Karelin, "Isomorphism of graphs and finite automata," *Engrg. Cybernetics*, vol. 1, pp. 124–129, 1968.

[93] B. Bollobas, *Random Graphs*. New York: Academic Press, 1985.

[94] L. Babai and L. Kucera, "Canonical labeling of graphs in linear average time," in *20th Annual IEEE Symposium on Foundations of Comp. Sci.*, (Puerto Rico), pp. 39–46, IEEE, 1979.

[95] R. M. Karp, *Relations between combinatorics and other parts of mathematics*, ch. Probabalistic Analysis of a canonical numbering algorithm for graphs, pp. 365–378. Providence, RI: Amer. Math. Soc., 1979.

[96] L. Babai, "On the complexity of canonical labeling of strongly regular graphs," *SIAM J. Comput.*, vol. 9, no. 1, pp. 212–216, 1980.

[97] A. T. Berztiss, "A backtrack procedure for isomorphism of directed graphs," *Journal ACM*, vol. 20, pp. 365–377, 1973.

[98] G. Levi, "A note on the derivation of maximal common subgraphs of two directed or undirected graphs," *Calcolo*, vol. 9, pp. 341–354, 1972.

[99] R. A. Jarvis, "A perspective on range finding techniques for computer vision," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 5, no. 2, pp. 122–139, 1983.

[100] A. Rosenfeld, *Advances in Computers*, vol. 27, ch. Computer Vision. Boston: Academic Press, 1988.

[101] M. M. Trivedi, *Encyclopedia of Science and Technology*, ch. Intelligent Robotic Systems, pp. 226–229. New York: McGraw-Hill, 1994.

[102] C. Thorpe, M. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the carnegie-mellon NAVLAB," *IEEE Trans. on Pattern Anal. Machine Intell.*, pp. 362–373, May 1988.

[103] C. A. Harlow, M. M. Trivedi, R. W. Conners, and D. Phillips, "Scene analysis of high resolution aerial scenes," *Optical Engineering*, pp. 347–355, March 1986.

[104] P. H. Eichel, E. J. Delp, K. Koral, and A. J. Buda, "A method for fully automatic definition of coronary arterial edges from cineangiograms," *IEEE Trans. Medical Imaging*, pp. 313–320, December 1988.

[105] L. S. Hibbard, J. S. McGlone, D. W. Davis, and R. A. Hawkins, "Three dimensional representation and analysis of brain energy metabolism," *Science*, pp. 1641–1646, 1987.

[106] A. M. Darwish and A. K. Jain, "A rule-based approach for visual pattern inspection," *IEEE Trans. on Pattern Anal. Machine Intell.*, pp. 56–68, January 1988.

[107] R. Kasturi, S. T. Bow, W. El-Masri, J. Shah, J. R. Gattiker, and U. B. Mokate, "A system for interpretation of line drawings," *IEEE Trans. on Pattern Anal. Machine Intell.*, pp. 978–992, October 1990.

[108] A. Okakzaki, T. Kondo, K. Mori, S. Tsunekawa, and E. Kawamoto, "An automatic circuit diagram reader with loop-structure-based symbol recognition," *IEEE Trans. on Pattern Anal. Machine Intell.*, pp. 331–341, May 1988.

[109] D. Marr, *Vision*. San Fransico, CA: Freeman, 1982.

[110] M. Okutomi, G. Medioni, and T. Kanade, "A multiple baseline stereo," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 4, pp. 353–363, 1993.

[111] S. B. Marapane and M. M. Trivedi, "Region-based stereo analysis for robotic applications," *IEEE Trans. on System, Man, and Cybernetics*, vol. SMC-19, pp. 1447–1464, November 1989.

[112] S. B. Marapane and M. M. Trivedi, "An active vision system for multi-primitive hierarchical stereo analysis and multi-cue depth extraction," in *Proceedings of the Sensor Fusion for Aerospace Systems Conference.*, (Orlando), SPIE, April 1993.

[113] S. B. Marapane and M. M. Trivedi, "Multi-primitive hierarchical (MPH) stereo analysis," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 16, pp. 227–240, March 1995.

[114] S. B. Marapane and M. M. Trivedi, "Experiments in active vision with real and virtual robot heads," *International Journal of Applied Intelligence, Special issue on Sensor Fusion*, vol. 5, pp. 237–250, July 1995.

[115] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," *Optical Engineering*, pp. 139–144, January/February 1980.

[116] D. Blostein and N. Ahuja, "Shape from texture: Integrating texture element extraction and surface estimation," *IEEE Trans. on Pattern Anal. Machine Intell.*, pp. 1233–1251, December 1989.

[117] A. K. Dalmia and M. M. Trivedi, "High-speed extraction of 3d structure of selectable quality using a translating camera," *Computer Vision and Image Understanding*, vol. 63, no. 3, 1996.

[118] A. K. Dalmia and M. M. Trivedi, "3-d structure extraction from image streams: A review," *IEEE Trans. on Pattern Anal. Machine Intell.*, To appear 1996.

[119] A. K. Dalmia and M. M. Trivedi, "Integrating depth from motion and binocular stereo approaches," *Machine Vision and Applications*, To appear 1996.

[120] G. Stockman and G. Hu, "Sensing 3-d surface patches using a projected grid," in *Proceedings Computer Vision and Pattern Recognition Conf.*, pp. 602–607, 1986.

[121] M. R. Ward, D. P. Rheaume, S. W. Holland, and J. H. Dunseth, "Production plant consight installations," Tech. Rep. GMR-4156, General Motors Research Laboratories, Warren, Michigan, August, 1982.

[122] H. P. Gadagkar, M. M. Trivedi, and T. N. Lassiter, "Versatile multi-modal system for surface profile measurements using a wrist-mounted laser device," in *Proceedings of the Sensor Fusion V Conference*, (Boston), pp. 466–474, SPIE, Nov. 1992.

[123] F. W. DePiero and R. L. Kress, "Design and in situ calibration of a structured light sensor," in *Proceedings of Intl. Conf. on Intelligent Robotics and Systems*, (Pittsburgh, PA), pp. 513–518, IEEE/RSJ, August 5-9 1995.

[124] B. L. Burks, F. W. DePiero, J. C. Rowe, C. B. Selleck, and D. L. Jacoboski, "Final results of the application of a structured light source for surface mapping of the fernald k-65 silos," in *Proceedings of The American Nuclear Society*, (Boston), ANS, June 1992.

[125] Y. Shirai and M. Suwa, "Recognition of polyhedrals with a range finder," in *Proceedings of the 2nd Int. Joint Conf. on Artificial Intell.*, (London), pp. 80–87, Sept. 1971.

[126] S. W. Holland, L. Rossol, and M. R. Ward, *Computer Vision and Sensor-Based Robotics*. New York: Plenum Press, 1978.

[127] G. Agin and T. Binford, "Computer description of curved objects," in *Proceedings of the 3rd Int. Joint Conf. on Artificial Intell.*, pp. 629–640, 1973.

[128] M. Oshima and Y. Shirai, "A scene description method using three dimensional information," *Pattern Recognition*, vol. 11, pp. 9–17, 1984.

[129] K. L. Boyer and A. C. Kak, "Color encoded structured light for rapid active ranging," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 9, no. 1, pp. 14–28, 1987.

[130] F. W. DePiero and R. L. Kress, "Camera calibration in a hazardous environment performed in situ with automated analysis and verification," in *Proceedings of ANS Fifth Topical Meeting on Robotics and Remote Systems*, (Knoxville, TN), ANS, April 1993.

[131] D. A. Simon, M. Hebert, and T. Kanade, "Real-time 3-d pose estimation using a high speed range sensor," Tech. Rep. CMU-RI-TR-93-24, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.

[132] B. L. Burks, F. W. DePiero, J. C. Rowe, C. B. Selleck, D. L. Jacoboski, and R. Markus, "Generation of 3 surface maps in waste storage silos using a structured light source," in *Proceedings of The Space Operations, Applications, and Research Symposium*, (Houston), NASA, 1991.

[133] M. Minou, *Theoretical and experimental studies on basic relations between real world and pictoral patterns and their generating constraints*. PhD dissertation, Kyoto University, Department of Information Science, Nov. 1982.

[134] M. D. Altscheuler, B. R. Altscheuler, and J. Toboada, "Laser electro-optic system for rapid 3-d topographic mapping of surfaces," *Optical Engineering*, vol. 20, no. 6, pp. 953–961, 1981.

[135] S. Inokuchi, K. Sato, and F. Matsuda, "Range imaging system for 3-d object recognition," in *Proceedings of the 7th International Conf. on Pattern Recog.*, pp. 806–808, 1984.

[136] K. K. Yeung and P. D. Lawrence, "A low-cost 3d vision system using space-encoded spot projections," in *Proceedings of The SPIE Conf. on Optics, Illumination and Image Sensing for Machine Vision*, pp. 160–172, SPIE, 1986.

[137] R. J. Popplestone, C. M. Brown, A. P. Ambler, and G. F. Crawford, "Forming models of plane-and-cylinder faceted bodies from light stripes," in *Proc. 4th Int. Joint Conf. on Artificial Intell.*, pp. 664–668, 1975.

[138] G. Bickel, G. Hausler, and M. Maul, "Triangulation with expanded range of depth," *Optical Engineering*, vol. 24, no. 6, pp. 975–979, 1985.

[139] M. Rioux, "Laser range finder based on synchronized scanners," *Applied Optics*, vol. 23, no. 21, pp. 3837–3855, 1984.

[140] F. W. DePiero and M. M. Trivedi, "3-d computer vision using structured light: Design, calibration and implementation issues," *Advances in Computers*, vol. 43, To Appear in 1996.

[141] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotic Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill, 1987.

[142] B. K. P. Horn, *Robot Vision*. Cambridge, MA: McGraw-Hill, 1984.

[143] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans ASME J. Basic Eng.*, vol. 83, pp. 35–45, 1960.

[144] J. A. Borrie, *Stochastic Systems for Engineers: Modeling, Estimation and Control*. New York: Prentice Hall, 1992.

[145] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic Press, 1989.

[146] G. R. Cooper and C. D. McGillem, *Probabalistic Methods of Signal and System Analysis*. Orlando, FL: Harcourt Brace Jovanovich, 1986.

[147] A. Isaguirre, P. Pu, and J. Summers, "A new development in camera calibration: Calibrating a pair of mobile cameras," tech. rep., Department of Computer and Information Science, GRASP Laboratory, University of Pennsylvania, Philadelphia, 1985.

[148] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore: John Hopkins University Press, 2nd ed., 1989.

[149] G. Strang, *Linear Algebra and its Applications*. New York: Academic Press, 2nd ed., 1980.

[150] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C.* Cambridge, New York: Cambridge University Press, 1988.

[151] G.-Q. Wei and S. D. Ma, "Two plane camera calibration: A unified model," tech. rep., National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Bejing, 1991.

[152] S. M. Thayer and M. M. Trivedi, "Residual uncertainty in three-dimensional reconstruction using two-planes calibration and stereo methods," *Pattern Recognition*, vol. 28, no. 7, pp. 1073–1082, 1995.

[153] I. M. Sobol, *A Primer for the Monte Carlo Method.* Boca Raton: CRC Press, 1994.

[154] R. M. Haralick and L. G. Shapiro, "Glossary of computer vision terms," *Pattern Recognition*, vol. 24, no. 1, pp. 69–93, 1991.

[155] P. J. Huber, *Robust Statistics.* New York: Wiley, 1981.

[156] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* New York: John Wiley and Sons, 1973.

