INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
On computing perceptual organization in computer vision

Sarkar, Sudeep, Ph.D.
The Ohio State University, 1993
ON COMPUTING PERCEPTUAL ORGANIZATION IN COMPUTER VISION

A Dissertation

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

by

Sudeep Sarkar, M.S, B.Tech.

* * * * *

The Ohio State University

1993

Dissertation Committee:
Professor K. L. Boyer
Professor A. Krishnamurthy
Professor Y. F. Zheng
Professor K. Passino

Approved by

Adviser
Department of Electrical Engineering
To my parents
ACKNOWLEDGMENTS

First of all I would like to thank Dr. Kim Boyer who has not only been my academic advisor but also has been a friend and a constant source of inspiration. Working with him is a stimulating and satisfying experience. I am grateful to Dr. Judea Pearl for writing an excellent book on Bayesian Networks which inspired a significant portion of my work. I also thank my committee members for their helpful comments which enhanced this dissertation.

Of course, this work would not have been possible without the critical, supportive, and friendly atmosphere of the SAMP Lab. Thanks are due to Kim, Dan, Steve, Tan, Jay, Mirza, Nitin, Kuntal, Subha, Ann, Gopa, and Xan. I also appreciate the efforts of Mike and Shane in maintaining a good bibliography.

The financial support from the NASA Center for the Commercial Development of Space and from the Ohio State University is acknowledged.
VITA

November 3, 1966 ...................... Born - Durgapur, India
May, 1988 .............................. B. Tech., Indian Institute of Technology, Kanpur, India
March, 1990 ............................ M. S., The Ohio State University, Columbus, U.S.A.

PUBLICATIONS


**FIELDS OF STUDY**

Major Field: Electrical Engineering
Studies in Computer Vision, Signal Processing, and Artificial Intelligence.
# Table of Contents

DEDICATION ................................................................. ii

ACKNOWLEDGMENTS .................................................. iii

VITA .............................................................................. iv

LIST OF TABLES .......................................................... x

LIST OF FIGURES ......................................................... xii

ABSTRACT ......................................................................... xx

CHAPTER

I Introduction ............................................................... 1

1.1 What? ................................................................. 2

1.2 Why? ................................................................. 5

II A Brief History of Perceptual Organization Research .......... 11

2.1 Classificatory Structure: A Proposal .............................. 14

2.1.1 2D organization .................................................. 16

2.1.2 3D organization .................................................. 28

2.1.3 2D + motion ....................................................... 33

2.1.4 3D + motion ....................................................... 35

2.2 Computational Techniques .......................................... 36

2.2.1 Group theory ...................................................... 37

2.2.2 Coding Theory ................................................... 39

2.2.3 Rule Based ......................................................... 40

2.2.4 Frequency Methods ............................................. 42

2.2.5 Energy/Probabilistic Optimization Methods ................. 43

2.2.6 Graph theory ...................................................... 50
III Overview ................................................................. 52

IV Preattentive Algorithm: Gestalt Graphs and Voting Methods ...... 58
  4.1 The Gestalt Graphs .................................................. 59
    4.1.1 First Level Organization ..................................... 60
    4.1.2 Second Level Algorithm .................................... 62
  4.2 Voting Scheme .......................................................... 65
    4.2.1 Offset Quantization ........................................... 67
    4.2.2 Handling General Compatibility Relations .............. 73
    4.2.3 Complexity Analysis ......................................... 74
    4.2.4 This is Not a Hough Transform ............................ 76
  4.3 Selecting the Organization Parameters ............................... 77
  4.4 Result on the Aerial Image ........................................ 81

V Attentive Algorithm: The Perceptual Inference Network ............ 83
  5.1 Knowledge Base: The Perceptual Inference Network .............. 85
    5.1.1 Belief Revision Using Modified Message Passing ......... 89
    5.1.2 Network Instantiation ....................................... 93
    5.1.3 Prediction of structures ..................................... 94
    5.1.4 Management of Expensive Computational Resources ...... 94
  5.2 An Example PIN for Perceptual Organization ....................... 95
    5.2.1 Manually Structuring the PIN ............................... 95
    5.2.2 Location Compatibility Functions, Comp ................. 98
    5.2.3 Conditional Belief Functions, g ........................... 102
    5.2.4 Prior Probabilities ........................................... 104
    5.2.5 Evidence Instantiation ....................................... 105
  5.3 Algorithmic Complexity ............................................ 106
  5.4 Results ........................................................................ 107
    5.4.1 Aerial Image ................................................... 108
    5.4.2 Performance Evaluation ...................................... 109

VI Structuring a Perceptual Inference Network ............................ 121
  6.1 Overview of the Algorithm .......................................... 123
  6.2 A Structural Representation of Organization ....................... 127
    6.2.1 The Random Parametric Structural Description .......... 128
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.2 Dissimilarity Measure</td>
<td>133</td>
</tr>
<tr>
<td>6.2.3 Structure of the Merged RPSD, $G_m$</td>
<td>135</td>
</tr>
<tr>
<td>6.2.4 Constructing the Mapping between RPSDs</td>
<td>138</td>
</tr>
<tr>
<td>6.3 Structuring the PIN</td>
<td>142</td>
</tr>
<tr>
<td>6.3.1 Dependency Detection</td>
<td>143</td>
</tr>
<tr>
<td>6.3.2 Next Level Formation</td>
<td>145</td>
</tr>
<tr>
<td>6.3.3 Root Node Decomposition</td>
<td>146</td>
</tr>
<tr>
<td>6.3.4 Expected Network Size</td>
<td>148</td>
</tr>
<tr>
<td>6.3.5 Desired Conditions</td>
<td>152</td>
</tr>
<tr>
<td>6.4 Estimating the Conditional Probabilities</td>
<td>153</td>
</tr>
<tr>
<td>6.4.1 Estimating $P(X = 1</td>
<td>U_1 = u_1, \cdots, U_n = u_n)$</td>
</tr>
<tr>
<td>6.4.2 Computing $f_{LOC}(l_k, l_{k_1}, \cdots, l_{k_n})$</td>
<td>158</td>
</tr>
<tr>
<td>6.5 An Example</td>
<td>162</td>
</tr>
<tr>
<td>6.5.1 The Target and Elementary Organizations</td>
<td>163</td>
</tr>
<tr>
<td>6.5.2 The Primitive and Relation Sets</td>
<td>164</td>
</tr>
<tr>
<td>6.5.3 Execution Statistics</td>
<td>168</td>
</tr>
<tr>
<td>6.5.4 Robustness</td>
<td>174</td>
</tr>
<tr>
<td>6.6 Performance on Real Images</td>
<td></td>
</tr>
<tr>
<td>6.6.1 Comparing Automatically and Manually Structured PINs</td>
<td></td>
</tr>
<tr>
<td>6.6.2 Performance on Real Data</td>
<td></td>
</tr>
<tr>
<td>6.6.3 Performance Evaluation</td>
<td></td>
</tr>
<tr>
<td>VII Attentive Algorithm: Resource Management</td>
<td>195</td>
</tr>
<tr>
<td>7.1 Theory</td>
<td>198</td>
</tr>
<tr>
<td>7.1.1 Control Algorithm</td>
<td>199</td>
</tr>
<tr>
<td>7.1.2 Information Theoretic Formalism</td>
<td>202</td>
</tr>
<tr>
<td>7.1.3 Return Condition</td>
<td>205</td>
</tr>
<tr>
<td>7.1.4 Final Hypothesis Resolution</td>
<td>206</td>
</tr>
<tr>
<td>7.2 Special Purpose Modules</td>
<td>207</td>
</tr>
<tr>
<td>7.2.1 Corner Analysis Module</td>
<td>207</td>
</tr>
<tr>
<td>7.2.2 Segment Analysis Module</td>
<td>208</td>
</tr>
<tr>
<td>7.2.3 Region Analysis Module</td>
<td>212</td>
</tr>
<tr>
<td>7.3 Results</td>
<td>214</td>
</tr>
<tr>
<td>VIII Evaluation</td>
<td>219</td>
</tr>
<tr>
<td>8.1 One Level of Organization</td>
<td>221</td>
</tr>
<tr>
<td>8.1.1 Blocks World</td>
<td>221</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Locational information format at each composite node of the example net.</td>
</tr>
<tr>
<td>2</td>
<td>Table showing the variation of the number of hypotheses of some salient nodes with parameter $t$ of the $Comp$ function. The last column lists the execution time for each value of $t$</td>
</tr>
<tr>
<td>3</td>
<td>The table of similarity coefficients, $C_{sim}$, for 9 random priors, each compared with equal prior assignments. The rows are for 4 different composite nodes of the PIN. P is Parallelogram, R is Ribbon, E is Ellipse, and C is Circle.</td>
</tr>
<tr>
<td>4</td>
<td>The table of similarity coefficients, $C_{sim}$, for 9 random perturbations of the constants, $C_{(u_1,\ldots,u_n)}$ compared with the chosen values. The rows are for 4 different composite nodes of the PIN. P is Parallelogram, R is Ribbon, E is Ellipse, and C is Circle.</td>
</tr>
<tr>
<td>5</td>
<td>The table of similarity coefficients, $C_{sim}$, for 9 random perturbations of the constants, $C_{(u_1,\ldots,u_n)}$ compared with the chosen values. The rows are for 4 different composite nodes of the PIN.</td>
</tr>
<tr>
<td>6</td>
<td>The target and elementary organizations used for the example</td>
</tr>
<tr>
<td>7</td>
<td>Primitive and Relation attributes used to construct the RPSDs of the structures of interest.</td>
</tr>
<tr>
<td>8</td>
<td>Different stages of the PIN structuring algorithm.</td>
</tr>
<tr>
<td>9</td>
<td>Root node breakup data. The names in bold refer to our starting root nodes. The others refer to the root nodes which were created in the recursive process.</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>54</td>
</tr>
<tr>
<td>6</td>
<td>56</td>
</tr>
<tr>
<td>7</td>
<td>61</td>
</tr>
<tr>
<td>8</td>
<td>63</td>
</tr>
<tr>
<td>9</td>
<td>66</td>
</tr>
<tr>
<td>10</td>
<td>68</td>
</tr>
<tr>
<td>11</td>
<td>69</td>
</tr>
<tr>
<td>12</td>
<td>71</td>
</tr>
<tr>
<td>13</td>
<td>74</td>
</tr>
</tbody>
</table>

1. Some experiments (adapted from Kanizsa) demonstrating the importance of perceptual organizations. (a) The vertical rectangle passes behind the striped rectangle, even when all of it is completely visible. (b) Symmetrization, by using mirror images of letters, makes word invisible. Formation of groupings based on closure, symmetry, and good continuation prevail over knowledge of letters. (c) Continuity of direction prevails over maximum symmetry.

2. Gestalt laws of grouping

3. Classificatory structure for perceptual organization

4. Note the competing organizations that are observed, analogous to an optimization problem stuck in a limit cycle.

5. Hierarchical Organization to analyze curvilinear structures

6. Perceptual Organization System Block Diagram

7. Algorithm executed to construct first level of the hierarchy

8. Algorithm executed to construct second level of the hierarchy

9. Typical parameter space configuration after voting has been done

10. (a) A quantized parameter space illustrating the effects of quantization. (b) Overlapping one dimensional quantized voting space

11. Using n quantizers for voting

12. Overlapping 2D voting spaces

13. General compatibility volume of the token A in a voting space
14 Measurements used to calculate the non-accidentalness of associations (after Lowe) ................................................... 77

15 Histogram of the orientations of line segments and their associated approximating probability distribution. ........................... 80

16 Organizations detected by the preattentive perceptual organization module. (a) Gray Level Image (b) Edge contours (c) Constant curvature segments (d) Closed boundaries (e) Edge strands (f) Parallel edge segments .............................................................................................................82

17 An Example Bayesian network ................................................... 85

18 An example of a Perceptual Inference Network ........................... 89

19 Block diagram of a composite node ........................................... 92

20 Algorithm at each composite node ........................................... 93

21 Perceptual Inference Network representing knowledge about some salient organizations ......................................................... 97

22 Outline of the compatible location generation algorithm for nodes N4, N9, and N15. ......................................................... 102

23 Organizations detected using the attentive module on the aerial image. (a) New closed boundary hypotheses (b) Parallelograms (c) Ellipses (d) Circles (e) Ribbons (f) Corners .......................................................110

24 The robustness of the equal prior assignment. This shows the plot of the probability assignments of features at different locations for 9 random prior assignments. The features considered are (a) Parallelogram (N23) (b) Ribbon (N13) (c) Ellipse (N6) and (d) Circle (N8). ................................................................. 114

25 The robustness of the choice of the constants, $C_{(u_1,\ldots,u_n)}$, involved in the specification of the conditional probabilities. This shows the plot of the probability assignments of features at different locations for 9 random perturbations of the constants. The features considered are (a) Parallelogram (N23) (b) Ribbon (N13) (c) Ellipse (N6) and (d) Circle (N8). ................................................................. 116

xiii
26 Functional form of the dependence of the conditional probabilities with $f_{geom}$. The exponential solid plot is the one that is chosen for the PIN. This is compared against the dot-dashed and dotted choices.

27 Performance plots depicting the flexibility of specification of the functional dependence of the conditional probabilities with $f_{geom}$. The features considered are (a) Parallelogram ($N23$) (b) Ribbon ($N13$) (c) Ellipse ($N6$) and (d) Circle ($N8$).

28 Flowchart of the PIN structuring algorithm.

29 The recursive RPSD matching algorithm.

30 A typical output PIN structure before the root node decomposition.

31 A typical output PIN structure after the decomposition of root nodes into elementary groupings. Some root node decompositions are not drawn to preserve clarity. The nodes with two incoming arrows denote a root node decomposed structure which is not drawn.

32 A simple PIN to illustrate the generation of compatible PSD. The pointers between the primitives of the PSDs at the neighboring PIN nodes are stored as paired lists.

33 Examples of the computed attributes for typical linear symmetry relations between two edge segments. (a) and (b) represent perfect cases. (c) and (d) represent relaxed cases.

34 An automatically structured PIN.

35 Isomorphic subparts of a PIN. The difference arises because of the order of breakup of the root node EPQR but the two structures have essentially the same behavior.

36 (a) Aerial image. (b) Edges detected in the aerial image. Organizations detected using the preattentive module on the aerial image and input to the PIN. (c) Closed rings of 4 segments (C4) (d) Edge strands of 3 segments (S3) (e) Straight symmetry (PS) (f) Curved symmetry (CS).
The target outputs of the automatically designed PIN (a) Parallelograms (b) Ellipses (c) Circles (d) Quadrilaterals (e) Ribbons (f) Triangles ................................................................. 182

The robustness of the equal prior assignment. This shows the plot of the probability assignments of features at different locations for 25 random prior assignments. The features considered are (a) Parallelogram (P) (b) Ribbon (R) (c) Ellipse (E) (d) Circle (C) (e) Quadrilateral (Q) and (f) Triangle (T) ................................. 189

The robustness of the choice of the constants $\Lambda_{(u_1,\ldots,u_n)}$ ($C_{(u_1,\ldots,u_n)}$). This shows the plot of the probability assignments of features at different locations for 25 random prior assignments. The features considered are (a) Parallelogram (P) (b) Ribbon (R) (c) Ellipse (E) (d) Circle (C) (e) Quadrilateral (Q) and (f) Triangle (T) ........................................... 192

Resource control algorithm ................................................................. 201

Segment and the area analyzed to provide evidence for it ................................................. 209

Determination of area to be analyzed to provide evidence for a region 211

Locations where special purpose modules were applied. (a) Corner module locations. (b) Segment analyzer module. (c) Region analyzer module locations ........................................... 216

Final resolved hypotheses for the aerial image. (a) Best set of hypotheses. (b) Best set of hypotheses overlaid on the edge image. (c) Second best set of hypotheses. (d) Second best set of hypotheses overlaid on the edge image .................................................. 218

Organizations detected by the pre-attentive perceptual organization module. (a) Gray Level Image (b) Edge contours detected (c) Closed boundary hypotheses (d) Edge strand hypotheses (e) Parallel edge segments draw as parallelograms (f) Parallelogram hypotheses detected as intersections of parallel strips ........................................................ 222

Organizations detected using the attentive module on the blocks image. (a) New closed boundary hypotheses (b) Set of parallelograms hypothesized (c) Ellipses predicted (d) Circles (e) Ribbons (no change from pre-attentive hypothesizing) (f) Corners hypothesized ......................... 224

xv
Locations where special purpose modules were applied. (a) Corner module locations. (b) Segment analyzer module. (c) Region analyzer module locations ........................................... 226

Final resolved hypotheses for the blocks image. (a) Best set of hypotheses. (b) Best set of hypotheses overlaid on the edge image. (c) Second best set of hypotheses. (d) Second best set of hypotheses overlaid on the edge image ........................................... 227

Organizations detected by the preattentive perceptual organization module. (a) Gray Level Image (b) Edge contours detected (c) Constant curvature segments (d) Parallel Symmetries (e) Closed boundary hypotheses (d) Edge strand hypotheses ........................................... 230

Organizations detected using the attentive module on the lab image. (a) Ellipses (b) Triangles (c) Circles (d) Quadrilaterals (e) Ribbons (f) Parallelograms ........................................... 231

Final resolved hypotheses for the laboratory image ........................................... 232

(a) Gray level image of a room. (b) Segmented edge contours. (c) Curved linear symmetries (lighter) for the edge segments (darker). (d) Straight linear symmetries. The symmetry axes are in lighter shade. (e) Strands of three segments. (f) Closed ring of four segments. The constituting edges are in lighter shade ........................................... 233

Organizations detected using the attentive module on the room image. (a) Ellipses (b) Triangles (c) Circles (d) Quadrilaterals (e) Ribbons (f) Parallelograms ........................................... 234

Final resolved hypotheses for the room image ........................................... 235

(a) Gray level image of a runway. (b) Segmented edge contours. (c) Curved linear symmetries. (d) Straight linear symmetries. The symmetry axes are in lighter shade. (e) Strands of three segments. (f) Closed ring of four segments. The constituting edges are in lighter shade ........................................... 237

Organizations detected using the attentive module on the runway image. (a) Circles (b) Quadrilaterals (c) Ribbons (d) Parallelograms ........................................... 238
Strong hypotheses of the first level which form the input to the preattentive algorithm of the second level. (a) The hypotheses (b) Line abstraction of the hypotheses (Symmetry Axes) (c) Point tokens (Centroid).

Preattentive cues of the second level of the description hierarchy. (a) Second Level parallel symmetries (b) Continuous parallel symmetries (c) Clusters of the centroids of the features at the first level.

Aerial Image (Left) used (482 x 482)

Aerial Image (Right) used (482 x 482)

Left Image (a) First section of the aerial image (250 x 250). (b) 428 constant curvature edge segments. (c) 11 curved linear symmetries. (d) 67 straight linear symmetries. The symmetry axes are in lighter shade. (e) 22 strands of three segments. (f) One closed ring of four segments. The constituting edges are in lighter shade.

Left Image (a) Second section of the aerial image (242 x 250). (b) 364 constant curvature edge segments. (c) 16 curved linear symmetries. (d) 29 straight linear symmetries. The symmetry axes are in lighter shade. (e) 5 strands of three segments. (f) 5 closed ring of four segments. The constituting edges are in lighter shade.

Left Image (a) Third section of the aerial image (250 x 242). (b) 370 constant curvature edge segments. (c) 10 curved linear symmetries. (d) 48 straight linear symmetries. The symmetry axes are in lighter shade. (e) 16 strands of three segments. (f) 5 closed ring of four segments. The constituting edges are in lighter shade.

Left Image (a) Fourth section of the aerial image (242 x 242). (b) 339 constant curvature edge segments. (c) 14 curved linear symmetries. (d) 27 straight linear symmetries. The symmetry axes are in lighter shade. (e) 13 strands of three segments. (f) One closed ring of four segments. The constituting edges are in lighter shade.

Left Image The first level description of the aerial image sections in terms of ellipses, circles, triangles, quadrilaterals, parallelograms, and ribbons. For (a) First section (250 x 250). (b) Second section (250 x 242) (c) Third section (242 x 250) (d) Fourth section (242 x 242).
The second level preattentive organizations for the left image. (a) Parallel symmetries (b) Continuous (symmetry) lines (c) Strands of symmetry axes (d) Clusters of similar features. Each feature is represented as a dot and lines are drawn between similar features. (Note that the images have been reduced for display.)

Right Image (a) First section of the aerial image (250 x 250). (b) 361 constant curvature edge segments. (c) 13 curved linear symmetries. (d) 62 straight linear symmetries. The symmetry axes are in lighter shade. (e) 13 strands of three segments. (f) 2 closed rings of four segments. The constituting edges are in lighter shade.

Right Image (a) Second section of the aerial image (242 x 250). (b) 409 constant curvature edge segments. (c) 6 curved linear symmetries. (d) 20 straight linear symmetries. The symmetry axes are in lighter shade. (e) 3 strands of three segments. (f) 4 closed ring of four segments. The constituting edges are in lighter shade.

Right Image (a) Third section of the aerial image (250 x 242). (b) 316 constant curvature edge segments. (c) 11 curved linear symmetries. (d) 44 straight linear symmetries. The symmetry axes are in lighter shade. (e) 24 strands of three segments. (f) 2 closed ring of four segments. The constituting edges are in lighter shade.

Right Image (a) Fourth section of the aerial image (242 x 242). (b) 440 constant curvature edge segments. (c) 5 curved linear symmetries. (d) 27 straight linear symmetries. The symmetry axes are in lighter shade. (e) 22 strands of three segments. (f) One closed ring of four segments. The constituting edges are in lighter shade.

The first level description of the aerial image sections in terms of ellipses, circles, triangles, quadrilaterals, parallelograms, and ribbons. For (a) First section (250 x 250). (b) Second section (250 x 242) (c) Third section (242 x 250) (d) Fourth section (242 x 242).

Right Image. The (a) 291 linear and (b) 432 point tokens used for the construction of the second level.
The second level preattentive organizations for the right image. (a) Parallel symmetries (b) Continuous (symmetry) lines (c) Strands of symmetry axes (d) Clusters of similar features. Each feature is represented as a dot and lines are drawn between similar features. (Note that the images have been reduced for display.) 259

A segment of a tree 270
ABSTRACT

The evolution of perceptual organization in biological vision, and its necessity in advanced computer vision systems, arises from the characteristic that perception, the extraction of meaning from sensory input, is an intelligent process. This is particularly so for high order organisms and, analogically, for more sophisticated computational models. By perceptual organization we refer to the ability of a vision system to organize detected features in images based on viewpoint consistency and other Gestaltic perceptual phenomena. This imparts robustness, efficiency, and a qualitative and holistic nature to vision.

Our computational paradigm aims to organize features into highly plausible sets of higher level geometric features which are present in images of objects belonging to a large number of domains. Our organizational philosophy is hierarchical, with complex organizations being formed from simpler ones. Each level of the hierarchy is constructed using voting methods, graph operations, and knowledge based reasoning in a new extension of the Bayesian network we call the Perceptual Inference Network. Analogous to theories in human vision, our strategy divides broadly into two parts: detecting regularities and similarities in the tokens (preattentive vision) and reasoning, based on a knowledge base built from past experience, to enable one to go beyond the information provided (attentive vision). The voting method
provides organizations based on Gestalt principles and the network reasons on those organizations to extract geometric features. The two steps of voting and evidential reasoning are repeated.

Previous approaches to perceptual organization have mostly been purely bottom up, without any top down knowledge base influence and therefore entirely dependent on the inputs, which may be imperfect. The knowledge base, besides coping with such input imperfections, also allows us to integrate multiple sources of information and to form a composite organization hypothesis.
CHAPTER I

Introduction

"Perception is not a mere passive recording of information impressed upon my sensory organs by the environment. Rather, it consists of an active construction by means of which sensory data are selected, analyzed, and integrated with properties not directly noticeable but only hypothesized, deduced, or anticipated, according to available information and intellectual capacities."

Gaetano Kanizsa

Perceptual organization can be defined as the ability to impose structural organization on sensory data, so as to group sensory primitives arising from a common underlying cause. This sort of organization lets us form object hypotheses with minimal domain knowledge and, therefore, minimal restrictions, beyond the fact that our world is not visually chaotic; it has structure and organization. The role of perceptual organization in vision, or any other sensory modality, is critical to success. It imparts robustness by making it more resistant to minor changes in the input and also brings computational efficiency to the perceptual process. The importance of finding organization in sensory data has long been recognized by researchers in
human visual perception, especially the Gestalt\(^1\) psychologists. Its role in computer vision was emphasized and demonstrated much later by researchers such as Witkin and Tenenbaum [1] and Lowe [2]. Recently, perceptual organization has been identified as one of the insufficiently emphasized areas in early computer vision and the US High Performance Computing and Communications (HPCC) initiative also identifies perceptual grouping as one its four problem areas [3].

1.1 What?

The Gestalt psychologists observed and emphasized the importance of organization in vision. They demonstrated that shapes have some elusive, immeasurable collective properties that do not appear when analyzed by their constituent parts. The Gestalt school of psychology, founded by Koffka [4], Kohler [5] and Wertheimer [6] in the early 20\(^{th}\) century, demonstrated the role of structure or organization by a number of self evident examples. Besides the convincing examples, their lucid and polemic writing style made their work very popular. They recognized what is computed in perceptual organization, but did not convincingly answer why it is calculated or how. This spurred several debates which obscured some of the original ideas and created a number of misconceptions. Kanizsa [7] offers some interesting comments about the popular misconceptions regarding Gestalt psychology, which are important to keep in mind as we try to incorporate Gestaltic behavior in computer vision systems. He translates Gestalt to “organized structure” as opposed to “form”, “aggregate” or

\(^1\)Gestalt is a German word which roughly translates to “organized structure”. Gestalt theory is a very general psychological theory that can be used to study and understand aspects of human behavior and experience.
"heap." The stress is on the concept of organization and on a whole that is *orderly*, *rule-governed*, and *nonrandom*. His other assertions are listed below:

1. Gestalt Psychology is *not* basically a psychology of perception. Gestaltists have experimented in perception [4], productive thinking [8], problem solving [9], memory and learning [10], and more. The Gestalt theory is a very general system of psychological concepts that can be used to understand virtually any aspect of experience and behavior. Thus, these principles should prove useful, in multiple ways, in building an autonomous robot exhibiting intelligent behavior.

2. Gestalt Psychology proposes that the "properties of the whole are not the result of a summation of those of the parts" and not quite the tenet that "the whole is greater than the sum of its parts." Thus, a part has properties that depend on the whole in which it is included. However, this does not necessarily imply that everything depends on everything else, which would certainly hinder scientific analysis. The dependence of parts on the whole has a varying nature. Gestalt psychology merely maintains that the parts of the whole are not completely independent and hence cannot be analyzed separately.

3. Gestalt psychology does not deny the importance of attention, intention, interest, attitude, and organizational factors in our perceptual experience.
4. Gestalt Psychology does not deny the influence of past experience on perception. However, it does deny the empiricist view that past experience is a universal explanatory principle. Wertheimer [6, pages 231-232] explicitly states that past experience helps in building the dependencies among various parts. For three events A, B, and C, if AB and C, but not BC, have become habitual (or "associated"), there is then a tendency for ABC to be perceived as AB and C.

5. The regularity of a structure is not just its symmetry. One of the basic principles of Gestalt psychology is that organization tends towards Prägnanz, i.e., the tendency of a process to realize the most regular, ordered, stable, and balanced state possible in a given situation. It is not just symmetric regularity (see Fig. 1(c)), stability, simplicity, and cohesion also play a role.

It is interesting to note that very few of the above aspects of Gestalt theory have so far been used in computer vision. As we shall see from the review in Chapter II, issues of attention (point 3), use of a knowledge base and learning (point 4), and criteria other than geometric regularity (point 5) are rarely considered. This is partly due to the semantic imprecision of the concepts of attention, intention, and attitude. However, these might be important in the context of task based or active vision.

Kanizsa [7] suggests that perception of form or shape can be divided into two processes. The primary process segregates the visual field into regions having spatial and temporal regularities. The secondary process, which helps us in going beyond
the information given, is the perceptual inference of totalization, of completion, of integration, of "filling in the gaps," i.e., of inferring that which is absent. This way of approaching the problem is beneficial in the context of machine vision; the second process should help us to bridge the gaps created by imperfect segmentation in the first or by occlusions in the scene. However, as Kanizsa illustrates, exploiting transparency, we might perceive organizations that are physically impossible or "meaningless" based on past experience (see Fig. 1(b)) but which are perceived because of the strong segmentation cues provided by the first process. This indicates that the second process is not the sole determining factor of organization.

Kanizsa observed that the completion of contours which are partially occluded is decided mainly by continuity in direction at the occluding point and the minimal distance criterion when the first criterion does not give rise to closed figures. Symmetry plays only a minor role. There is also a preference for convex forms over symmetric ones (see Fig. 1(c)).

1.2 Why?

It is natural to ask why organization in vision is so important. The answer is neither simple nor self evident and has far reaching effects in general cognition because we also seem to use this principle in other forms of perception. The functional role of

---

2Most researchers in computer vision consider a Euclidean space for their perceptual organizer. However, there is considerable evidence that the human visual system is not Euclidean. Watson [11], postulated that the visual spaces are Riemann spaces to explain the interaction between the elements of the visual field. Using this he was elegantly able to explain a number of visual illusions and figural after effects, such as the Poggendorff, Enclosure, Muller-Lyer, Ponzo, Wundt-Hering, Orbison, Zollner, and Delboeuf illusions, and the Kohler, Willems, and Delboeuf figural after effects.
Figure 1: Some experiments (adapted from Kanizsa) demonstrating the importance of perceptual organizations. (a) The vertical rectangle passes behind the striped rectangle, even when all of it is completely visible. (b) Symmetrization, by using mirror images of letters, makes word invisible. Formation of groupings based on closure, symmetry, and good continuation prevail over knowledge of letters. (c) Continuity of direction prevails over maximum symmetry.
organization is very difficult to frame. However, this does not diminish its importance. Any theory of perceptual organization should explain not only organization in vision, but also in other cognitive activity like learning (where the organization of knowledge in the form of "chunks" is well known), music appreciation, and so on.

Many have attempted formulating a reason for the importance of perceptual organization. One of the theories which appeals due to its simplicity and elegance is that of non-accidentalness, also known as the principle of common-cause, or coincidence explanation. This was independently proposed by Witkin and Tenanbaum [1], Lowe [2], and Rock [12] in explaining the preference for one perceived form over another. The low probability of the chance occurrence of a particular organization imparts a very high significance to it if found. This implies a single causal entity for those features so organized. In vision we are most interested in spatio-temporal coherence or regularity.

The importance of organization can be made more explicit using the following mathematical language. Let Causality denote the event that a set of features are part of (or caused by) the same object and let Organization denote the event that the features have some organization among themselves. Then we are interested in the probability that the features come from the same underlying cause, given that we have, found some organization among them: \( P(\text{Causality}|\text{Organization}) \). Using Bayes rule we can write:

\[
P(\text{Causality}|\text{Organization}) = \frac{P(\text{Organization}|\text{Causality})P(\text{Causality})}{P(\text{Organization})} \quad \text{(1.1)}
\]
Each of the three terms on the right hand side of the equation denotes an important concept. $P(Causality)$ is the prior probability that a set of features come from a common cause. This in some way captures the meaningfulness of a scene. In an entirely random world this would be very low. In actual life this is quite high. $P(Organization)$ is the prior probability of organization being present among a set of features. The value of this decreases as the complexity of the organization increases. This captures the probability of accidental occurrence of the organization. $P(Organization|Causality)$ is the probability that we will observe an organization among a set of features given that they come from the same cause. This captures the fact that matter is coherent and behaves according to some fixed law (disclosed or yet to be disclosed). This value is high. Thus we see that we can safely infer causality from those organizations which have a very low prior probability of accidental occurrence and a very high probability of being exhibited by matter. Both terms must be considered. Considering one alone will not suffice. For example, a given organization may have a very low probability of occurrence owing to its sheer complexity. However, if the probability of that structure being a part of an object is also low, it will still be less than useful. Similarly, there might be relations or organizations that occur frequently in real objects. But if their random occurrence probability is also high, then their significance is diminished.

Depending on the world we have, our set of salient organizations will differ. The Gestalt psychologists have found a set of properties which are important in the perceptual organization of images in our real world. They are proximity, continuity,
Figure 2: Gestalt laws of grouping
similarity, closure, symmetry, and the new properties [13] of common region and connectedness (see Fig. 2). These properties are exhibited by any salient organization and so we should search for groupings exhibiting these properties. The importance of proximity was demonstrated empirically by Brunswick and Kamiya [14] using several stills from a motion picture. The number of adjacent straight and parallel lines were counted along with their separation. The pairs of lines were then classified according to what they represented in the scene. A correlational analysis on the separation gap between features on objects and the gap between objects yielded a significantly high correlation coefficient. This means that $P(Proximity|Causality)$ is very high. In summary, organization provides a very good indicator of common causality and the saliency of an organization depends on its probability of accidental occurrence and its probability of being part of an object.