[157] G. Strang, *Introduction to Applied Mathematics.* Wellesley, MA: Wellesley-Cambridge Press, 1986.

[158] P. W. Holland and R. E. Welsh, "Robust regression using iteratively reweighted least squares," *Comm. Stat.*, vol. A, no. 6, pp. 813–828, 1977.

[159] R. C. Gonzalez and P. Wintz, *Digital Image Processing.* Reading, MA: Addison-Wesley, 2nd ed., 1987.

[160] K. Sengupta and K. L. Boyer, "Information theoretic clustering of large model bases," in *Proceedings of Conf. Comput. Vis. Pattern Recognition*, pp. 174–179, IEEE, 1993.

[161] D. H. Ballard, "Animate vision," *Artificial Intelligence*, vol. 48, pp. 57–86, 1991.

[162] H. P. Gadagkar and M. M. Trivedi, "An integrated system for active exploration using contact and non-contact sensors," in *Proceedings of The IEEE IROS '92 Conference*, (Raleigh, NC), IEEE, July 1992.

[163] K. Ikeuchi and M. Hebert, "Task Oriented Vision," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Raleigh, NC), pp. 2187–2194, 1992.

[164] R. J. Wilson and L. W. Beineke, eds., *Applications of Graph Theory,*, ch. Chemical Applications of Graph Theory, pp. 328–363. No. 1, London: Academic Press, 1979.

[165] B. Baumgart, "A polyhedral representation for computer vision," in *National Computer Conference*, pp. 589–596, AFIPS, 1975.

[166] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1991.

[167] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*. New York: McGraw-Hill, 1968.

[168] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[169] J. O. Berger, *Statistical Decision Theory and Bayesian Analyses*. New York: Springer-Verlag, 1985.

[170] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press, 1976.

[171] J. E. D. Jr. and R. B. Scnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, New Jersey: Prentice-Hall, 1983.

[172] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. Reading, MA: Addison-Wesley, 1974.

[173] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming Theory and Algorithms*. New York: John Wiley and Sons, 1979.

[174] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. MIT Press, 1990.

APPENDICES

APPENDIX A

ADDITIONAL TEST RESULTS FOR RPT WITH SYNTHETIC SCENES

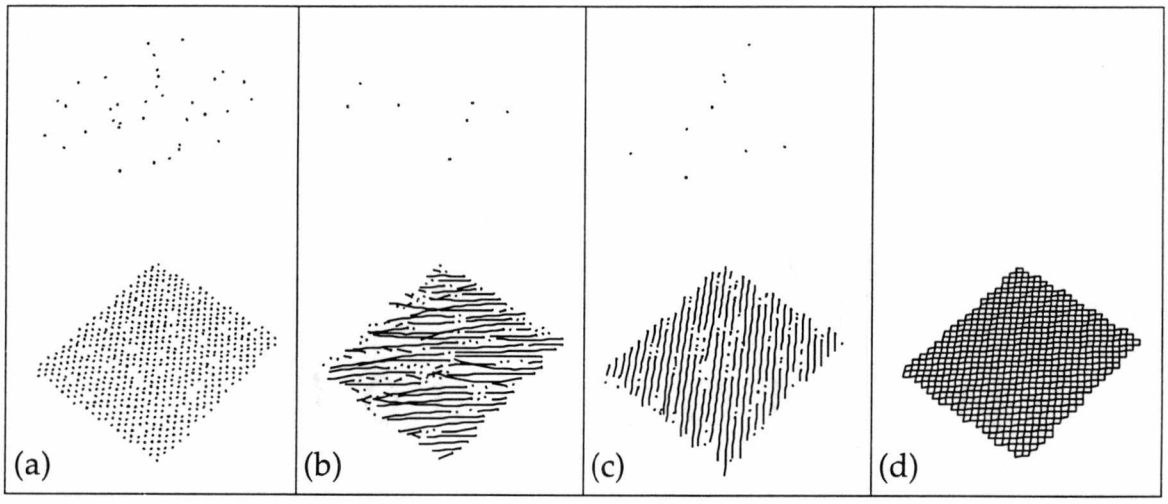


Figure A.1: Typical measurements and processing results generated for a planar surface bounded by jump discontinuities. The measurements, X-Strips, Y-Strips and final fitted surface are shown in (a)-(d), respectively. The original measurements contained zero-mean Gaussian noise with σ_m = 0.010 in. and spike noise occurring with a probability of 5 percent. This synthetic scene had a SNR of 7db.

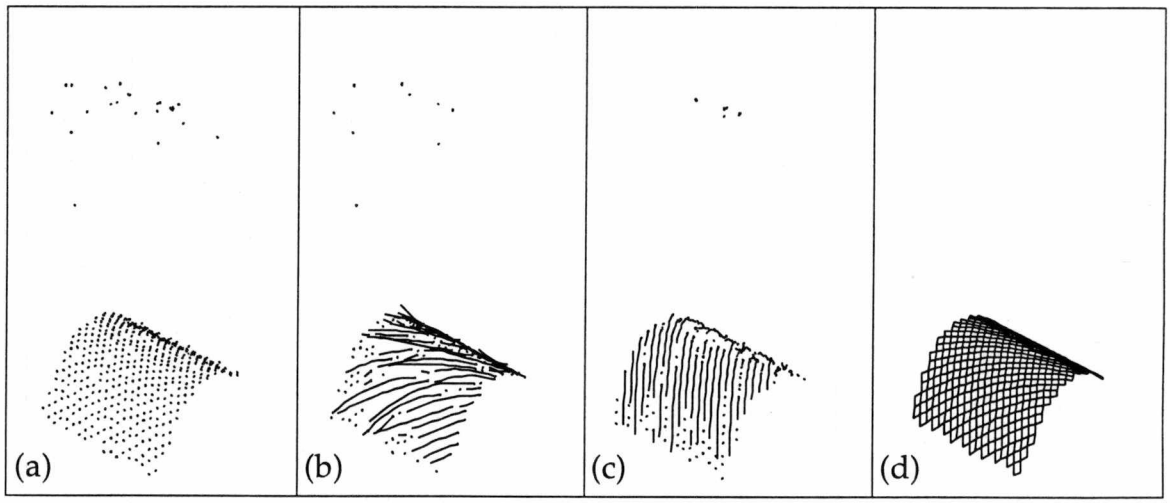


Figure A.2: Typical measurements and processing results generated for a second order surface bounded by jump discontinuities. The measurements, intermediate results and final fitted surface are shown in (a)-(d), respectively. This synthetic scene had a SNR of 5db.

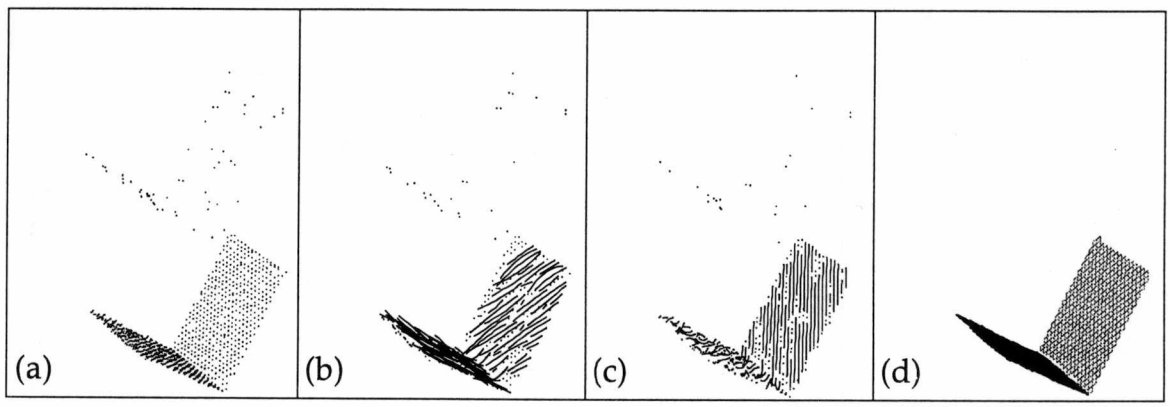


Figure A.3: Typical measurements and processing results generated for a two first order surfaces joined with a crease discontinuity. The measurements, intermediate results and fitted surfaces are shown in (a)-(d), respectively. This synthetic scene had a SNR of 13db.

Table A.1: Results from Monte Carlo analysis of RPT when segmenting a scene containing a single object having a first order surface bounded by jump discontinuities. 100 Trials were run at each noise level (1000 trials in all). The SNR for this set of tests varied from 7.1 to 7.7 db.

Std. Dev. of Gaussian Noise (inches)	Percent Trials OK	Centroid Error (Percent)	Volumetric Error (Percent)	Duration On-Line (Sec)	Duration Off-Line (Sec)
0.010	96	0.4 ± 1.0	0.1 ± 0.2	0.007 ± 0.0002	1.4 ± 0.3
0.020	97	0.3 ± 0.7	0.1 ± 0.1	0.008 ± 0.0004	1.5 ± 0.3
0.030	99	0.3 ± 0.7	0.1 ± 0.1	0.007 ± 0.0001	1.4 ± 0.3
0.040	100	0.3 ± 0.9	0.1 ± 0.1	0.007 ± 0.0001	1.5 ± 0.4
0.050	100	0.4 ± 0.8	0.1 ± 0.1	0.007 ± 0.0001	1.5 ± 0.4
0.060	98	0.6 ± 1.4	0.2 ± 0.1	0.007 ± 0.0001	1.6 ± 0.5
0.070	97	0.3 ± 0.4	0.2 ± 0.1	0.008 ± 0.0003	1.7 ± 0.4
0.080	97	0.3 ± 0.8	0.2 ± 0.1	0.009 ± 0.0006	1.9 ± 0.4
0.090	100	0.5 ± 1.1	0.2 ± 0.2	0.009 ± 0.0005	2.0 ± 0.4
0.100	96	0.4 ± 0.5	0.2 ± 0.2	0.009 ± 0.0004	1.9 ± 0.4

Table A.2: Results from Monte Carlo analysis of RPT when segmenting a scene containing a single object having a second order surface bounded by jump discontinuities. 100 Trials were run at each noise level (1000 trials in all). The SNR for this set of tests varied from 5.2 to 5.5 db.

Std. Dev. of Gaussian Noise (inches)	Percent Trials OK	Centroid Error (Percent)	Volumetric Error (Percent)	Duration On-Line (Sec)	Duration Off-Line (Sec)
0.010	90	0.1 ± 0.1	0.1 ± 0.1	0.009 ± 0.0001	1.4 ± 0.3
0.020	89	0.2 ± 0.1	0.1 ± 0.1	0.009 ± 0.0014	1.5 ± 0.4
0.030	95	0.2 ± 0.1	0.1 ± 0.1	0.009 ± 0.0002	1.4 ± 0.3
0.040	95	0.2 ± 0.1	0.1 ± 0.1	0.009 ± 0.0002	1.5 ± 0.3
0.050	99	0.2 ± 0.2	0.1 ± 0.1	0.010 ± 0.0005	1.6 ± 0.4
0.060	99	0.2 ± 0.2	0.2 ± 0.1	0.009 ± 0.0002	1.6 ± 0.4
0.070	98	0.2 ± 0.2	0.2 ± 0.1	0.010 ± 0.0004	1.7 ± 0.4
0.080	96	0.3 ± 0.2	0.2 ± 0.2	0.011 ± 0.0008	1.9 ± 0.4
0.090	99	0.3 ± 0.2	0.3 ± 0.2	0.011 ± 0.0001	1.9 ± 0.4
0.100	100	0.3 ± 0.2	0.3 ± 0.2	0.011 ± 0.0001	1.9 ± 0.4

Table A.3: Results from Monte Carlo analysis of RPT when segmenting a scene containing a crease discontinuity separating two planar surfaces. 100 Trials were run at each noise level (1000 trials in all). The SNR for this set of tests varied from 13.2 to 13.3 db.

Std. Dev. of Gaussian Noise (inches)	Percent Trials OK	Centroid Error (Percent)	Volumetric Error (Percent)	Duration On-Line (Sec)	Duration Off-Line (Sec)
0.010	99	0.5 ± 0.3	0.7 ± 0.5	0.009 ± 0.0003	2.6 ± 0.6
0.020	97	0.8 ± 2.3	1.1 ± 4.0	0.009 ± 0.0001	2.6 ± 0.6
0.030	89	0.5 ± 0.2	0.7 ± 0.3	0.009 ± 0.0001	2.6 ± 0.6
0.040	96	0.7 ± 2.3	1.6 ± 8.3	0.009 ± 0.0005	2.8 ± 0.8
0.050	84	0.9 ± 2.3	1.4 ± 5.1	0.009 ± 0.0001	2.8 ± 0.5
0.060	75	1.2 ± 3.0	2.8 ± 11.3	0.009 ± 0.0002	2.8 ± 0.7
0.070	64	1.2 ± 3.0	1.8 ± 6.2	0.009 ± 0.0004	3.0 ± 0.7
0.080	68	2.4 ± 5.2	4.8 ± 12.9	0.009 ± 0.0005	3.0 ± 0.7
0.090	62	2.1 ± 4.4	2.8 ± 6.0	0.009 ± 0.0004	3.0 ± 0.7
0.100	46	2.3 ± 4.8	5.6 ± 17.2	0.009 ± 0.0005	3.1 ± 0.9

Table A.4: Results from Monte Carlo analysis of RPT when segmenting a scene containing curvature discontinuities. 100 Trials were run at each noise level (1000 trials in all). The SNR for this set of tests varied from 15.1 to 15.7 db.

Std. Dev. of Gaussian Noise (inches)	Percent Trials OK	Centroid Error (Percent)	Volumetric Error (Percent)	Duration On-Line (Sec)	Duration Off-Line (Sec)
0.010	98	1.0 ± 0.7	3.0 ± 2.7	0.007 ± 0.0002	2.0 ± 0.2
0.020	93	1.2 ± 0.8	4.2 ± 5.8	0.007 ± 0.0003	2.1 ± 0.2
0.030	90	1.5 ± 0.8	9.5 ± 10.8	0.007 ± 0.0004	2.1 ± 0.2
0.040	85	2.0 ± 1.1	12.3 ± 13.1	0.008 ± 0.0006	2.3 ± 0.2
0.050	82	3.5 ± 2.9	15.3 ± 14.6	0.008 ± 0.0003	2.2 ± 0.2
0.060	76	5.2 ± 5.5	18.0 ± 16.8	0.008 ± 0.0004	2.3 ± 0.2
0.070	74	4.7 ± 2.2	14.8 ± 11.5	0.008 ± 0.0001	2.2 ± 0.2
0.080	69	7.6 ± 6.1	21.3 ± 20.2	0.009 ± 0.0006	2.4 ± 0.3
0.090	63	8.7 ± 6.2	24.5 ± 20.8	0.009 ± 0.0005	2.5 ± 0.3
0.100	58	11.0 ± 6.9	23.9 ± 22.4	0.009 ± 0.0001	2.6 ± 0.4

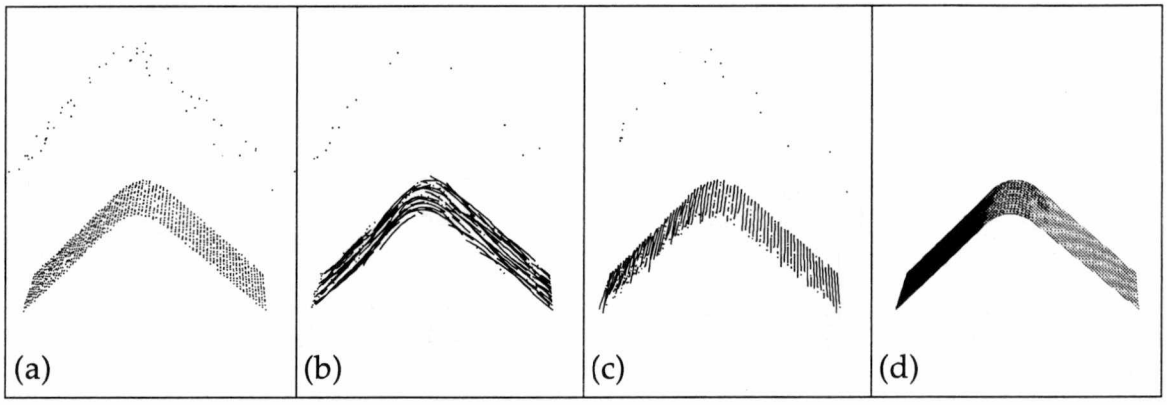


Figure A.4: Typical measurements and processing results generated for an object containing curvature discontinuities. The measurements, intermediate results and fitted surfaces are shown in (a)-(d), respectively. This synthetic scene had a SNR of 15db.

APPENDIX B

ADDITIONAL FIGURES OF OBJECTS USED IN INTEGRATED TESTING

Note that in the sets of figures depicting the segmented regions of objects and their graphs, the nodes are labeled p_i and c_i, indicating planes and cylinders, respectively. There is a correspondence between the physical locations of the nodes and the centroid of each segmented region. For example, see Figures B.3 and B.4.

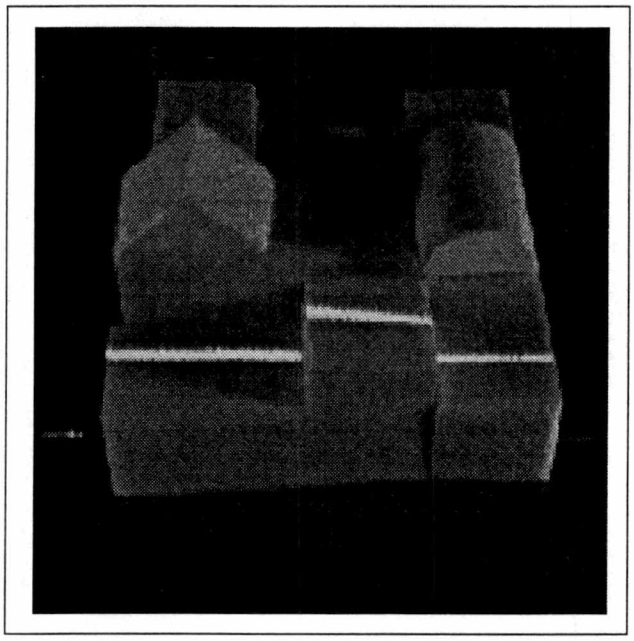


Figure B.1: Object #2.

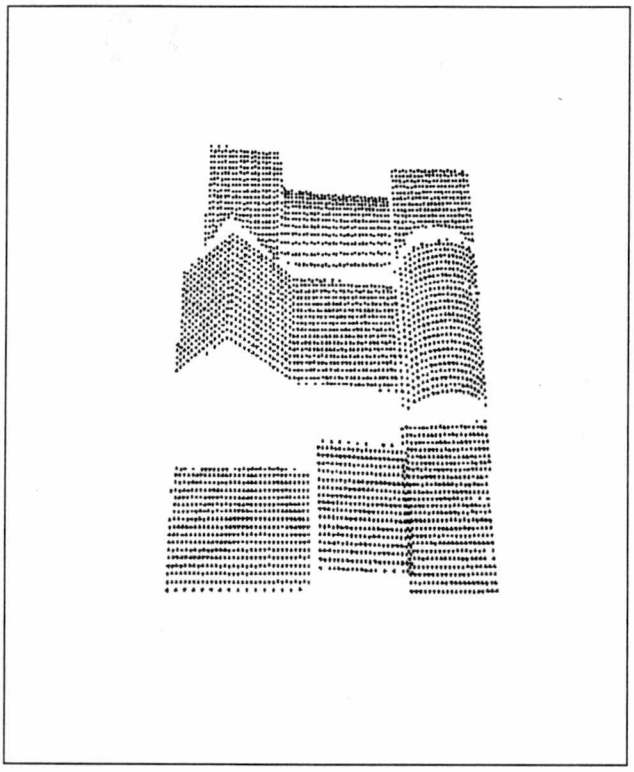


Figure B.2: Measurements captured by PRIME of Object #2.

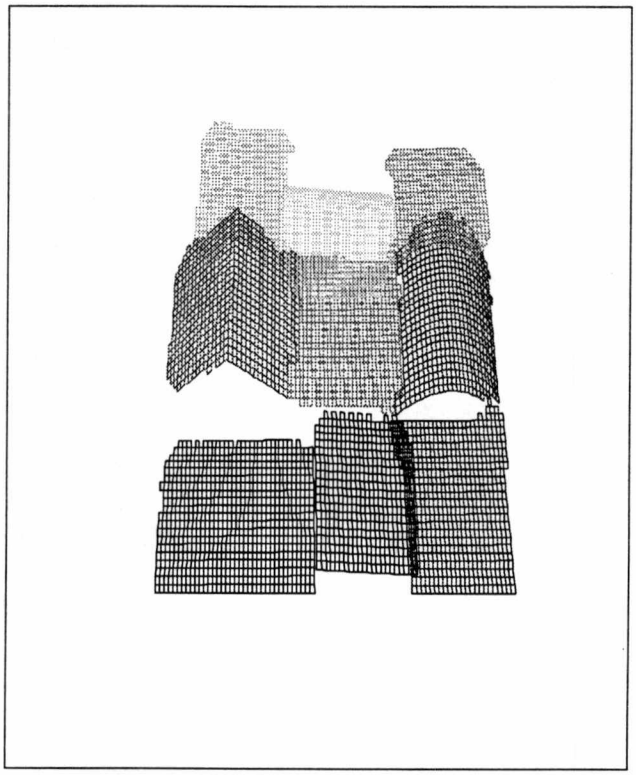


Figure B.3: Surfaces fitted by RPT for Object #2.

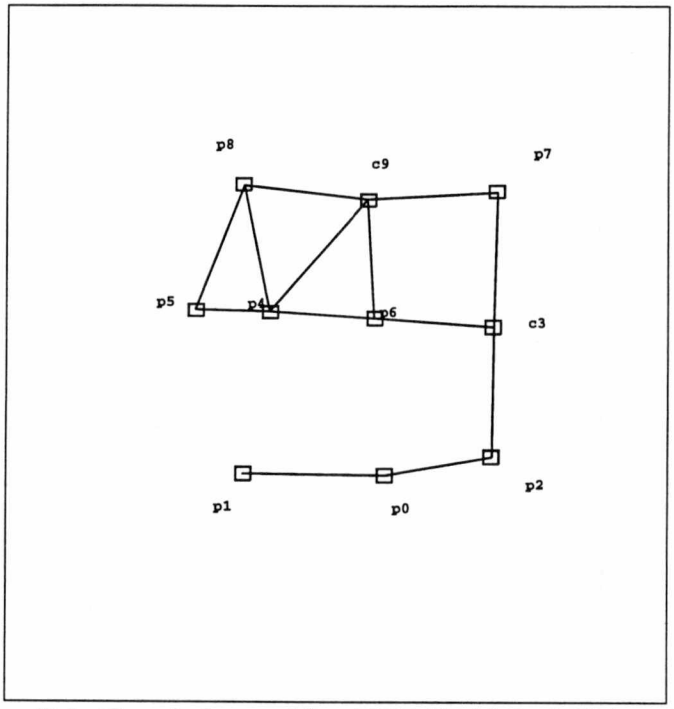


Figure B.4: Graph formed by GEOGRAM of Object #2.