In the next chapter we review the work in computer vision which employ perceptual organization to their advantage. In the process, we identify open areas of perceptual organization research. The overview of the proposed perceptual organization paradigm is in Chapter III. Chapter IV concerns the preattentive module consisting of the graph theoretic operations and the voting operations. We develop the attentive parts: the PIN, structuring of PINs, and resources management in Chapters V, VI, and VII respectively. At various stages in our discussion we present results on one aerial image to illustrate the working principles. More results and extensive tests are presented in Chapter VIII. We conclude with directions for future work in Chapter IX.
CHAPTER II

A Brief History of Perceptual Organization Research

"Progress, far from consisting in change, depends on retentiveness .... Those who cannot remember the past are condemned to fulfill it."

George Santayana

The role of perceptual organization is best illustrated by the very interesting experiment reported by Smith [15]. A perceptually random stimulus was presented to subjects who were asked to reproduce it. The reproduced set was used as stimulus to a second set of subjects who were also asked to reproduce what they were shown. After about 12 cascades the final reproduction had a definite structure to it. The subjects had (gradually) imposed organization on an entirely random stimulus! The experiment indicates the importance of organization in human perception. The human visual system values organization to such a degree that it attempts to find (or impose) it even when none actually exists. Clearly this heuristic behavior has evolved in a world which, generally, is very much structured.

In computer vision, Marr [16] suggested incorporating groupings based on curvilinearity into larger structures in the primal sketch. Grouping points into parame-
terized curves using Hough transforms [17] has received a lot of attention. Though the method has been generalized to handle a very large class of curves, it suffers from the fact that it ignores proximity. Witkin and Tenenbaum [1] recognized the broad implications of perceptual grouping methods for computational theories in machine vision in providing an interpretation more in keeping with its qualitative and holistic character. Perceptual organization can impart the robust, qualitative, and holistic nature of human perception to the frail, quantitative, and local character of most current algorithms in computer vision. The perceived structure provides us with a more meaningful description than do quantitative local surface characteristics.

An important observation that can made from this review is that the role of perceptual organization (in the true Gestaltic sense) has been very minor if not altogether absent in computer vision research \(^1\). The efficacy of perceptually significant structures in model based vision has been shown by Lowe and others. The general approach entails detecting simple organizations, like parallel lines and rectangles, followed by model based matching. In this scenario, the organizations impose constraints to prune the search tree. Although these are good, solid results, we feel that perceptual organization has an even wider role to play in vision and hope that this review will also offer pointers to further research.

In the present state of affairs, a variety of researchers pursuing various specific goals have advanced numerous interesting computational approaches to organization based on different sets of mathematical tools. This has engendered a proliferation

---

\(^1\)This is not to say that work has not been done but that the full potential of perceptual organization has not been realized.
of research on perceptual organization embodying many approaches over many levels, with no particular agreement on the terminology and vocabulary. In short, the term has become somewhat ambiguous, meaning different things to different people. Therefore, in this chapter we will not only review prior work but also organize the research already accomplished in perceptual organization and propose a standard nomenclature with which to discuss existing and future work. The classificatory structure for perceptual organization work and the computer vision work is reviewed in this light in Section 2.1. Section 2.2 discusses the various computational techniques that have been used to compute organization.
2.1 Classificatory Structure: A Proposal

The concept of perceptual organization can be applied to a number of cognitive activities, not just to vision. In vision, it can be applied at a number of stages and use different types of features. This state of affairs gives rise to some ambiguity in the term "perceptual organization." In this section we propose a classificatory structure for perceptual organization in computer vision strictly from a dimensionality, domain, and I/O point of view. We do not classify algorithms per se.

Perceptual groupings differ from one another with respect to the types of features they comprise and the dimensions over which the organizations are sought [1, page 521]. We use these two factors as two axes in our classificatory structure. Fig. 3 presents the classificatory structure we advocate. It should become clearer as we proceed. One axis represents the dimensions over which organization is sought: 2D, 3D or 2-1/2 D, 2D plus motion, and 3D (2-1/2 D) plus motion. The other axis denotes the feature types to be organized stratified by layers of abstraction: signal level, primitive level, structural level, and assembly level. Although somewhat self explanatory, the definitions for these will become clearer presently. Thus one can talk of "2D signal level" perceptual organization or "2D+time structural level" organization and we in the computer vision community will have a common understanding.

The features to be organized are classified into four categories: signal, primitive, structural, and assembly. The signal level pertains to organizing the raw signal. For example, the gray level image in 2D, the range image in 2-1/2 D, the motion sequence
in 2D plus motion, and the range sequence in 3D plus motion. The delineation of the next two types is based on the "dimensionality" of the feature with respect to the domain of organization. The criteria of dimensionality, although not mathematically strictly defined here, refers to the number of parameters that are needed to define a feature. For example, a contour segment is one dimensional and a ribbon is two dimensional.

The primitive level deals with organizing signal level extracted features into lower dimensional manifestations in the organizing field. For example, constant curvature segments and region boundaries are 1D manifestations in 2D, surfaces are 2D manifestations in 3D. Hence, constant curvature segments and surfaces constitute primitive level organizations in their respective domains. At the structural level the organized features have the same dimensionality as that of the space in which they are being organized. Ribbons and closed regions are 2D manifestations in 2D and so are structural level features for 2D organization. The assembly level is concerned with further organizing the structural level features. Organizations such as parallel sets of ribbons or boxes constitute the assembly level for 2D grouping.

Let's consider the layout of Fig. 3. The information in each box of the matrix is arranged as follows. The first row lists some of the typical features to be organized at this level and dimension set. The second row lists some typical output organizations from modules at this level and dimension. The third row lists, by author names, some of the representative work in this area. None of these lists are exhaustive; this is just a sampling to convey the ideas. Clearly, not all categories in our classificatory
structure have representative work to cite. Those categories lacking exemplars are areas for future work.

2.1.1 2D organization

We first consider 2D organization. At the signal level we are mainly concerned with organizing pixels or interest points into extended regions, edge chains, or dot clusters. Most of the work in edge detection and region segmentation fall into this category. Edge detection organizes points which are possible object boundaries by looking for those exhibiting certain photometric characteristics. Region segmentation groups pixels into regions using mainly proximity and similarity.

At the primitive level we organize edge chains and regions. Most of the work in contour segmentation falls under this category, in which we search for contiguous edge pixels similar in such attributes as curvature and contrast. The organized features are of dimensionality one. For example, constant curvature segments, segments broken at points of high curvature, and region boundaries.

The features produced at the primitive level are then organized at the structural level, where we look for structures which are of dimensionality two such as ribbons, corners, merges, polygons, closed regions, and strands among them. These basic structures are then organized at the assembly level to identify arrangements of these. So we may look for groups of rectangles and groups of ellipses. Incidentally, in all cases we take "assembly level" perceptual organization to be unbounded from above. That is, if the situation permits, this can continue indefinitely.
Signal Level

This level involves the most basic form of organization. Here our primitives (or tokens) may be points of high gradient, as in edge detection [18, 19, 20, 21, 22], or points of constant gray level as in region segmentation [23, 24].

Liou, Chiu and Jain [24] use the term signal level perceptual organization to mean the partitioning of gray level or range images into regions. Since the term is very descriptive, we have adopted it for our classificatory structure and extend the notion to dynamic scene analysis, to which they allude, as well. Their contributions are threefold. First, they suggest a parallel, single pass algorithm in contrast to iterative region segmentation schemes. Secondly, they successfully integrate edge and region information. Thirdly, they explicitly recognize the low level process as a class of perceptual organization. The starting point of their algorithm is an image or range map to which an edge detector is applied. A set of α-images are created by thresholding the gradient magnitude at α and identifying the connected components. The range of thresholds to use is either determined from the image or provided externally. These α-images are evaluated in parallel by fitting surfaces to the partitions and analyzing the residuals. Poor partitions are deleted, giving a set of filtered α-images which are inclusive "ORed" to produce the final segmentation, along with the fitted surface model. The algorithm is simple and has been shown to work on a wide variety of images. The Gestalt principles of similarity, continuity, and connectedness are used in the organizational process.
Zucker [25] studied the problem of contour inference from a collection of dots, which is an instance of the problem of early orientation selection. Algorithms for extracting orientation information are devised by taking a differential geometric point of view. Ahuja and Tuceryan [26] propose a novel way to extract structure from dot patterns by integrating information from multiple constraining agents using probabilistic relaxation. The geometric structure of an arrangement of dots is represented in terms of the geometric properties of their Voronoi neighborhoods 2. Gestalt criteria, such as border and curve smoothness, serve as constraints in the process. The output is a set of perceptual segments of dots that group together and are classified as being interior to a region, bordering a region, or isolated. The strength of the approach, obvious from the results presented, derives partly from the use of a small number of adaptive thresholds.

Reed and Wechsler [28] use a joint spatial/spatial frequency representation, specifically the Wigner distribution, to do texture segmentation and Gestalt grouping. The use of frequency domain concepts to explain the Gestalt phenomena in visual perception is discussed in [29, 30]. Their results are in good agreement with those predicted by the Gestalt laws for very simple test images. The findings of Beck, Sutter, and Ivy [31] also suggest that at least tripartite segmentation is primarily a function of the spatial frequency components. However, they point out that the effect of geometric attributes and spatial relations such as elongatedness, contour

\[^2\text{Voronoi neighborhoods are an elegant way to compute neighbors of points not only in a 2D space but also for higher dimensions. However, the computational complexity grows with the number of dimensions. In fact it is } O(d^3N^{[d/2]} \log N) + O(dN^{[d/2]+1}) \text{ where } d \text{ is the number of dimensions and } N \text{ is the number of points [27].}\]
smoothness, and contour alignment on texture segmentation cannot be explained in terms of low spatial frequency differences and suggest the existence of a symbolic grouping process.

**Primitive Level**

This level involves the organization of curves into primitives which are manifestations of dimensionality less than two such as linear segments, constant curvature segments, and basic region primitives (like convex regions boundaries). Fischler and Bolles [32] studied the problem of perceptual organization in the partitioning of curves. They concluded that the partitioning problem does not have a unique definition and depends on the purpose, data representation, and trade off between error types. They identified two important criteria for segmentation: the stability of the partitions under minor perturbations and the Gestalt principles of simplicity, conciseness, and completeness of the description. Based on these principles, they suggest two algorithms for curve partitioning. One of the algorithms analyzes the deviations of the curve from a chord that is iteratively advanced along it. The algorithm has two parameters: the length of the chord and the deviation tolerance. The second algorithm fits lines and circular arcs to segments. It selects a small number of seed segments from the curve, analyzes line or arc fits to them, and the good fits are then grown as far as possible. After a large number of seeds have been grown, a histogram of the start and end points is used to select critical points, starting with the point that occurs most often.
Lowe [33] organizes curves at multiple scales of Gaussian smoothing and corrects for curve shrinkage before segmenting. The criterion for segmentation is the rate of change of curvature; this is a form of the Gestalt principle of continuity. Thus, a contour with slowly changing curvature is classified as smooth and is not partitioned. Wuescher and Boyer [34] study the partitioning of curves based on constancy of curvature using voting methods. Each point on the curve votes for a range of curvature and the constant curvature segments are extracted iteratively beginning with the largest.

Zucker, David, Dobbins and Iverson [35] tackle the problem of curve detection in a novel fashion using the principle of continuity. They have two distinct phases: first inferring the tangent (orthogonal to the local gradient) field and then inferring a covering of the field using global splines. The discrete tangent field represents the trace of the curve together with a coarse estimate of the tangent and curvature at those points. The field is calculated by minimizing a functional of position, tangent, and curvature. In the second stage a spline covering of the tangent field is sought. The initial splines are those dictated by the tangent field. These later migrate according to an energy function which is minimized. The global curves are recovered to subpixel accuracy.

Straight line extraction in 2D based on perceptual factors of proximity and continuity is presented by Boldt et al. [36]. They consider four important issues in the development of the algorithm. Firstly, the global structure be constructed from a local process. They advocate a locally parallel hierarchical grouping process. Sec-
ondly, the grouping process should aid in data abstraction into an easily accessible structure. Thirdly, the procedure should be computationally efficient. Lastly, the organization should provide a multiscale geometric description of the image tokens. In their system symbolic line tokens are used and geometric relations such as collinearity, proximity, and similarity are used to control the grouping process. Each step of the algorithm involves linking, optimization, and replacement of token groups by new tokens having additional emergent properties. The number of levels in the hierarchy is controlled by the neighborhood size of each token considered at each step. The neighborhood size is controlled, in turn, by the amount of search one is willing to do at each step. As the grouping cycle proceeds, shorter line tokens are replaced by longer line tokens. In the linking process a global, directed link graph is created in which the set of vertices represent all the line segments in the image at the given scale, and the arcs represent the links. Each link denotes that the two line segments satisfy appropriate Gestalt properties. In the optimization step all paths in the line graph of length appropriate to the scale and containing the line under consideration are generated. Each path is a possible replacement hypothesis and is evaluated using a least squares fit. Each group whose fit error is low is replaced by a new line token in the replacement step. And the cycle continues. The starting point is a Laplacian image with gradient estimates at each point. The results are quite impressive with 6 to 7 hierarchical steps.

The above results are extended by Dolan and Weiss [37, 38] using the same philosophy of hierarchical organization and principles of perceptual organization such
as proximity and smooth continuation to identify co-curving and curvilinear structures. The system is iterative, at least in concept, over a range of perceptual scale, or increasing neighborhood size at every step. They construct an association graph under some compatibility relation defined between two tokens. As demonstrated, they link single pixels based on proximity and orientation to form a graph and enumerate all paths (of length 5 pixels) of the subgraph within a perceptual window (of 0.5 pixels). These paths are called strands. The strands are classified into straight, infection, cusp, corner, conic, and noise. These are then evaluated and replaced to create the next level. The identification of curvilinear structure at multiple scales is also investigated by Saund [39]. He primarily uses the Gestalt principles of proximity and continuity.

**Structural Level**

This is the level that most people mean when they use the term “perceptual organization” and is the one where the most work appears. Line or region tokens found at the primitive level (straight lines, curved lines, and extended regions) are organized into a variety of 2D shapes. The most sought after organization is parallelism. Others include rectangles, circles, convex strands, ribbons, closed figures, and symmetrical figures. Most often these organizations are used to index a model base to speed up recognition. Although the detection of more complex features is possible, very few attempts are reported.
Lowe, who has conducted much of the pioneering work on perceptual organization in machine vision [2, 40], did his most of his work in this category. In the SCERPO system, he used grouping to find collinear and parallel lines. He advocates an organization strategy based on a viewpoint invariance condition and the non-accidentalness constraint. That is, the perceptual features should remain stable over a wide range of viewpoints and be sufficiently constrained so that accidental instances of particular spatial relationships are unlikely to arise. He ranks perceptual groups according to their significance. The significance of a grouping is inversely proportional to the a priori probability of the occurrence of that event. The negative logarithm of the probability, which is the amount of information gained by observation of that event, is a good measure of the significance. To build the grouping, one has to search all possible pairs of tokens and calculate the grouping significance. This is computationally expensive, so Lowe limits the search to a small neighborhood of the token in a parameter space. To increase the efficiency, he indexes his segments in a grid-like data structure according to the endpoint positions. The matrix can further be indexed according to orientation and length. Though he gains temporal efficiency, this scheme requires lots of storage space.

Henikoff and Shapiro [41] use the significant relation of a triple, which is a convex strand of three line segments, to do model-based vision via consistent labeling.McCafferty [42] has made a unique contribution by formulating the grouping problem in perceptual organization as an energy minimization problem, which he then solves using simulated annealing. His formulation handles lines and regions in the same
framework. The energy functional involves terms related to continuity, similarity, closure, and proximity. The relative contribution of each term can be adjusted using multiplicative constants to allow for higher level interactions. Results on synthetic and simple real images are demonstrated. The idea is unique in the sense that it recognizes the need for the influence of higher level knowledge and has a means of doing so, even though it is not clear how this will be achieved in practice.

Mohan and Nevatia [43] use perceptual organization concepts to detect and describe buildings in aerial images. They recognize the usefulness of the structural relationships made explicit by perceptual organization in complex image understanding. They call the emergent structures \textit{collated features}, which are described by the generic shape of the target objects in the scene. All reasonable feature groupings are first detected and the promising ones are then selected by a constraint satisfaction network. They demonstrate impressive results on real scenes. Buildings are approximated by rectangles or combinations thereof. They first detect linear features, which are then grouped into parallels. Parallel collation with aligned endpoints triggers the formation of a U structure. Two U structures trigger the formation of a rectangle hypothesis. The search for these structures is done exhaustively. They extend the results to curved segments in [44]. Co-curvilinearity is detected by imposing continuity and proximity constraints and exhaustive search. This algorithm is used by Chung and Nevatia [45] in their hierarchical stereo system with explicit occlusion detection. A hierarchy of edges, curves, and ribbons are created from each image and matched.
Structures such as rectangles are important cues for buildings in aerial images and are used in several implementations, for example [46, 47]. In [47] Heurtas and Nevatia search for rectangle hypotheses starting from local contour tracing techniques. Shadows are used to confirm hypotheses and estimate the height of the buildings. In [48], Heurtas et al. detect runways in aerial images. They form hypotheses by looking for extended rectangular shapes, or extended strips. These are verified by looking for expected markings and the like.

Computing salient structures from local characteristics is suggested by Sha’ashua and Ullman [49] using a locally connected network. The output is a saliency map, a representation emphasizing conspicuous locations. The idea is to model preattentive vision. To restrict the computational demands, only local structural saliency is considered; global saliency such as symmetry is ignored. The local saliency measure depends on two factors: the total length of the curve and its smoothness. The global optimum is obtained by recursive local optimization, similar to dynamic programming. Each pixel in the image represents a computational element with \( k \) local communication links. The final output is a set of smoothed traced curves with gaps filled in and associated saliency measures. As is evident, the Gestalt principles of proximity and continuity play a large role in this formalism.

Horaud et al. [50] describe a method to compute, by exhaustive search, an intermediate level description in terms of curves and rectangles by grouping image features into abstract structures useful for object recognition [2], or for structural stereo [51, 52]. Their vocabulary of description includes linear and curved contours,
junctions, and local symmetries like parallels, ribbons, and parallelograms. The input to the process is a list of straight line segments.

Nitzberg and Mumford [53] organize regions to detect occlusions using energy minimization. The energy functional is biased toward constant, larger regions with smooth boundaries. The algorithm incorporates three stages: finding edges and T junctions, hypothesizing combinations, and minimizing the energy functional combinatorially. They demonstrate the potential to detect occluding surfaces in real images and on Kanizsa's triangle. Williams [54] detects occluding contours using linear integer programming over a set of constraints. From detected lines, corners and collinear lines are identified. Nearby endpoints are joined using virtual lines, and the intersections of virtual lines are also identified. Each corner and vertex has a fixed number of constraints and the optimal solution is found using the Simplex algorithm. The constraints derive from physical validity and involve occlusion polarity, consistency with the image data, and human visual preferences such as that for closed, convex figures. Impressive results are demonstrated on simple Kanizsa patterns. How well it performs in real images is not discussed.

Rearick, Frawley and Cortopassi [55] organize binary regions into cusps, loops, and elongated regions. At the next level they look for adjacency of different types, and then for unions of structures. They demonstrate results only for synthetic occluded objects. They speculate that intelligent behavior is more likely to originate from simple grouping processes than from previously non-existent symbolic representations. In [56] Quan and Mohr use perceptual groups to hypothesize matches in
two motion sequences. The organizations they compute are skewed and Euclidean parallelism, and collinearity; they work with straight lines.

Kanatani [57] proposes a hierarchical model to detect geometric configurations by first hypothesizing a configuration and then determining how much the original edge must be displaced to support it. It is not clear how the hypotheses are formed.

Among other work related to structural level perceptual organization is a rich body of literature on symmetry detection, e.g. [58, 59, 60, 61, 62, 63]. Van Gool et al. [64] report a novel way of detecting similarities in a curves using an "arc-length space".

**Assembly Level**

The assembly level builds on the structural organizations found in the previous level and extracts organizational information about them. Typically this will involve finding organizations of rectangles, or ribbons, or closed figures. Organizations such as rectangles with aligned symmetry axes are very significant, for example as an indicator of city blocks in aerial images. Little work has been done in this regard.

We should also point out (again) that assembly level organization is unbounded from above. This is mostly a language restriction. In our lexicon, an organization of lines of rectangles into blocks of lines of rectangles (for instance) would be just an additional layer of assembly level organization.

Although they don't use the term, the work of Rao and Nevatia [65] is a typical example of assembly level perceptual organization. They search for 2D ribbons among edge segments. The symmetry of ribbons renders them perceptually signif-
icant and a useful descriptor of shape. The algorithm involves two steps: forming ribbon hypotheses from edge fragments, and their verification using sophisticated geometric reasoning methods. Using the ribbons a scene graph is constructed and ribbon intersections are found by searching for cycles in the graph. They then look for supercycles, or combination of cycles, which correspond to a group of joints in the physical object. Thus, they form compound object hypotheses. Impressive results on real and synthetic data are reported.

Rosin and West [66] extract surfaces of revolution by perceptually grouping ellipses. First, ellipses with parallel axis of symmetry are grouped together. Then the Hough transform groups those ellipses whose centers lie on a straight line. The confidence in the resulting grouping is assessed by looking for line segments or edge pixels that are equidistant from the center line.

2.1.2 3D organization

In 3D organization our input is a 3D world map or a 2-1/2 D sketch obtained by active ranging or by reconstructing the world from a stereo pair. Thus the signal primitives may be range points, or points in 3D space. These are organized into elemental 2D surface patches, surface discontinuities, or clusters of points in 3D space. The work in range image segmentation falls into this category. The surface patches or clusters of points in 3D space are then organized into co-parametric surface patches or cluster groups at the primitive level. Fitting biquadrics, and splines falls under this heading. At this level we will have groups of surface patches which possibly come from the same underlying object surface. At the structural
level we organize the co-parametric surface patches and their junctions into useful surface combinations which are 3D manifestations, such as convex groups, parallel surfaces, and orthogonal surfaces. This helps us to form part hypotheses. These (possible) object parts are then organized based on proximity or other geometric constraints into object hypotheses at the assembly level.

**Signal Level**

This level organizes 3D or 2-1/2 D points, as in range image segmentation. A lot of work is being done in this area; we can mention only a few examples. The common thread among the work is that they organize range points into regions which are constant according to some parameter, for example curvature. Brady *et al.* grouped range points into regions using local curvature properties. Besl and Jain [67] also used curvature to form seed regions and then fitted variable order surface models to grow the regions.

**Primitive Level**

Work at this level involves organizing signal surface patches into primitives. Mirza and Boyer [68], however, chose biquadric patches as their primitives. The segmentation is done using robust statistical techniques. The method has a certain elegance in that parameter estimation, surface discontinuity detection, and the joining of co-parametric surfaces are all done together.

Parvin and Medioni [69] solve the problem of boundary contour completion in range images. Though they do not explicitly cast it as a perceptual organization
problem, they solve an organizational problem with good results. Pointwise detection of surface discontinuities such as occlusion and orientation changes lead to broken and open boundaries. Before any meaningful analysis can be done, we need closed boundary contours. The construction of closed boundaries is formulated as an energy minimization problem involving three terms: one for the binding energy between two compatible curve segments, a second denoting the binding energy between a curve segment and a junction, and a third capturing the binding between two curve segments to form a new vertex. A dynamic network finds a suboptimal solution to the energy minimization problem over the whole image. The network solves a set of differential equations capturing the changing importance of different features as the solution evolves. The idea is to derate contributions from features which do not take part in the organization as the network iterates. This is modeled as a first order system with time constants determining how the weights change. Thus, we can change the importance of, say, long segments over short segments by choosing a time constant proportional to their lengths. The perceptual significance of corners is also appropriately encoded in the system. The system can be said to implement the Gestalt principles of proximity, continuity, and closure. The results on range images are impressive. The enclosed regions are described using various surface descriptors and matched with a range image of the same scene from another view to develop a 3D model.
Structural Level

Work at this level groups surfaces into 3D organizations. Pentland [70] suggests using superquadrics. He argues that they capture the physical regularities of the objects and describes the perceptual organization that people impose on the world. The claim is substantiated with examples and a method to compute the parameters; the idea is certainly promising. Fisher [71] uses an intermediate representation, called surface clusters, to bridge the conceptual distance between the segmented image surface and the objects, to help focus on distinct regions in the image and to gain a computational advantage. The work can be said to span both the structural and the assembly level. The surface clusters are formed by organizing surface patches produced by the primitive level organization. The process involves three steps: locally classifying surface region connectedness, forming primitive surface clusters from connected groups, and merging primitive clusters. The adjacent surfaces are classified as sharing a connecting or segmenting boundary. Connectivity is assumed to hold across convex shape boundaries and not across concave and crack shape boundaries. Connectivity also does not hold across occlusions except in the case of laminar surfaces. To form clusters a graph is built with the surface patches as the nodes and the links denoting connections or boundaries. Connected components in the graph forms our primitive surface clusters. These surface clusters are grouped in two stages to form the final organization. In the first level primitive clusters

---

3Crack boundary between surfaces are coincidental alignments of surfaces or part of a flush contact boundary.

4Laminar surfaces are thin surfaces which may occlude each other but be part of the same surface, e.g. a hollow cylinder.
are grouped into equivalent depth clusters. To this end, he forms a graph with the primitive clusters as the nodes and the links denoting the relation that the primitive clusters have surfaces which are mutually occluding or a pairs of contiguous surfaces which are segmenting. The connected components of this graph form the equivalent depth clusters. In the next level, the equivalent depth clusters are then grouped to form depth merged surfaces in a similar manner using a graph constructed by considering occluding relationships between them and considering all subsets of the nodes of each of the connected components. These surface clusters form object level chunks. Thus, this work also spans assembly level organization. The organization is achieved, as we can see, based on the Gestalt principles of proximity and surface continuity embedded in the relationships of convex, concave, or occluding surface junctions.

Fisher's work can be considered an extension of work by others, including Sugihara [72] and Kak et al. [73]. Fisher's formulation is more clearly Gestaltic. One of its drawbacks, as acknowledged by Fisher [71], is the large number of surface clusters produced. We feel that these can be reduced by considering other Gestalt criteria such as symmetry in the form of parallel or orthogonal surfaces, and closure.

Assembly Level

As mentioned above, Fisher's work [71] also contains elements of assembly level organization. Little else currently appears in this category.
2.1.3 2D + motion

Here we consider perceptual organization in a sequence of 2D images. Signal level organizations will include moving streams of interest points, edges, or regions to create optic flow vectors, or edge point correspondences between frames. Finding correspondences can be looked upon as finding that organization having the least distortion, retaining the most "parallelism" among the image sequences [1]. Primitive level organization here involves organizing the optic flow field into significant flow patterns like swirls, vortices, sinks, sources, and saddles. Geometric edge structures, like ribbons and polygons, which persist over time are also considered. At the structural level we consider organizations of these basic flow patterns to form part hypotheses. The assembly level organizes the above found structures into significant groups to form object hypotheses.

Although the importance of motion in vision was recognized by the psychologist Gibson [74, 75], who coined the term "optic flow," and by numerous computer vision researchers, the role of perceptual organization in image motion analysis has been minimal. This contributes, we feel, to the fragile nature of some algorithms. Perceptual organization will shift the emphasis from a quantitative recovery of structure to a more reliable and fast qualitative recovery of structure. Murray and Buxton [76], provide an excellent study of the state of art in motion. The reader is referred there for details; we will mention work relevant to perceptual organization.

Murray and Buxton [76] recognized the very important role of motion segmentation as a precursor to recovering structure from motion in that it identifies groups
of motion vectors in the image which arise from a single moving entity in the scene. They observe that "...segmentation using motion properties alone proves to be a difficult and, we suggest, largely unsolved problem in computational vision". We feel that the automatic grouping of scene motion vectors can provide a good starting point. Studying patterns in the motion field can definitely reveal various qualitative behaviors. Consider the motion of specularities [77] or of extremal boundaries [78]. Borjesson and Hofsten [79, 80] studied the perception of 3 dot motion patterns. They concluded that translatory motion in depth is evoked by the concurrent (convergent/divergent) relative motion of the points while rotation is induced by parallel relative motion. They also found that under certain conditions the visual system is able to split up a complex motion pattern into simple relative patterns. These studies indicate the importance of organization in visual motion perception.

Signal Level

Signal level organization in the 2D plus time environment involves the computation of optic flow, point wise or feature wise. Point wise flow estimation techniques typically involve constraints like smoothness, which is related to the Gestalt principle of continuity. The final organizations are optic vectors or lines.

Primitive Level

At this level one organizes the optic lines into more meaningful structures. Verri and Poggio [81] argue for a qualitative description of optic flow and suggest descriptions in term of saddles, sinks, sources, and periodic attractors. They draw analogy with
dynamical systems and point out that these features are structurally stable. This implies perceptual significance. A good example of organization in qualitative vision is the optic flow divergence pattern's use in collision avoidance [82]. Divergence patterns are highly symmetric and hence very significant. Carlsson [83] studied the information content of geometric structures in retinal flow fields. The planarity constraint along with tracking made it possible to compute unique values for the direction of motion and surface orientation considering just the symmetry property of the flow field.

Hoffman and Flinchbaugh [84] infer motion by organizing moving dots using the planarity assumption. First, they divide the dots into groups of 2 or 3 elements. The planarity assumption, or the assumption that the dots are moving in a plane, is used to test each group for planar motion. These local groupings are combined to form a globally consistent interpretation.

2.1.4 3D + motion

This is the area where the least amount of work has been done and we are necessarily a bit speculative. The concepts should crystallize as the field matures. Here we look for organization in 3D space plus motion, as in dynamic vision. Organizations found here are highly significant because of the high dimensionality of the space. Our signal level primitives are moving points in 3D space, or changing 2-1/2 D or 3D sketches. These are organized into 3D optic vectors or segmented into surface patches having similar motion. The primitive level organizes them into optic lines, or co-parametric surfaces so as to group surfaces hidden because of occlusions or missing because of
noise. At the structural level we look for surfaces which exhibit regularity in 3D shape and motion and group them. Assembly level organization forms groupings of articulated objects, or groups of objects exhibiting similar motion.

2.2 Computational Techniques

In the previous section we saw how the work in perceptual organization can be classified based on the types of features being organized and the domain of organization. Works which are classified together can be distinguished by computational technique. Detecting organization is a very difficult problem. An exhaustive enumeration of all possible arrangements of perceptual tokens is computationally hopeless and intellectually very unsatisfying. A variety of techniques have been proposed to compute perceptual organization. These differ in their applicability, from the highly specialized and domain dependent to the very general. In this section we outline the formalisms of some of the different approaches; the reader is directed to the original work for more detail. This selection is intended to sample the spectrum of approaches but is not an exhaustive review. For each technique we chose a representative work in 2D structural level organization, where most of the work in perceptual organization has been done. We might add that there is a rich body of work in signal level organization with very similar techniques. For example, the relaxation method is used not only at the structural level but also in signal level organization (e.g. in region segmentation). Since reviews of work at the signal level may be found in various surveys of edge detection, region segmentation, optic flow [85], and range image segmentation [67], we do not consider them here.
2.2.1 Group theory

Palmer [86] proposes a unified framework for perceptual organization, using the concept of local invariance over a group of Euclidean similarity transformations. This is conducive to a parallel implementation and is constructed from many spatial analyzers that are related to each other by similarity transformations. He recognizes that shape constancy is one of the enigmatic problems of perceptual organization. Two figures differing only in position, orientation, size, sense (reflection), or some combination of these are defined to have the same shape. Euclid built shape constancy into his geometry in the form of “similar” figures. Palmer suggests detecting shape constancies using Euclidean similarity group transformations, which can easily be extended to 3D projective transformation groups. The properties that are invariant with respect to these transformations are the set of invariant properties that can be used to recognize objects [87, 88]. Thus angle size, relative length of lines, number of lines, number of angles, closedness, connectedness, and the like are all invariant over the similarity group. The figural goodness or the rank for a particular structure of an organization is measured by its transformational invariance. So good figures have greater transformational invariance (symmetry) than do bad figures. The problem with such a measure is that it lays too much stress on symmetry and as we noted earlier, Pragnanz is more than symmetry.

The laws of Gestalt organization can also be cast as group transformations. The law of proximity is formulated as the grouping preference with the ordering imposed by a metric on a set of local translations in the pattern. Organization by similarity
can also be cast in similar fashion as a group of transformations involving rotation, dilation, reflection, and translation. The elements (tokens) having similar translational distances among themselves are grouped together. This is akin to visualizing the 2D stimulus array as being projected into a higher dimensional space that includes factors other than position such as color, size, orientation, etc. The elements having similar features will be clustered in this space. The concept of transformational distance is extended to cover factors such as continuity, by considering pairs of points. Each pair of points, with an associated position and direction, is projected into a position and orientation space. All the points lying on a continuous contour will tend to be close together in this space. The use of a temporary 3D space (the rho-space) is also suggested by Chen and Hsu [89] to explain line continuation in human visual perception.