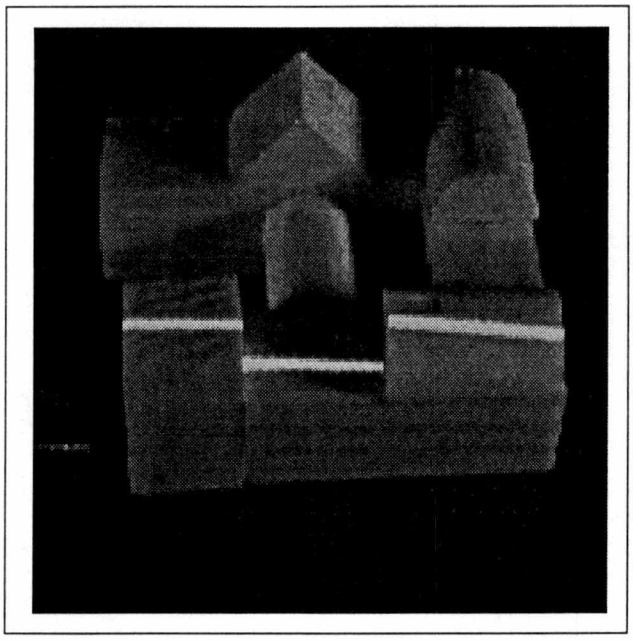


Figure B.5: Object #3.

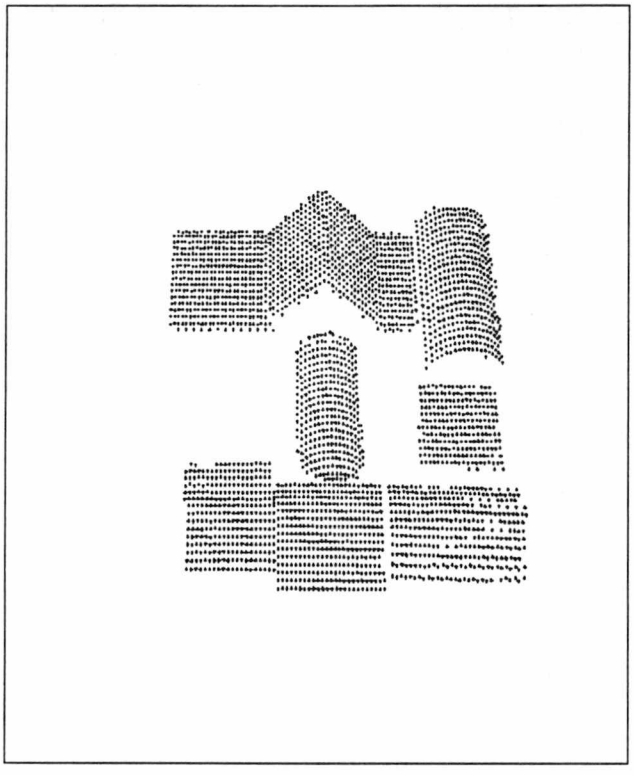


Figure B.6: Measurements captured by PRIME of Object #3.

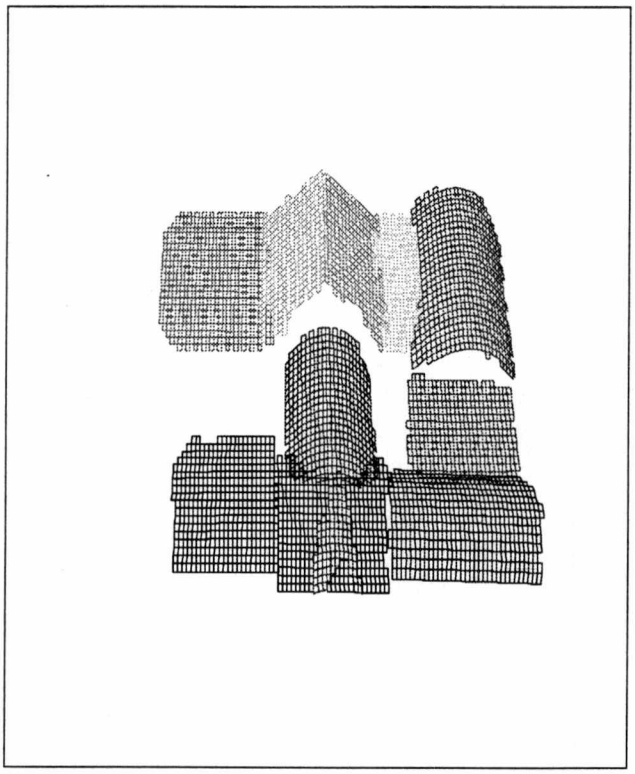


Figure B.7: Surfaces fitted by RPT for Object #3.

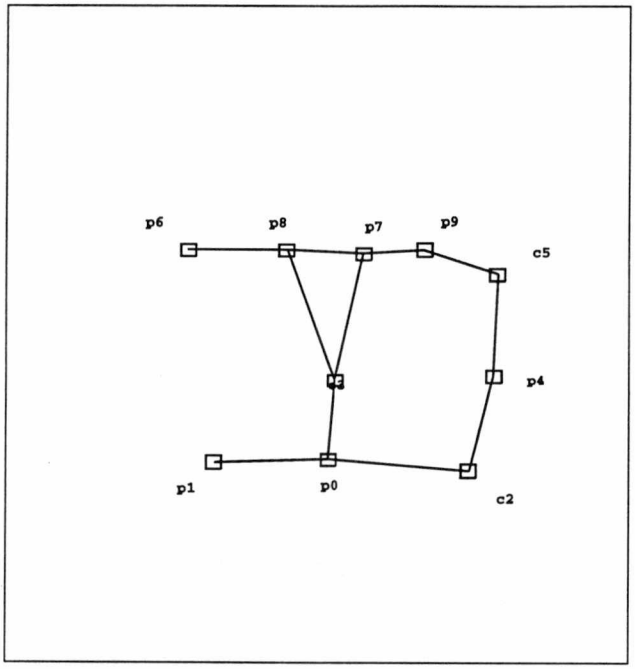


Figure B.8: Graph formed by GEOGRAM of Object #3.

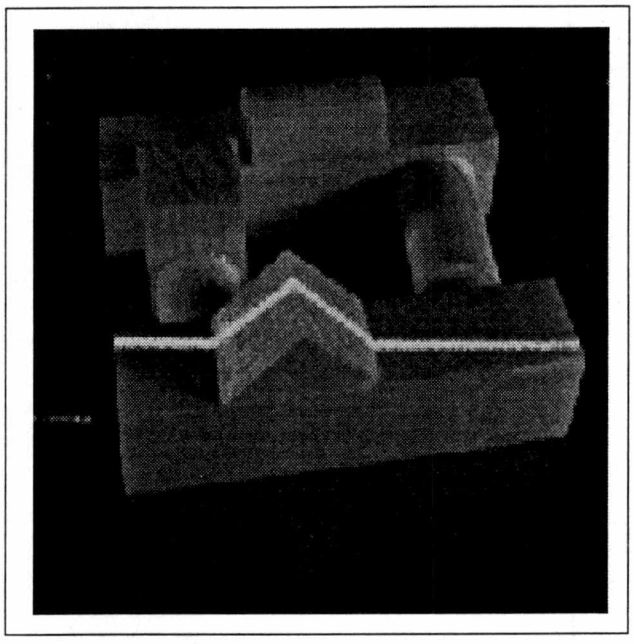


Figure B.9: Object #4.

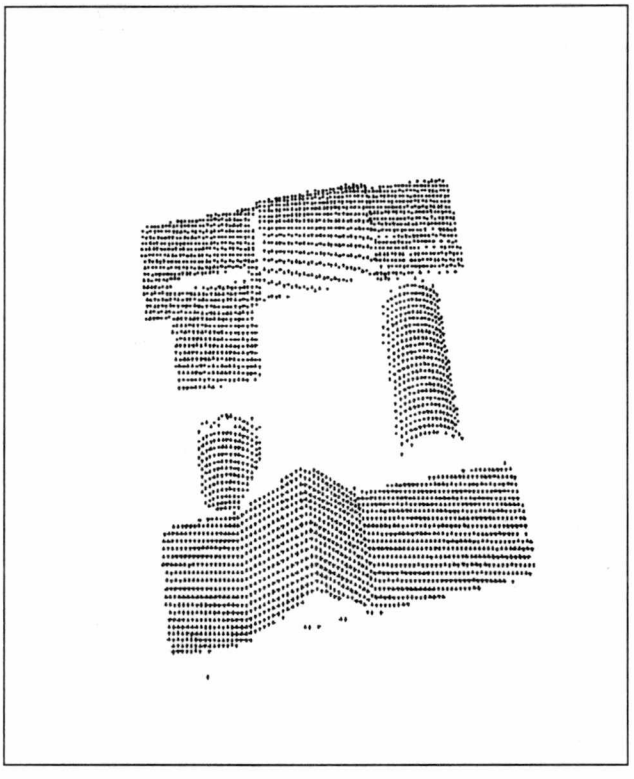


Figure B.10: Measurements captured by PRIME of Object #4.

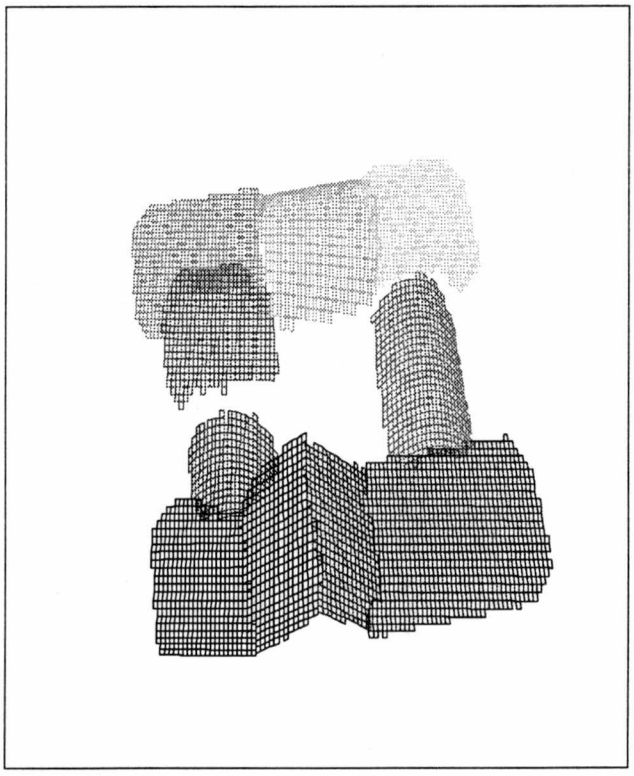


Figure B.11: Surfaces fitted by RPT for Object #4.

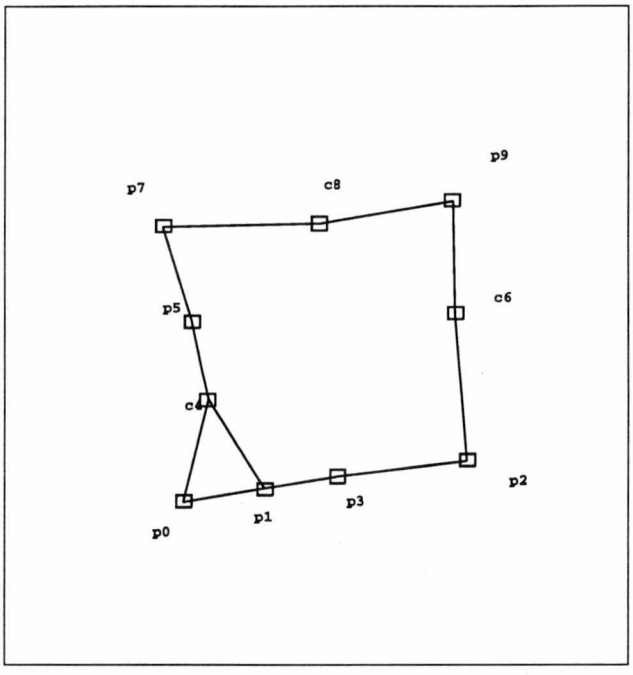


Figure B.12: Graph formed by GEOGRAM of Object #4.

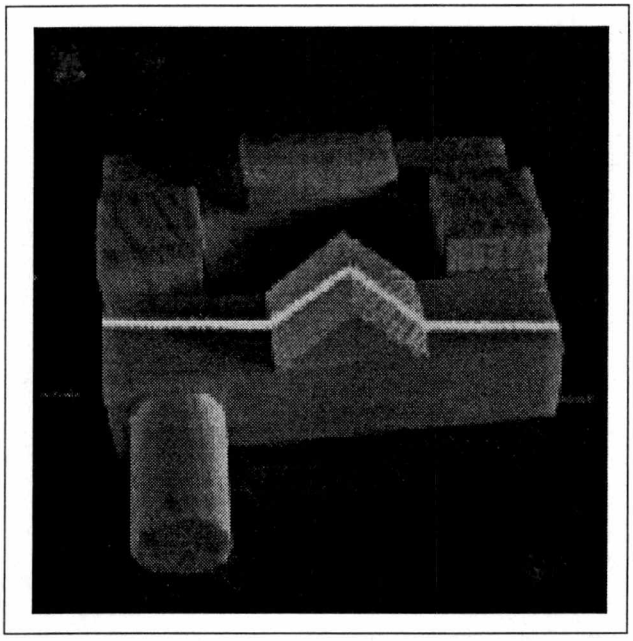


Figure B.13: Object #5.

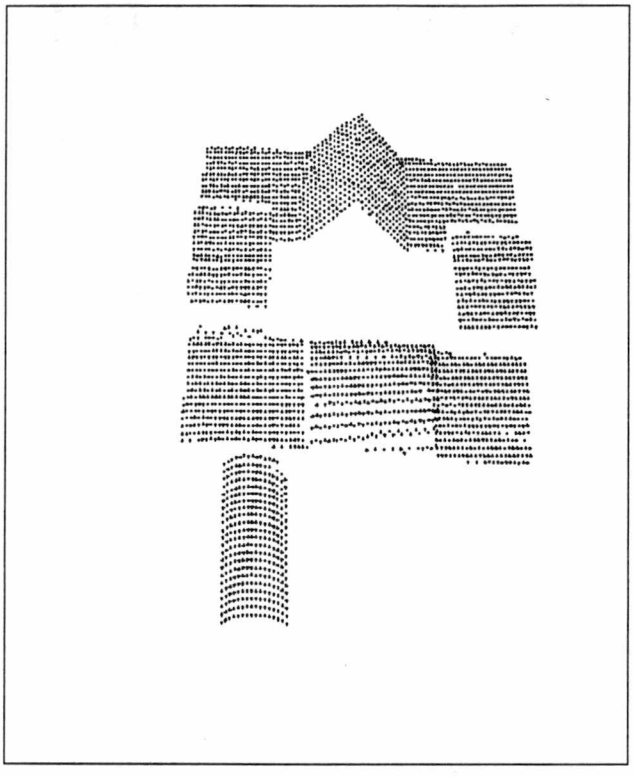


Figure B.14: Measurements captured by PRIME of Object #5.

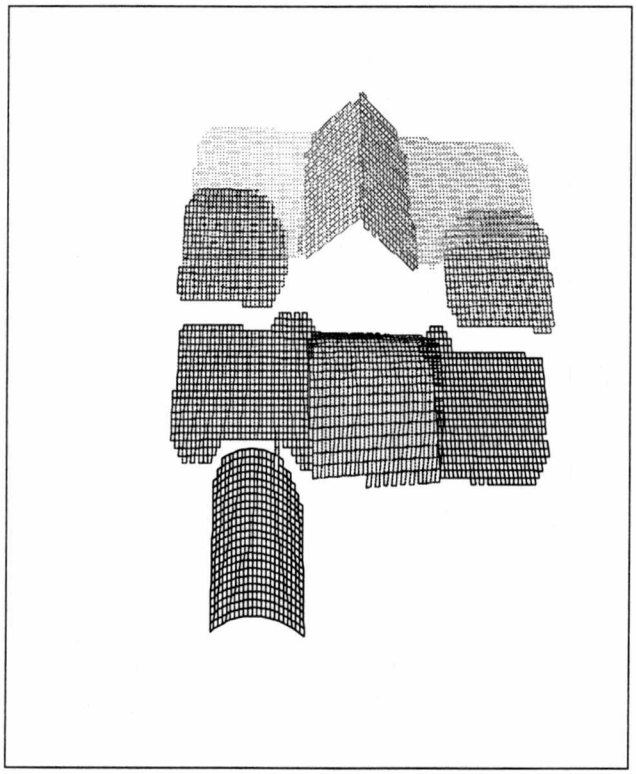


Figure B.15: Surfaces fitted by RPT for Object #5.

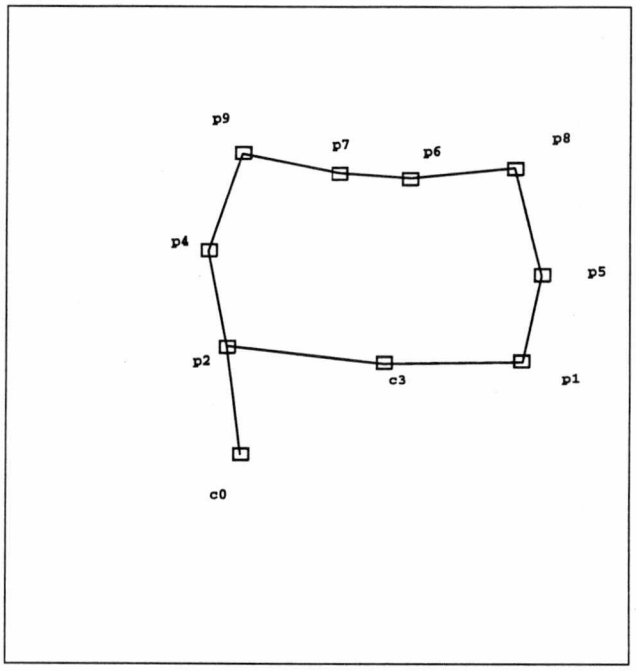


Figure B.16: Graph formed by GEOGRAM of Object #5.

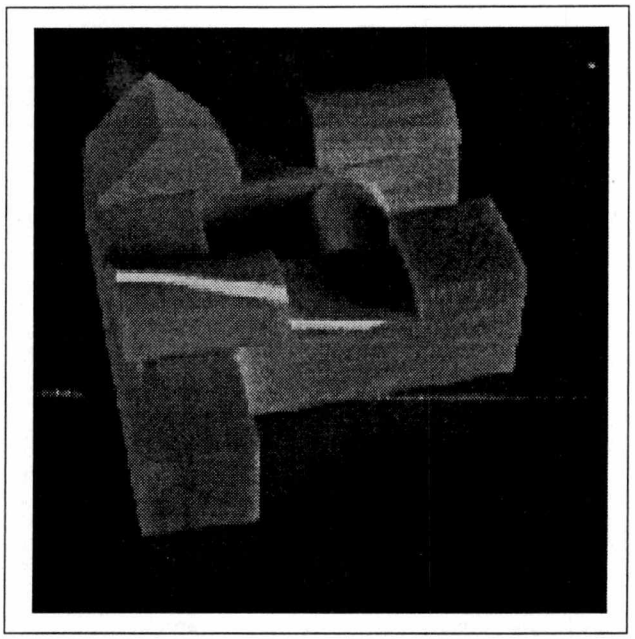


Figure B.17: Object #6.

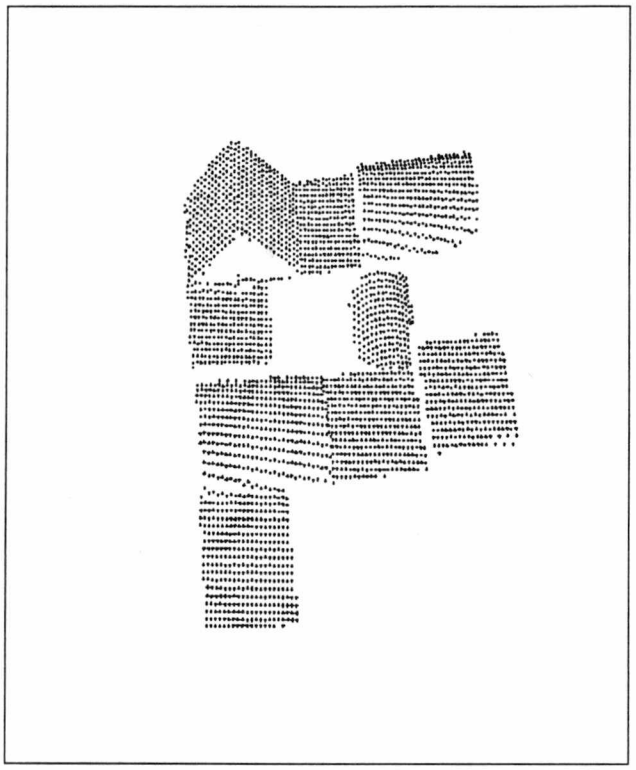


Figure B.18: Measurements captured by PRIME of Object #6.

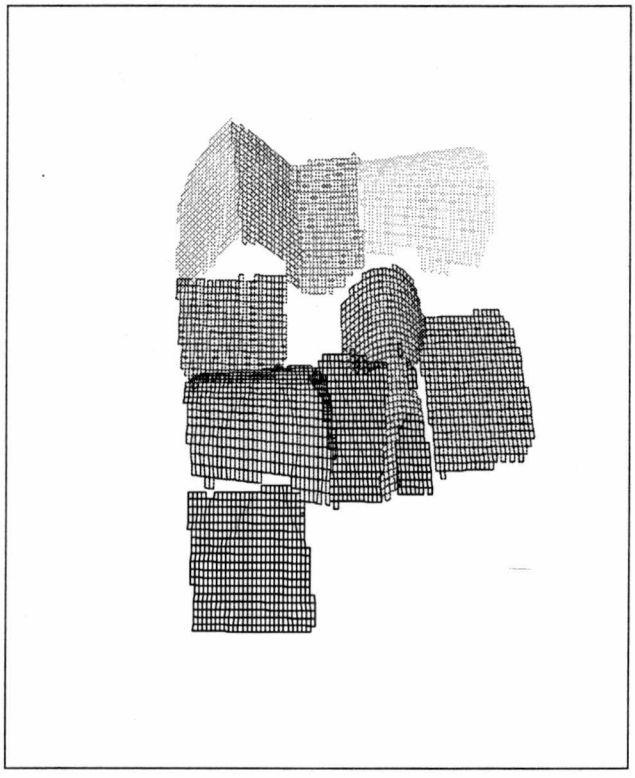


Figure B.19: Surfaces fitted by RPT for Object #6.