Leyton [90] postulates that a highly constrained form of analyzer structure, based on a transformation and an algebraic decomposition of that group, can explain a number of perceptual organization phenomena. In some ways this is similar to that of Palmer, where transformation groups are used to analyze patterns composed of many individual shapes. However, Leyton uses this idea to analyze highly complex shapes, providing a theory of perceptual shape encoding. Given a stimulus set, it is encoded by an analyzer system structured by the transformation group composed of three subgroups of a true deformation, shear, and rotation and the group's algebraic structure constitutes the perceptual organization of the stimulus. The usefulness of this formalism arises from its simplicity and elegance.
2.2.2 Coding Theory

There have been attempts to reduce some of the Gestalt principles to quantitative information-theoretical statements to provide a way to compute them. The percept that can be encoded with the fewest parameters will be preferred. Regular, symmetrical configurations are, from the standpoint of information theory, redundant and so can be encoded economically.

Leeuwenberg [91] proposed some measures based on coding theory to measure structural information and to predict the saliency and the similarity between patterns. Figural goodness is reduced to figural simplicity, implying that perceptual organization is guided by a 'minimal coding principle'. One of the arguments against information-theoretic approaches in perceptual organization is that a universal set of codes which works on all figures has not been found and thus the results obtained are different. Leeuwenberg demonstrates impressively high correlation between the subjective complexity and the calculated structural information content of patterns according to his scheme. Even the subparts of the codes of the total figure appear to correspond with subparts that the subject conceives of in the total figure. He uses the coding of the figure to determine its distinctive salience and contrast salience. Distinctive salience is the contrast between the features of a pattern with respect to those of the background pattern and contrast salience is the difference between alternative interpretations of the same pattern. Based on his codes he also measures the similarity between two patterns. These measures are taken as indicative of structural "goodness" or Pragnanz and can be used to rank the perceptual struc-
tures hypothesized in machine vision. Complex patterns are described as composed of simple patterns, thus implying a hierarchical nature of organization. The lower most level is usually angles, intersections and lines/curves. Pomerantz [92] deduces from a set of controlled experiments that configural features such as angles and intersections may be the primitive units of visual pattern analysis.

Barnard [93] suggests a computational solution to the problem of interpreting an image as a projection of rectangular forms. The Gestalt justification for the search of rectangular forms is that they can be described using an orthogonal basis. An orthogonal basis can be specified as two vectors and the handedness and is symmetrical, whereas complete specification is needed in the general case.

2.2.3 Rule Based

Lawton and McConnell [94] suggest computing qualitatively significant perceptual groupings by the repeated application of rules. They suggest a format for describing the rules, image structures, and relationships to which they apply. A hierarchical organizational structure is suggested with increasing neighborhoods. To control the combinatorics, only "interesting" structures are passed to the next higher level. Interestingness is evaluated by considering size, contrast, extent, type of similarity, and the number of grouping the structure is associated with. They suggest the concept of a label plane associated with different structures. A label plane is an abstract image in register with a sensor surface where each pixel consists of a list of pointers to all objects which occur at and occupy that pixel. As new groups are formed they deposit their points in the label plane according to their spatial
arrangement. The algorithm starts with an initial arrangement of groups on the label plane which may be extracted from edge detection, motion and stereo, or shape description processes. Similarity relationships between objects are created, based on local or global features and relations. These are sorted according to interestingness. Grouping rules are applied to the most interesting to produce new groups, or merge groups. A nice feature of this algorithm is that they do not use a single measure for shape significance, instead they use various measures such as group membership scores, number of attributes of the group, time since the group was created and so on to prune the groups. This framework is very useful in building a rule based perceptual organization system.

Nazif and Levine [95] built an expert system for low level image segmentation. This enabled them to bridge gaps and to deal with the fragmentation of edge features from noise. To this end they stored knowledge about low level processes and a set of control rules. Metarules were used to infer the order in which the knowledge rules were applied and attention rules determined the processing path. The processing strategy is controlled by still higher level rules. Although they do not explicitly use the term, their rules incorporate the essential features of perceptual organization such as collinearity and parallelism. The method has the disadvantage of requiring that rules be specified for all sorts of special cases, which is generally tedious and error prone. Such systems do not generalize well; they tend to be very much domain dependent and representation specific.
2.2.4 Frequency Methods

Frequency domain methods in vision are primarily motivated by neurophysiological evidence for frequency selective cortical cells. The applicability of this method has been restricted to texture groups. Perceptual organization in textured images can be looked upon as low pass filtering [31]. However, Janez [29] has evidence for high frequency components in grouping. One of the disadvantages of frequency based methods is that they are global in nature, contrary to the fact that visual discrimination is a more local process. Besides, as Reed and Wechsler [28] point out, Gestalt grouping depends on both shape similarity (a spatial property) and organization (a spectral property). This suggests joint spatial and spatial-frequency characterizations of the 2D signal. These are methods that indicate frequency content in localized spatial regions and include techniques like the spectrogram, difference of Gaussian representations, Gabor representations, and Wigner distributions. Reed and Wechsler [28] argued that the Wigner distribution has superior resolution capabilities.

Reed and Wechsler [28] proposed a four stage algorithm. The first stage is a pre-processing antialiasing step. Next is the computation of the Wigner distribution followed by a stage of data reduction in which we select the frequency content at the frequency containing maximum energy for each pixel. The last stage is low pass filtering implemented as relaxation followed by grouping of similar valued pixels. Results on textured images were demonstrated. They also show that the results are in conformity with some of the Gestalt principles. Beck et al. demonstrate the cor-
relation between human segmentation of tripartite textured images and the outputs of a bank of 2D Gabor filters, which are direction sensitive Fourier transforms.

2.2.5 Energy/Probabilistic Optimization Methods

Sometimes perceptual organization is formulated as an energy minimization problem. This has origins in the early Gestalt days. One the mechanisms suggested to explain perceptual organization was based on an interacting electric field settling to an optimal state. The idea drew inspiration from the "soap bubble" solution of the shape. The idea of an electric field failed to gain much acceptance because of a lack of neurophysiological support. But mechanisms based on the minimization principle have proven very successful in computing perceptual organization in computer vision, where one is not restricted by neurophysiology. The presence of an underlying optimization process in perceptual organization is particularly evident in the Marroquin pattern (see Fig. 4) where competing organizations are observed. Typically, one associates energies to binary interactions of tokens to capture their compatibility in terms of various Gestalt criteria and tokens are linked so that the total energy is minimized. The method has the advantage of computing global percepts from local interactions. It may be pointed out that the method can be used at each level of a hierarchical formulation.

We consider energy and probabilistic formulations together because of their equivalence in some cases, e.g. Markov Random Fields and Gibbs' energy distributions [96]. In a probabilistic formulation local interactions are modeled as low order conditional probabilities or Markov Random Fields. The optimization of the
Figure 4: Note the competing organizations that are observed, analogous to an optimization problem stuck in a limit cycle.

...total energy or probability can be done in a number of ways. We review the best known methods next. Some of the less investigated techniques like genetic algorithms [97] are not considered here.

Neural Networks

Grossberg and Mingolla [98] hypothesize a system of neural dynamics to explain a wide variety of perceptual grouping and segmentation phenomena. Their hypothetical system has three components: a boundary contour system, a feature contour system, and an object recognition system. The boundary contour system is responsible for pre-attentive perceptual grouping to generate a coherent boundary structure. It consists of two subsystems: one sensitive to the image contrast, which forms an in-
put to the second, a competitive-cooperative network encoding spatially short range competitive and long range cooperative interactions between edge elements. Feedback between the competitive and co-operative stages synthesizes a global context sensitive segmentation from among the many possible groupings of local features.

Neural networks can be used to solve parallel optimization problems. Details regarding the formulation of the method and its convergence can be found in the neural network literature [99, 100, 101, 102]. The method is used in computer vision as follows. A preprocessing stage first extracts primitive symbolic tokens. The interactions (support or conflict) among them are modeled using energy values and the problem of finding a set of stable perceptual groups is posed as an energy minimization problem. The neural network is used to find a local energy minimum. To illustrate the method we concentrate on the excellent work of Mohan and Nevatia [43].

As discussed earlier, Mohan and Nevatia find organizations such as lines, U structures, and rectangles by constrained search. This detection stage generally produces a large number of supporting and conflicting groupings (or collations). The neural network helps to select the best subset of these. Let us call the original set of organizations, the initial set of hypotheses, \( H = \{h_i\} \). To each hypothesis or organization we assign a confidence value, \( 0 \leq V(h_i) \leq 1 \), which changes with iteration and settles to a final value. The binary interaction between organizations, or hypotheses, is modeled using the energy function, \( T_{ij}V_iV_j \). \( T_{ij} \) models the type and strength of the interaction; \( T_{ij} > 0 \) for supporting interaction and \( T_{ij} < 0 \) for
conflicting interaction. The "input" to each hypothesis, denoted by \( I(h_i) = I_i \), is the sum of the evidence for that organization based on photometric and geometric information. The "energy" of a singleton hypothesis is modeled as \( I_i V_i \), measuring the contribution to the total evidence from hypothesis \( h_i \). Finding the best set of hypotheses is now formulated as the assignment of confidence values, \( V_i \), which maximizes

\[
\sum_i \sum_j T_{ij} V_i V_j + \sum_i I_i V_i \tag{2.1}
\]

The double sum represents the total energy of interaction and the second is the sum of the energies in the individual hypotheses. This is solved using the Hopfield network. Each of the organizations, or hypotheses, is a node or "neuron". The relationships between the organizations define the link weights between the nodes. The link weights depend on the type of organizations connected and were determined empirically. To avoid cases where an organization is selected because it is the only grouping of its component edges, a "winner take all" type of network is superimposed on the Hopfield net. For more detail the reader is referred to [43]. In [44] they use the same formalism to find the best groupings of ribbons.

Parvin and Medioni [69] use neural networks to find boundary groupings in range images. The interaction variables are corners, junctions, and segments. The constraints among them are captured using link weights, as before. However, they use another nice aspect of neural networks: the time constant. In a neural network, each node can be shown to solve a first order differential equation with an associated
time constant and the node response can be modeled as a solution to a differential equation. The previous work used the same time constant for all organizations. However, Parvin and Medioni use a different value for each organization depending on the length of the segments involved. Longer segments compete longer in the network, while shorter segments settle to equilibrium faster. They achieve impressive results on range images.

Relaxation Labeling

As discussed earlier, Ahuja and Tuceryan [26] propose an efficient algorithm to organize dot patterns. First the geometric structure is represented in terms of the Voronoi neighborhoods of the dot patterns. Geometric properties such as compactness, area, elongation, eccentricity, distance measure, squeezedness (density), and the Gabriel measure for the neighborliness (dependent on the number of neighbors and how far they are) of a point are calculated for the Voronoi polygons. The goal is to label each as: interior of a region, border of a region, curve, or isolated dot. This is achieved through the cooperation of a number of “expert” modules over three levels. At the first level, the interior identification, border identification, and curve identification modules independently label the dots as being in the interior, on the border, or belonging to a curve, respectively. At the second level we have the interior correction and border correction modules which use the outputs of the interior and border modules to correct the labels based on global consistencies. In the third level a module combines the interior and border results. Another module
labels the curves based on the curve identification module and border correction module results to form the final interior/border combination results.

Each module uses probabilistic relaxation [103] for inference. The initial probability vector assignments for the dots are computed based on the local compatibilities and the initial labeling is done based on the Voronoi polygon attributes. The interior identification is formulated as a relaxation process on the labels INTERIOR and NONINTERIOR. Similarly, the border identification module labels dots as BORDER and NONBORDER and so on. The correction processes uses the labels CHANGE or NOCHANGE. The reader is referred to [26] for the actual assignments of compatibilities in the relaxation process. The probability updating formula is the one suggested in [103]. Impressive results are shown on various dot patterns. The algorithm does not use a large number of thresholds and the authors suggest explicit ways of calculating them.

Regularization

Trytten and Tuceryan [104] describe an algorithm based on energy minimization to perform curvilinear grouping of incomplete edge contours or edgels. This is a very important task in intermediate level vision because most high level algorithms expect complete, closed contours whereas low level algorithms usually give fragmented edge contours or edgels. The algorithm completes and smooths boundaries and detects discontinuities in curvature. A Delaunay graph of all the edge points is first created and only that subgraph corresponding to the edgel end points is kept.
This defines a neighborhood for the endpoints. This is used to hypothesize about possible extensions of an edgel. These are evaluated using an energy term encoding local curve smoothness and discontinuity. The edgels are sorted according to their length and we start with the largest. The edge contour under consideration is grown like a crystal, by choosing the neighboring edgel, based on the Delauny subgraph, which decreases the total fit energy the most. The process continues until it cannot decrease the energy any further. A new edgel is then considered and the process of crystal growing repeats. The algorithm has reasonably good performance but, as the authors point out, has problems in detecting obtuse angles.

**Simulated Annealing**

McCafferty [42], as seen in Section 2.1.1, formulates the grouping problem in perceptual organization as an energy minimization problem, which he then solves using simulated annealing. His energy formulation allows for the interaction of a higher level module. The ability to identify objects on the basis of a limited set of features is evidence of high level influences [105]. The influence of top down sources, such as from the conscious conceptual knowledge, on the perceptual process is recognized by Perkins [106]. The perceiver is thought to make use of the Gestalt criteria of rectangularity, symmetry and parallelism [106], using a constraint handling process with relaxation to a minimum state\(^5\), or a production system that triggers rules when some salient features are detected [107]. Freuder [108] also acknowledges the

\(^{5}\)The dynamic nature of perception is evident from the Marroquin pattern [16]. Some believe it to be the result of multiple minimum “energy” states.
role played by knowledge in vision inferring of features missing because of noise or occlusion.

2.2.6 Graph theory

The use of graphs to extract Gestalt percepts was proposed by Zahn [109]. He used a family of graph theoretical techniques based on the minimal spanning tree to describe several kinds of dot clusters. Though the results were demonstrated on 2D dot clusters, the method is applicable for higher dimensions or general metric spaces. The minimal spanning tree captures the local neighborhood of a point. Different neighborhoods are possible for each point: $k$ nearest neighbors, absolute distance neighbors, and Voronoi neighbors. Ahuja [110] offers a good review of the various neighborhood definitions.

Graphical representations are particularly suited for perceptual organization. The explicit nature of the stored information allows for the ready extraction of higher order features. In this work, we use such representations to build an hierarchical description of the scene. We create graphs for the Gestalt criteria: proximity, continuity, common region, and closure using voting methods. Extracting various graph structures such as cliques, cycles, paths, and trees gives us continuous lines, closed boundaries, or continuous curves. These latter organizations are integrated using a class of geometric knowledge base called Perceptual Inference Networks. The PIN is based on Bayesian Networks and encodes information about geometric organizations. It hypothesizes organizations in the presence of uncertain information and integrates multiple sources of information. More on this later.
Huttenlocher and Wayner [111] use graph representations to extract convex edge groupings. Convexity, as pointed out earlier, has been found by psychologists to have very high perceptual significance and hence is a good form of organization to look for. They start with linear edge segments and triangulate to get a local neighborhood definition of the end points. In this respect they are similar to Trytten and Tuceryan [104]. However, the latter work with curvilinear segments and not just linear segments. From the local neighborhood graph a local convexity graph, with the segments as nodes, is constructed by keeping the best convex pairs \(^6\) of neighbors using a cost function based on distance and convexity angle. This ensures, as the authors show, that each vertex can belong to at most two cycles. A path in the local convexity graph will correspond to convex chains, and cycles give convex polygon hypotheses. The work demonstrates the power of graphical representations in encoding spatial relationships and extracting organizations. In [112] Wayner discusses interfacing the above convex grouping module with a model based matching module. The grouping module provides the model base with a manageable number of hypotheses to consider.

To summarize, we reviewed the work in perceptual organization in machine vision and suggested a framework to help direct future research and to provide a consistent lexicon. The role of organization in vision is indisputable and many artificial systems are starting to use it, sometimes without recognizing it as such. We hope that this would generate new work in this area and help to focus future work.

\(^6\)Convexity among segments may be anticlockwise or clockwise.
"By thinking of our brains as informational processing systems we can gradually dispel the fog and pick our way across the great divide discovering how it might be that our brains produce all the phenomenon."

Daniel Dennett

Perceptual organization provides significant computational leverage and can do so over several layers of abstraction. The sophistication of a vision system, on the whole, lies largely in the sophistication of its perceptual organization processes. In biological systems, visual capability is developed to a degree appropriate to the ambulatory capabilities of the organism, i.e., the degree to which it can use the information to acquire food, elude danger, or otherwise restructure its environment. The performance of a machine vision system is coupled to the chosen problem domain, which is analogous. It is the recognition, development, and exploitation of perceptual organization concepts, models, paradigms, and computational techniques that brings efficiency to machine vision systems. Perceptual organization allows us to assign computational resources intelligently, which is also important in biological systems because of the relative expense of neural tissue from an evolutionary
standpoint. As we shall see, perceptual organization in computer vision uses computational resources effectively to extract organizations from which features are hypothesized, instead of the expensive alternative of directly applying feature detectors over the entire image.

We recognize the following three characteristics of perceptual organization. First, that it is essentially symbolic in nature [113] with a top down and a bottom up component. Secondly, this organization should proceed in a hierarchical fashion, with coherent global structures gradually emerging from very local features. Thirdly, this process produces an information reduction in the encoding of a given structure, because of the hierarchical nature of the abstraction. This sort of hierarchical approach was also followed in [36, 37]. However, in recognizing the problem of computational complexity they considered a large number of hierarchies. Our computational model enables us to overcome the computational complexity issue and allows us to achieve the same degree of structural organization with fewer hierarchies.

A typical hierarchy of organizations is shown in Fig. 5. In the zeroth level of the hierarchy are constant curvature edge tokens extracted using an edge detector such as in [21] and segmented using the algorithm described in [34]. The choice of circular segments is not arbitrary; a large class of curves can be quite accurately described by circular arc primitives. In aerial images, this is particularly appropriate because cultural features tend to be approximately straight (e.g., buildings, runways) or piecewise circular (e.g., roads, railways). In [33], Lowe considers the segmentation of curves into constant curvature segments and also suggests their usefulness in ex-
Figure 5: Hierarchical Organization to analyze curvilinear structures
tracting perceptual structures. The next level of the hierarchy consists of continuous lines, ribbons, mergings, strands, and closed figures. These organizations are further organized in the next level of the hierarchy into continuous ribbons, parallel ribbons, closed ribbons and strands of ribbons. The hierarchy can be continued to many levels. For example, the next highest level may consider organizations of polygons, ellipses, circles, and parallel ribbons. Our system is implemented to Level 2 as shown in the figure. Typical structures are drawn in the figure as examples.

Tokens at each level are described by a list of photometric and geometric attributes and are associated with virtual line or dot tokens. For constant curvature segments the virtual token is the same as the edge element; for complex structures, such as ribbons and rectangles, the virtual tokens are the axes of symmetry. Circularly symmetric structures such as circles are represented as dots with the attributes inherited from the circle. Thus the organization of tokens at each level of the hierarchy can be looked upon as an organization of (virtual or real) line and dot tokens. However, the virtual tokens have more attributes associated with them than do simple line tokens. Since at each level of the hierarchy we represent the organization in terms of virtual line or dot tokens with inherited attributes, we can use an appropriate algorithm recursively to construct the hierarchy.

Each level of the hierarchy is constructed using voting methods, graph operations, and knowledge based reasoning in a new extension of the Bayesian network we call the Perceptual Inference Network (see Fig. 6). Regularities in the image tokens based on the Gestaltic principles of proximity, similarity, smooth continuity, and
Tokens from low level vision e.g. edges

Voting Methods implementing Gestalt Principles

Graph Theoretic Operations on the associations found by voting

Hierarchical Description of the scene

Perceptual Inference Using Bayesian Networks.

Vision Algorithms
1. Low level
2. Shape from X
   (a) stereo
   (b) contour
   (c) texture
   (d) shading

Figure 6: Perceptual Organization System Block Diagram

closure are detected by the voting methods which effect a search procedure among the image tokens and are shown to be superior to conventional techniques. The Gestaltic associations among tokens are represented by a set of Gestalt graphs, which lets us apply sophisticated graph theoretic techniques. The knowledge base, encoded in the form of a Perceptual Inference Network, helps us to go beyond the data to predict undetected features and to integrate multiple sources of information.
For example, a rectangle has the property that it is closed, it is made up of two sets of parallel lines, and it has right angled corners. These properties, detected in a preattentive phase, provide cues for the presence of a rectangle and the network then enables us to hypothesize its presence, even in the absence of some of the above features, and to generate expectations for such. The hypotheses of the PIN are confirmed or rejected by selectively using a set of computational resources.

Analogous to theories in human vision, our strategy divides broadly into two parts: detecting regularities and similarities in the tokens (preattentive vision) and reasoning, based on a knowledge base built from past experience, to enable one to go beyond the information provided (attentive vision)\(^1\). The voting method provides organizations based on Gestalt principles and the network reasons on those organizations to extract geometric features. The two steps of voting and evidential reasoning are repeated. The type of organizations considered at each level gets more and more complicated. After two or three levels we can consider generic part descriptions and so the role of voting gradually diminishes, replaced by more reasoning on sets of features. Thus, there is a gradual transition from intermediate to high level vision processes.

In the next chapter we present the preattentive module consisting of the graph theoretic operations and the voting operations. We develop the attentive parts: the PIN, structuring of PINs, and resources management in Chapter VI, VII, and VIII respectively. In our discussion, at various stages we present results on one aerial image.

\(^1\)We do not claim that our system models the human perceptual system.
CHAPTER IV

Preattentive Algorithm: Gestalt Graphs and Voting Methods

"...others refuse to do so, because in their opinion only quantitative observations are scientific. ...Being in love with figures and curves, they will keep away from the true source of new ideas and new problems in a youthful science; a broad outlook upon its subject matter."

Köhler

The preattentive perceptual organization module tries to extract organizational cues from the features at the previous level in a bottom up fashion without the aid of a knowledge base. It consists of a search component implemented using voting methods and a structure extraction component based on graph theory. An important aspect of our algorithm is the representation of the associations at every stage in the form of Gestalt graph; each symbolic token is represented as a node in an attributed graph. This enables us to extract various types of structures by standardized graph analysis algorithms. Graph structures, such as cliques, connected components and cycles, represent different types of perceptual structures, which can therefore be easily extracted. Our results give concreteness to these ideas. We use the efficient
and generic voting method to search for the associations represented in the Gestalt graphs. The same code finds associations among various forms of tokens over the full range of levels (from points to high order structures).

We present the concept of a Gestalt graph in Section 4.1. Section 4.2 presents the voting method used to construct these graphs. The rationale for the choice of the organizational parameters is presented in Section 4.3. Section 4.4 display results on a test image.

4.1 The Gestalt Graphs

We use various graph theoretic operations to extract the preattentive organization cues from the primitives at the previous level. The Gestalt laws of organization, proximity, continuity, similarity, closure, symmetry and the new properties [13] of common region and connectedness, suggest a few basic forms, which we will use throughout the hierarchy to describe the current organization. For example, a ribbon is a symmetrical arrangement of tokens with a continuous portion of the segments proximal and sharing a common region.

To compute the Gestalt we consider five basic graphs: proximity, end point nearness, continuity, similarity, and common region. We call these the Gestalt graphs. The nodes of each graph represent the tokens from the previous level and the arcs (which may be weighted) denote the existence of the various relationships. The proximity graph is a graph whose links join nodes representing tokens having points close together with similar orientation. The nodes of the end point nearness graph represent the endpoints of the edge tokens and the links join endpoints that are
close or that are parts of the same contour. The links of the continuity graph connect nodes, representing the edge tokens, which are possible continuations based on orientation and intercept, or on curvature and center.

The similarity graph encodes the fact that two tokens share common photometric or geometric properties. The common region graph is graph of tokens sharing a common region. Because similarity criteria are weak at the bottom level and because common region criteria are weak at higher levels, we use the common region graph only for the bottom level and similarity graph at all subsequent levels.

4.1.1 First Level Organization

Using various graph theoretic operations, we compute from the Gestalt graphs more complex organizational hypotheses. Fig. 7 depicts how the various organizations are calculated for the first level of the organizational hierarchy. We outline the procedures here:

- **Continuous Segments**: We "AND" the continuity, end point nearness, and common region graph to form a graph of tokens representing the relationship that the tokens shares a common region, have similar orientation or curvature and center, and their endpoints are close. To "AND" two graphs we AND the adjacency matrices of the two graphs. We detect connected components and form complete subgraphs using the nodes in each component. The resultant graph with complete subgraphs is "AND"ed with the continuity graph. Cliques in the resulting graph form continuous line hypotheses, such as continuous straight lines, co-circular arcs, and circles.
Figure 7: Algorithm executed to construct first level of the hierarchy.

- **Closed Figures**: The links of the end point nearness graph join line segment endpoints. Each link is quantified by the Euclidean distance between the endpoints, if they correspond to endpoints of different segments, or a negative number if they are from the same segment. The minimal spanning tree of this graph defines a neighborhood structure [109] and the corresponding fundamental cycles form our closed figure hypotheses.
• **Strands:** The shortest paths between nodes of valency one in the endpoint nearness graph, i.e. free endpoints, form our strand hypotheses.

• **Parallels:** The proximity and the common region graphs are “ANDed” to construct a graph from which parallel structures such as ribbons and mergings are extracted. The links of the resulting graph join tokens having portions of segments which are close together, parallel, and which share a common region.

• **Junctions:** We detect junctions by considering the endpoint nearness graph with the links corresponding to the edge segments deleted. The connected components of the remaining graph form our junction hypotheses.

• **Intersections:** Intersections among segments are detected by writing out the line segments on an array and identifying crossings. This particularly simple realization of the voting method (to be discussed next). Intersections of the virtual lines (symmetry axes) representing a ribbon can give us various quadrilateral hypotheses. The junctions and the intersections are do not play a major part in our present implementation of the perceptual organization system. But, future work can make efficient use of these forms of organizations too.

### 4.1.2 Second Level Algorithm

The above discussed the graph theoretic preattentive algorithms to build the first level of the hierarchy. Since tokens at all levels are represented by lines or dots, the basic structure of the algorithm is preserved at other levels and involves the
same basic set of codes. The line tokens for the first level are the linear edge segments. The organizations at other levels are abstracted as linear and point tokens using the symmetry axes and the centroid, respectively. Each linear or point token is associated with attributes of the organization it represents. Fig 8 shows the algorithm for the second level. The diagram is interpreted in exactly the same way as the first level.

Note that we use the same set of Gestalt graphs and graph theoretic algorithms except for some minor changes in the interconnections. Unlike in the first level where we use the end point nearness graph to find continuous tokens, in the second level we
use the proximity graph. At the second level we need to relax the spatial separation requirement for continuity so as to group parallel lines which are far apart. The separation considered in the proximity graph is more relaxed than that for the end point nearness graph.

Although at present we detect just one type of point organization, future work might involve others. We use the end point nearness graph and the attribute similarity graph (Fig. 8) to search for groups of point tokens which are near and have similar characteristics.

Thus, we see that given the core Gestalt graphs we can extract more complex organization cues using various graph enumeration algorithms. The Gestalt graphs can be constructed using a variety of procedures. Our problem is to find all pairs of tokens satisfying some compatibility relation, $\mathcal{R}$, defining the links of the graph. Mathematically, we have (similar to that used by [37]):

$$\mathcal{R} = \{(t_i, t_j) | (a_1(t_i, t_j) < A_1) \land (a_2(t_i, t_j) < A_2) \land \cdots \land (a_n(t_i, t_j) < A_n)\} \quad (4.1)$$

Tokens $t_i$ and $t_j$ are compatible if the difference in attributes, $a_k(t_i, t_j)$ is less than some specified $A_k$, $\forall k$. The attributes depend on the token type under consideration. Instead of searching for pairwise compatibility by considering all possible pairs we propose a computationally efficient voting scheme which identifies tokens "to be linked" using all token points without brute force search.
4.2 Voting Scheme

Our problem is to find all pairs of tokens satisfying some compatibility relation, $\mathcal{R}$. A brute force search over all token point pairs is obviously impractical. Instead, we build global consensus over the image using a voting scheme. Consider a four dimensional space of $x$ coordinate, $y$ coordinate, slope ($\theta$), and curvature ($\kappa$) in which to do the organization. In this space each point on an edge token votes for all points in the parameter space satisfying the compatibility relation, $\mathcal{R}$. Each vote is tagged with the edge token number. If points from two edge tokens vote for the same location in parameter space, then those two tokens will be, in some way, associated. The exact form of association is determined later. Fig. 9 depicts a typical 3D voting space. In this example tokens A and B are associated while C is isolated. Although drawn as parallelepipeds, the exact voting region shape depends on the nature of the compatibility function. Intersecting voting regions identify associations by indicating a non-empty set of points compatible with more than one token.

The voting space must be constructed carefully. First, we have to consider quantization. Suppose we quantize the $x$ and $y$ coordinates into pixel units, and slope ($\theta$) in degrees. In the digital domain curvature varies from 0 to 1, in units of pixels. (We are not concerned with subpixel accuracy.) The number of quantization levels depends on the minimum difference in curvature which is significant. For the present discussion, we choose ten levels arbitrarily.
Figure 9: Typical parameter space configuration after voting has been done

For this very fine quantization, the parameter space requires approximately $1800NK$ cells to represent it, where $NK$ is the size of the image. This is approximately 400Mega cells for a $512 \times 512$ image! Another disadvantage with this sort of strategy is that it requires multiple memory accesses for each point because each point must vote for all the points in the parameter space which satisfy the compatibility relation with it. This increases the execution time. Choosing a coarser quantization reduces the memory access but there is a tradeoff in the form of those compatibility relations that can be handled without post processing. To reduce the memory size and access requirement, we use a coarser quantization scheme, but with an enhancement, as we show next. This allows us to vote faster using less memory.
4.2.1 Offset Quantization

Coarser quantization reduces the memory requirement; we quantize the space so that each token has just one memory access. Each point votes for the bin in which its parameter lies. The program then scans the parameter space for bins receiving votes from multiple tokens, thus identifying associations. However, we have to be careful of parameter values falling on bin boundaries; noise can perturb the vote in either direction. This is illustrated for a two dimensional case in Fig. 10(a). Let the maximum tolerable difference in each parameter value be known as its constraint distance. Here the dimensions $\alpha_1$ and $\alpha_2$ have been quantized into multiples of $b$ and $a$, suggesting that these are their respective constraint distances. Note, however, that although points $A$ and $B$ lie within the constraint distance in each dimension, they do not occupy a common bin and, according to this simplistic approach, would not be associated.

To overcome this problem, we can quantize each dimension twice as shown in Fig 10(b) for one dimension. In this manner, two features lying within the constraint distance will are likely to occupy a common bin in at least one of the quantizers. Crosses $B$ and $C$ in Fig. 10 fall in different bins according to the first quantizer but fall together in the second. This sort of offset quantization scheme is used in one dimension by Burns et al. [22] and Boyer et al. [114] to form regions of constant gradient direction in finding straight lines. We generalize this idea to $N$ dimensions and to multiple order offset quantization, using more than two interleaved quantizers.

Even with offset quantization it is possible that points within a distance, $d$, of
Figure 10: (a) A quantized parameter space illustrating the effects of quantization. (b) Overlapping one dimensional quantized voting space

one another are placed into different bins in both quantizers. For example, consider points $A$ and $C$ in the diagram. However, we observe that points within a distance $\frac{d}{2}$ will always fall into a common bin in one of the quantizers. Thus, for a 1D overlapping quantization scheme, the bins will surely group all points less than $d/2$ units apart. This suggests quantizing to twice the constraint distance, but it will also group some points up to twice the constraint distance apart. However, it is easy to search this now very restricted space to delete unwanted associations. This is not as expensive as it appears, because our search space is very small compared to the original space comprising all possible combinations of the tokens. Furthermore, we can merge this step with the search step needed for general compatibility relations, which is discussed later.
The simplest scheme of offset quantization, as described above, has one quantizer shifted by \( d/2 \) with respect to the other (Fig. 10(b)) for uniform quantizers of binwidth \( d \). Extending this notion, we can construct \( n \) quantizers with each quantizer shifted by \( \frac{d}{n} \) with respect to the adjacent one as illustrated in Fig. 11. We call this scheme \( n^{th} \) order offset quantization. For this scheme we advance the following proposition:

**Proposition:** Let \( \{t_1, t_2, ..., t_n\} \) be a set of tokens which are to be grouped together according to a given compatibility relation. Let the compatibility relation be of the form: \( \{(t_i, t_j)\mid |\delta(t_i) - \delta(t_j)| < d\} \), where \( \delta(\cdot) \) is an attribute function over the space of tokens. Also let a set of \( n \) uniform quantizers, with unit of quantization \( d \) (as in Fig. 11) be defined. Then, there may exist a pair of tokens satisfying the compatibility relation, with \( d > |\delta(t_i) - \delta(t_j)| > (1 - \frac{1}{n})d \), but not grouped together.