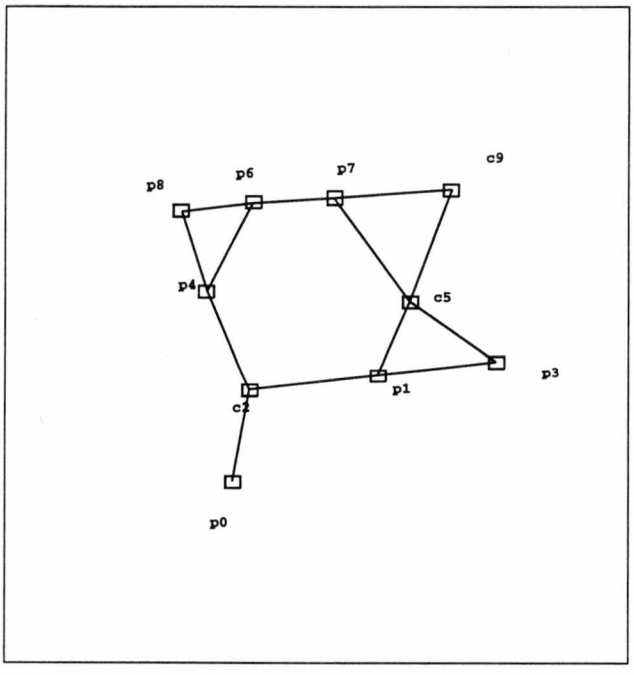


Figure B.20: Graph formed by GEOGRAM of Object #6.

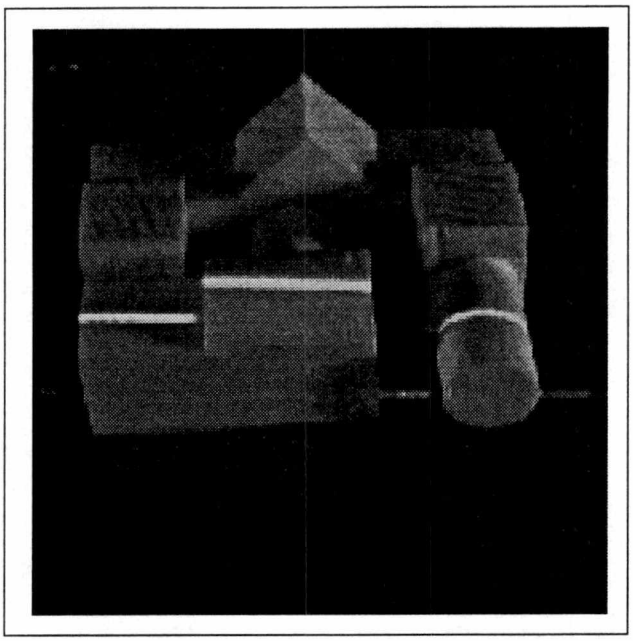


Figure B.21: Object #7.

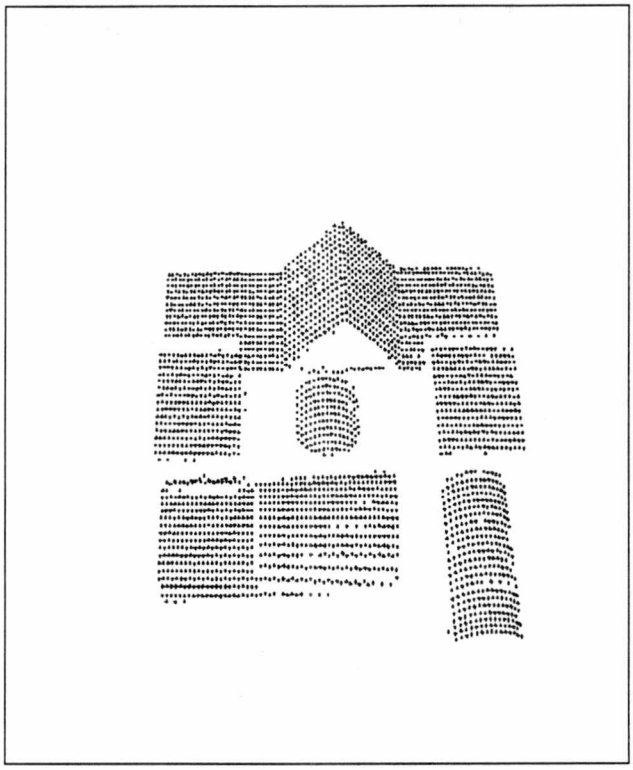


Figure B.22: Measurements captured by PRIME of Object #7.

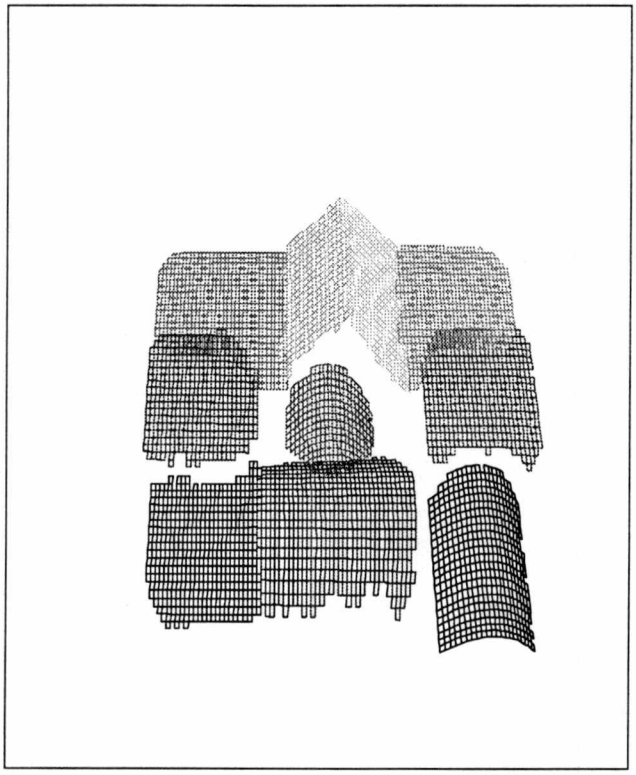


Figure B.23: Surfaces fitted by RPT for Object #7.

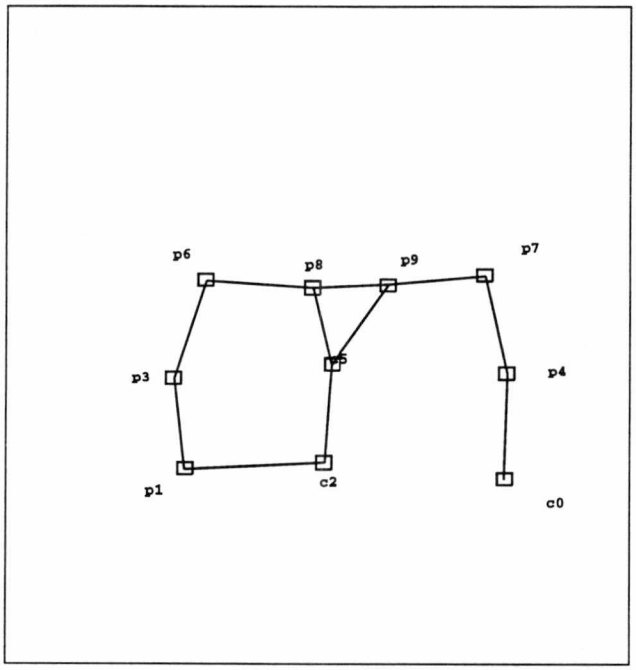


Figure B.24: Graph formed by GEOGRAM of Object #7.

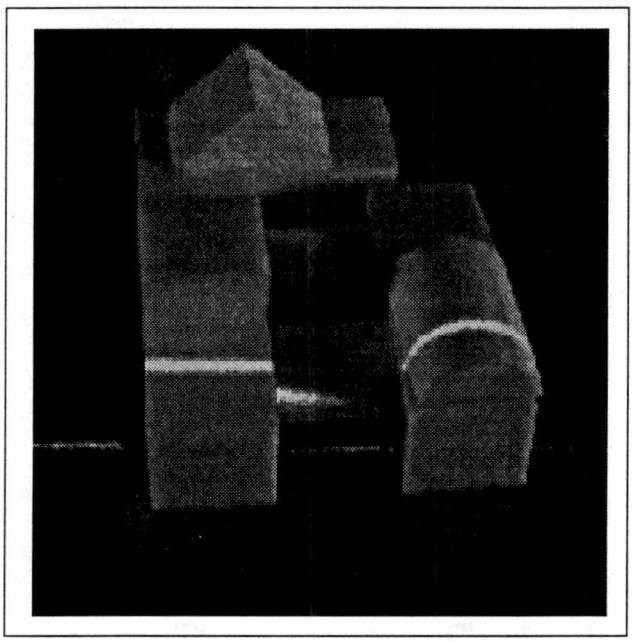


Figure B.25: Object #8.

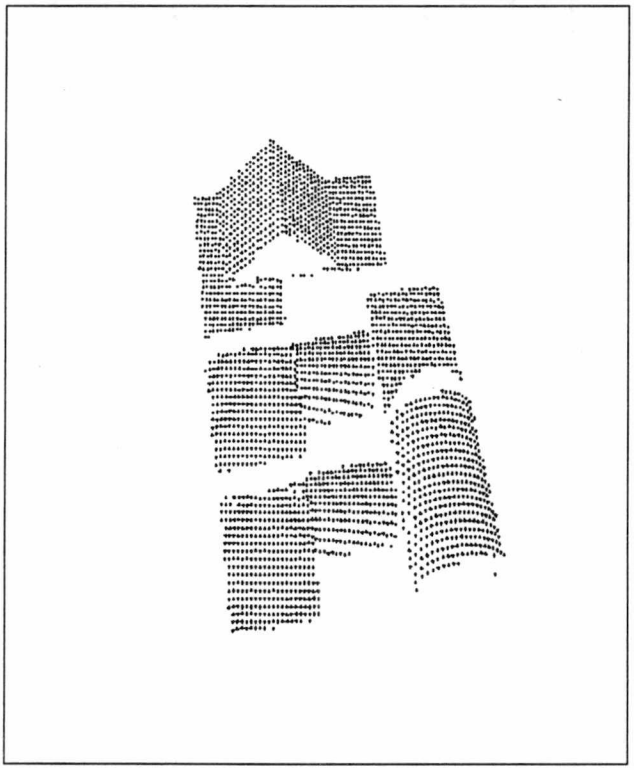


Figure B.26: Measurements captured by PRIME of Object #8.

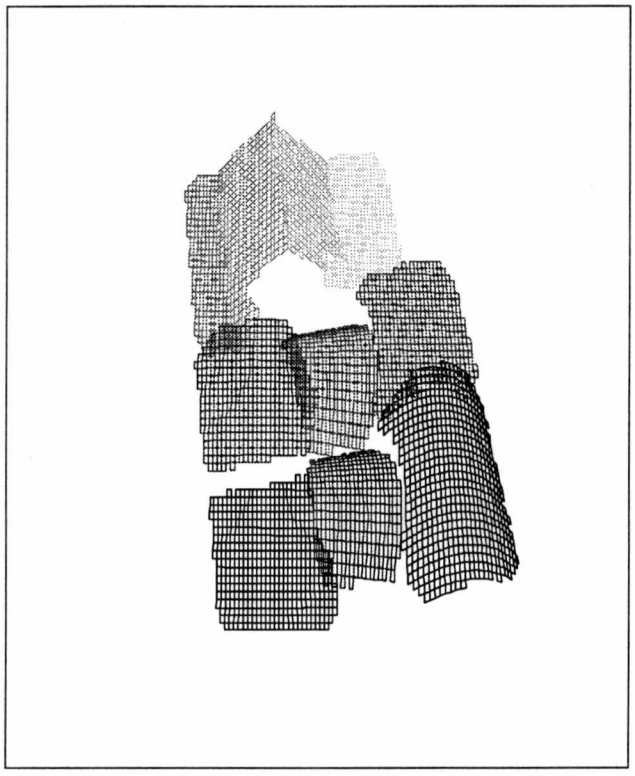


Figure B.27: Surfaces fitted by RPT for Object #8.

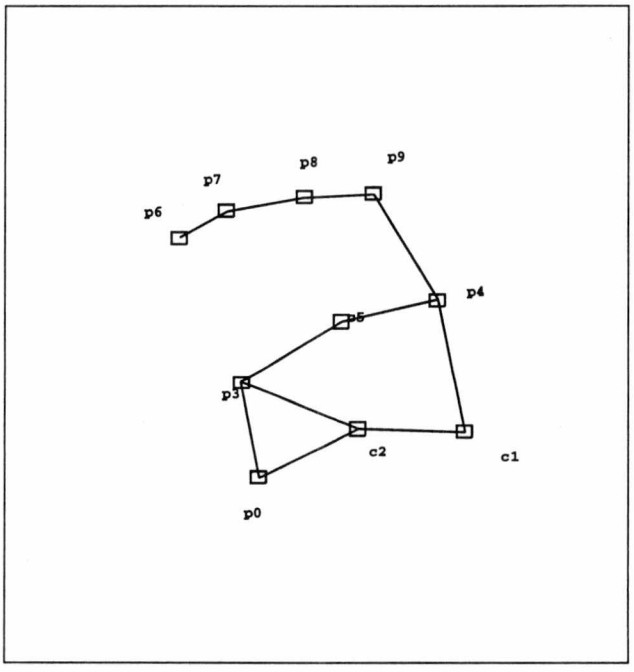


Figure B.28: Graph formed by GEOGRAM of Object #8.

APPENDIX C

ADDITIONAL FIGURES OF OBJECTS USED FOR TESTING RPT

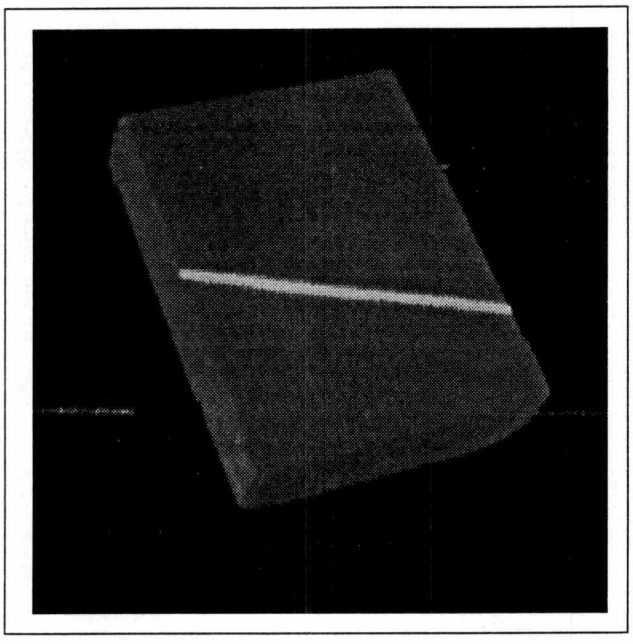


Figure C.1: Typical scene used to test RPT in the presence of jump discontinuities bordering a planar surface.

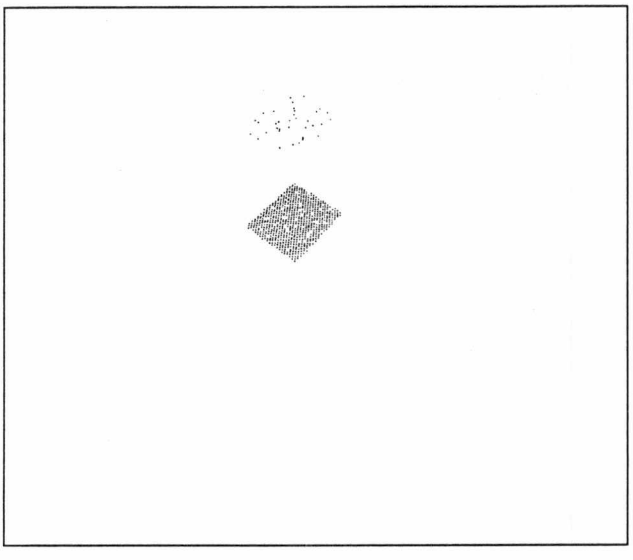


Figure C.2: Measurements collected by PRIME of a planar surface.

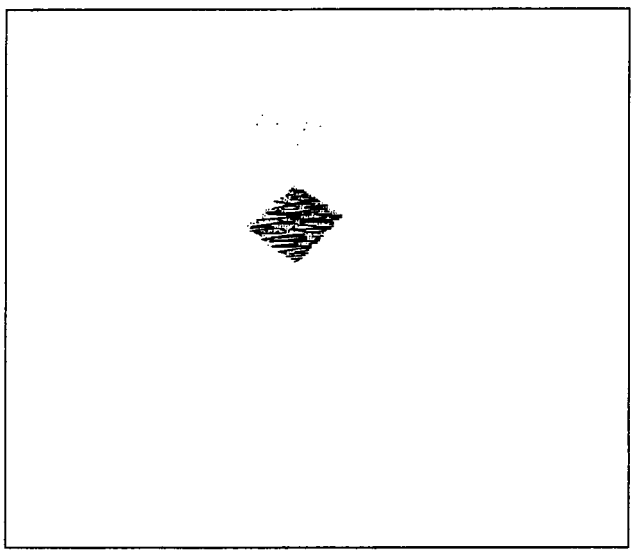


Figure C.3: X-Strips formed by RPT across a planar surface.

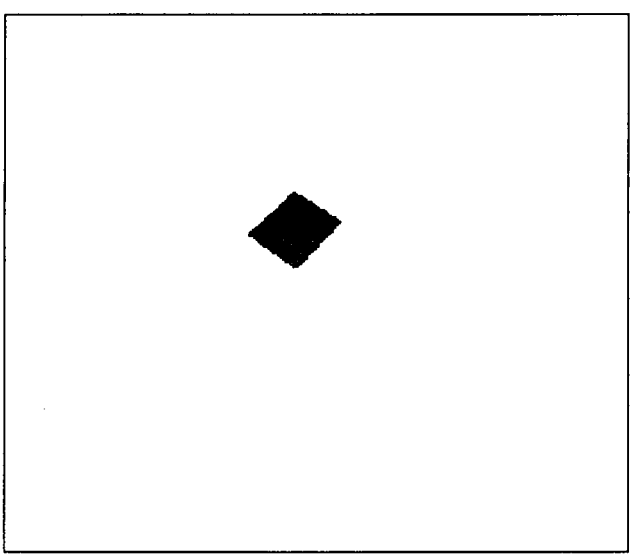


Figure C.4: Surface fitted by RPT to a planar surface.

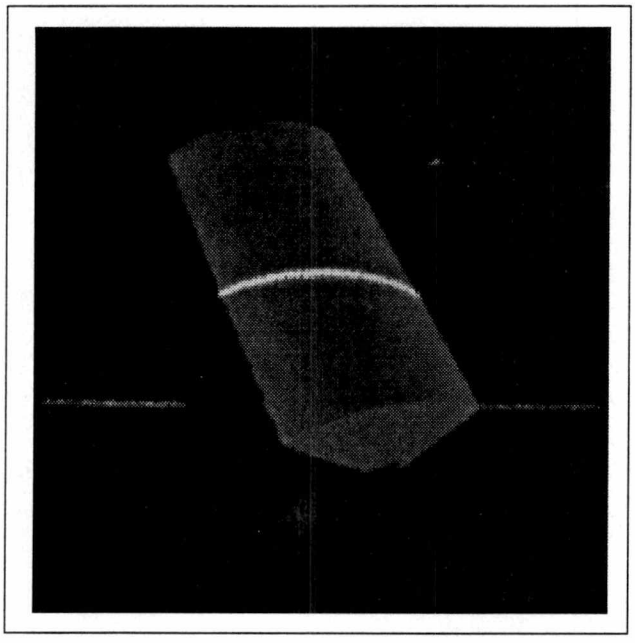


Figure C.5: Typical scene used to test RPT in the presence of jump discontinuities bordering a 2nd order surface.

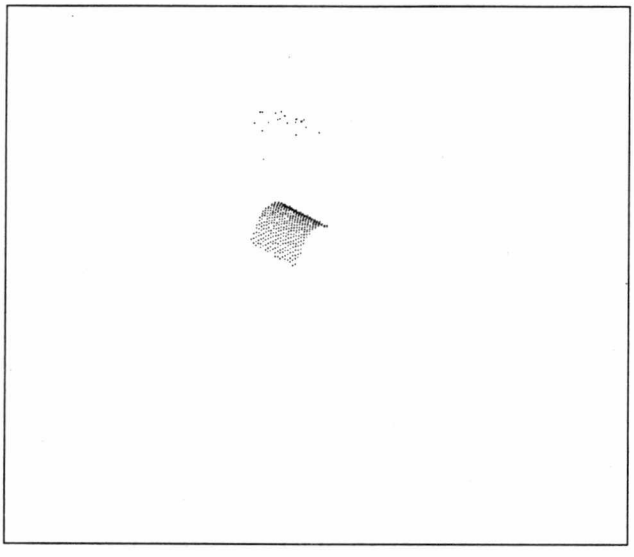


Figure C.6: Measurements collected by PRIME of a 2nd order surface.

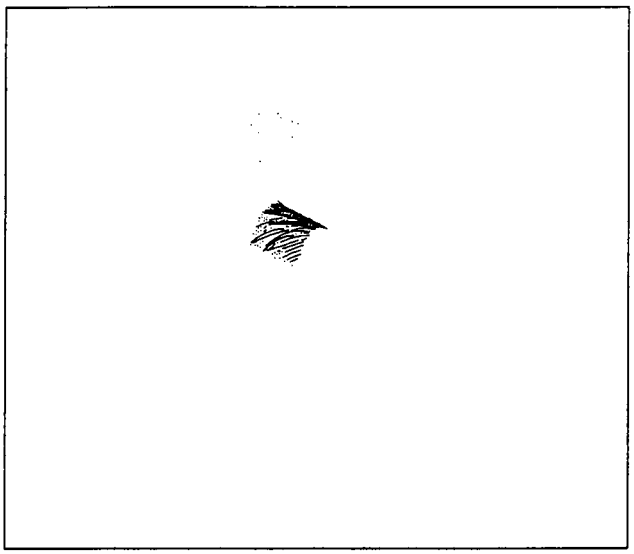


Figure C.7: X-Strips formed by RPT across a 2nd order surface.