**Argument:** Tokens are grouped if and only if they fall into at least one common bin, so to prove our proposition we have to show only that there may be a pair of tokens satisfying the compatibility relation which do not vote for the same bin in
any of the quantizers. Suppose we have two tokens, \( t_1 \) and \( t_2 \) with \( \delta(t_1) = x_1 \) and \( \delta(t_2) = x_2 \). The token \( t_1 \) will vote for bins of indices \( b_k^1 \) given by:

\[
b_k^1 = \left\lfloor \left( x_1 + \frac{k\frac{d}{n}}{d} \right) \right\rfloor (4.2)
\]

where \( k = 0, 1, 2, 3, ..., n - 1 \) for quantizers \( q_0, q_1, ..., q_{n-1} \) respectively. \( \lfloor x \rfloor \), the floor of \( x \), denotes the largest integer smaller than \( x \). Note that the \( i^{th} \) bin of quantizer \( q_k \) extends from \( (di - k\frac{d}{n}) \) to \( (d(i + 1) - k\frac{d}{n}) \). We have similar expressions for \( t_2 \).

Now, let \( x_1 = d + \epsilon_1 \frac{d}{n} \) and \( x_2 = d - \epsilon_2 \frac{d}{n} \), where \( 0 < \epsilon_1, \epsilon_2 < 1 \) and \( ((\epsilon_1 + \epsilon_2) < 1) \) so that \( (1 - \frac{1}{n})d < (x_1 - x_2) < d \). Then,

\[
\begin{align*}
b_k^1 &= \lfloor \left( (d + \epsilon_1 \frac{d}{n}) + k\frac{d}{n} \right)/d \rfloor = \lfloor (1 + \frac{\epsilon_1 + k}{n}) \rfloor = 1 \\
b_k^2 &= \lfloor \left( (d - \epsilon_2 \frac{d}{n}) + k\frac{d}{n} \right)/d \rfloor = \lfloor (1 + k - \epsilon_2)/n \rfloor = 0 \\
& \quad (\forall k = 0, 1, ..., n - 1) \text{ and } (0 < \epsilon_1, \epsilon_2 < 1) (4.3)
\end{align*}
\]

Thus we see that a pair of tokens such as these will vote for different bins in every quantizer, even though their attributes differ by less than \( d \). Actually, we can be sure that tokens will be grouped together only if their attributes differ by less than \( (1 - \frac{1}{n})d \), which is the maximum overlap among the quantizers. \( \Box \)

So for \( n \)-level uniform quantization with binwidth \( d \) not all the associations satisfying the compatibility relation with tolerance \( d \) will be grouped together. That is, there may be valid associations with attribute difference less than \( d \) that are not grouped. But we are sure to pick up all associations with attribute difference less than \( (1 - \frac{1}{n})d \). So we could set this to the constraint distance and solve for an adjusted binwidth. This will “over-associate” tokens, and we then delete the improper associations in a step we call “screening.” In general, \( n \) is chosen such that \( \frac{d}{n} \) is the least significant unit of measurement for that particular dimension.
Figure 12: Overlapping 2D voting spaces.

For an \( N \)-dimensional voting space, each dimension may be quantized differently, according to the needs of the problem. However, for our purposes we use a two level quantization (i.e. \( n = 2 \)) in each dimension. Since each dimension is quantized twice we need \( 2^N \) voting spaces, or distinct sets of accumulators for an \( N \) dimensional parameter space\(^1\).

To illustrate, consider a voting region in two dimensions quantized as shown in Fig. 12. For a two dimensional parameter space we need 4 voting spaces, as shown in the figure. The dashed lines show the relative quantizer offsets. Observe that although points \( A, B \) and \( C \) are in separate bins in voting space 1, they lie in the

\(^1\)All the possible combinations of quantization schemes is \( 2^N \), and this is now the number of memory accesses per token.
same bin in voting space 4. Thus with a coarser parameter space, compared to the finer quantization considered earlier, we need four accumulator arrays but the memory requirement reduces to $\frac{1600}{D, \Theta, \Psi} M$ for a $512 \times 512$ image, where $D, \Theta, \Psi$ are the number of quantizers along the spatial, slope, and curvature dimensions. Typically this is around 32K cells as compared to 400M cells for the finer quantization scheme.

The memory required by this scheme increases exponentially with parameter space dimension. It can be modified, however, to produce only a linear rate of increase. Suppose the memory of the system restricts the parameter space dimension to, say, $K$. Then, we vote in a subspace of $K$ dimensions first, followed by the next $K$, and so on, until cover the complete parameter space. The associations from each subspace are retained as the adjacency matrix of an ordinary, non-directed graph for which the nodes represent tokens and the arcs represent the binary relation of association within the subspace. These binary matrices are simply "AND"ed to compute those associations supported by all dimensions of the parameter space. We will illustrate with an example.

Assume that there are four dimensions: $x$ and $y$ coordinate, slope, and curvature. Let the first subspace have just the $x$ and $y$ coordinates as dimensions. Thus, this voting will group those edge tokens which are physically close in the image. The other space has slope and curvature as its dimensions. Edge tokens which have sections of similar slope and curvature will be associated in this subspace. Now, if we "AND" the pairs of edge tokens generated by these two voting processes we get pairs of tokens which satisfy the complete compatibility relation. Of course, we could
build four one dimensional voting spaces, but we consume time fusing the results and need more memory to store the adjacency matrices. The reduced memory per space may be offset by that of the adjacency matrices.

4.2.2 Handling General Compatibility Relations

The procedure as discussed thus far is based on a compatibility relation of the form of Eq. 4.1. For a general compatibility relation, this procedure works fine except we need some post processing.

Consider a compatibility relation of the following form:

$$\mathcal{R} = \{(t_i, t_j)|f(d(t_i, t_j), \theta_i, \theta_j, \kappa_i, \kappa_j) = 0\}$$

(4.4)

as opposed to the conjunctive form of Eq. (1). We define the compatibility volume as the set of all points in the parameter space which satisfy the compatibility relation with respect to a fixed point in that space. We consider only compatibility relations such that the compatibility volume is simply connected. This is not likely to be a real restriction in practice for a well defined parameter space. The shaded volume depicted in Fig. 13 is the compatibility volume of the point A in a 3D space. The compatibility volume corresponding to our previous example is a 4-dimensional parallelepiped. The algorithm, as presented so far, works for hyperparallelepiped compatibility volumes. For a general compatibility volume, we find a parallelepiped which completely encloses the volume as shown in Fig. 13. Mathematically this means we find the support of the function in each dimension. Then we conduct voting using the compatibility relation corresponding to the enclosing parallelepiped.
After the associations have been hypothesized, we delete those pairs which do not satisfy the strict compatibility relation. This is also part of the screening process mentioned above. The search space is limited by considering only those token pairs that satisfy the enclosing (weaker) relation.

4.2.3 Complexity Analysis

To analyze the complexity of the method we compute the time spent in voting, scanning the voting space, and (sometimes) in performing logical operations on adjacency graphs. We will consider these costs separately. Let the time spent in voting per token be $t_v$, and let $t_s$ and $t_l$ be the scanning and logical times per operation, respectively; generally $t_v \geq t_s \geq t_l$. Suppose we have $N$ tokens, which are associating in a $D$-dimensional space with $k$ quantizers per dimension. Let the number of bins in dimension $i$ be $q_i$. If we have a general compatibility relation
to satisfy (as above), then we screen the resulting associations by computing the compatibility relation over a very restricted number of tokens.

Let the time for such a comparison be $t_{\text{comp}}$. Then the total amount of time for voting, considering all dimensions at once is:

$$
\text{Total Time } (T) = \left\{ \begin{array}{l}
\text{time to vote} \\
+ \text{time to scan the voting space} \\
+ \text{time to screen the associations}
\end{array} \right\}
$$

$$
T = N t_v + \left( \prod_{i=1}^{D} q_i \right) k^D t_s + N_{\text{assoc}} t_{\text{comp}}
$$

$$
T < N t_v + (qk)^D t_s + N_{\text{assoc}} t_{\text{comp}}
$$

where $q = \max(q_i)$ and $N_{\text{assoc}}$ is the number of associations in the enclosing parallelepiped. Strictly speaking $N_{\text{assoc}} \propto N^2$, hence the algorithm is $O(N^2)$. However, generally $N_{\text{assoc}} = O(N)$, giving a speedup when compared to brute force search. The exponential dependence on the number of dimensions can be reduced, as explained earlier, by considering a few dimensions at a time. However in this case we incur an extra cost in fusing the results by logical operations. Let us assume we restrict each voting space to be of dimension $d$, and let $\frac{D}{d} = m$, then the time taken is given by:

$$
\text{Total time } (T) = \left\{ \begin{array}{l}
\text{time taken to vote} \\
+ \text{time taken to scan the } m \text{ voting spaces} \\
+ \text{time taken to fuse the results} \\
+ \text{time to screen the associations}
\end{array} \right\}
$$

$$
T(d) < \frac{D}{d} \left[ N t_v + (qk)^d t_s \right] + (\frac{D}{d} - 1)N^2 t_i + N_{\text{assoc}} t_{\text{comp}}
$$

Then $d$ is chosen so that $T(d)$ is minimized. Since $d (= 1, 2, \ldots, D)$ is discrete, this needs to be done graphically, by plotting $T$ as a function of $d$ and choosing other parameter values appropriate for the problem. So $d = 1$ may not always be
the right choice. However, for a parallel implementation the $t_v$ and $t_i$ terms are constant and $d$ depends on the amount of hardware one is willing to devote. Note that the exponential dependence of the time on the dimensions is reduced from $D$ to $D/m$ and as $m \to D$ the dependence becomes linear.

4.2.4 This is Not a Hough Transform

The concept of voting employed in this approach may, quite naturally, lead the reader to infer a relationship to the Hough Transform. In fact, once past the notion of voting, the two approaches are conceptually very different and bear no real relationship to one another. They are designed to solve different problems, and consequently, use their voting spaces quite differently. Part of the difficulty lies in our own terminology, voting, which arises from language limitations. One may conclude that we count the votes (as in a Hough Transform), but we do not. We scan the voting space to identify tokens having compatible attributes. Rather than identifying instances of structures of a specific class from raw token data (as in a Hough Transform), we associate instances of a given class of token to infer the presence of a higher order class of token formed as an aggregate of the more primitive tokens.

On a mechanical level, while one looks for peaks in a Hough space, we do not. Quantization in a Hough space is crucial in terms of the accuracy of the parameters to be determined, but in our case the quantization index can be coarser. The Hough Transform ignores spatial and geometric context, generally due to dimensionality limitations, while we explicitly incorporate this information.
4.3 Selecting the Organization Parameters

The parameters to choose in perceptual organization are those of the compatibility relation. How far (in parameter space) do we search for associations? What is the orientation tolerance for collinearity or parallelism? We suggest choosing the parameters to reflect the least degree of significance tolerable\(^2\). Lowe [2] suggests using the non-accidental nature of the associations as a measure of their significance, thus measuring the significance of an association or relation by calculating its probability of occurrence. The former is inversely related to the latter [40]. We use his basic formulation, with some adjustments.

\(^2\)In effect, we err on the side of over association, since it is easier to recover from this condition.
We measure the proximity of two line segments of lengths, \( l \) and \( l_1 (l < l_1) \), by \( r \); where \( r \) is the separation between the two segments, as shown in Fig. 14. A longer line will have potential joins over a larger distance than a shorter line. Hence, the distance between two lines is normalized by the length of the shorter one to arrive at a scale independent measure. The expected number of endpoints to be considered for joining, \( N_{\text{join}} \), is:

\[
N_{\text{join}} = 2\rho \frac{r^2}{l_1}
\]  

(4.7)

where \( \rho \) is a unitless constant specifying the expected scale-independent density of line segments. Since the measure of significance is relative, the absolute value of this does not matter. As does Lowe, we choose it to be 1. The proximity factor is squared because we are considering an area; a square in the case of an digital implementation. We choose our search area so that \( r \leq l_{\text{average}} \), where \( l_{\text{average}} \) is the average length of the tokens (straight lines, ribbons) at the current level of hierarchy.

We proceed along similar lines to choose the angular tolerances for parallelism and collinearity. We consider parallelism first. Let \( l_1 \) be the length of the shorter line and \( l_2 \) be the length of the longer line; \( s \) is the separation between the two lines, measured as the distance perpendicular to the smaller line from its midpoint to the larger line. Then the proximity factor for parallelism is given by \( \frac{s}{l_1} \) (c.f. the argument above on endpoint proximity).

Another factor to consider is the angular separation between the two segments. Lowe takes this to be uniformly distributed. This is not, strictly speaking, correct.
If, as he assumes, the line orientations themselves are uniformly distributed, then the difference of orientations will have a triangular distribution function, which is the convolution of two identical uniform distributions. We calculate the significance of angular separation as follows. Construct a histogram of the orientations of all the line segments and examine its modes. This gives the major modes of token orientation. Around each mode assume a Gaussian density, and estimate its variance, $\sigma^2$ (Fig. 15). Assuming the orientations of different lines to be independent and identically Gaussian distributed, the difference between two orientations will also be Gaussian distributed with variance $2\sigma^2$. Of course, the identical distribution assumption is not always true but if we are associating two lines, they ought to have come from the same distribution. Since the algebraic difference is Gaussian distributed, the probability distribution of the absolute difference is given by:

$$p(\theta) = \frac{1}{\sqrt{\pi} \sigma} e^{-\frac{\theta^2}{(2\sigma)^2}} \quad \theta \geq 0 \quad (4.8)$$

The expected number of lines within a given separation and maximum tolerable angular difference of $\theta_{\text{max}}$ is:

$$E = \left( \frac{2Ds l_2}{l_1^2} \right) \int_0^{\theta_{\text{max}}} p(\theta) d\theta \quad (4.9)$$

Taking the expectation over the entire image we substitute $l_{\text{average}}$ for $l_1$ and $l_2$. We choose $s \leq l_{\text{average}}$. Thus we consider an area of $l_{\text{average}}^2$ when looking for parallelism and consider all angular separations less than $\sigma_{\text{average}}$ (i.e. $\theta_{\text{max}} = \sigma_{\text{average}}$), the average of the variances of the different modes; where $\int_0^{\sigma_{\text{average}}} p(\theta) d\theta \approx$
Figure 15: Histogram of the orientations of line segments and their associated approximating probability distribution.

0.5211, an estimate which works well. In fact, for most cases $\sigma \approx 20^\circ$; which is perceptually satisfactory.

To consider groupings based on collinearity, we proceed similarly except that our proximity factor is $\frac{g(\pi + h)}{h}$, where $g$ is the size of the gap. The angular separation tolerance remains the same.

We have tried to tie the parameters of perceptual grouping to meaningful entities, so that their selection does not seem arbitrary. It may be noted that the notions of collinearity, parallelism and proximity also hold for higher level groupings. However, they will involve a larger number of attributes. We proceed along similar lines as for orientation and use the variance of the attributes as the upper limit on the associable difference.
4.4 Result on the Aerial Image

The above algorithm has been implemented in C and tested on a variety of images with very good performance. Results in perceptual organization are difficult to display, because they are in a symbolic form. A single pictorial form might not convey the full meaning. Ideally, we want to display each organization in a different image, which is not practical. As a compromise we display each type of organization in a separate image. We have also not distinguished strong organizations from the weaker ones in these images; there has been no selection process.

The edges were detected by the optimal zero crossing operator [21] at $\gamma = 1.5$ and the contours were segmented into 103 constant curvature segments as shown in Fig. 16(b). The preattentive module detected 47 closures, 9 strands, 12 ribbons, and 16 parallels as shown in Figs. 16(c), (d), (e) and (f). Note the closed hypotheses corresponding to the lake. Notice how the significant features in the image were extracted.

The preattentive algorithm was coded in C. The total CPU execution time on Sparc IPX is 35 sec.

The structures detected in the preattentive part are used as evidence in the attentive part, discussed next. More complex organizations are inferred from these simple preattentive organizational cues.
Figure 16: Organizations detected by the preattentive perceptual organization module. (a) Gray Level Image (b) Edge contours (c) Constant curvature segments (d) Closed boundaries (e) Edge strands (f) Parallel edge segments
CHAPTER V

Attentive Algorithm: The Perceptual Inference Network

"Units in experience go with functional units in the underlying physiological process"

Köhler

The attentive part of perceptual organization have explicit knowledge about various Euclidean geometric structures and enables us to organize our preattentive cues. According to our understanding, the use of an explicit knowledge base in perceptual organization is rare if not unique. Our main contribution in this regard is the formalism based on Bayesian networks [115] for geometric knowledge base representation. The knowledge base allows us to infer missing data and gain immunity against noise processes. We propose an efficient methodology, based on sound probabilistic semantics and amenable to multiproccessing hardware, to reason over spatial data. To our knowledge, this is the first application of Bayesian networks to intermediate and low level vision. The network formalism also enables us to provide an efficient framework for the integration of multiple sources of information, such as corner detectors, straight line detectors, circle detectors, and ribbon detectors.
The use of Bayesian networks is not new. Binford et al. [116] and Chelberg [117] each use knowledge bases encoded in a Bayesian net to recognize objects in range images in the SUCCESSOR system. Recently, Mann and Binford [118] presented the new version of the SUCCESSOR system using the concept of aggregation nodes to separate interactions between parts of the Bayesian net. Their work differs from ours in two ways. First, theirs is a high level system to infer complete objects and therefore has a network of parts corresponding to each object. Second, we do not agree with the structure of the Bayesian network as used in [117]. Chelberg suggests that an object causes its parts, so the links should be directed from the objects to their parts. This leads to the conclusion that the parts are conditionally independent, given the object or, more precisely, that the existence of each object part is a conditionally independent event, given the existence of the entire object. For example, this would suggest that given the observation of a table, the observation of each leg and the top would be (five) statistically independent events. We do not agree.

Jensen et al. [119] use the context modeling capabilities of Bayesian networks for image interpretation tasks. Munck-Fairwood [120] uses a Bayesian network for probabilistic 3D inference from 2D data. Rimey and Brown [121] proposed a task oriented vision system using Bayesian networks in a utility based decision framework.

In Section 5.1 we propose the new network formalism which is clarified through an example network in Section 5.2. Section 5.3 evaluates the computational complexity of the network formalism. Results on a real image are presented in Section 5.4.
By attentive organization we refer to the active process of detecting structure by inference, hypothesis, and confirmation. The process involves a knowledge base of basic information encoded in the form of a Bayesian network. For perceptual organization, we feel a knowledge base aware of geometric figures such as circles, ellipses, rectangles, polygons, and the like is appropriate. A geometric feature can be constructed from various cues, such as combinations of vertices and sides; the whole feature is then inferred from the parts detected. This enables us to detect features in the presence of noise and occlusion.

\[
P(x_1, \ldots, x_9) = P(x_9|x_7)P(x_8|x_7)P(x_7|x_4)P(x_4|x_2, x_3)P(x_6|x_5)P(x_5|x_3)P(x_3|x_1)P(x_2|x_1)P(x_1)
\]

Figure 17: An Example Bayesian network

5.1 Knowledge Base: The Perceptual Inference Network
Bayesian networks, also known as belief networks, influence networks, or causal networks, are directed acyclic graphs with nodes representing propositions (or random variables) and arcs signifying direct dependencies as quantified by conditional probabilities. An example Bayesian network is shown in Fig. 17. Bayesian networks do not assume independence among features, rather, they encode the dependencies among features. This can be seen by comparing the factoring of the joint density with the structure of the network in the figure. Consider the following simple example of two lines. Two distinct lines, \( l_1 \) and \( l_2 \), are independent. However, they become dependent once we know that they are parallel, because that suggests that they may share a common underlying cause. Mathematically, denoting the event of parallelism as \( I \), we have \( P(l_1, l_2) = P(l_1)P(l_2) \) but \( P(l_1, l_2|I) \neq P(l_1|I)P(l_2|I) \).

This information is encoded in a Bayesian net by 3 nodes: 2 nodes represent the lines and the third node represents the parallelism relation; links are drawn from the line nodes to the parallel relation node. More details regarding Bayesian Networks are presented in Appendix I.

**Extension: The Perceptual Inference Network**

Our problem is to integrate information about various spatial features to form composite hypotheses. Let us say we have \( m \) features \( f_1, \ldots, f_m \) and \( n \) locations \( l_1, \ldots, l_n \). Let us form our random variables as \( l_i^f \) taking two values, 0 (false) and 1. Although the links are directed, probabilistic messages are passed both ways. The message in the direction of the link is termed \( \lambda \) and the other way is \( \pi \). \( \lambda \) and \( \pi \) denote the bottom up and top down components, respectively [115].

\footnote{Although the links are directed, probabilistic messages are passed both ways. The message in the direction of the link is termed \( \pi \) and the other way is \( \lambda \). \( \lambda \) and \( \pi \) denote the bottom up and top down components, respectively [115].}

\footnote{It is interesting to note that in the theory of structural descriptions of models [122, 123] it is assumed that the relations are conditionally independent given the primitives and not the other way around. Our assumption is consistent with this.}
The event \( \{ l^f_i = 1 \} \) denotes the fact that feature \( f_j \) occurs at location \( l_i \). Our aim is to formulate an efficient means of updating the probabilities of these random variables based on the evidence. Given a set of features at some locations, expectations for other features at different locations are formed. We want to devise an efficient means to do so. In total we have \( mn \) random variables. One way of updating is by specifying a joint probability distribution of \( mn \) random variables, which is very difficult if not impossible.

We make the problem tractable by exploiting the conditional dependencies inherent among the variables. Features which are dependent tend to be close together spatially. In the context of a hierarchical system this assumption is generally true. Dependencies among distant features are captured at higher levels of the hierarchy. The local dependency structure will make the connections among the random variables sparse. Let us say we can construct a directed acyclic graph among the \( mn \) variables with the links quantified by the conditional probabilities among the variables. This will form the Bayesian network which we can update using the message passing discussed in [115]. The disadvantage of this solution is its high structural complexity. We have \( mn \) nodes and typically \( mn - 1 \) connections. Yet many nodes will not be used because there will be no evidence to support the feature at the location represented by the node. We will take advantage of this sparseness.

The sparse nature of the above Bayesian network can be exploited to reduce the structural complexity further. The sparseness arises because some random variables are independent from another or conditionally independent. Besides being
sparse, the network will also have a number of similar substructures because the
same feature (type) at different locations will have similar neighborhood structure.
The manner in which a feature affects the probabilities of another feature is depen­
dent on their relative location and is independent of their absolute locations. We
can group together some of the random variables representing the same features
at various locations. These random variables share the common factor that they
are the same feature, \( f_j \), but in different locations. Hence they will also share a
common neighborhood structure (in the network) with other such groupings. The
group of random variables \( l_{i_1}^{f_j}, \ldots, l_{i_k}^{f_j} \) form a composite random variable represen­
ting the events that feature \( f_j \) occurs at \( l_i \) through \( l_k \). The grouping will certainly
reduce the number of nodes of the network, but the computational complexity of
each node increases. However, the increase is less than the reduction in structural
complexity, as we shall see. Note that we might not be able to group all the random
variables representing a feature. For example, the grouping of \( l_{i_1}^{f_j}, \ldots, l_{i_k}^{f_j} \) might not
be possible if it leads to the formation of a cycle in the undirected graph structure.
This concept will become clearer when we discuss a concrete example.

Grouping the nodes will create various other issues, such as message passing,
updating of probabilities, and the construction of conditional probability matrices.
We address these issues next. We refer to the Bayesian net with these incorporated
features as a Perceptual Inference Network (PIN) (see Fig. 18).

As an aside, it may be noted that it is also possible to group nodes representing
a set of features at a fixed location. However the number of composite nodes in such
a case would be equal to the number of possible locations, which is typically large. Also the network structure would be dependent on the size of the image. Such a grouping based on a fixed locations may be useful when we have small number of locations to consider.

5.1.1 Belief Revision Using Modified Message Passing

Each composite node (Fig. 19) represents a collection of random variables denoting the presence of a feature at a set of locations. The common characteristic is the
type of feature each random variable represents. They differ only in the location of the feature. We have to specify the conditional probability of a feature at a particular location given the parents at compatible locations. The original formulation of message passing in Bayesian network will suffice for this case. However, we need to check for compatible locations when the message arrives. The conditional probabilities that need to be specified are of the following form:

\[ P(l_x^i | l_{ui1}^i, \ldots, l_{um}^i) \quad \text{for } i, l = 1, \ldots, n \]  

where \( l_x^i \), \( l_{um}^i \) denote the locations of the features \( X \) and \( U_m \) at \( l_x \), and \( l_{um} \), respectively. We use upper case to refer to random quantities and lower case to refer to specific values.

The specification of the conditional probability can be further simplified if we assume that the conditional probability can be factored into two functions \( g \) and \( Comp \), one depending only the feature type, and the other being a locational compatibility function:

\[ P(l_x^i | l_{ui1}^i, \ldots, l_{um}^i) = g(x | u_1, \ldots, u_m)Comp(l_x, \{l_{ui1}, \ldots, l_{um} \}) \]  

The function \( g \) is a conditional function which expresses the degree of belief in a feature, \( X = x \), given the existence of other features, \( U_1 = u_1, \ldots, U_m = u_m \). This, in general, will not be a probability function because it is multiplied by the \( Comp \) function to form the conditional probabilities. For lack of a better name, we call \( g \) the conditional belief function. This function is independent of the locational information and is affected only by the definition of the features. Thus, the feature
“triangle” depends on three lines and three vertices. We can assert a degree of confidence in the existence of the triangle based just on the presence of subsets of the lines or vertices and the attributes of the formed triangles. The locational compatibility of line or point features with a particular triangle is captured by the \textit{Comp} function.

The form of the \textit{Comp} function is a matter of choice and depends on the feature sought. The range of the function should be chosen such that $P$ is a conditional probability function. The support of \textit{Comp}, that is, the domain over which it is non-zero, depends on the feature represented by the composite node. We choose a binary valued function which is unity (true) when the locations of the features lie within a locational compatibility tolerance. The choice of the tolerance is an open issue depending on the amount of positional error one can accept. The belief in the features at locations not compatible with the incoming message locations (where $\text{Comp} = 0$, or false) are unchanged.

Assuming the form of the conditional probability to be as in Eq. 5.2 means that, instead of storing a set of conditional matrices, we need to store only the much smaller $g$ matrix and \textit{Comp} function definition. As for the other belief parameters, instead of storing one set, the composite node keeps lists of belief parameters for each location separately. Since the beliefs in features at locations not compatible with a particular location are not changed, we can decompose the processing at the node into two modules (see Fig. 19). The first module computes spatial compatibility and chooses the belief parameters of the compatible features for updating. Spatial
compatibility can be computed using a function, a look up table, or an associative memory.

If there is no match for a message, then a new entry is created in the table, obviating the spawning of a new node as would be required in a brute force Bayesian network implementation. The flow diagram is shown in Fig. 20. The belief updating is done in the second module by a shared computational resource [115]. To prevent multiple updates from the same direct evidence, each message is also tagged with the identity (ID) of the elementary evidence on which it is based. Thus, duplicate confirmations are avoided by checking the evidence ID at the composite node, as well as the location. This computational resource is time shared by the features at different locations and is shielded from the network by the compatibility computation module (again, see Fig. 19) unlike previous formulations of Bayesian Networks.
5.1.2 Network Instantiation

The composite nodes and associated links of the PIN are instantiated \textit{a priori}. The root composite nodes are given equal prior probabilities. At the start, each of the composite nodes has a NULL instantiated location table. As new evidence is introduced into the PIN, locational entries are made at the composite nodes corresponding to the hypothesized structure.
5.1.3 Prediction of structures

As mentioned earlier, on receiving a message from a neighbor the locational module of a composite node generates a set of compatible locations. If a feature already exists at the predicted location, its associated probabilities are updated. Otherwise, a new locational entry is created. These locational entries form predictions of the structures corresponding to the respective composite nodes. Although we may lack direct evidence for a given structure, it can often be inferred from messages received at the corresponding composite node. The algorithm to generate compatible locations depends on the feature type. This idea will become clearer with an example PIN in Section 5.2s.

5.1.4 Management of Expensive Computational Resources

Many (most) machine vision algorithms, such as symmetry detection and recovery of shape from shading, are very expensive computationally. It is to our advantage to apply these algorithms opportunistically, only when there is promise of success or a need to resolve ambiguity. The formalism of this chapter supports this. The evidence for one feature at a particular location may also raise expectations for other features at other locations. A utility maximization module can decide, based on the resulting probabilities, which computational modules (if any) can offer the maximum advantage. We present such a module in the next chapter.
5.2 An Example PIN for Perceptual Organization

The Perceptual Inference Network formalism, as presented, is general and has the potential for being used to reason and integrate spatial information in a variety of problem domains. The choice of features over which to reason depends on the domain in question. In the case of perceptual organization as implemented here, this depends on the organizations of interest, which vary with level within the organizational hierarchy (Fig. 5). The zeroth level of the hierarchy consists of constant curvature segments. The first level consists of ribbons, closed figures, strands, and parallelograms. The second level includes parallel ribbons, strands of ribbons, closed cycles of ribbons, regular arrangements of polygons, and intersections. To demonstrate the viability of this approach, we will concentrate on the first level of the hierarchy with PIN composite nodes representing closure, segments, strands, ellipses, circles, ribbons, parallels, and corners.

5.2.1 Manually Structuring the PIN

Given the nodes of the network we have to decide on the links in the network. If we had correlation data between the random variables representing our nodes, we could have used the algorithm presented in [115]. As a first cut, we use heuristic means and manually construct our PIN. The next chapter presents a comprehensive theory of Bayesian network construction for visual tasks. We also compare our manually structured PIN with the automatically structured one in the next chapter. In the process of the manual construction we used the following heuristics:
• The direct dependencies among features are represented by links going from nodes representing less organized features to those representing features having more organization. A set of features of less structure making up a feature of more structure is said to cause the feature of more structure. This encodes the fact that the constituent features are dependent, once the more organized feature is detected.

• The distance (in terms of links) between two nodes of the PIN representing two geometric figures should be low if the figures are very similar. For example, a rectangle node and a trapezoid node should be closer than a rectangle node and a circle node.

• Any cycles in the PIN are broken by introducing pseudo nodes representing intermediate organizations. That is, we intentionally introduce a bit of node redundancy to avoid cycles. We show examples of this below.

An example Bayesian network is shown pictorially in Fig. 21. The network has 23 nodes, each representing a feature type, as depicted in the figure. We now point out some key aspects of the network structure.

• Node N23 denotes the concept of a parallelogram, which is formed from the concept of a trapezoid and the attribute that the "non parallel" sides of the trapezoid are parallel.

• A trapezoid, N21 is a quadrilateral with a pair of the sides parallel, represented by nodes N19 and N20, respectively.
A quadrilateral is a polygon, node $N_{15}$, having four sides. A polygon with 3 sides is a triangle, node $N_{16}$. An equilateral triangle, node $N_{18}$, is a triangle with appropriate symmetry, node $N_{17}$.

The feature of closure, node $N_{3}$, with corners at particular locations, node $N_{14}$, forms polygonal hypotheses.
• The features of ellipse, node N6, and circles, node N8, are formed from appropriate symmetry, nodes N5 and N7, respectively.

Note that we have three corner nodes (N10, N11, N14) corresponding to different spatial groupings. Placing them into a single node would render the network cyclic. Also note that the features represented by N4, N9, and N15 could have been formed from the concepts of strand (N1) and constant curvature segments (N2) directly, bypassing node N3. However, the network would then have cycles. Introducing N3 keeps the network acyclic.

5.2.2 Location Compatibility Functions, Comp

As we saw before, each node of the perceptual inference network has a locational compatibility function, Comp, defined to calculate the compatibility of a message from a child or parent with its own location set (see Eq. 5.2). If the location is instantiated, then the belief parameters associated with that location are updated. Otherwise, the new location is entered in the list of present locations. To check for compatibility, each node generates a set of compatible locations based on the spatial information sent by the parent or child. Then it does a point by point locational match to ascertain how far the new hypothesized location is from previously noted locations. Compatibility is ascertained based on a proximity tolerance. For our experiments this is 5 pixels. The choice of this value depends on the spatial resolution of the system. The robustness of the choice is studied in Section 5.4.2.

Before going into the details of the matching process, we briefly discuss the format of the locational information at each node. This format is peculiar to this
Table 1: Locational information format at each composite node of the example net.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1 (Strand)</td>
<td>((start) (end) ((Segment), · · ·, (Segments)))</td>
</tr>
<tr>
<td>N2 (Segment)</td>
<td>((start) (end) (radius, center) ((x y), · · ·))</td>
</tr>
<tr>
<td>N3 (Closed Token)</td>
<td>((Segment), · · ·, (Segment))</td>
</tr>
<tr>
<td>N4 (All Curves)</td>
<td>((x1 y1) (radius, center) (x2 y2) (radius, center)...)</td>
</tr>
<tr>
<td>N5 (Symmetry)</td>
<td>(center orientation)</td>
</tr>
<tr>
<td>N6 (Ellipse)</td>
<td>((center) (major axis param) (major, minor axis size))</td>
</tr>
<tr>
<td>N7 (Radial Symmetry)</td>
<td>(Center, radius)</td>
</tr>
<tr>
<td>N8 (Circle)</td>
<td>(Center, radius)</td>
</tr>
<tr>
<td>N9 (Mixed Closed)</td>
<td>((x1 y1) (radius, center) (x2 y2) (x3 y3)...)</td>
</tr>
<tr>
<td>N10 (Corner)</td>
<td>(x y)</td>
</tr>
<tr>
<td>N11 (Corner)</td>
<td>(x y)</td>
</tr>
<tr>
<td>N12 (Curve symmetry)</td>
<td>(axis, width)</td>
</tr>
<tr>
<td>N13 (Ribbon)</td>
<td>(curve1, curve2)</td>
</tr>
<tr>
<td>N14 (Corner)</td>
<td>(x y)</td>
</tr>
<tr>
<td>N15 (Polygon)</td>
<td>((x1 y1) (x2 y2) · · · (xn yn))</td>
</tr>
<tr>
<td>N16 (Triangle)</td>
<td>((x1 y1) (x2 y2) (x3 y3))</td>
</tr>
<tr>
<td>N17 (Symmetry)</td>
<td>(center orientation)</td>
</tr>
<tr>
<td>N18 (Equilateral)</td>
<td>((x1 y1) (x2 y2) (x3 y3))</td>
</tr>
<tr>
<td>N19 (Quadrilateral)</td>
<td>((x1 y1) (x2 y2) (x3 y3) (x4 y4))</td>
</tr>
<tr>
<td>N20 (Parallel)</td>
<td>(line1 line2)</td>
</tr>
<tr>
<td>N21 (Trapezoid)</td>
<td>((x1 y1) (x2 y2) (x3 y3) (x4 y4))</td>
</tr>
<tr>
<td>N22 (Parallel)</td>
<td>(line1 line2)</td>
</tr>
<tr>
<td>N23 (Parallelogram)</td>
<td>((x1 y1) (x2 y2) (x3 y3) (x4 y4))</td>
</tr>
</tbody>
</table>

implementation and is certainly not general; one can choose any suitable format.

Table 1 gives abstract representations of the data structures which store the locations of points and attributes characterizing a particular feature. The spatial information of the strands (N1), constant curvature segments (N2), and the closed tokens (N3) are stored pointwise. Only the best fitting circular arc and/or straight line end points are stored for the "all curves" closed figure (N4), the mixed curve straight
line closed figure (N9), and the polygon (N15). Similar structures are stored for the other features. The ellipse node (N6) and circle node (N8) store the relevant ellipse and circle parameters.

Computing compatible locations for nodes like the triangle, the trapezoid, or the parallelogram nodes is straightforward; we just match the end points. For N6 and N8 we consider the best fitting ellipse or circle, respectively. Nodes N4, N9, and N15 deserve special attention. For a message received at N4 from N6, we segment the ellipse at those points where the latus rectum intersects the boundary. This generates the constant curvature segments for node N4. To generate compatible locations for spatial information from node N3 to N4, N9, and N15, we proceed as follows. The locational information for node N3 is a list of constant curvature segments detected by the low level contour segmentation algorithm [34] after edge detection. We need to repeat the segmentation of each closed chain of pixels into constant curvature parts to generate the appropriate set of compatible arcs or straight lines.

The process of calculating compatible locations for N4, N9, and N15 for a message from N3 is depicted in Fig. 22. To begin, we compute the possible intersection points of the arcs or straight lines, for example I1, I2, I3, and I4 in Fig. 22. Arcs and straight lines connecting a subset of intersections will form our best decomposition into arcs or straight lines. Thus, for example, a better description of the closed figure in Fig. 22 may be arcs connecting I4 to I1, I1 to I2, and I2 to I4, without involving I3. To choose this optimal subset we transform the problem into the shortest path
problem in graph theory. The set of possible intersections form the nodes in a graph whose links denote the best fitting arc or line between the intersection points. The weight of a link from node $i$ to $j$, $f_{ij}$, is the least square fit error considering the part of the cycle between intersections $i$ and $j$. Note that $f_{ij} \neq f_{ji}$ because they represent the directed fit error between $i$ and $j$ and therefore involve different points.

To choose the best decomposition we select the cycle which starts at a point and ends at that same point having the best fit index. The fit index is defined as the total fit error divided by the number of intersection points constituting that cycle. We search for the optimal cycle by listing the shortest paths starting and ending at a common node and choose the one having the minimum fit index. We use the standard graph algorithm of Floyd and Warshall [124], based on dynamic programming, to calculate all shortest paths between all pairs of nodes.

The optimal segmentation found above forms our compatible location for a message from $N3$ to nodes $N4$, $N9$, or $N15$. The algorithm for the three nodes, $N4$, $N9$, and $N15$, is the same except for the types of primitives we fit between the intersection points to set up our graph. For node $N4$ we consider circular arcs. For node $N9$ we consider both the circular and straight lines and choose the one having the minimum error. We compute the compatible locations for node $N15$ by considering straight line fits between points. To compute compatible locations for the quadrilateral node ($N19$), we consider the four largest lines of the polygons as constituting the quadrilateral boundary, for messages coming from node $N15$. Similarly, we only consider the three largest sides of the polygon for the triangle node.
The conditional belief function, \( g \), (see Eq. 5.2) captures the belief in the specified feature type, given the status of the corresponding parent nodes. The conditional probabilities of the underlying Bayesian network are partly characterized by \( g \) and partly by the locational compatibility function, \( \text{Comp} \), discussed before. We consider the following general form for the conditional belief functions:

\[
g(X = 1|u_1, \ldots, u_n) = C_{(u_1,\ldots,u_n)}(1 - e^{-f_{\text{geom}}})
\]  

(5.3)
where \( X \) is the composite node under consideration, \( u_1, \ldots, u_n \) are the values taken by the random variables represented by \( X \)'s parent composite nodes, and \( C(u_1, \ldots, u_n) \) is a value which depends on the state of the parents. We found from experimentation that the absolute value of the constant is less important than the relative values of the constant for different parent states. The second factor captures the geometric information in the primitive. The exponential is chosen to limit this factor to the interval \((0, 1)\). \( f_{\text{geom}} \) is a factor based on the size of the features and some defining relations, like parallelism for a parallelogram. Qualitatively speaking, the larger the feature, or the more strictly it satisfies the defining relation of a feature type, the larger the value of the belief. Other functional forms with this qualitative behavior may also suffice. In Section 5.4.2 we investigate the robustness of the form of the conditional probability function.

In the present implementation \( f_{\text{geom}} \) is calculated in the following manner. For the nodes representing some form of closed convex figure \((N3, N4, N9, N13, N15, N16, N18, \text{and } N19)\) we use the following form:

\[
f_{\text{geom}} = \frac{\left(\prod_{i=1}^{n} l_i\right)^{1/n}}{l_{av}}(\delta_{\text{convex}})
\]

(5.4)

where \( l_i \)'s are the length of the \( n \) constituting segments of the closed figure, \( l_{av} \) is a normalizing constant and is chosen to be the average length of segments in the image, and \( \delta_{\text{convex}} \) is 1 if the figure is convex and 0 if not. Clearly, it would be a useful enhancement to measure the degree of convexity on a continuum of some sort, but
this form suffices for the present. The geometric mean in the numerator is chosen to suppress the spurious generation of hypotheses like a quadrilateral with one side whose length is zero, which we noted were formed with an arithmetic average. The trapezoid node, $N21$, and the parallelogram node, $N23$, has the same expression for $f_{geom}$ as that above, with extra multiplicative terms of the form $\cos^2(\theta_{diff})$, where $\theta_{diff}$ is the angular difference between lines, to penalize for nonparallelness in the constituting sides. For the ellipse node, $N6$, we choose $f_{geom} = |b/a - 1|\sqrt{Area/l_{av}}$, where $b$ and $a$ are the minor and major axis, respectively. Circle node, $N8$, has $f_{geom} = \sqrt{Area/l_{av}}$.

5.2.4 Prior Probabilities

One of the common arguments against Bayesian probability theory is that we must always specify prior probabilities. In our case, we have to specify the prior probabilities of the root nodes. We do not see the specification of prior probabilities as a disadvantage, rather it enables one to incorporate already acquired knowledge in a very concise manner. The prior probabilities, for our example, are the prior probabilities of occurrence of the root node features like strands, constant curvature segments, corners, and symmetry. There may be domains where the occurrence of, say, corners is very low; this knowledge can be very concisely incorporated in the reasoning process through the assignment of estimates of prior probabilities. In the absence of any prior knowledge we assume equal priors, a solution which maximizes the entropy of the distribution. This is the best we can do in such a situation. It is significant that the effect of prior probabilities in a probabilistic system decreases as
new evidence is gathered. Thus, minor differences in prior probability assignment do not matter significantly in the long run. This is shown in Section 5.4.2.

5.2.5 Evidence Instantiation

The pieces of evidence for our case are the organizations detected in the preattentive phase. These include strands, constant curvature segments, parallelograms, and ribbons. Each piece of evidence activates the node belonging to the corresponding feature type. This evidence is virtual in the sense that it conveys a graded degree of belief about the underlying features. The introduction of virtual evidence can be modeled as a dummy node $Z$ posting a message $P(Z = z | x)$ to the network node $X$. The message is the conditional probability of state represented by the evidence given the state of the node, $X$. For our case, this will be an estimate of the probability of the detected feature. This probability can be measured in terms of the photometric and geometric evidence we have for the feature. For a closed boundary we set the message equal to the ratio of the total length of the segments to the length of the perimeter of the hypothesized closed boundary. For parallels it is the fraction of overlap. Constant curvature segments are assigned confidence measures according to the straight line or arc fit error they exhibit.

Note that although our present evidence is based only on geometric properties, this need not be. Since we start with edge segments, which are necessarily geometric entities, and since the process of perceptual organization is primarily geometric, the emphasis is on geometric properties. However, photometric properties may be used to confirm or reject a hypothesis. This will involve the intelligent management of
computational resources, because the gathering of photometric evidence is generally expensive. As shown in Fig. 6, special purpose visual algorithms form a different block of our system, discussed in Chapter VII.

5.3 Algorithmic Complexity

In this section we analyze the proposed perceptual organization system from the standpoint of algorithmic complexity and present some timing data. Since the preattentive module is analyzed in Chapter IV, we restate the results here.

The complexity is computed with respect to the input size, which is the number of tokens, $N$. The preattentive algorithm is depicted in Fig. 7. As we can see the algorithm consists of two parts: construction of the Gestalt graphs using voting and the extraction of structure using graph theoretic operations. The voting algorithm was shown to have computational complexity of $O(N^2)$. However, the average case complexity is $O(N_{assoc})$, where $N_{assoc}$ is the expected number of associations, which is generally $O(N)$. Thus voting methods are better than brute force search, which is always $O(N^2)$.

The graph theoretic operations used are those to AND graphs, find connected components, compute all shortest paths, calculate the minimal spanning tree, and search for maximal cliques. ANDing two graphs is an $O(N^2)$ operation, but with a small multiplicative constant. We find connected components by depth first search, which is $O(N)$. The Floyd and Warshall algorithm to compute all shortest paths is $O(N^3)$, and the minimal spanning tree algorithm is $O(N^2)$. Finding cliques is $NP$-complete but we are fortunate in that the clique size in our graph, the one to
find continuous segments, is very small, typically 4 to 5 vertices. Thus the clique
finding algorithm is not the bottleneck. In practice, the execution time is dominated
by the polynomial all shortest paths algorithm.

The next part of our algorithm is the attentive part implemented in the Percep­tual Inference Network (PIN). Since a PIN is a form of Bayesian network except
with a locational matching module, we use the complexity results of Bayesian Net­
works. Inference using general Bayesian Networks is $NP$-hard [125]. However, if
we use a singly connected network, then the inference time is on the order of the
diameter of the network. Assume that we have $N$ evidence tokens. The instantia­
tion of each token will take constant time to propagate. But at each node we have
to do a locational match which is $O(\log N)$ when efficient data structures such as
quad trees are used. Thus, taking all the $N$ tokens together, theoretically, belief
revision will take $O(N \log N)$ time when using efficient data structures to store the
locational information at each node. In the present implementation the location
match is $O(N)$ and therefore belief revision is $(N^2)$.

5.4 Results

We implemented the network shown in Fig. 21, from nodes $N1$ to $N23$, representing
knowledge up to the concept of a parallelogram, ribbons, circles, ellipses, polygons,
and closed figures. The code was written in C++ and runs on Sun Sparc-IPX. The
PIN is instantiated first with a NULL locational list at each composite node. Each
composite node is an object and the associated methods are the message handling
algorithms. As discussed in the pervious chapter the perattentive algorithm, the vot-
ing and the graph theoretic modules, detect various relations like parallelism, curved symmetry, closure, and strands in a purely bottom up fashion. These relations are introduced as evidence for the appropriate nodes of the PIN, initiating a sequence of message passing. The output is the set of salient hypotheses represented by some nodes of the PIN like that for parallelograms, quadrilaterals, ellipses, circles, and ribbons with associated probabilities.

In this section we present results using the output presented in the last chapter on the aerial image. We also demonstrate performance characteristics of the PIN with respect to the chosen parameters.

5.4.1 Aerial Image

The results are on the aerial image shown in Fig. 16(a) (Chapter IV). The scene has a number of linear and curvilinear structures of interest, which is typical of aerial scenes. The detected edges are shown in Fig. 16(b). These edges were segmented into constant curvature primitives (shown in Fig. 16(c)) and given as input to the voting and graph theoretic modules. The latter produced organizations such as parallels, ribbons, closures, and strands, each associated with a probabilistic degree of confidence. There were 103 constant curvature segments, 47 closures, 9 strands, 12 ribbons, and 16 parallels as shown in Figs. 16(c), (d), (e) and (f).

The relations detected by the preattentive voting and graph theoretic modules are introduced as evidence into the PIN. After the network settles to equilibrium, which takes 72 seconds on a Sun Sparc-IPX, we have various organization hypotheses with associated probabilities. Since it is impractical to give a complete listing of
the node probabilities, we show some of the organizations as images. Fig. 23 depicts a set of such organizations. The complete set of closure hypotheses represented by node $N3$ of the PIN are shown in Fig. 23(a). The predicted sets of parallelograms, circles, ellipses, ribbons and corners are shown in Figs. 23(b)-(f). There are 67 closures, 26 parallelograms, 58 ellipses, 58 circles, and 11 ribbons. Note that these are just hypotheses based on the relational evidence detected by the attentive module. Expensive photometric based custom feature detectors can now be intelligently applied on the hypothesized image regions based on the associated probabilities. This is discussed in the next chapter.

5.4.2 Performance Evaluation

On reaching equilibrium, the PIN produces hypotheses of various organizations, each associated with a probability value. In this section we investigate the robustness of output probability values with respect to the chosen system parameters. The PIN is specified by its graphical structure, the conditional probabilities, and the prior probabilities of the root nodes. The conditional probabilities are specified by the function $g$ (Eq. 5.2) and the locational compatibility function $Comp$ (Eq. 5.2). The function $g$ (Eq. 5.3) in turn is a product of a constant and a functional form. We study the robustness of the PIN probability outputs with regards to these constants, functional form, and the $Comp$ function. In addition, we also probe into the necessity of equal priors assumption for the root nodes of the PIN.
Figure 23: Organizations detected using the attentive module on the aerial image. (a) New closed boundary hypotheses (b) Parallelograms (c) Ellipses (d) Circles (e) Ribbons (f) Corners
Locational Compatibility Function, $Comp$

For these experiments, the support of the locational compatibility function (the domain over which its value is one) is $t = 5$ pixels. This choice depends on the necessary spatial resolution. A large value of $t$ means that features that are spatially distant can be declared to be compatible. This reduces the ability to resolve nearby features and tends to produce fewer hypotheses. However, as we shall see, the reduction in the number of hypotheses is not drastic.

For this discussion we focus on the variation of the number of hypotheses of a few salient nodes which are distributed over the network: $N23$ (Parallelogram), $N13$ (Ribbon), and $N6$ (Ellipse), as $t$ ranges from 3 to 15 pixels. The results are shown in Table 2. Note that the variation of the number of hypotheses is low around the chosen value of the parameter ($t = 5$). Only for very high values (like 15) is the number significantly different. However, the execution time, which depends on the number of introduced pieces of evidence, is approximately constant. From these results we infer that the PIN is robust with respect to the locational tolerance parameter $t$.

Prior probabilities

The prior probabilities specified for the PIN are those of the root node features. In the absence of information to the contrary, we assume equal priors for the root nodes. We now want to consider the robustness of this assumption by comparing the behavior of the final PIN probabilities for different priors.
Table 2: Table showing the variation of the number of hypotheses of some salient nodes with parameter $t$ of the $Comp$ function. The last column lists the execution time for each value of $t$

<table>
<thead>
<tr>
<th>$t$</th>
<th>Parallelogram ($N_{23}$)</th>
<th>Ribbon ($N_{13}$)</th>
<th>Ellipse ($N_6$)</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>27</td>
<td>11</td>
<td>64</td>
<td>69</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>11</td>
<td>63</td>
<td>71</td>
</tr>
<tr>
<td>5</td>
<td>26</td>
<td>11</td>
<td>58</td>
<td>72</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>11</td>
<td>51</td>
<td>74</td>
</tr>
<tr>
<td>8</td>
<td>22</td>
<td>10</td>
<td>44</td>
<td>67</td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>10</td>
<td>43</td>
<td>75</td>
</tr>
<tr>
<td>15</td>
<td>17</td>
<td>9</td>
<td>36</td>
<td>78</td>
</tr>
</tbody>
</table>

As an indicator of performance we will study the behavior of the probability assignments of nodes $N_{23}$ (Parallelogram), $N_{13}$ (Ribbon), $N_6$ (Ellipse), and $N_8$ (Circle). The behavior of the other nodes is similar. Since the random variables are binary we consider the probability of just one state, namely that of existence. For each composite node we plot the probabilities associated with the feature at different locations and see how they change with different priors.

Notice that the relative values of probabilities are far more important than their absolute assignments. Therefore, we use a normalized correlation measure to quantify the similarity between two probability assignments. Let $\{P_i^1|i=1,\ldots,n\}$ denote one set of probability assignments for $n$ locations and $\{P_i^2|i=1,\ldots,n\}$ the other. The similarity $C_{sim}$ between the two sets is computed by
Table 3: The table of similarity coefficients, $C_{sim}$, for 9 random priors, each compared with equal prior assignments. The rows are for 4 different composite nodes of the PIN. P is Parallelogram, R is Ribbon, E is Ellipse, and C is Circle.

<table>
<thead>
<tr>
<th></th>
<th>$C_{sim}$ for 9 sets of random priors with equal priors</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.998 0.998 0.983 0.999 0.991 0.995 0.9943 0.9952 0.9995</td>
</tr>
<tr>
<td>R</td>
<td>0.999 0.998 0.999 1.0 0.996 0.998 1.0 1.0 0.998</td>
</tr>
<tr>
<td>E</td>
<td>0.998 0.999 0.995 0.984 0.987 0.995 0.998 0.997 0.995</td>
</tr>
<tr>
<td>C</td>
<td>0.999 0.999 0.998 0.995 0.995 0.997 0.998 0.999 0.995</td>
</tr>
</tbody>
</table>

$$C_{sim} = \frac{\sum_{i=1}^{n} p_{i}^1 p_{i}^2}{\|P^1\| \|P^2\|}$$

(5.5)

where the denominator is the product of the norms of the two probability vectors.

We started from 9 random assignments of prior probabilities and compared each result with that resulting from equal prior assignment. The results are shown in Fig. 24 and Table 3. The plots of Fig. 24 are of the final probabilities at a composite node versus the feature at different locations. Note that the relative assignments of the probabilities are preserved most of the time for different priors, although the absolute values vary. In other words, it is the shape of the plots that interest us, rather than their vertical positions. This is also validated by the calculated similarity values $C_{sim}$ for the 9 different priors. The consistently high values imply that the equal prior assumption is not constraining.
Figure 24: The robustness of the equal prior assignment. This shows the plot of the probability assignments of features at different locations for 9 random prior assignments. The features considered are (a) Parallelogram ($N23$) (b) Ribbon ($N13$) (c) Ellipse ($N6$) and (d) Circle ($N8$).
Table 4: The table of similarity coefficients, $C_{\text{sim}}$, for 9 random perturbations of the constants, $C_{(u_1,...,u_n)}$ compared with the chosen values. The rows are for 4 different composite nodes of the PIN. P is Parallelogram, R is Ribbon, E is Ellipse, and C is Circle.

<table>
<thead>
<tr>
<th></th>
<th>$C_{\text{sim}}$ for 9 random perturbations</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.990 0.993 0.980 0.999 0.999 0.987 0.998 0.992 0.999</td>
</tr>
<tr>
<td>R</td>
<td>0.999 1.0 0.999 0.999 0.999 1.0 0.999 0.999 0.998</td>
</tr>
<tr>
<td>E</td>
<td>0.978 0.979 0.976 0.992 0.999 0.999 0.988 0.997 0.991</td>
</tr>
<tr>
<td>C</td>
<td>0.996 0.996 0.982 0.996 0.990 0.993 0.996 0.999 0.996</td>
</tr>
</tbody>
</table>

Conditional Probability Constants

One of the components of the conditional probability at each node (see Eq. 5.2) is the function $g$ which in turn involves (see Eq. 5.3) a multiplicative $C_{(u_1,...,u_n)}$. These numbers have been chosen empirically from the interval $[0,1]$ and in this section we study the robustness of the choice. As a performance evaluation criterion we again consider the probability assignments of four salient nodes in the network, namely $N_{23}$ (Parallelogram), $N_{13}$ (Ribbon), $N_{6}$ (Ellipse), and $N_{8}$ (Circle). We added uniformly distributed (between -0.25 and 0.25) noise to the chosen values, constrained to the valid interval of $[0,1]$, and studied the final probability distributions. The similarity between this final probability distribution is again compared with that for the chosen constants using Eq. 5.5. We ran 9 different trials and the results are shown in Fig. 25 and Table 4.

The plots of Fig. 25 are of the probability of existence of features versus their location index for different random perturbations. From the plots we can readily
Figure 25: The robustness of the choice of the constants, $C(u_1, \ldots, u_n)$, involved in the specification of the conditional probabilities. This shows the plot of the probability assignments of features at different locations for 9 random perturbations of the constants. The features considered are (a) Parallelogram ($N_{23}$) (b) Ribbon ($N_{13}$) (c) Ellipse ($N_{6}$) and (d) Circle ($N_{8}$).
Figure 26: Functional form of the dependence of the conditional probabilities with $f_{geom}$. The exponential solid plot is the one that is chosen for the PIN. This is compared against the dot-dashed and dotted choices.

We observe the qualitative similarity of the relative probability assignments for different values of the constant, $C(u_1, \ldots, u_n)$. This is further reinforced by the quantitative similarity index, $C_{sim}$, shown in Table 4 which compares the similarity of the probability assignments for 9 noise added $C$ values with that for the chosen values. Thus, we infer that the results of the PIN are indeed robust against perturbations of the values, $C(u_1, \ldots, u_n)$. 
Table 5: The table of similarity coefficients, $C_{\text{sim}}$, for 9 random perturbations of the constants, $C_{(u_1, \ldots, u_n)}$, compared with the chosen values. The rows are for 4 different composite nodes of the PIN.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>$C_{\text{sim}}$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dot-dashed function</td>
<td>Dotted function</td>
<td></td>
</tr>
<tr>
<td>Parallelogram</td>
<td>0.9923</td>
<td>0.9905</td>
<td></td>
</tr>
<tr>
<td>Ribbon</td>
<td>0.9983</td>
<td>0.9968</td>
<td></td>
</tr>
<tr>
<td>Ellipse</td>
<td>0.9856</td>
<td>0.9173</td>
<td></td>
</tr>
<tr>
<td>Circle</td>
<td>0.9829</td>
<td>0.9512</td>
<td></td>
</tr>
</tbody>
</table>

**Functional Dependence of the Conditional Probabilities**

The definition of $g$ in the conditional probability expression (Eq. 5.2) also involves a function involving a geometric term, $f_{\text{geom}}$ (see Eq. 5.3). We have chosen an exponential function for our purpose (see solid plot in Fig. 26). The function maps values from $[0, \infty)$ to the range $[0, 1)$. To study the dependence of the PIN on this functional form we employ the same performance evaluation scheme as before, namely considering the final probability distribution of some salient nodes for different functional forms. We have considered two different piecewise linear functional forms which saturate at one as shown in Fig. 26. The dot-dashed line and the dotted line define a boundary for other possible monotonic mappings which can be expected to have more amenable behavior than the bounding piecewise linear mappings.

The final probability plots are shown in Fig. 27. The dot-dashed and the dotted plots correspond to the dot-dashed and dotted functions shown in Fig. 26. From the plots we can see that the qualitative similarity between the relative probabilities
Figure 27: Performance plots depicting the flexibility of specification of the functional dependence of the conditional probabilities with $f_{geom}$. The features considered are (a) Parallelogram ($N23$) (b) Ribbon ($N13$) (c) Ellipse ($N6$) and (d) Circle ($N8$).
is significant, although not so much as we saw in the case of the other parameters. The associated qualitative similarity index, $C_{\text{sim}}$, is shown in Table 5. The similarity values are for the dot-dashed and dotted functions with the chosen exponential function. Note the high values of the index. For the ellipse composite node ($N6$) the indices seem to vary more (0.9173 - 0.9856) than previously encountered. Considering the wide range of distortion we have considered on our original function, the performance is quite stable.
CHAPTER VI

Structuring a Perceptual Inference Network

"Probability theory is not really about numbers;
it is about the structure of reasoning."

_G. Shafer_

In this chapter we present a theoretical formalism to structure perceptual inference networks instead of manually designing them as in the last chapter. Since PINs are a form of Bayesian network we might be tempted to use algorithms to automatically structure Bayesian Networks like in [115, 126] from pairwise joint probability distributions or pairwise correlation data. However, we cannot easily acquire such information about our random variables since for spatial reasoning these variables are typically objects, parts or organizational relations. The primary heuristic used to structure Bayesian Networks in computer vision has been the notion of "causality" based on PARTS-OF relation derived from an hierarchical description of an object [117, 127] or from higher level context of a scene [121]. The parts of an object are the nodes and the links are directed into the parts. However, such notions lead to multiply connected (non tree structured) networks when we consider a _set of objects_ or structures. The problem of reasoning on a general non-tree structured
Bayesian Network is \(NP\)-hard \([125]\). Agosta \([127]\) explores the use of such multiple connected network in visual tasks. However, we avoid the combinatorial problem by insisting on a tree structure.

Our aim is to design a Bayesian Network (or specifically a PIN) to enable us to integrate information about a set of elementary feature forms to hypothesize about a set of target object hypotheses \(^1\). We also suggest a way to estimate the numeric conditional probabilities of the network. Since, our decisions involve object descriptions, we use the concept of Random Parametric Structural Descriptions (RPSDs) and entropic distance measures among them to formulate the solution to the structuring problem.

The starting point of the PIN structuring algorithm is the definition of a set of target organizations which constitute a level of the description hierarchy and a set elementary groupings about which we can provide direct evidence using the preattentive modules. The PIN provides us a means to reason about the target organizations based on the evidence for the elementary groupings. The algorithm starts from the target organizations and iteratively constructs intermediate organizations which are simpler than the starting ones till we reach the elementary groupings. At each stage, as we shall see, the organizations are modeled as Random Parametric Structural Descriptions (RPSDs).

\(^1\)We use the terms objects, parts, structures, groupings, and organizations rather loosely. The notion of parts and objects are relative to the application. The terms structures, groupings, and organizations are used in the context of perceptual organization to denote entities exhibiting some degree of regularity. Since our application area is perceptual organization, the "objects" of interest are structures, groupings, or organizations.
Section 6.1 overviews our structuring algorithm. We present the formalism of a Random Parametric Structural Description (RPSD) in Section 6.2. Sections 6.3 and 6.4 discuss how the PINs are structured and conditional probabilities estimated. We present the performance characterization of our algorithm in Sections 6.5 and 6.6.

6.1 Overview of the Algorithm

The structure of the PIN reflects the dependencies among organization types. Our problem is to find such a dependency structure among the chosen set of salient groupings or organizations with the constraint that the network be tree structured (ignoring link directions). Fig. 21 depicts the manually structured PIN. Note how we have tried to keep similar structures in the same branch of the tree. The conditional probabilities were also chosen by hand. We designed this PIN using the following heuristics.

- The direct dependencies among groupings are represented by links going from groupings exhibiting less organization to groupings having more organization; the grouping with less organization is a part of the grouping with more organization. Note that we differ on the link directions from other prevalent views which would direct links from groupings with more organization to those with less organization, or from objects to parts. We consider the object as an abstraction of a relation among the parts, which are its primitives. The primitives, or parts, are assumed to be independent while the relations, or objects, are assumed to be conditionally independent given the primitives. As we shall
see in Section 6.2.1, this assumption is consistent with the well established theory of structural descriptions [122, 51, 123] which also assumes the conditional independence of the relations given the primitives. We use this notion of conditional independence of relations to decide on the link direction: from part to object.

- The distance (in terms of links) between two nodes of the PIN representing two geometric figures should be low if the groupings are very similar. For example, the rectangle node and the trapezoid node are closer in the PIN than the rectangle node and the circle node. This should ensure that the size of the PIN remains small.

In this work we formalize these heuristics and offer an algorithm for PIN construction. In addition to the heuristics, we impose the constraint that the PIN nodes have similar neighborhood structure. That is, the number of parents and children of each node are to be approximately constant throughout the network. This constraint should prove useful in hardware implementation where node (processor) uniformity is an advantage. We enforce the simplest neighborhood structure of at most two parents and two children.

The overall PIN structuring algorithm is depicted in Fig. 28. It constructs the PIN iteratively, level by level. The starting point of the algorithm is a set of target organizations. At each iteration it constructs, from the present level, the next (above) level of organizations which are simpler than those of the present level. In keeping with our first heuristic, the organizations formed for the next level are the
Input set of target organizations

Compute the similarity graph based on entropic measures over the organizational models of interest. The measure reflects structural similarity between organizations.

Find a maximal covering of the similarity graph. This gives optimal pairwise similar organizations.

Merge the similar organizations preserving the non-null mappings. The merged models form a parent.

Form the second parent of each model by considering the null mappings.

Yes

Construct the next level

Any more models to be merged?

No

Input set of elementary organizations

Decompose the root nodes into elementary organizations which can be readily detected.

Figure 28: Flowchart of the PIN structuring algorithm.
parents or the causes of the organizations in the present level. The next level is constructed in two steps: dependency detection and next level formation.

In the dependency detection step we determine the organizations that are structurally similar and, thus, can be considered to have a common parent. This tends to keep the nodes representing similar organizations close to one another in the network, in accordance with our second heuristic. The first step in dependency detection is the construction of a similarity graph whose nodes represent the current level organizations and whose links are weighted with a structural similarity measure. The second step determines the groups of similar organizations. Since we impose the constraint that each node in the PIN can have at most two children and two parents, we search for similar groups of size two. As we shall see, this search can be transformed into the graph theoretic problem of maximal cover.

The next level formation is concerned with building the actual structure of the parents. The structurally similar pairs are merged to form a common parent having the characteristics present in both organizations. The second parent of each organization consists of parts which are not in the first (common) parent but are a part of the organization. This idea will get clearer as we develop it.

The above two steps of dependency detection and next level formation are embedded in an iteration loop which ends when organizations can no longer be merged. The root nodes are then expanded into a set of elementary organizations for which we can provide direct evidence. Thus, the network grows from its target organizations to its elementary (source) features.
We need to represent the organizations concisely to quantify the concept of structural similarity between them and to construct the parents of similar organizations. We use the formalism of Random Parametric Structural Descriptions (RPSDs) [51, 123], discussed in the next section, for this. In Section 6.3 we discuss the PIN structuring algorithm in detail.

6.2 A Structural Representation of Organization

As we saw in the previous section, a similarity measure between organizations is needed to determine those which should have a common parent. To formulate such a measure we have to represent organizations in a concise manner. We also need the ability to form the next level of the PIN by constructing the representations of a parent organization from those of its children. The optimal representation would exhibit the following properties:

1. The representation should be able to handle imprecise specifications of organizations. For example, a parallel organization can be defined to consist of two almost straight lines whose slopes are within a tolerance and which has significant overlap; we should be able to represent such a structure. This type of representation will make the system robust to noise and occlusion problems.

2. It should be easy to define a meaningful quantitative similarity measure between two representations.

3. There must be a methodology to infer parent representations from those of the children.
4. The representations must be explicit (declarative) so that they can be easily modified.

The formalism of Random Parametric Structural Descriptions (RPSDs) has all the above properties; this section presents the fundamentals. Because the present discussion on structural descriptions is brief, the reader is referred to [51, 123] for details. The reader is also requested to bear with the mathematics; they are essential to justify the mathematical expressions used in our PIN structuring algorithm. The most important points of the discussion are the form of the entropic similarity measure between RPSDs and the manner in which a merged RPSD is formed from two others. The former is used as the similarity measure between two organizations and the latter is used to construct the parent representations at each iteration of the PIN structuring algorithm. Section 6.3 discusses, in greater depth, the application of these RPSD concepts in the structuring algorithm.

6.2.1 The Random Parametric Structural Description

A random parametric structural description (RPSD) is a pair, \( G = (P, R) \), in which \( P \) is a set of random primitives and \( R \) is a set of random parametric relations. A random primitive, \( P \), is a primitive feature in the organization and is characterized by a set of attributes, where the value of each is a discrete random variable. Each primitive also has associated with it a probability of nullity. This is the probability that the primitive is absent from the structure altogether. A random parametric relation, \( R \), is a named N-ary parametric relation over the random primitive set such that the parameter values of the attributes associated with each N-tuple are
discrete random variables. Also associated with each $N$-tuple is a probability of nullity. That is, the probability of its being absent from the relation altogether. In general, $\mathcal{R}$ is a set of named relations $\{\mathcal{R}_1, \cdots, \mathcal{R}_k\}$, each of a different type and characterized by different parameters.