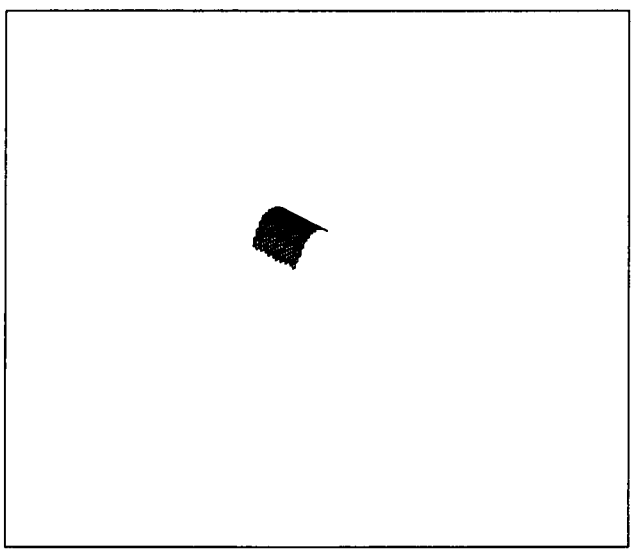


Figure C.8: Surface fitted by RPT to a 2nd order surface.

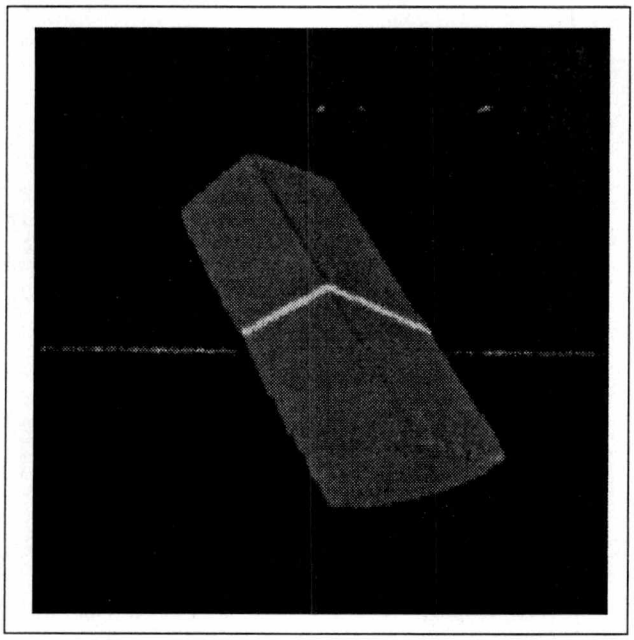


Figure C.9: Typical scene used to test RPT in the presence of a crease discontinuity separating two planar surfaces.

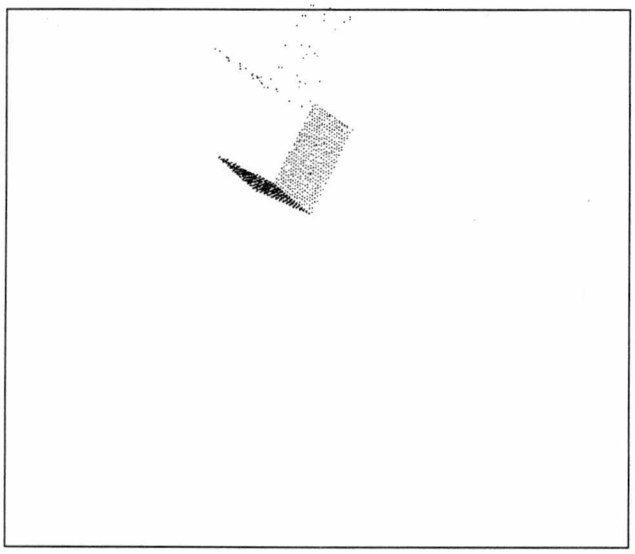


Figure C.10: Measurements collected by PRIME for tests with crease discontinuities.

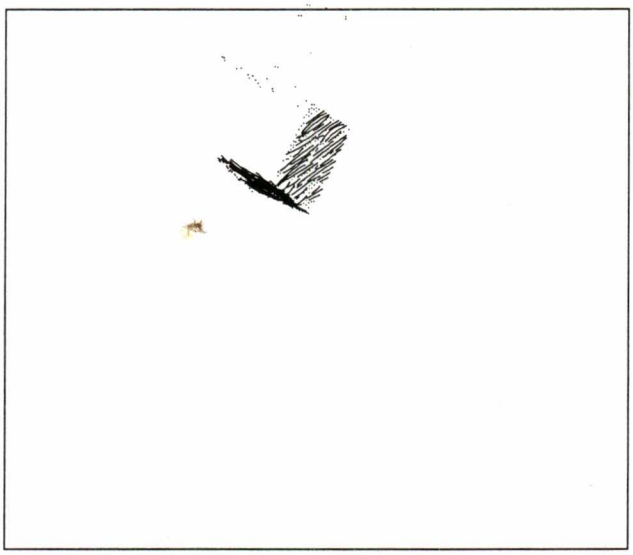


Figure C.11: X-Strips formed by RPT across a crease discontinuity.

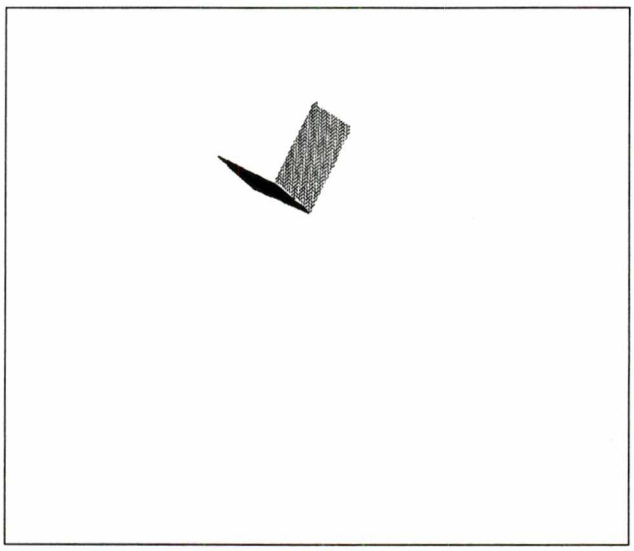


Figure C.12: Surface fitted by RPT bordering a crease discontinuity.

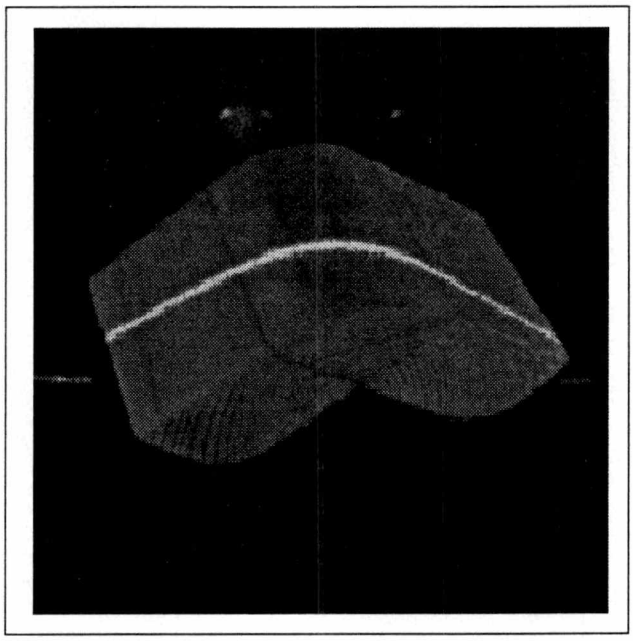


Figure C.13: Typical scene used to test RPT in the presence of a curvature discontinuity.

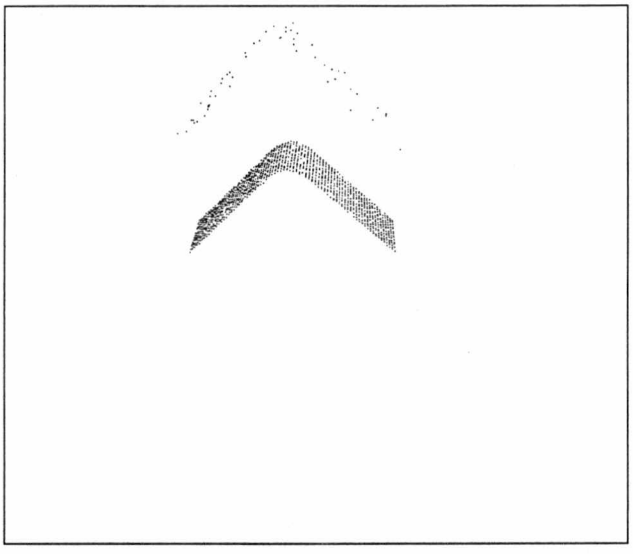


Figure C.14: Measurements collected by PRIME of surfaces separated by curvature discontinuities.

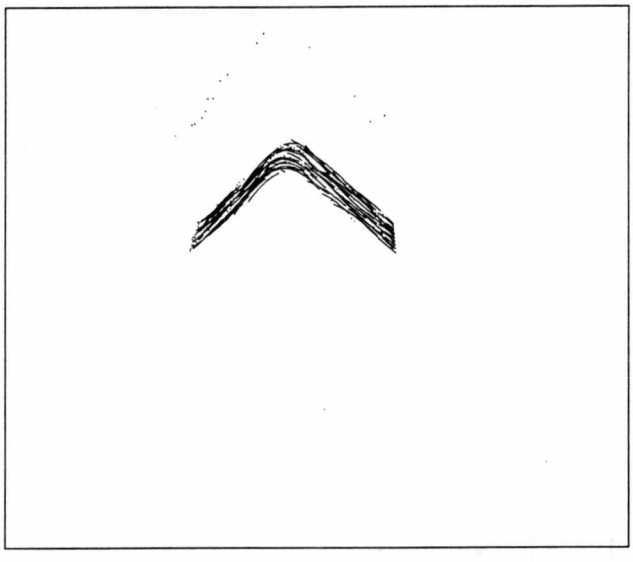


Figure C.15: X-Strips formed by RPT across a curvature discontinuity.

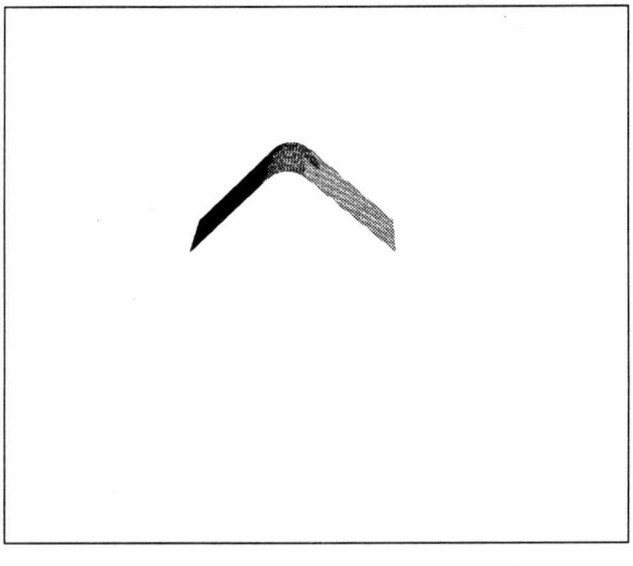


Figure C.16: Surfaces fitted by RPT to regions separated by curvature discontinuities.

VITA

Frederick William DePiero is originally from Pittsburgh, PA. He received his B.S. and M.S. degrees in Electrical Engineering from Michigan State University in 1985 and 1987. Fred worked as a Development Associate at Oak Ridge National Laboratory until 1993. While there he was involved in a variety of real-time image processing projects including a high-compression video transmission system for remote driving and various structured light measurement systems. At that time, his research goals focused on real-time performance and sensor calibration techniques.

Fred is currently a Research Associate and Ph.D. Candidate at the University of Tennessee. His structured light work continues on a high performance measurement system for railway maintenance and on a multipurpose ranging system for industrial applications. His research interests include efficient methods of graph matching and range image segmentation, as well as range sensor development.