In the context of perceptual organization, the primitives are the tokens to be organized e.g., constant curvature segments of the edge segments. The relations of interest are closure, continuity, and symmetry. An organization is represented by a structural description using these primitives and relations. The RPSD formalism lets us relax the definitions of the relations and the primitives by defining their attributes to be random variables taking on a range of values. For example, a relaxed definition for parallel lines is possible by considering the parallelism relation to be a random parametric relation with the angle between the lines being a random attribute with a probability mass function centered about zero.

For mathematical convinience we make the following assumptions which can be assured (at least approximately) by proper system design:

- The random primitives are mutually independent. The values of distinct primitive attributes are also independent. The parameter values associated with a tuple in a given relation are independent of the values assigned to any other tuple in that relation, and to any other tuple in any other relation.

- The \textit{a priori} probability of nullity is known for each primitive and for each relation tuple in which none of the constituent primitives is null. The probability of nullity for a given relation tuple having null constituents is one.
A particular outcome event of an RPSD is a Parametric Structural Description (PSD). A PSD is a deterministic pair $G = (P, R)$ with $P$ representing the set of primitives and $R$ representing the set of named parametric N-ary relations \{$R_1, \ldots, R_k$\} over $P$. The primitives are characterized by their attribute values and the relations by their parameter values. The RPSD corresponds to a priori world knowledge, while the deterministic PSD describes a particular observation. Thus, an observed set of parallel lines would be represented as a PSD; a specific outcome of the RPSD describing parallelism.

PSD Outcome Probability

We define the similarity between two RPSDs using an entropic measure (developed later) requiring knowledge of the probability distribution of the entire RPSD. To compute the probability of a specific PSD outcome, the interprimitive mapping function $h : P \rightarrow \mathcal{P} \cup \emptyset$, stored as a matchlist, is required. This associates with each member of the primitive set from the observed PSD a single random primitive from the RPSD. We discuss the construction of the mapping in Section 6.2.4; for now suppose it is given. For notational convenience, we assume that the primitives of the PSD and RPSD are indexed such that $h(p_i) = p$, with $p_i \in \mathcal{P}$ and $p_i \in P$.

The probability of a given outcome structure under a mapping $h$ is given as:

$$Pr_h[G] = Pr_h[P]Pr_h[R|P] \quad (6.1)$$

Note that the inverse mapping may be incomplete. That is, it may be that $h^{-1}(p) = \emptyset$ for some $p \in \mathcal{P}$. This corresponds to a null outcome for random primitive $p$, indi-
cating that a portion of the organization was not observed (occluded or whatever). For the sake of convenience, we denote the event in which the outcome of a primitive $p_i$ is null mathematically as $p_i = \emptyset$ and the complementary event as $p_i \neq \emptyset$.

The probability of a non-null outcome primitive assuming attribute independence is given by:

$$\Pr_h(p_i) = \prod_j \Pr[a_j(p_1) | p_i \neq \emptyset]$$

(6.2)

where $a_j$ denote the $j^{th}$ attribute of the primitive $p_1$. The probability of the entire set of primitives as a joint event is the product of the probabilities of the null and the non-null outcomes:

$$\Pr_h(P) = \prod_{i \in h^{-1}(p_i) = \emptyset} \Pr[p_i = \emptyset] \times \prod_{i \in h^{-1}(p_i) \neq \emptyset} \prod_j \Pr[a_j(p_1) | p_i \neq \emptyset] \Pr[p_i \neq \emptyset]$$

(6.3)

To compute the probability of $R$ as an outcome of $\mathcal{R}$, conditioned on $P$, we form the composition of the outcome tuples with $h$ to produce a set of image tuples of random primitives: $R_k \circ h = \{(p_1, \ldots, p_n) | (p_1, \ldots, p_n) \in R_k\}$. We call these inferred tuple pairs as relation pairs or matches. In the composition we retain the parameter values assigned to each nonnull observation tuple. The quantity $\Pr_h[R_k | P]$ represents the probability that all null tuples (tuples in the random relation having no counterpart in the outcome relation) are indeed null, and that all nonnull tuples are nonnull and possess their assigned parameter values. Let $r_i$ be
a member tuple in random parametric relation $\mathcal{R}_k$ and let $f_j$ be a parameter value assigned to $r_i$, the inverse image of the tuple $r_i$. Our foregoing assumptions along with mutual independence assumptions of the named relations (again, by design) now allow us to write:

$$
\Pr_h[R|P] = \prod_k \Pr_h[R_k|P]
$$

(6.4)

$$
= \prod_k \prod_{i \not\in p, r_i = \emptyset} \Pr_h[r_i = \emptyset] \prod_{i \not\in p, r_i \neq \emptyset} \Pr_h[f_j(r_i)|r_i \neq \emptyset] \Pr_h[r_i \neq \emptyset]
$$

(6.5)

**RPSD Entropy**

The uncertainty in the probabilistic description of an RPSD can be quantified by its entropy which is written as the sum of primitive and relational entropies [123]:

$$
H[\mathcal{G}] = H[\mathcal{P}] + H[\mathcal{R}|\mathcal{P}].
$$

This concept is used in the next section to formulate the structural similarity measure. Using the independence assumptions we have:

$$
H[\mathcal{G}] = \sum_{p \in \mathcal{P}} H[p] + \sum_k \sum_{r_i \in \mathcal{R}_k} H[r_i]
$$

(6.6)

$H[p]$ and $H[r_i]$ represent the entropy of random primitive $p$ and random tuple $r_i$, given no constituents null, respectively. So,

$$
H[p] = -\Pr[p = \emptyset] \log \Pr[p = \emptyset] - (1 - \Pr[p = \emptyset]) \log \{(1 - \Pr[p = \emptyset])
$$

$$
+ (1 - \Pr[p = \emptyset]) \sum_n \sum_j \Pr[a_j(p) = v_n] \log \{\Pr[a_j(p) = v_n]\}
$$

(6.7)
The entropy of a random tuple follows similarly. Let $C$ represent the event that all constituents of tuple $r_i$ are nonnull and $c_i = p[C]$ its probability. Then,

$$H[r_i] = -c_i \Pr[r_i = \emptyset|C] \log \{\Pr[r_i = \emptyset|C]\}$$

$$-c_i (1 - \Pr[r_i = \emptyset|C]) \log \{1 - \Pr[r_i = \emptyset|C]\}$$

$$-c_i (1 - \Pr[r_i = \emptyset|C]) \sum_j \sum_n \Pr[f_j(r_i) = u_n|C] \log \{\Pr[f_j(r_i) = u_n|C]\}$$

where \{u_n\} represents the range of parameter functions $f_j$.

6.2.2 Dissimilarity Measure

Following Wong and You [122] and Sengupta and Boyer [123], we use the concept of increment of entropy as a measure of the structural dissimilarity between two RPSDs. Given two RPSDs $G_1$ and $G_2$ corresponding to two organizations, $M_1$ and $M_2$, these two RPSDs are merged to yield RPSD $G_m$ representing a merged organization $M$. The dissimilarity function $\text{DISS} (G_1, G_2)$ is:

$$\text{DISS} (G_1, G_2) = H(G_m) - \frac{\Pr[M_1]H(G_1) + \Pr[M_2]H(G_2)}{\Pr[M]}$$

Note that $\Pr[M_1]$ and $\Pr[M_2]$ are the a priori appearance probabilities of organizations $M_1$ and $M_2$, respectively, and $\Pr[M] = \Pr[M_1] + \Pr[M_2]$. It can be shown [123] that $\text{DISS} (G_1, G_2) < \Sigma_r (1 + a_r) + \Sigma_p (1 + a_p)$ where $a_r$ and $a_p$ are
the number of attributes of the $i^{th}$ merged relation and $j^{th}$ merged primitive. The summations are defined over all the merged relations and primitives. The measure, $DISS$, increases with the dissimilarity of the RPSDs. Thus a structural similarity measure, $SSM$ in $[0, 1]$ can be formulated as follows:

$$SSM(G_1, G_2) = \left(1 - \frac{DISS(G_1, G_2)}{\sum_{r_i}(1 + a_{r_i}) + \sum_{p_i}(1 + a_{p_i})}\right) \frac{N - N_0}{N}$$  (6.10)

The second multiplicative term represents the ratio of the number of non-null primitive matches and relation pairs $^2$, $N - N_0$, to the total number of matches, $N$, between the RPSDs. Without this term, the entropic measure is meaningful only for comparing RPSDs which are similar in size in terms of the total number of primitives and relations. The match between two RPSDs differing greatly in their graphical size will produce many null mappings. In this case the second term ensures that the similarity measure is very low. Indeed if none of the primitives are matched then the value of the structural similarity measure is zero.

To compute the similarity we need to form the merged RPSD, $G_m$. This involves two steps: constructing the correct interprimitive mapping between the two RPSDs, $G_1$ and $G_2$, and computing the necessary probabilities of the merged models primitives and relations. We seek the mapping which minimizes the increase in entropy(Eq. 6.9).

Constructing the interprimitive mapping is NP hard, but the approximate algorithms developed in [123, 128] run faster in the expected case. However, they

---

$^2$Remember that the relation pairs refer to the non-null tuples formed by composition of the relations with the interprimitive mapping function.
give monomorphic matches. We present in Section 6.2.4 a new N-ary mapping algorithm which gives a many to one mapping between the primitives sets. But first, we consider how the probabilities of the merged models are computed for a given mapping.

6.2.3 Structure of the Merged RPSD, \( \mathcal{G}_m \)

The aim of merging the RPSDs, \( \mathcal{G}_1(\mathcal{P}_1, \mathcal{R}_2) \) and \( \mathcal{G}_2(\mathcal{P}_2, \mathcal{R}_2) \), is to obtain a synthesis \( \mathcal{G}_m(\mathcal{P}_m, \mathcal{R}_m) \) which is a general random description encompassing the variations of both. To compute the merged description we assume, for now, the existence of a mapping between the two RPSDs expressed in the form of a pairwise listing (the matchlist) of corresponding primitives and inferred relation pairs. The unmatched primitives and relations are paired with a NULL primitive (a primitive with probability of nullity one). We note that, although in the present context the structure of the merged RPSD is needed to compute the similarity measure \( SSM \), these ideas are also used in the PIN structuring algorithm to construct the representations of the parents.

We form a merged, or synthesized, primitive (relation) corresponding to each pair of matched primitives (relations). (Note that we construct the matchlist between the primitives and the relation pairs are inferred by composition.) For example, for the matchlist \( \{(p_1, p'_1), (p_2, p'_1), (r_1, r'_1), (NULL, p'_2)\} \) the merged description will have three random primitives and one relation. For a many to one mapping, as are the first two maps in this case, we have to ensure that the independence assumption between the primitives holds for the merged RPSD. We cannot form merged
primitives directly corresponding to \((p_1, p'_1)\) and \((p_2, p'_2)\) because they will not be independent. As a solution, we physically divide the primitive \(p'_1\) into \(M\) equal parts where \(M\) is the mapping cardinality of the primitive. Thus, for this example we divide \(p'_1\) into two \((M = 2)\) parts \(p'_{1d}\) and \(p''_{1d}\) and we consider the 2:1 mappings as two 1:1 mappings: \((p_1, p'_{1d})\) and \((p_2, p''_{1d})\). A separate merged primitive is formed for each of these 1:1 mappings.

The probability distributions of the attributes of divided primitives are constructed according to the attribute type. For example, a random variable representing the length of the divided primitives, \(l_M\), is related to the length of the original primitive, \(l\), by: \(l_M = l/M\) and the probability distributions for the length attribute of the divided parts are \(Pr(l_M = L) = MPr(l = LM)\). However, for attributes like curvature the probability distributions remain the same as that for the undivided one.

The merged primitives are defined as follows. Suppose that the primitives \(p^1\) from \(G_1\) and \(p^2\) from \(G_2\) are to be merged to give primitive \(p\) of \(G_m\). Then it can be shown [123] that the probability of nullity of the merged primitive, \(\eta_p\) is:

\[
\eta_p = \frac{Pr[M_1]\eta_{p^1} + Pr[M_2]\eta_{p^2}}{Pr[M]} \tag{6.11}
\]

where \(\eta_{p^1}\) and \(\eta_{p^2}\) are the probabilities of nullity of \(p^1\) and \(p^2\), respectively. Next we need expressions for the probability densities for the attributes assigned to the merged primitive. Let the outcome of the \(i\th\) attribute for the primitives \(p, p^1\), and \(p^2\) be denoted be \(a_i(p), a_i(p^1)\), and \(a_i(p^2)\), respectively. Then:
\[
Pr(a_i(p) = u_n) = \frac{\sum_{j=1}^{2} Pr[M_j](1 - \eta_{p_i})Pr(a_i(p^j) = v_n)}{Pr[M](1 - \eta_p)} \quad (6.12)
\]

We have similar expressions for relation merging. Suppose we merge relations \( r^1 \) and \( r^2 \) to form \( r \). Let the constituent primitives of \( r^1 \) be \( p_1^1, \ldots, p_n^1 \) and those for \( r^2 \) be \( p_1^2, \ldots, p_m^2 \). Note that, in general the number of constituent primitives of the matched relations might be different, as for the closure relation. The merged relation, \( r \), will be over the merged primitives, \( p_1, \ldots, p_{\text{max}(m,n)} \). The subscript \( \text{max}(m,n) \) is a consequence of the manner in which we merge the primitives; not only do we form a merged primitive for every one to one primitive match but we construct \( N \) merged primitives for an \( N \) to 1 match. The probability of nullity of the merged relation is given by:

\[
\eta_r = \frac{\eta_{r^1}Pr[M_1](1 - \eta_{p_1^1})\cdots(1 - \eta_{p_n^1}) + \eta_{r^2}Pr[M_2](1 - \eta_{p_1^2})\cdots(1 - \eta_{p_m^2})}{Pr[M](1 - \eta_{p_1})\cdots(1 - \eta_{p_{\text{max}(m,n)}})} \quad (6.13)
\]

The probability of nullity of the merged relations is weighted sum of the probabilities of nullity of the matched relations; the weights are determined by the probabilities of existence of the primitives constituting the relations.

The probability density for the \( i \)th parameter, \( f_i(r) \), of the merged relation is given by:

\[
Pr(f_i(r) = u_n) = \frac{\sum_{j=1}^{2} Pr[M_j](1 - \eta_{r^1})(1 - \eta_{p_1^1})\cdots(1 - \eta_{p_{(m,n)}^1})Pr(f_i(r^j) = u_n)}{Pr[M_1](1 - \eta_{p_1})\cdots(1 - \eta_{p_m}) + Pr[M_2](1 - \eta_{p_1^2})\cdots(1 - \eta_{p_{\text{max}(m,n)}})} \quad (6.14)
\]

It is the weighted sum of the probability distributions of the matched relations with weights, again, determined by the probability of existence of the primitives constituting the relations.
6.2.4 Constructing the Mapping between RPSDs

As mentioned earlier, we need to construct a mapping between the primitive sets of two RPSDs to compute the synthesized RPSD. The optimal mapping is that which gives the least increase in entropy (Eq. 6.9). Since the entropy of an RPSD can be expressed as the sum of the entropies of the primitives and the relations (Eq. 6.6) we can express the dissimilarity measure as the sum of the entropy changes associated with merging each primitive match and relation pair (inferred by composition). This suggests a recursive formulation of the mapping problem (Fig 29).
Before going into the details of the algorithm, we define a few terminologies.

- $L^{(N)}$ is the set of partial matches between the primitives of the two RPSDs, $G_1$ and $G_2$, at the $N^{th}$ recursion stage. At the start of the recursion we have $N = 0$ and $L^{(0)} = \emptyset$.

- $G_1^{(N)}$ is the subgraph of $G_1$ comprised of the primitives not in $L^{(N)}$. Every relationship tuple in $G_1$ which is also a subset of the primitives of $G_1^{(N)}$, is a tuple of $G_1^{(N)}$. In other words, $G_1^{(N)}$ is the unmatched part of $G_1$ with the matching defined by $L^{(N)}$. At start, $G_1^{(0)} = G_1$. We have similar expressions for $G_2$.

- $G_1^{(N,p)}$ is the part of $G_1^{(N)}$ consisting of the primitive $p$ ($p \in G_1^{(N)}$) and all relation tuples of $G_1^{(N)}$ which contain $p$. We call $G_1^{(N,p)}$ the fractional structural description of $G_1^{(N)}$ with respect to the primitive $p$. We have similar expressions for $G_2$.

- $MP$ (the matchpool) is the set of possible primitive matches not in $L^{(N)}$.

- In a structural description, two primitives are neighbors if they appear together in a tuple of any invoked relation.

At stage $N$ of the recursion we have a partial list of primitive matches, $L^{(N)}$, which defines the subgraphs $G_1^{(N)}$ and $G_2^{(N)}$ not yet matched. The cost of computing the match between $G_1^{(N)}$ and $G_2^{(N)}$ is the structural dissimilarity between them (Eq. 6.9) and this can be recursively computed as follows (also see Fig. 29):
\[
\text{DISS}(\mathcal{G}_1^{(N)}, \mathcal{G}_2^{(N)}) = \min_{(p_1^1, p_2^1) \in \mathcal{MP}} \left[ \text{DISS}(\mathcal{G}_1^{(N,p_1^1)}, \mathcal{G}_2^{(N,p_2^1)}) + \text{DISS}(\mathcal{G}_1^{(N+1)}, \mathcal{G}_2^{(N+1)}) \right]
\]

(6.15)

where \((p_1^1, p_2^1)\) is a possible matchpool pair. The dissimilarity between two RPSDs (or their subgraphs), \(\mathcal{G}_1^{(N)}\) and \(\mathcal{G}_2^{(N)}\), is the minimum of the sum two dissimilarity measures. The first is the dissimilarity between the fractional structural descriptions of \(\mathcal{G}_1^{(N)}\) and \(\mathcal{G}_2^{(N)}\) with respect to potential primitive pair \((p_1^1, p_2^1)\) where \(p_1^1 \in \mathcal{G}_1^{(N)}\) and \(p_2^1 \in \mathcal{G}_2^{(N)}\). This term is determined by the primitive match \((p_1^1, p_2^1)\) and tuple pairs inferred by composition using the matchlist \(L^{(N+1)}\). And the second term is the dissimilarity between \(\mathcal{G}_1^{(N+1)}\) and \(\mathcal{G}_2^{(N+1)}\) which are the subgraphs of \(\mathcal{G}_1^{(N)}\) and \(\mathcal{G}_2^{(N)}\) excluding the primitives \(p_1^1\) and \(p_2^1\), respectively. \(\mathcal{G}_1^{(N+1)}\) and \(\mathcal{G}_2^{(N+1)}\) are defined by the matchlist at stage \(N+1\) given by: \(L^{(N+1)} = \{L^{(N)} \cup (p_1^1, p_2^1)\}\). For the second term we recursively use Eq. 6.15.

The recursion stops when \(\mathcal{G}_1^{(N)} = \emptyset\) and \(\mathcal{G}_2^{(N)} = \emptyset\). At the start, \(N = 0\), \(L^{(0)} = \emptyset\), \(\mathcal{G}_1^{(0)} = \mathcal{G}_1, \mathcal{G}_2^{(0)} = \mathcal{G}_2\). On termination, the best matchlist is constructed by the union of the best matchpool pair at each stage.

Note that, if the minimum in Eq. 6.15 is computed only over the first term then we implement the hill climbing strategy. The adopted strategy is more like beam search. The computational complexity of this method is exponential since we have to investigate branches corresponding to all possible matchpool pairs available at each stage of the recursion. To cut down on the computational costs we restrict the matchpool to a constant number of relationally consistent matches, \(b\), selected according to the following heuristics. The most likely type of mapping between the
primitives of RPSDs is one to one. A many to one mapping is less likely, followed by NULL mapping. The primitives which are neighbors (see above definitions) of the primitives in $L^{(N)}$ are good candidates for the matchpool since they have associated constraints and the chance of backtracking on them is low. This consideration of neighbors makes the RPSD mapping algorithm akin to crystal growing.

These heuristics are implemented as set of five filters run in sequence at each stage, $N$, of the recursion to produce the matchpool $MP$ relationally consistent with the matched list $L^{(N)}$. If one filter does not produce a matchpool we go to the next. If none of the filters hypothesize a match then $MP = \emptyset$ and the recursion ends. The matchpool filters are:

- **Filter 1**: Choose *one to one* consistent matches between primitives which are neighbors of those in $L^{(N)}$.
- **Filter 2**: Choose *one to one* consistent matches between primitives which are *not neighbors* of those in $L^{(N)}$.
- **Filter 3**: Choose *many to one* consistent mappings between primitives which are neighbors of those in $L^{(N)}$.
- **Filter 4**: Choose *many to one* consistent mappings between primitives which are *not neighbors* of those in $L^{(N)}$.
- **Filter 5**: Choose the *null pairing* of unmatched primitives and relations. The relation null maps are returned when we have exhausted the primitive null maps.
At startup, Filter 2 is activated because the Filter 1 produces a NULL matchlist. Filter 1 just considers the neighbors of $L^{(N)}$ and at start $L^{(0)} = \emptyset$. The matchpool pairs created by Filter 2 are recursively grown, as discussed before, and the best grown match is selected. The process is analogous to beam search and is efficient but does not guarantee the globally optimum solution. However, we have found the sub-optimal solution to be acceptable for our test cases.

To summarize, in this section we outlined the graph based, probabilistic RPSD formalism used for representing the organizations in the PIN. Among the advantages of RPSD representations is the ease of defining a similarity measure between two RPSDs (Eq. 6.10) based on structural entropies. In subsection 6.2.3, we outlined the procedure to compute a merged RPSD representation from the mapping between the primitives of two RPSDs. This procedure is used by the PIN structuring algorithm to construct the intermediate levels of the PIN.

6.3 Structuring the PIN

As sketched in Section 6.1, the PIN structuring algorithm builds the PIN layer by layer. PIN construction can be broken up into three parts: the dependency detection among groupings, the construction of the next level, and the decomposition of the root nodes (Fig. 28). The first two steps, dependency detection and next level formation, are embedded in an iterative loop and the last step, root node decomposition, is executed when the iteration ends. We start the iterative process with a set of target organizations and construct its parent structures to form the next level. This process continues until we can no longer form new parents. A
Figure 30: A typical output PIN structure before the root node decomposition.

typical output of the PIN structuring algorithm at this stage is shown in Fig. 30. For the present, please ignore the shaded distinctions between the PIN nodes. The root nodes are then broken into a set of elementary organizations for which we can provide direct evidence. All the organizations in the PIN are represented by RPSDs and the entropic measure $SSM$ is used to group similar organizations.

6.3.1 Dependency Detection

We would like similar organizations to be close (in the graph sense) to one another in the PIN structure. To ensure this, at each iteration the algorithm groups similar organizations at a level and assigns a common parent to them. The formulation of the structural similarity measure ($SSM$, Eq. 6.10) is useful in this regard. We
represent each organization as an RPSD and the SSM between two RPSDs quantifies the structural similarity between the two corresponding organizations. We seek groupings of organizations which are high in this measure. Recall the imposed constraint of two parents and two children for each PIN node, to ensure structural uniformity across the PIN. Because of this we are interested only in pairs of organizations which are similar. However, in the process of determining them we must take care to maintain the tree structure of the network.

The parents of nodes at the current level form the next (higher) level; we cannot pair organizations which might form cycles (ignoring the link directions) in the PIN. For example, in Fig. 30 we cannot pair nodes 11 and 13 at Level 2 to form Level 3. To distinguish between such valid and non valid pairing at a given level we consider the nodes in the connected components of the PIN as grown to that level. In Fig. 30, nodes 1, 2, 8, 9, and 10 form one connected component at Level 2. Clearly, if we pair nodes from the same connected component we will get cycles; we avoid pairing nodes in the same connected component. Note that, the iterative loop terminates when we have just one connected component.

We cast the search for the similar organization pairs as the maximal cover problem from graph theory by forming a similarity graph whose nodes represent the connected components at a particular level. The links of the similarity graph are quantified by the maximum entropic similarity measure (Eq. 6.10) between the corresponding organization RPSDs. We also impose the constraint that based on the primitive matches between the RPSDs we be able to infer (by composition) at least
one non-null relation pair. As we shall see, this ensures that we can form a meaningful common parent; a common parent has characteristics common to both children.

The problem of determining the optimal pairings of organizations to form the next level can now be cast as finding the maximal covering of the similarity graph, i.e. finding a set of links of the similarity graph with the maximum total weight such that each node of the similarity graph is covered by at most one link. We find the optimal covering by recursively enumerating all the possibilities in a complete graph and choosing the cover with maximum total weight. Note that in the case of an odd number of connected components we will have one isolated node; the corresponding unmatched nodes are carried over to the next level of the PIN.

6.3.2 Next Level Formation

The organizations at the next level (the parents of the present level) represent the inferential “causes” of those in the present level. Evidence for the parent organizations lets us make inferences concerning the children. For each optimal pair of organizations (as above) we construct a common parent to capture the substructure common to them and a pair of second parents to capture the features unique to each. The common substructure is extracted using the optimal matchlist constructed when we computed the entropic similarity measure between the organization RPSDs. The primitive matches and non null relation pairings (non-null results of the composition of the interprimitive mapping between the RPSDs with the relational tuples) determine the common part between the two organizations. The mergings of the non null paired relations and the primitives formed by merging those participating
in the non-null paired relations form the RPSD of the common parent. The merging algorithm is the same as that in Section 6.2.3.

The second parent of each organization represents the part which is characteristic to the organization and is not captured by the common parent. This is again constructed using the matchlist but now we merge the null paired relation tuples (relation tuples with null outcomes for composition with the best interprimitive mapping function between the RPSDs) and their primitives. If there are no relation tuple is paired with NULL then the second parent is NULL.

All the parents of the groupings at a particular level compose the next level of the PIN. The process of merging and parent construction is repeated for the next level. The iteration terminates when it can no longer merge any nodes without violating the tree structure of the PIN.

6.3.3 Root Node Decomposition

The PIN structuring algorithm outlined so far will produce a network as shown in Fig. 30. The first parents are shown in dark and the second parents are shown in a lighter shade. The arrow directions on the links denote the dependency structure; from the parent to the child. The root nodes have no links entering them and can be termed the root causes of all the structures or organizations. Since we recursively extract substructures of the organizations, the nodes grow simpler with increasing level. In particular, the root nodes tend to be simpler than the initial groupings. To start the reasoning process we have to provide evidence for (usually) the root nodes of the PIN. However, the evidence we can provide may be in terms of still
Figure 31: A typical output PIN structure after the decomposition of root nodes into elementary groupings. Some root node decompositions are not drawn to preserve clarity. The nodes with two incoming arrows denote a root node decomposed structure which is not drawn.

more elementary structures than the root nodes we have at this point. We need to infer the root node groupings from these elementary forms.

Suppose we have $n$ elementary structures, $E_1, \ldots, E_n$, for which we can easily provide evidence. We need to construct evidence for a root node, say $R_i$, from these elementary structures. To achieve this in the framework of the PIN, we decompose the root node to form a directed acyclic tree with the root node as the leaf (see
The decomposition is done recursively. We try to construct two parents or causes of the root node, \( R_i \), with an elementary grouping as one of the parents. To search for the elementary grouping which is most similar to the root node, we again proceed in the framework of RPSDs and the similarity measures defined above. The elementary nodes are described using RPSDs and the one with the maximum similarity with the root node \( R_i \) is chosen as one parent. The second parent of the root node is constructed by the synthesis of relation tuples merged with NULL and the primitives of \( R_i \). The process is repeated for the second parent of \( R_i \) until we can no longer find an injective mapping from the primitives of any elementary structure to those of the root node. Paraphrasing, we terminate when we find that the root node does not contain any of the elementary groupings. These “left over” root nodes represent organizations simpler than any of the elementary structures and therefore will not be supported by direct evidence.

With the decomposition of the root nodes, the structuring of the PIN is now complete. We have nodes for which we can provide direct evidence from our information source (the image) and the action of the PIN will infer the target groupings which make up the lowest level.

6.3.4 Expected Network Size

The size of the PIN depends on both the iterative and the root node decomposition processes. To get a feel for the total number of nodes in the PIN at the end of the iterative stage, suppose that we start with \( N \) target nodes. As we construct each level, the number of connected components decrease; in fact it is halved at each level.
Thus, if we are at level $k$ then the number of connected components, $N_{f}^{k} = N_{f}^{k-1}/2$ or $N_{f}^{k} = N/(2^{k})$. The number of nodes at level $k$ is related to that at level $k - 1$ by $N_{k} = N_{k-1}^{k-1}/2 + N_{k-1}^{k-1}$, the sum of the number of common and second parents. There are $\log N$ levels in the PIN structure before root node decomposition. Thus the total number of nodes at the end of the iteration stage is:

$$N_{\text{iteration}} = N + \sum_{k=1}^{\log N} \frac{3}{2}(\frac{N}{2^{k-1}}) = 4N - 3$$

(6.16)

The number of nodes created at the root node decomposition step is difficult to formalize exactly. However, we can derive average case estimates using the following assumptions.

1. We measure the structural size of an RPSD by the number of primitives ($n_{p}$) and relations ($n_{r}$). Since the number of relations increases faster than linearly with the number of primitives, a good measure of the structural complexity of an RPSD is the number of relations, $n_{r}$.

2. The average number of relations for the elementary nodes is $n_{r}^{e}$ and that for the target nodes is $n_{r}^{t}$.

3. The root decomposition process can be modelled as recursively grouping the relations of a root node into simpler subsets of relations. Thus a root node with $n_{r}$ relations is first broken into an elementary node with $n_{r}^{e}$ relations and a node with $(n_{r} - n_{r}^{e})$ relations which is again recursively decomposed. Each recursive step produces two new root nodes. This process continues for $\frac{n_{r}}{n_{r}^{e}}$
steps. Thus the expected number of extra nodes generated per root node with \( n_r \) relations is \( 2\left(\frac{n_r}{n_r^2} - 1\right) \).

4. We assume that on an average the composition of the relational tuples with interprimitive mapping function (produced in the PIN construction) produces null value for fraction \( f \) of the relations. \( f \) varies proportionally with the average similarity between the RPSDs. A low value for \( f \) signifies that on an average the RPSDs have a low similarity value, SSM, between them.

Recall that at each level of the iterative PIN construction stage we form a first parent and the second parent for each RPSD. The first parent is formed from the non-null paired relations and the second parent is formed from the null paired relations. We first derive expression for the average number of relations in the RPSDs at different levels of the PIN. At start (or zeroth level) we have \( N \) target RPSDs with an average of \( n_r^t \) relations. At the first level we have \( N/2 \) first parents and \( N/2 \) second parents. Since we assume that the average mapping between the primitives of two RPSDs produce a fraction \( f \) of null relation pairings, the second parents would have \( fn_r^t \) relations and the first parents would have \( (1 - f)n_r^t \) relations. Thus, the average number of relations, \( n_r^2 \) in an RPSD at the first level is:

\[
\begin{align*}
  n_r^1 &= \frac{N/n_r^t + f(1-f)n_r^t}{3N/2} \\
        &= \frac{(1+f)n_r^t}{3} 
\end{align*}
\] (6.17)
Generalizing, the average number of relations in an RPSD at the $k^{th}$ level, $n_r^k$, is:

$$n_r^k = \left( \frac{1 + f}{3} \right)^k n_r^i$$  \hspace{1cm} (6.18)

Observe that since $f$ is a fraction and is always less than 1, $n_r^k$ decreases with the PIN level $k$. The complexity of the RPSDs decrease with the PIN level. After the iterative stage, the number of root nodes at each level is given by $N/2^{(k-1)}$. Using our third assumption above and Eq. 6.18 we have the following expression for the expected number of extra nodes generated by the root node decomposition stage.

$$N_{\text{root}} = \sum_{k=1}^{\log N+1} \frac{N}{2^{(k-1)}} \left[ 2 \left( \frac{1+f}{3} \right)^k \frac{n_r^i}{n_r^e} - 2 \right]$$

$$= \frac{4Nn_r^i}{n_r^e} \sum_{k=1}^{\log N+1} \frac{1}{6} \left( 1 + f \right)^k - 2N \sum_{k=1}^{\log N+1} \frac{1}{2^{(k-1)}}$$  \hspace{1cm} (6.19)

The expected number of total nodes in the PIN is: $N_{\text{total}} = N_{\text{iteration}} + N_{\text{root}}$.

$$N_{\text{total}} = \frac{4Nn_r^i}{n_r^e} \left[ 1 - \frac{1}{6} \left( \frac{1+f}{6} \right)^{\log N+1} \right] - 1$$  \hspace{1cm} (6.20)

From the expression we can see that $N_{\text{total}}$ is $\mathcal{O}(N)$ given everything else remain constant. The other factors effecting $N_{\text{total}}$ are $\frac{n_r^i}{n_r^e}$ and $f$. $\frac{n_r^i}{n_r^e}$ is a measure of the discrepancy between complexities of the target nodes and the elementary nodes. If we are asked to construct a PIN to reason on a complex set of target hypotheses based on evidence for a set of simple features, $\frac{n_r^i}{n_r^e}$ would be high and, as expected, $N_{\text{total}}$ will increase. $N_{\text{total}}$ also increases with $f$ which is the expected number of null pairings for the relations when matching two RPSDs. $f$ is a rough estimate of the average similarity between the RPSDs. If the target set consists of RPSDs which
are very different from each other then \( f \) would be high and \( N_{total} \) would increase. Thus, we see the expected size of a PIN depends not only on the number of target nodes but also on the relative complexity of the target and elementary nodes and the similarity between the target nodes.

### 6.3.5 Desired Conditions

Given a set of target and elementary organizations, the structuring algorithm designs a PIN. In this section we consider the constraints of the design algorithm. Are we guaranteed an answer for any combination of target organizations and any set of direct evidence sources? While sufficient conditions for this are difficult to enunciate, it is possible to list some desirable characteristics for the target and the elementary nodes.

Ideally we want a PIN structure to have a small number of nodes and to be singly connected. For the final PIN to be singly connected we require that the network, after the iterative stage, also be singly connected. A necessary condition is that the RPSD at the top most level (the \( \log N^{th} \) level where \( N \) is the number of target nodes) after the iterative stage have atleast one relation. Recall, the number of relations participating in an organization RPSD decrease with the PIN level. Using the estimate of the number of relations in an RPSD at the \( k^{th} \) level of the PIN (Eq. 6.18), we have the following necessary condition.

\[
\left( \frac{1+f}{3} \right)^{\log(N)} n_r^t \geq 1
\]

\[
f \leq 1 - 3 \left( \frac{1}{n_r^t} \right)^{1/\log(N)}
\]  \( (6.21) \)

where \( f \) is the average fraction of null outcomes of the relational compositions for
the best mapping between the primitives of two RPSDs, \( n_r \) is the average number of relations in the target organizations, and \( N \) is the total number of target nodes. Observe that since \( f > 0 \), we can derive the upper limit on \( N \) for a particular target node complexity, \( n_r \) and vice versa. In a real case, we expect \( f \), which is a crude measure of the average similarity between the target organizations, to be high. Accordingly, the operational maximum limit \( N \) will be higher than that for \( f = 0 \).

Another desirable characteristic of a PIN structure is that the number of nodes be small. Eq. 6.20 of the previous section gives the expected number of nodes in an automatically designed PIN. The number \( N_{\text{total}} \) is proportionally dependent on the ratio \( \frac{n^i_r}{n^i_f} \) for a given \( N \) and \( f \). The fraction \( \frac{n^i_r}{n^i_f} \) is a measure of the discrepancy between complexities of the target and the elementary organizations. As mentioned earlier, we desire that this discrepancy be as small as possible. Of course we cannot expect the complexities of elementary features be the same as that of the target organizations; it defeats the purpose to infer complex structures from simple ones. However, this complexity gap between the target and elementary organizations cannot be too wide.

6.4 Estimating the Conditional Probabilities

The output of the structuring algorithm is a PIN of the form shown in Fig. 31. To complete the network specification we also need to determine the conditional probabilities at the nodes. The conditional probability at a node represents the conditional belief in an organization given the presence of its parent organizations.
Ideally, we would conduct a very large number of experiments or empirical studies to determine the number of times a particular structure is associated with another. This is infeasible\(^3\) so earlier we proposed the following form for the conditional probabilities.

\[
P(l_k^X = 1|l_{k_1}^{U_1} = u_1, \ldots, l_{k_n}^{U_n} = u_n)
= f_{LOC}(l_k, l_{k_1}, \ldots, l_{k_n})P(X = 1|U_1 = u_1, \ldots, U_n = u_n)
= f_{LOC}(l_k, l_{k_1}, \ldots, l_{k_n})C(u_1, \ldots, u_n)f_{GEOM}
\]

where \(l_k^X\) is a binary random variable representing feature \(X\) at location \(l_k\) and the \(l_{k_i}^{U_i}\) are its parents with the appropriate binary state \(u_i\). The conditional probability is a product of two terms: a locational compatibility term \(f_{LOC}\) and the term \(P(X = 1|U_1 = u_1, \ldots, U_n = u_n)\) which depends on the states of the parents. \(f_{LOC}\) determines the locational compatibility of an incoming message with feature \(X\) at location \(l_k\). In our implementation this is binary; one if the location is compatible and zero if not.

The term \(P(X = 1|U_1 = u_1, \ldots, U_n = u_n)\) denotes the confidence in inferring the presence of feature \(X\) given the states of the parents independent of location. The states of the parents are binary; a value of one \((U_i = 1)\) denotes the presence of the parent \(U_i\) and a value of zero denotes the absence. This conditional term is again considered to be a product of two terms: \(C(u_1, \ldots, u_n)\) and \(f_{GEOM}\). \(C(u_1, \ldots, u_n)\) is a number measuring the confidence of inferring \(X\) based only on the knowledge of states of the parents. Thus, if the states of the parents are all zero then the value of the constant is very low and if all states of the parents are one then the value of

\(^3\)Although there have been some studies of this type [14] they considered only simple relations like proximity.
the constant is high. The value for intermediate assignments of states depend on the similarity of the parents at state one with \( X \). \( f_{\text{GEOM}} \) measures the geometric compatibility of the particular instance of \( X \) with the definition of \( X \). Thus, highly divergent parallel lines would have lower values compared to perfectly parallel ones. In our previous implementation, \( C(u_1, \cdots, u_n) \) and \( f_{\text{GEOM}} \), which depend on the PIN node type, were chosen by hand.

### 6.4.1 Estimating \( P(X = 1|U_1 = u_1, \cdots, U_n = u_n) \)

Using the formulation of structural descriptions we can provide a theoretical underpinning to the choice of \( P(X = 1|U_1 = u_1, \cdots, U_n = u_n) \) while preserving its essential characteristics. Let us denote the random structural description (RPSD) for the features \( X, U_1, \cdots, U_n \) by \( G_X, G_{U_1}, \cdots, G_{U_n} \), respectively. The specific instances of these structural descriptions (PSDs) are represented as \( G_X, G_{U_1}, \cdots, G_{U_n} \). \( P(X = 1|U_1 = u_1, \cdots, U_n = u_n) \) measures the certainty of the presence of a structure given the states (presence or absence) of its parents. Instead of directly specifying this quantity we work with the likelihood ratio of feature \( X \):

\[
\Lambda(X|U_1 = u_1, \cdots, U_n = u_n) = \frac{P(X = 1|U_1 = u_1, \cdots, U_n = u_n)}{P(X = 0|U_1 = u_1, \cdots, U_n = u_n)} \quad (6.23)
\]

Analogous to our earlier form for \( P(X = 1|U_1 = u_1, \cdots, U_n = u_n) \), we factor this likelihood ratio into a product of two likelihood ratios: one considering just the states of the parents and the other considering the geometry of the actual outcome of \( X \). Thus we have

\[
\Lambda(X|U_1 = u_1, \cdots, U_n = u_n) = \Lambda_{(u_1, \cdots, u_n)} \Lambda_{\text{GEOM}} \quad (6.24)
\]
This product form is justified because the two evidence sources, the states of the parents and the geometric structure, can be considered to be independent. The likelihood ratio $\Lambda(u_1,\ldots,u_n)$ represents the odds based on the knowledge of just the states of the parents. As in the previous case for $C(u_1,\ldots,u_n)$, the value of $\Lambda(u_1,\ldots,u_n)$ should be high if the states of the parents are mostly one and low if they are mostly zero. Note that while the estimates for $C(u_1,\ldots,u_n)$ were restricted to be between 0 and 1, $\Lambda(u_1,\ldots,u_n)$ are in the range 0 to $\infty$.

To estimate $\Lambda(u_1,\ldots,u_n)$, we first separate the evidence in favor and the evidence against inferring $X$. We consider the zero state for a parent to be evidence against and the one state to be evidence in favor. All the evidence in favor is grouped and as is the evidence against $X$. Thus, we represent the joint event of the parents, $(U_1 = u_1,\ldots,U_n = u_n)$, by the occurrence of two events $U^1 = 1, U^0 = 0$. $U^1$ is the combination of the parents with state one and $U^0$ is the combination of the parents with state zero. This "combination" is effected in the RPSD formulation as follows. The event $U^1$ is represented by the RPSD $G_{U1}$ formed by merging the RPSDs of the parents whose states are one and the event $U^0$ is represented by $G_{U0}$ constructed by merging the parents whose states are zero. The KPSDs $G_{U1}$ and $G_{U0}$ are compact representations of the joint event of the parent's states.

To ascertain the effect of $G_{U1}$ and $G_{U0}$ on $\Lambda(u_1,\ldots,u_n)$, we again use the RPSD structural similarity measure. Intuitively, the odds should increase with the similarity of $G_X$ to $G_{U1}$, the structure of the merged RPSDs of those parents in state one. And, the odds should decrease with increasing $SSM(G_X,G_{U0})$. Thus, if we
have positive evidence (state one) for parent organizations which are similar to \( X \) then the odds increase and *vice versa*; it is easier to infer structures from parents which are structurally very similar. Among the variety of possible combinations of the two similarity measures, we chose the following form:

\[
\Lambda(u_1, \ldots, u_n) = \frac{SSM(G_X, G_{U1})}{SSM(G_X, G_{U0})}
\]  

(6.25)

where \( SSM \) is the structural similarity measure of Eq. 6.10. The likelihood ratio decreases with the similarity of the structure \( X \) to the parents with state zero and increases with the similarity to the parents with state one.

The second likelihood ratio, \( \Lambda_{GEOM} \), considers the geometry of the particular outcome. The more strictly a particular outcome satisfies the definition of the organization, the higher the value of \( \Lambda_{GEOM} \). For example, with continuity, the more continuous the constituent lines are, higher the confidence in inferring continuity. We again use the RPSD formulation to quantify this confidence. Let an outcome organization be represented by the parametric structural description (PSD), \( G_X \), and the associated RPSD by \( G_X \). The RPSD represents a probabilistic definition of the organization \( X \) in terms of primitives and relations. As seen in Section 6.2, the attributes of the primitives and the relations are random variables. The form of the joint probability distribution of the RPSD is given by Eqs. 6.1, 6.3, and 6.5.

The mode of a discrete probability distribution represents the value which is most probable. A primitive or relation attribute equal to the mode of the distribution can be considered to satisfy the definition of that particular attribute to the
greatest extent. Similarly for an entire RPSD, outcomes with primitive and relation attribute values equal to the modes of their respective distributions can be said to be archetypes of the organization represented by the RPSD. We quantify this as follows.

\[
\Lambda_{GEOM} = \frac{(P(G_X = G_X))^{1/(n_p + n_r)}}{(1 - (P(G_X = G_X))^{1/(n_p + n_r))}^{-1}}
\]

(6.26)

where \(P(G_X = G_X)\) is the probability of the particular outcome PSD \((G_X)\), \(P_{mode}(G_X)\) is the maximum of the joint probability distribution of the RPSD and \(n_p, n_r\) are the number of primitives and relations, respectively. The normalization with respect to the number of primitive and relations is done to get a number closer to one because computing \(P(G_X = G_X)\) requires the multiplication of many numbers less than one. Note that the value of the likelihood ratio is maximum when the attributes of the primitives and relations correspond to the modes of their probability distributions and is a minimum when the probabilities are zero. \(\Lambda_{GEOM}\) is one when all the primitive and relation attributes have values whose corresponding probabilities (as specified by their probability densities) are \(1/2\) that of the mode. Mathematically, \(\Lambda_{GEOM}\) is monotonic with \(P(G_X = G_X)\), the probability of the PSD outcome.

6.4.2 Computing \(f_{LOC}(l_k, l_{k_1}, \ldots, l_{k_n})\)

The term \(f_{LOC}(l_k, l_{k_1}, \ldots, l_{k_n})\) in the conditional probability expression (Eq. 6.22) measures the compatibility of the parents’ locations with that of the node feature. For our experiments, this is a binary function taking value one when the locations are compatible and zero otherwise. One consequence of factoring out the locational
compatibility is the simplicity of the message handling protocol. On receiving a message from its neighbors, a PIN node generates a compatible structure hypothesis of its own type based on the neighbor's feature. The generated structure is then matched for location against the structures present at the PIN node. In the event of a locational match the associated probabilities are updated and messages are sent out to the other neighbors. Otherwise, a new entry is made at the PIN node and messages are sent.

In the manually designed PIN the compatible structure generating and the matching algorithms depended on the node type because of the nonuniform nature of the feature representation scheme. Using RPSD representations throughout the PIN, as we do here, allows us to define a uniform protocol for message handling. The feature (or organization) instances at each PIN node are represented as parametric structural descriptions (PSD), which are outcomes of the RPSD representing the node organization type. A PSD is a structure isomorphic to an RPSD and is associated with the outcome attributes of the primitive and relations. We use the PSD also for storing locational information of an organization instance by associating locations with each primitive and relation.

On receiving a message, a node generates a compatible PSD based on the PSD of the organization instance at the PIN node which sent the message. For this we need to know the mapping between the primitives of the PSDs at the two nodes, the current node and the node which sent the message. These mappings are derived from the RPSD matching results in the PIN structuring algorithm and are stored as
Figure 32: A simple PIN to illustrate the generation of compatible PSD. The pointers between the primitives of the PSDs at the neighboring PIN nodes are stored as paired lists.
paired lists of the primitives. Fig. 32 illustrates the process using a simple example. The PSDs representing organization instances at the PIN nodes $X, U_1,$ and $U_2$ are shown inside circles. The pointers between the primitives of the PSDs are stored as pairwise lists. If node $X$ receives a message from parent $U_1$, then a compatible PSD at $X$ is partially completed using the information from the PSD at $U_1$. Primitives A, B, C, D and relations AB, BD, CD, CA are copied.

We complete this partially defined compatible PSD by invoking filling functions. These functions are dependent on the choice of the primitive or relations types of the RPSDs. Each primitive type is associated with a predefined function to estimate its locations and attributes from a subset of the relations. Similarly, we have functions to infer the locations and the parameters of the relations from primitives. These filling functions may be able to infer primitives (relations) in the absence of some relations (primitives) by making suitable assumptions. For example, we might be able to infer the relationship of closure even if one of the primitives is missing by assuming a default shape (e.g. straight line) for the missing primitive.

The filling functions go through the list of primitives and relations of the compatible PSD and checks if missing primitives or relations can be inferred. Thus in Fig. 32, the a relation filling function infers the 3-ary relations ABF and ACE from the knowledge of the primitives A, B, C, and D. Based on the inferred location of this relation, we estimate the locations of the primitive F or E. The process continues until we can no longer infer any primitive or relation of the compatible PSD. As one can notice, the compatible PSD generation algorithm is dependent on
just the small set of predefined filling functions and, unlike the manually structured PIN, independent of the specific nature of each PIN nodes.

After we construct the compatible PSD associated with the primitive and relation locations, we match it with those stored at the PIN node. The matching between PSDs is conducted using the parameters and locational information of the PSD relations; relations whose locational information could not be inferred in the filling stage are ignored. The parameters of the relations are first compared to see if they agree. This lets us prune PSDs with very different values for the relational parameters. A stricter point by point match is performed for the surviving PSDs using the locations of their relations.

6.5 An Example

In this section, we lead the reader through the automatically structuring of an actual PIN. We also present performance studies quantifying the robustness of the proposed algorithm to input data. Earlier we structured a PIN manually (Fig. 21) to integrate information from preattentive modules. In the present example, we design a PIN to serve the same purpose but do so automatically. In addition to the advantage that we can now structure this PIN (and therefore others) automatically, we show in the next section that the automatically structured PIN performs at least as well as the manually structured one.
Table 6: The target and elementary organizations used for the example

<table>
<thead>
<tr>
<th>Target Organizations</th>
<th>Target Organizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Ellipses</td>
</tr>
<tr>
<td>Alias</td>
<td>E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Elementary Organizations</th>
<th>Elementary Organizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Closed Set of 4 segments</td>
</tr>
<tr>
<td>Alias</td>
<td>C4</td>
</tr>
</tbody>
</table>

6.5.1 The Target and Elementary Organizations

The starting point of the structuring algorithm is the definition of the target structures or organizations of interest and the elementary structures or features for which we can provide evidence. In the present context of perceptual organization we reason on a set of regular geometric forms based on the evidence for a set of preattentive features. These organizations are listed in Table 6. A ribbon is a closed set of four edge segments which has a dominant curved parallel symmetry axis; the other target organizations have their usual meaning. A closed set refers to a collection of edge segments which form a closed boundary. A strand is a chain of edge segments which do not close. Note that since the RPSDs, as formulated, cannot handle variable numbers of primitives or relations, we have to model the closed sets of edges separately, according to the number of edge segments. We found this simpler and more effective for the present study than generalizing the formulation of the RPSDs to handle different numbers of primitives.
Table 7: Primitive and Relation attributes used to construct the RPSDs of the structures of interest.

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Type</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant Curvature</td>
<td>CC</td>
<td>Primitive</td>
<td>Relative Length ( (l_1/l_{av}) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Relative Curvature ( (l_i/R) )</td>
</tr>
<tr>
<td>Closure</td>
<td>N-ary Relation</td>
<td>Convexity = (\sqrt{\frac{\text{Area}}{\text{Area of Convex Hull}}}))</td>
<td></td>
</tr>
<tr>
<td>Strand</td>
<td>N-ary Relation</td>
<td>Convexity = (\sqrt{\frac{\text{Area}}{\text{Area of Convex Hull}}}))</td>
<td></td>
</tr>
<tr>
<td>Linear Symmetry</td>
<td>Sym</td>
<td>Binary Relation</td>
<td>Axis Curvature ( (l_i/R) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Profile curvature</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Divergence Angle</td>
</tr>
<tr>
<td>Circular Symmetry</td>
<td>Sym-C</td>
<td>Unary Relation</td>
<td>Radius</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Circularity (fit error)</td>
</tr>
</tbody>
</table>

6.5.2 The Primitive and Relation Sets

We describe the RPSDs of the target and the elementary organizations using edge segments as primitives and the relations of closure, strands, linear symmetry, circular symmetry, and continuity, as shown in Table 7. Edge primitives are segments of approximately constant curvature and are described by two attributes: relative length and relative curvature. Relative length is defined as \( l_i/l_{av} \) where \( l_i \) is the length of the primitive and \( l_{av} \) is the average length of primitives in the image. This attribute captures the relative size of the primitive with respect to other image features. Relative curvature is defined as \( l_i/(2\pi R) \) where \( R \) is the radius of curvature of the edge segment. The measure is dimensionless and captures the perceptual curvature of the segment. A straight line has value 0 and a full circle has value 1. The value for an arc depends on its length; for small lengths relative to the radius
Figure 33: Examples of the computed attributes for typical linear symmetry relations between two edge segments. (a) and (b) represent perfect cases. (c) and (d) represent relaxed cases.

of curvature the value is closer to that of a straight line.

Closure is an N-ary relation which exists if a string of edge segments form a closed region. It is described by its convexity, the square root of the ratio of the area of the figure to that of the convex hull of the figure. The measure ranges from 1 for a completely convex shape to 0 for a very convoluted figure. The strand relation is also N-ary and exists if a string of edge segments form a chain which does not close. It is also described by its convexity; to compute the area of a strand we connect the end points with a straight line.
We use two types of symmetry relations: linear and circular. Linear symmetry is a binary relation and exists if there is a line of symmetry between the constituting edge segments. To allow for a relaxed definition of the symmetry relation, we describe it using three attributes. The first is the relative curvature of the symmetry axis and is computed as for edge primitives. The second is the ratio of the distance between the centers of the edge segments to the radius of the symmetry axis; this is zero for straight segments and for concentric ribbons. The third is the divergence angle between the segments measured as the angle between the line joining the end points of one segment and the line joining the points on the other segments which are closest to the end points of the first segment. Again, for a perfect case this is zero. The value of these attributes for typical linear symmetries are shown in Fig. 33.

Circular symmetry is a unary relation described by the radius of the best fitting circle and the associated least square fit error which measures the circularity of the edge points. Continuity is a binary relation measuring the angle between the edge segments near the end points on a (0 1) scale; 0 for perfect continuity (180°), and 1 for an acute corner (0°).

In the framework of RPSDs, the relations defined above are relaxed by considering their attributes to be random variables. Thus for linear symmetry, allowing for non zero values for the second and third attributes makes for a relaxed definition of linear symmetry or parallelness. However, the exact probability distributions of these attributes are still unknown. For our experiments, we estimated these distri-
Figure 34: An automatically structured PIN

...butions based on some statistical data, like the histogram of the relative curvature of the edge segments, but mostly relied on subjective judgment. We did conduct experiments, presented below, to study the affect of these choices on the final PIN structure and discovered that the final structure of the net is relatively insensitive to the exact probability definitions of the RPSDs.
6.5.3 Execution Statistics

Starting from the elementary and target organizations, the PIN shown in Fig. 34 was designed. Note the naming convention we have used. The name of a node, other than the target and elementary nodes, is the concatenation of the names of its children. The name of the second parent of a node is the concatenation of the name of that node and "n", to signify a merging with null. Thus, the node EPQRn is the second parent of the node EPQR which, in turn, is the first parent of the nodes EP and QR, and so on. From the list of names of the PIN nodes we can recover the structure of the network.

Network Structure

The total number of nodes in the network is 36 (Fig. 34) as compared to 23 in the manually designed PIN (Fig. 21). The nodes associated with the elementary evidence are listed in the table inset in Fig. 34.

Table 8 gives information about various stages of the structuring algorithm. There are three levels of construction in this example. At the starting level, ellipse was paired with parallelogram, triangle with circle, and quadrilateral with ribbon as they were found to be the most similar pairings among the target nodes. For the next level, the parents of level 0 participated in the process: EP, TC, QR, Tn, Cn, Rn. Note that we do not have the second parents En, Pn, and Qn for the nodes E, P, and Q respectively. This is because no relation of E, P, or Q was mapped to NULL in the RPSD mapping process, i.e., the basic RPSD structure of the nodes
Table 8: Different stages of the PIN structuring algorithm.

<table>
<thead>
<tr>
<th>Level</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Starting Nodes</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Parents</td>
</tr>
<tr>
<td>1</td>
<td>Starting Nodes</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Opt. Pairings</td>
</tr>
<tr>
<td></td>
<td>Parents</td>
</tr>
<tr>
<td>2</td>
<td>Starting Nodes</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Structural Similarities</td>
</tr>
<tr>
<td></td>
<td>Opt. Pairings</td>
</tr>
<tr>
<td></td>
<td>Parents</td>
</tr>
</tbody>
</table>
is easily captured by one parent and a second parent is unnecessary (see Section 6.3.2).

At the next level, similarity values were computed only for valid pairs; the invalid pairs like (TC Tn) were rejected as they formed a cycle (ignoring link direction) in the PIN. At this level there were 3 connected components: (EP), (TC Tn Cn), and (QR Rn). The best pair was (EP QR). For the last level we had 5 parents in two components: (EPQR EPn) and (TC Tn Cn). The most similar valid structure between the two components was (EPQRTC). Note that the pairing (TC EPn) was not chosen despite its high value of similarity because the composition of the relation tuples with the interprimitive mapping between the RPSDs did not produce any non-null tuples; only the primitives were matched and all the relations were mapped to NULL. Thus, the common merged RPSD in this case had no relations, only primitives! Since this is not meaningful, we ignore such results. At the end of the iterative construction process the PIN had 15 nodes.

The root nodes at this stage were Tn, Cn, Rn, EPn, EPQRTC, and EPQRn. Nodes Cn, EPQRTC, and Rn represent circular symmetry, closure, and curved symmetry, respectively; the other three denote complex organizations. Tn denotes a collection of three corners, EPn denotes 4 primitives with two linear symmetry relations, and EPQRn is a more complex structure with 4 primitives, 4 strand relations, and 4 corner relations. To enable the introduction of direct evidence to the PIN we recursively decompose the root nodes into the elementary feature groupings (listed in Fig. 6) using the algorithm in Section 6.3.3. Table 9 gives data
Table 9: Root node breakup data. The names in bold refer to our starting root nodes. The others refer to the root nodes which were created in the recursive process.

<table>
<thead>
<tr>
<th>Root Node</th>
<th>Similar Elementary Grouping</th>
<th>New Root Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cn</td>
<td></td>
<td>Cn</td>
</tr>
<tr>
<td>Tn</td>
<td>(Corner 0.44)</td>
<td>TnCorner, Tnn</td>
</tr>
<tr>
<td>Tnn</td>
<td>(Corner 0.52)</td>
<td>TnnCorner, Tnnn</td>
</tr>
<tr>
<td>Tnnn</td>
<td>(Corner 0.85)</td>
<td>Tnnn</td>
</tr>
<tr>
<td>Rn</td>
<td>(CS 0.98)</td>
<td>Rn</td>
</tr>
<tr>
<td>EPQRTC</td>
<td>(C4 0.98)</td>
<td>EPQRTC</td>
</tr>
<tr>
<td>EPn</td>
<td>(PS 0.47)</td>
<td>EPnPS, EPnn</td>
</tr>
<tr>
<td>EPnn</td>
<td>(PS 0.93)</td>
<td>EPnn</td>
</tr>
<tr>
<td>EPQRn</td>
<td>(S3 0.32)</td>
<td>EPQRnS3, EPQRnn</td>
</tr>
<tr>
<td>EPQRnn</td>
<td>(S3 0.35)</td>
<td>EPQRnnS3, EPQRnnn</td>
</tr>
<tr>
<td>EPQRnnn</td>
<td>(S3 0.38)</td>
<td>EPQRnnnS3, EPQRnnnn</td>
</tr>
<tr>
<td>EPQRnnnn</td>
<td>(S3 0.42)</td>
<td>EPQRnnnnS3, EPQRnnnnn</td>
</tr>
<tr>
<td>EPQRnnnnn</td>
<td>(Corner 0.35)</td>
<td>EPQRnnnnnCorner, EPQRnnnnn</td>
</tr>
<tr>
<td>EPQRnnnnnn</td>
<td>(Corner 0.40)</td>
<td>EPQRnnnnnnCorner, EPQRnnnnnn</td>
</tr>
<tr>
<td>EPQRnnnnnnn</td>
<td>(Corner 0.56)</td>
<td>EPQRnnnnnnnCorner, EPQRnnnnnnn</td>
</tr>
<tr>
<td>EPQRnnnnnnnn</td>
<td>(Corner 0.94)</td>
<td>EPQRnnnnnnnn</td>
</tr>
</tbody>
</table>
about the breaking of the root nodes. The table lists the similarity values (SSMs) for the elementary node most like the root node and contained in it. Since we do not have an elementary grouping for circular symmetry the root node Cn was not decomposed. We cannot provide direct evidence for Cn. The other nodes were recursively decomposed into a number of simpler root nodes. Note that because the nodes Rn and EPQRTC were isomorphic to CS and C4, respectively, they were not decomposed. At the end of the process we have a total of 36 nodes in the PIN, of which 14 are root nodes.

Conditional Probabilities

As we saw in Section 6.4, the specification of PIN conditional probabilities involves $A^{*}_{(u_1,\ldots,u_n)}$ and $A_{GEOM}$. $A_{GEOM}$ depends on the particular realization of the structure and its form was discussed in Section 6.4. We present the estimates of $A^{*}_{(u_1,\ldots,u_n)}$ for this example in Table 10. A likelihood ratio greater than one denotes odds in favor of inferring the structure; a value less than one denotes odds against. Note that the value of $A^{*}_{(u_1,\ldots,u_n)}$, as expected, is always greater than one when the state(s) of the parent(s) are one and is near zero when states are zero. For intermediate states, the value depends on the structural similarity between the node and each of its parents. For example, the node EPQR has two parents, EPQRTC and EPQRn. Since EPQRn is structurally very similar to EPQR, the likelihood ratio is greater than one when the state of EPQRn is one.
Table 10: The estimated conditional likelihood ratios, $\Lambda_{(u_1,u_2)}$.

<table>
<thead>
<tr>
<th>Node</th>
<th>Parent1 $U_1$</th>
<th>Parent2 $U_2$</th>
<th>$\Lambda_{(u_1,u_2)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0 0)</td>
</tr>
<tr>
<td>E</td>
<td>EP</td>
<td></td>
<td>0.022</td>
</tr>
<tr>
<td>EP</td>
<td>EPQR</td>
<td>EPn</td>
<td>0.024</td>
</tr>
<tr>
<td>EPQR</td>
<td>EPQRTC</td>
<td>EPQRn</td>
<td>0.051</td>
</tr>
<tr>
<td>TC</td>
<td>EPQRTC</td>
<td></td>
<td>0.301</td>
</tr>
<tr>
<td>T</td>
<td>TC</td>
<td>Tn</td>
<td>0.057</td>
</tr>
<tr>
<td>Tn</td>
<td>TnCorner</td>
<td>Tnn</td>
<td>0.024</td>
</tr>
<tr>
<td>Tnn</td>
<td>TnnCorner</td>
<td>Tnnn</td>
<td>0.029</td>
</tr>
<tr>
<td>C</td>
<td>TC</td>
<td>Cn</td>
<td>0.099</td>
</tr>
<tr>
<td>EPQRn</td>
<td>EPQRnS3</td>
<td>EPQRnn</td>
<td>0.017</td>
</tr>
<tr>
<td>EPQRnn</td>
<td>EPQRnnS3</td>
<td>EPQRnnn</td>
<td>0.013</td>
</tr>
<tr>
<td>EPQRnnn</td>
<td>EPQRnnnCorner</td>
<td>EPQRnnnn</td>
<td>0.010</td>
</tr>
<tr>
<td>EPQRnnnn</td>
<td>EPQRnnnnS3</td>
<td>EPQRnnnnn</td>
<td>0.009</td>
</tr>
<tr>
<td>EPQRnnnnn</td>
<td>EPQRnnnnnCorner</td>
<td>EPQRnnnnnn</td>
<td>0.014</td>
</tr>
<tr>
<td>EPQRnnnnnn</td>
<td>EPQRnnnnnnCorner</td>
<td>EPQRnnnnnnn</td>
<td>0.019</td>
</tr>
<tr>
<td>EPQRnnnnnnn</td>
<td>EPQRnnnnnnnCorner</td>
<td>EPQRnnnnnnnCorner</td>
<td>0.017</td>
</tr>
<tr>
<td>QR</td>
<td>EPQR</td>
<td></td>
<td>0.007</td>
</tr>
<tr>
<td>Q</td>
<td>QR</td>
<td></td>
<td>0.012</td>
</tr>
<tr>
<td>R</td>
<td>QR</td>
<td>Rn</td>
<td>0.025</td>
</tr>
<tr>
<td>EPn</td>
<td>EPnP5</td>
<td>EPnn</td>
<td>0.022</td>
</tr>
<tr>
<td>P</td>
<td>EP</td>
<td></td>
<td>0.026</td>
</tr>
</tbody>
</table>
Design Time

The PIN construction algorithm is code in C++ and runs on a Sparc IPX. The total CPU time for the design of the example PIN is 114 sec. Most of the time was consumed finding optimal mappings between the primitives of the RPSDs; there were 227 calls to this routine. The complexity of the optimal mapping algorithm is controlled by the size of the search tree branching factor which is the matchpool size. In the present implementation it was restricted to 6 primitive matches.

6.5.4 Robustness

The inputs to the PIN structuring algorithm are the RPSDs of the target and elementary organizations. In this section we investigate the sensitivity of the output PIN structure and the estimates of the conditional probabilities with respect to the probability specifications of the RPSD primitives and relations.

There are two types of probability values we specify for each primitive or relation: the probability mass functions (pmf) of the attribute values and the probability of nullity. For the present study, we perturbed the chosen values of the probabilities and compared the resulting PIN with the unperturbed one. We added uniformly distributed noise over \([a, b]\) to the chosen values of the probabilities of nullity and the pmfs. For a pmf we added noise to each value of the discrete mass function and normalized the resultant pmf to retain its probabilistic validity. To quantify the amount of distortion caused by the addition of noise to a pmf we define \(d_{pmf}\). If
Figure 35: Isomorphic subparts of a PIN. The difference arises because of the order of breakup of the root node EPQR but the two structures have essentially the same behavior.

\( \{ p_i | i = 1, \cdots, n \} \) is one pmf and \( \{ q_i | i = 1, \cdots, n \} \) is the other then:

\[
d_{pmf} = \sum_{i=1}^{n} |(p_i - q_i)|
\]

(6.27)

The range of \( d_{pmf} \) is \([0, 2]\); 0 if the pmfs are the same and 2 if they are completely different.

To study the impact of the RPSD perturbations on the PIN structure, we compare the resulting PIN both structurally and quantitatively with the noiseless case. We use the mean absolute change in the conditional probabilities as the quantitative measure. Since locational and geometric factors in the conditional probability expression are dependent on the particular instances of the PIN node structures (Eqs. 6.22 and 6.24), their contributions to the conditional probabilities are difficult to study. For our comparisons we chose \( \Lambda_{geom} = 1 \) and \( f_{LOC} = 1 \). We study just the effect on the conditional probabilities due to the estimates \( \Lambda_{u_1, \cdots, u_n} \).
For structural comparisons we consider the total number of nodes in the PIN, node types and the actual structure of the interconnections. If two nodes, one from each of the two PINs (noiseless and noisy) have the same names, then because of the nature of our naming scheme we know that the two have the same interconnections. However, two PINs with slightly different interconnections might also be considered to have the same structure. An example is shown in Fig. 35. This can happen only at the root node decomposition step when we break up a root node in slightly different sequence and does not affect the behavior of the network. PINs with only this type of simple difference are classified as having the same structure.

Table 11 summarizes the experimental results in four columns. For the first column we added noise distributed uniformly over [0, 0.1] to the probability mass functions (pmfs) of the target node primitive and relation attributes and normalized as described above. There were 15 different runs and as the results show, the output PIN structure was quite stable. All 15 PINs were similar to the noiseless case. The starting pmf difference, \( d_{pmf} \), between the noisy and noiseless cases was 0.52 and the average \( d_{pmf} \) for all the final nodes was the same. Even in the presence of a large distortion in the pmfs, all 15 runs resulted in 36 final nodes. There was no difference in the basic nodes, i.e. the nodes neglecting root node decomposition. Out of the 15 instances only 4 had a difference in the order of the breakup of one root node. The average change in conditional probability estimates was 0.01 which is very small compared to the distortion a PIN can tolerate. The next section presents results demonstrating the robustness of the PIN output with respect to uniform noise in
Table 11: PIN structure comparisons of the noisy case with the noiseless one. The number in [] following a value indicates the number of instances with that particular value. Basic Nodes refer to the PIN nodes before root node breakup. In experiments 1 and 2 uniform noise was added to the pmfs of the target and elementary nodes. In experiments 3 and 4 only the probabilities of nullity of the primitives and relations of the target and elementary nodes were perturbed. In experiment 5 both the pmfs and the probabilities of nullity of the target and elementary nodes were perturbed.

<table>
<thead>
<tr>
<th>Experiments</th>
<th>Pmf</th>
<th>Probability of nullity</th>
<th>Both</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Noise</td>
<td>U[0, 0.1]</td>
<td>U[0, 0.2]</td>
<td>U[0, 0.2]</td>
</tr>
<tr>
<td>No. of outcome PINs</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>No. of PINs similar to the noiseless case</td>
<td>15</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>Av. $d_{pmf}$ for target nodes</td>
<td>0.52</td>
<td>0.76</td>
<td>0.0</td>
</tr>
<tr>
<td>Av. $d_{pmf}$ for all nodes</td>
<td>0.52</td>
<td>0.72</td>
<td>0.16</td>
</tr>
<tr>
<td>Av. probability of nullity for nodes at start</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>Av. change in cond. prob</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
</tr>
</tbody>
</table>
[0.25 0.25] distortions of the conditional probabilities; the relative values of the conditional probabilities are more important than their absolute quantities.

In the second experiment we increased the range of the noise to [0, 0.2] with an average $d_{pmf}$ of the target nodes of 0.76. This is the algorithm's tolerance in pmf distortion. From the table we can see that out of 15 instances 14 had the correct structure. However, the change in the conditional probability estimates of 0.01 is still well within tolerance.

In the third and fourth experiments we perturbed just the probabilities of nullity of the target and elementary RPSD primitives and relations. We added noise uniformly distributed over [0, 0.2] to the previous probability of nullity values of 0.1 resulting in an average probability of nullity of 0.2, which is a large increase. Out of 15, only 8 had the proper structure but the change in the conditional probability estimates was low at 0.02. On decreasing the average probability of nullity to 0.05, experiment 4, the changes in conditional probabilities (0.01) were again within tolerance and 13 instances resulted in the right structure. Thus, we found the structuring algorithm to be more sensitive to the probability of nullity values than to the pmfs of the attributes.

In the fifth experiment we added noise distributed uniformly in [0, 0.1] to both the pmfs and the probabilities of nullities of the target and elementary nodes. The starting pmf difference, $d_{pmf}$, between the noisy and noiseless cases was 0.51 and the average probability of nullity was 0.15. Although there were no differences in the basic nodes of the 15 instances, only 12 of the final PINs were similar to the
noiseless one. However, the average change in the conditional probability estimates was again small, 0.01.

In comparison to the structure, the estimates of the conditional probabilities were found to be more stable. However, for modest changes in the starting probability specifications both the structure and conditional probability estimates of the algorithm are quite robust. Of course, the PIN should vary if the known conditions vary sufficiently.

6.6 Performance on Real Images

The inputs to the PIN are the elementary structures detected using a bottom up preattentive module. The PIN, acting as the attentive module, allows us to go beyond the input data to hypothesize target organizations. We tested the automatically structured PIN on a number of images. We present summarized results for some images in Table 12 and detailed results on an aerial image (Fig. 36(a)). The edges were detected using the optimal zero crossing operator (OZCO) [21] (Fig. 36(b)). The elementary pieces of evidence are 15 closed rings of 4 edge segments (C4), 2 edge strands (S3), 12 curved symmetries (CS), and 16 straight symmetries (Fig. 36(c)-(f)).

The preattentive elementary cues, each associated with a probability value based on geometric and photometric considerations, are introduced to the PIN at the appropriate root nodes. After the network settles to equilibrium (which takes 46 seconds) we have the target hypotheses shown in Fig. 37. There are 2 ellipses, 8 triangle, 13 circles, 22 quadrilaterals, 14 ribbons, and 18 parallelograms. Probability
Figure 36: (a) Aerial image. (b) Edges detected in the aerial image. Organizations detected using the preattentive module on the aerial image and input to the PIN. (c) Closed rings of 4 segments (C4) (d) Edge strands of 3 segments (S3) (e) Straight symmetry (PS) (f) Curved symmetry (CS)
values are associated with the hypotheses measuring the confidence of inferring the feature from the elementary evidence.

6.6.1 Comparing Automatically and Manually Structured PINs

In this section we compare a manually structured PIN with the automatically structured PIN designed to accomplish the same task. The manually structured PIN used is that used in our earlier work on perceptual organization (see Fig. 21). Like the automatically structured PIN, the manually structured PIN is designed to infer target organizations like ellipses, circles, quadrilaterals, ribbons, triangles, and parallelograms based on preattentive cues like strands, closure, linear symmetry, and corners. We compare these PINs in terms of structure, ease of design, and performance on real data.

Structural Comparison

Although the manually structured PIN (Fig. 21) has fewer nodes, 23 compared to 36 in the automatic design, the structure of the manual design is not necessarily simpler. Its nodes do not have a uniform neighborhood structure (see node \( V_3 \) in Fig. 21) whereas all nodes in an automatically structured PIN are guaranteed to have a uniform structure of at most 2 parents and 2 children. Nonuniformities also exist in the representations of structures at the nodes. For example, node \( N_1 \) in the manually designed PIN represents the collection of all possible strands in a closed ring of edge segments. While the corresponding forms in the automatically designed
Figure 37: The target outputs of the automatically designed PIN (a) Parallelograms (b) Ellipses (c) Circles (d) Quadrilaterals (e) Ribbons (f) Triangles
PIN are separated into distinct nodes. Although this increases the total number of nodes, each node is simpler and the locational compatibility computations are less complex. While the locational compatibility computation in the manually designed PIN requires a point by point match of the primitive locations, the automatically structured PIN employs the more efficient strategy outlined in see Section 6.4.2.

In addition to the uniformity of the computational units and the locational compatibility function of the automatically structured PIN, the feature representation and conditional probability specification are also uniform. Unlike the special representations used for each node of the manually designed PIN, every instance of an organization in an automatically designed one is uniformly represented as a Parametric Structural Description (PSD). The conditional probabilities were chosen by hand in the manually structured PIN, but in an automatically structured one we adopt the uniform strategy described in Section 6.4.1.

The uniformity of structure and simplicity of organization representation in an automatically structured PIN results in an overall computational performance increase. Despite the increased number of nodes, the execution times for the same tasks (Table 12) are less for the automatically designed one. For a parallel implementation we can estimate the speedup by dividing the serial execution time by the number of nodes suggesting that the automatically designed network would be, in fact, faster. Of course, the automatically designed PIN will involve more parallel hardware, but this is easily offset by the uniformity and simplicity of the computational units.
Ease of Design

The inputs for the automatic design are the RPSD specifications of the elementary and target organizations from which the structure of the PIN and the conditional probabilities are automatically deduced. But in the manual design, the entire structure and all conditional probabilities must be specified by hand. Manually specifying the structure for small PINs might be efficient but for larger nets it is difficult.

Besides automatically structuring the PIN, we now have a uniform strategy to estimate the conditional probabilities. As we saw in Section 6.4, the PIN conditional probabilities involve a locational compatibility function and a second factor which depends on the states of the parent organizations (Eq. 6.22). In a manual design, we specify these separately for each node and, in general, they will vary from node to node. In an automatic design, not only is the form of the second factor automatically estimated from the RPSDs but the locational compatibility computation strategy is also uniformly specified (Section 6.4.1).

6.6.2 Performance on Real Data

In this section we compare the performance of the manually and the automatically structured PINs for the same task in terms of speed and final target organization hypotheses. For this test we use the image of Fig. 36. Fig. 37 shows the target node outputs for the automatically designed PIN and Fig. 23 displays those of the manually designed PIN. Note that, although the important organizations in the image have been hypothesized by both the PINs, there are a significant number of
Table 12: Comparison of the target outputs of the manually designed and automatically designed PINs. The triplets listed from the second to the sixth rows specify the number of the organizations hypothesized by the manually designed PIN, by the automatically designed PIN, and by both PINs, respectively. The last row lists the similarity index $C_{sim}$ for the relative probability assignments for the common hypotheses of the two PINs.

<table>
<thead>
<tr>
<th>Image</th>
<th>Aerial</th>
<th>Blocks</th>
<th>Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (Manual)</td>
<td>(72 46)</td>
<td>(87 55)</td>
<td>(83 67)</td>
</tr>
<tr>
<td>Ellipses ($N6/E$)</td>
<td>(58 2 2)</td>
<td>(42 2 2)</td>
<td>(92 5 5)</td>
</tr>
<tr>
<td>Circles ($N8/C$)</td>
<td>(58 13 12)</td>
<td>(42 13 9)</td>
<td>(92 25 25)</td>
</tr>
<tr>
<td>Ribbons ($N13/R$)</td>
<td>(11 11 11)</td>
<td>(7 9 8)</td>
<td>(1 6 1)</td>
</tr>
<tr>
<td>Quadrilaterals ($N19/Q$)</td>
<td>(26 22 19)</td>
<td>(30 27 21)</td>
<td>(57 46 46)</td>
</tr>
<tr>
<td>Parallelograms ($N23/P$)</td>
<td>(26 18 17)</td>
<td>(30 23 19)</td>
<td>(57 42 41)</td>
</tr>
<tr>
<td>Relative Prob.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_{sim}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(N6 E)</td>
<td>0.89</td>
<td>0.93</td>
<td>0.57</td>
</tr>
<tr>
<td>(N8 C)</td>
<td>0.89</td>
<td>0.91</td>
<td>0.84</td>
</tr>
<tr>
<td>(N13 R)</td>
<td>0.99</td>
<td>0.98</td>
<td>1.0</td>
</tr>
<tr>
<td>(N19 Q)</td>
<td>0.88</td>
<td>0.94</td>
<td>0.77</td>
</tr>
<tr>
<td>(N23 P)</td>
<td>0.87</td>
<td>0.88</td>
<td>0.90</td>
</tr>
</tbody>
</table>

very unlikely hypotheses generated in the manually designed PIN for ellipses, circles, and triangles, albeit with very low probabilities (which the figures cannot display).

This increased number of false or weak hypotheses increases the computational and storage overhead at each node.

Table 12 presents quantitative comparisons in terms of common hypotheses generated in the two networks. The first column displays the results for the aerial image. The execution times for the automatically designed PIN is less than the manual one. The second to the sixth rows list the number of target organizations hypothesized by the manually designed, by the automatically designed, and by both PINs.
We also studied the nature of the final probability assignments for the common organizations hypothesized by both PINs. For example, suppose the manually designed PIN assigned \( \{P_i^{\text{manual}} | i = 1, \ldots, N\} \) for \( N \) instance of a given organization and the automatically designed one assigned \( \{P_i^{\text{auto}} | i = 1, \ldots, N\} \) for the same organization. Then Eq. 5.5, reproduced here, measures the similarity.

\[
C_{\text{sim}} = \frac{\sum_{i=1}^{N} P_i^{\text{manual}} P_i^{\text{auto}}}{||P^{\text{manual}}|| ||P^{\text{auto}}||}
\]  

(6.28)

where the denominator is the product of the norm of the two probability vectors. \( C_{\text{sim}} \) is 1 when the relative assignments are perfectly preserved and 0 when none are preserved.

The quantitative measure, \( C_{\text{sim}} \), for the common target hypotheses are also tabulated in Table 12. \( C_{\text{sim}} \) is consistently high signifying that the relative probability assignments of the manually designed PIN and the automatically designed PIN are similar.

The second and the third columns of Table 12 list the performance comparisons for two other images. The numbers warrant conclusions similar to those for the aerial image.

### 6.6.3 Performance Evaluation

At equilibrium, the PIN hypotheses of various organizations are associated with probability values. In this section we investigate the robustness of the output probability values of the automatically designed PIN with respect to the conditional probabilities and the prior probabilities of the root nodes. The condi-
tional probabilities (Eq. 6.22) are specified by the functions $f_{LOC}(l_k, l_{k+1}, \ldots, l_{kn})$ and $P(X = 1 \mid U_1 = u_1, \ldots, U_n = u_n)$. The function $P(X = 1 \mid U_1 = u_1, \ldots, U_n = u_n)$ is in turn specified by a constant and a functional form dependent on the geometry of the feature (Eq. 6.24). We study the robustness of the automatically designed PIN probability outputs with regards to these constants, functional forms, and the $f_{LOC}$ functions. In addition, we also look into the equal prior assumption for the root nodes of the PIN. For these investigations we follow the earlier methodology to evaluate the performance of the manually structured PIN.

**Locational Compatibility Function, $f_{LOC}$**

The choice of the support of the locational compatibility function (the domain over which its value is one) depends on the necessary spatial resolution. The large value for the size of the support, $t$, means that the features that are spatially distance are declared to be compatible. This reduces the ability to resolve nearby features, thereby producing fewer hypotheses. However, as we shall see the reduction in the number of hypotheses for the automatically designed PIN is not drastic.

We focus on the variation of the number of hypotheses of the target nodes as support $t$ ranges from 3 to 15 pixels. The results are shown in Table 13. The variation in the number of hypotheses is low over a wide range of support sizes. From these results we infer that like the manually designed PIN, the automatically designed PIN is robust with respect to the locational tolerance parameter $t$. 
Table 13: Table showing the variation of the number of hypotheses of some salient nodes with parameter $t$ of the $f_{LOC}$ function.

<table>
<thead>
<tr>
<th>$t$</th>
<th>Ellipse (E)</th>
<th>Triangle (T)</th>
<th>Circle (C)</th>
<th>Quadrilateral (Q)</th>
<th>Ribbon (R)</th>
<th>Parallelogram (P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>8</td>
<td>13</td>
<td>22</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>8</td>
<td>13</td>
<td>22</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>7</td>
<td>13</td>
<td>21</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>7</td>
<td>10</td>
<td>19</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>7</td>
<td>10</td>
<td>19</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>19</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>15</td>
<td>10</td>
<td>13</td>
</tr>
</tbody>
</table>

**Prior Probabilities**

The prior probabilities specified for the PIN are those of the root node features. In the absence of information to the contrary, we assume equal priors for the root nodes. Here we consider the robustness of this assumption by comparing the behavior of the final PIN probabilities with different priors.

As an indicator of performance we study the behavior of the probability assignments of the target nodes, E, T, C, Q, R, and P. Since the random variables are binary we consider the probability of just one state, namely that of existence. For each composite node we plot the probabilities associated with the features at different locations and see how they change with different priors. Since the relative values of the probabilities are far more important that their absolute assignments, we use the normalized correlation based measure of Eq. 5.5 to quantify the similarity between two probability assignments.
Figure 38: The robustness of the equal prior assignment. This shows the plot of the probability assignments of features at different locations for 25 random prior assignments. The features considered are (a) Parallelogram (P) (b) Ribbon (R) (c) Ellipse (E) (d) Circle (C) (e) Quadrilateral (Q) and (f) Triangle (T).
We started from 25 random assignments of prior probabilities and compared each result with that resulting from equal prior assignment. The prior probabilities were assigned random values from a uniform distribution in $[0, 1]$. The results are shown in Fig. 38 and Table 14. The plots of Fig. 38 are of the final probabilities at the composite node versus the feature at different locations. Note that the relative assignments of the probabilities are preserved most of the time for different priors, although the absolute values vary. In other words, it is the shape of the plots that interest us, rather than their vertical positions. This effect is also quantitatively validated by the high value for the average similarity index $C_{sim}$ between the random prior and equal prior cases, as shown in Table 14.

Conditional Probability Constants

The conditional probability at each node (Eq. 6.22) involve a function $P(X = 1|U_1 = u_1, \ldots, U_n = u_n)$ which in turn involves a constant $\Lambda(u_1, \ldots, u_n)$. These constants are determined by the RPSD definitions of the node organizations (Eq. 6.25). In this section we study the sensitivity of the PIN outputs with respect to these constants. As performance criteria we again consider the probability assignments of the target nodes for various perturbations of the likelihood ratios $\Lambda(u_1, \ldots, u_n)$.

$\Lambda(u_1, \ldots, u_n)$ is a likelihood ratio and the significance of a perturbation depends on its value; a change to a small likelihood value is more significant than a similar change to a large likelihood value. To circumvent this problem we employ the following strategy. We convert a likelihood ratio into probability measure by $C(u_1, \ldots, u_n) =$
Table 14: The table of average similarity coefficients, $C_{\text{sim}}$, for random perturbations of PIN parameters, compared with the noiseless case. The first row lists the $C_{\text{sim}}$ of the target hypotheses for randomized priors. The second row corresponds to the randomized constants in the conditional probability expression. The rest of the rows list the $C_{\text{sim}}$ for randomized pmfs of the RPSDs at each PIN node.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Noise type</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>E</th>
<th>C</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priors</td>
<td>$U[0, 1]$</td>
<td>0.9994</td>
<td>1.0</td>
<td>0.9996</td>
<td>0.9995</td>
<td>0.9972</td>
<td>0.9991</td>
</tr>
<tr>
<td>$C_{(u_1, \ldots, u_n)}$</td>
<td>$U[-0.25, 0.25]$</td>
<td>0.9997</td>
<td>0.9976</td>
<td>0.9990</td>
<td>0.9960</td>
<td>0.9924</td>
<td>0.9950</td>
</tr>
<tr>
<td>RPSDs</td>
<td>$U[0, 0.1]$</td>
<td>0.9989</td>
<td>1.0000</td>
<td>0.9993</td>
<td>0.9994</td>
<td>0.9889</td>
<td>0.9996</td>
</tr>
<tr>
<td></td>
<td>$U[0, 0.2]$</td>
<td>0.9981</td>
<td>0.9999</td>
<td>0.9984</td>
<td>0.9990</td>
<td>0.9768</td>
<td>0.9992</td>
</tr>
<tr>
<td></td>
<td>$U[0, 0.3]$</td>
<td>0.9969</td>
<td>0.9998</td>
<td>0.9980</td>
<td>0.9985</td>
<td>0.9681</td>
<td>0.9986</td>
</tr>
<tr>
<td></td>
<td>$U[0, 0.4]$</td>
<td>0.9957</td>
<td>0.9998</td>
<td>0.9970</td>
<td>0.9985</td>
<td>0.9725</td>
<td>0.9984</td>
</tr>
</tbody>
</table>

To this probability measure, $C_{(u_1, \ldots, u_n)}$, we add uniformly distributed noise between -0.25 to 0.25, constrained to the valid interval of [0, 1]. Finally, this perturbed $C_{(u_1, \ldots, u_n)}$ is converted to likelihood values according to

$$\Lambda_{(u_1, \ldots, u_n)} = \frac{C_{(u_1, \ldots, u_n)}}{1-C_{(u_1, \ldots, u_n)}}.$$

The final distribution of the perturbed case is again compared with the unperturbed case. We ran 25 different trials and the results are shown in Fig. 39 and Table 14.

From the plot of Fig. 39 we can readily observe the qualitative similarity of the relative probability assignments for different values of the constants, $\Lambda_{(u_1, \ldots, u_n)}$. This is corroborated by the quantitative similarity index, $C_{\text{sim}}$, shown in Table 14 which compares the similarity of the probability assignments for perturbed $\Lambda$ values with the unperturbed case.

*Note that the present $C_{(u_1, \ldots, u_n)}$ is analogous that used in the manually structured PIN (Eq. 6.22).
Figure 39: The robustness of the choice of the constants $A_{(u_1, \ldots, u_n)} \ (C_{(u_1, \ldots, u_n)})$. This shows the plot of the probability assignments of features at different locations for 25 random prior assignments. The features considered are (a) Parallelogram (P) (b) Ribbon (R) (c) Ellipse (E) (d) Circle (C) (e) Quadrilateral (Q) and (f) Triangle (T).
Functional Dependence of the Conditional Probabilities

The definition of $P(X = 1|U_1 = u_1, \ldots, U_n = u_n)$ in the conditional probability expression (Eq. 6.22) also involves a function ($\Lambda_{geom}$, Eq. 6.24) which depends on the geometry of an organization instance. These geometric functions, $\Lambda_{geom}$, are tied to the RPSD outcome probabilities of the node organization outcomes according to Eq. 6.26. To study the dependence of the PIN on $\Lambda_{geom}$ we employ the same performance evaluation scheme as before, namely considering the final probability distribution of the target nodes for different values of $\Lambda_{geom}$. To generate perturbed values of $\Lambda_{geom}$, we add noise to the probabilities used in the RPSDs of the organizations, namely the probabilities of nullity of the primitives and relations, and the probability mass functions (pmfs) of the primitive attributes and the relational parameters. After adding noise, uniformly distributed in $[0, b]$, we normalize the pmfs.

We found out that, in general, a small change in the pmfs, and hence a perturbation in the $\Lambda_{geom}$, result in drastic changes in the probabilities of the PIN hypotheses. However, if we carefully preserve the support of the pmfs, i.e. we add noise only to the non-zero probabilities, then the outputs of the PINs are quite stable for large distortions in the pmfs. This is evidenced by the high values of similarity indices as listed in Table 14 when uniform noise in $[0, 0.1]$, $[0, 0.2]$, $[0, 0.3]$, and $[0, 0.4]$ was added only over the pmf support. Thus, we deduce that although the PIN outputs are sensitive with respect to the support of the pmfs, they are robust with respect to the exact probability profile chosen over a fixed support. The support of the
pmfs places stricter restrictions on the primitive and relational attribute values in an RPSD than the probability profiles. A change in the hard limits of the attributes can result in significant changes in what a RPSD models. In fact, we expect PIN using RPSDs with different supports to behave differently.

In this chapter, we presented an algorithm based on entropic similarity measures and random parametric structural descriptions to design a Perceptual Inference Network. As confirmed by our experiments, the algorithm is reasonably stable with respect to minor perturbations of the input parameters. We also presented results on real images and made comparisons of an automatically designed PIN with a manually designed one for the same perceptual organization task. We found the performance of the automatically structured PIN to be very similar to the manually structured one, if not better.
CHAPTER VII

Attentive Algorithm: Resource Management

“Feature detection or discrimination only has to be made once. That is, once a particular “observation” of some feature has been made, by a specialized, localized portion of the brain, the information content thus fixed does not have to be sent somewhere else to be rediscriminated by some “master” discriminator. ... There is no Cartesian Theatre”

Daniel Dennett

The role of the management scheme is depicted in Fig. 6 and is built around the concept of the Perceptual Inference Network (PIN) developed in Chapter V. Our aim is to generate an hierarchical description of the scene. This is done using two modules: the pre-attentive and the attentive. The pre-attentive module, described in Chapter IV provides evidence in terms of primitive organizations like parallelism, continuity, closure, and strands. The attentive organization (Chapter V) is done using PIN which integrates information from the preattentive module and hypotheses in the presence of noise and occlusions. The output set of hypotheses is large and redundant. A set of lines is described as a parallelogram and/or ellipse and/or circle. There is ambiguity in such an description. Our strategy is to use special
purpose modules to resolve the ambiguous hypotheses and generate a comprehensive description of the scene. We probe the original grey level image for evidence to resolve the ambiguities. Although the algorithm is demonstrated using simple organizations, it is easily extendible to complex shapes.

The present formalism is useful not only in the context of perceptual organization but also in advancing the solution to the problem of computer vision which is not possible by using a single processing module but instead involves the intelligent coordination and judicious management of a number of specific visual task algorithms. For example, any one shape from X algorithm cannot solve the $2\frac{1}{2}$D sketch problem nor can edge detection provide all the necessary information for recognition; texture and shading information are important in elucidating the underlying surface structure. Although more needs to be done, there has been significant progress in the design of efficient specific task modules solving specific visual tasks. Along with this success comes the realization that there is a trade off among the accuracy of results, the knowledge of domain parameters, and the number of constraints one can impose. The generalization of algorithms to various domains is typically accompanied by inaccuracy and inefficiency. Thus the solution lies not in the generalization of different algorithms but in the efficient management and integration of specific visual task modules. Such paradigms are scarce in computer vision; our present work is a step in that direction.

The integration of visual modules that we seek differs from that usually considered in the Marr paradigm. In the Marr paradigm, integration occurs among
various shape from X modules on an algorithmic level by reformulating the constraints and the underlying theory. For a good review and a stimulating discussion of such approaches the reader is referred to [129]. Our mode of integration is not on the algorithmic level but involves the judicious management of existing algorithms. The success of our approach relies on the opportunistic use of visual task modules which are not necessarily restricted to shape from X algorithms. The type of visual tasks we consider may include various symmetry analyzers, region analyzers, corner detectors, shape from X, texture analyzers, road analyzers, and so on. This is not to imply that algorithmic integration is unnecessary; it can make our task easier but certainly cannot be the full solution. We believe that, although the algorithmic integration of a small number of modules is possible, such integration of all visual task modules is effectively impossible.

We engage probabilistic reasoning and information theory to manage such visual task modules opportunistically. Levitt et al. [130], in their INTACTS system, also employ a control strategy based on utility theory and Bayesian network inference. Candidate processing actions are selected by utility computations based on the estimated value and the cost of each action. The value of each action is calculated as the expected increment in evidential value achieved at the parent hypotheses due to the action. The choice is biased to those actions that change the probabilities in the network the most. Executable actions include providing terrain support, the use of high resolution sensors, and actions to structure the network dynamically. Recently, Rimey and Brown [121] used Bayesian nets and maximum expected utility decision
rules to select a sequence of visual operations. The Bayesian nets was built on high level primitives such as utensils, cups, and plates. The visual operations involved camera foveation, cup detection, napkin detection and the like. They provide a solution to “where to look next?,” an important question in the context of active vision.

We present the theory behind the management scheme in Section 7.1. In Section 7.2, we briefly present the visual modules used in our results, which are presented in Section 7.3 on the aerial image.

7.1 Theory

The Perceptual Inference Network (PIN) outlined in the previous chapter allows us to integrate and make inferences from multiple sources of information such as parallelism, corners, closure, and so on. The primary information is integrated to form composite organized hypotheses like parallelograms, circles, ellipses, or ribbons. The integration of multiple sources of information is achieved in an probabilistic framework which is also computationally efficient. In this section we investigate ways of refining the hypotheses in a PIN by the intelligent use of special purpose modules. The PIN framework is very amenable to the control of a battery of special purpose modules. We can associate the modules with the nodes of the PIN as providing evidence for it. For example, a corner analysis module can be associated with the nodes representing corners, \((N10, N11, N14)\) in Fig. 21 and \((EPQRnnnCorner, EPQRnnnnCorner, EPQRnnnnnCorner, EPQRnnnnnn, TnnCorner, TnCorner, Tnn)\) in Fig. 34.
The use of special purpose modules is not rare in Computer Vision. There are a number of algorithms which are designed with specific tasks in mind. For example, in domains where circles are the object of interest, the use of a special purpose circle detector is warranted. Or one might have special purpose road marking detector. These special purpose tasks tend to be computationally very expensive and have very limited applicability. Hence, it is highly desirable that we use them very judiciously. We ideally want to apply them only when and where we expect highest return.

The norm for "highest return" is a matter of choice. In keeping with our probabilistic approach we chose an entropy (or information theoretic) based norm. We want to use the special purpose module which gives us greatest degree of entropy reduction per unit computational resource. The measure of computational resource for a special purpose module involves the computational time, the use of special purpose hardware, amount of storage, and so on. Thus, our aim is to reduce the uncertainty in the final hypothesis set to the greatest extent while using the least possible amount of computational resources.

### 7.1.1 Control Algorithm

Our goal is to produce the best description of the image using a set of salient organization. For example, we might be interested in parallelograms, quadrilaterals, ellipses, circles, and ribbons. The set of Bayesian nodes representing these structures of interest forms the target set; \((N23, N19, N6, N8, N13)\) in Fig. 21 and \((P, Q, E, T, C, R)\) in Fig. 34. Our special purpose modules will be associated with the composite nodes representing the structures that they are designed to look for. For
illustration, consider that we have special purpose modules for corners, contours, and regions associated with composite nodes: \((N10, N11, N14, N2, N3)\) in Fig. 21 and the evidence nodes in Fig. 34.

The algorithm is depicted in Fig. 40. Conflict arises when we have more than one organizational hypothesis for the same location in the image and the associated probabilities are nearly equal. Thus, there is an ambiguity in the description. For example, we might describe a set of edges as an ellipse, a rectangle, or a ribbon. If the probabilities of the respective hypotheses are very different then it is easy to resolve such a conflict; that is not true if they are close. For these experiments, if 

\[
\frac{p_1}{1-p_1} < 2 \cdot \frac{p_2}{1-p_2} \quad \text{and} \quad \frac{p_2}{1-p_2} < 2 \cdot \frac{p_1}{1-p_1},
\]

the likelihood ratio of one hypothesis is within twice the likelihood ratio of the other, then they are declared to be close together. Other schemes are possible, but this seems to work pretty well.

We wish to apply our computational resources to minimize the size of the conflict(s) as described below. The size of a conflict is determined as follows. First, we form a locational conflict graph whose nodes denote the target hypotheses and links denote conflicts among them. We detect such conflicts by looking for locational overlap and nearly equal probabilities, as above. If there is a locational conflict between two hypotheses and the probabilities are not close (decided as above) then the node representing the hypothesis with lower probability is removed from further consideration. The locational conflict graph is next broken into connected components. The size of a conflict is determined by the size of the associated component. We will apply the special purpose computational modules until the components of
For every connected component:
  Is component small?

Yes

Target Hypotheses (T1, ..., Tn)

Create Locational Conflict Graph

Create Information Matrix

Rank Sensors

Apply resources

Resolve hypotheses by clique analysis

Final Output

Figure 40: Resource control algorithm.
the conflict graph are below a specified size or we have exhausted the information from our computational modules. The choice of the minimum tolerable size depends on the final conflict resolution scheme one chooses. For example, our conflict resolution scheme involves clique detection, which is NP complete. Therefore, large components make the final process impractical; a maximum component size of 5 suffices for our purposes.

7.1.2 Information Theoretic Formalism

In this section we present the means by which the special purpose computational modules are used to reduce the uncertainty in the hypothesis set. Suppose we have a set of target hypotheses \( T_1, \ldots, T_n \), involved in locational conflict. This conflict will be resolved by using each appropriate module to provide additional evidence to the corresponding network sensory nodes. Let the Bayesian nodes in these sensory composite nodes be \( X_1, \ldots, X_m \).

The total uncertainty in hypothesis \( T_i \) can be measured by the entropy \( H(T_i) \). The uncertainty in \( T_i \) given the knowledge of the sensory node \( X_j \) is \( H(T_i|X_j) \). The expected entropy reduction with the knowledge of \( X_j \), or the mutual information between \( T_i \) and \( X_j \), is

\[
I(T_i, X_j) = H(T_i) - H(T_i|X_j)
\]

(7.1)

Substituting standard expressions for the entropy in terms of the discrete probabilities of the events, we have

\[
I(T_i, X_j) = \sum_{T_i = t_i} \sum_{x_j = x_j} P(t_i, x_j) \log \frac{P(t_i, x_j)}{P(t_i)P(x_j)}
\]

(7.2)
Note that the mutual information is itself the relative entropy between the joint distribution and the product of the marginals.

Computation

The mutual information can be calculated in a PIN by observing that $P(t_i, x_j)$ is actually the notation $BEL(t_i, x_j)$ in the context of a Bayesian network and that $BEL(t_i, x_j) = BEL(x_j | t_i) BEL(t_i)$. Thus to calculate the mutual information between $T_i$ and $X_j$ we need to have $BEL(t_i), BEL(x_j)$, and $BEL(x_j | t_i)$. The values $BEL(t_i)$ and $BEL(x_j)$ are present the nodes of a PIN at equilibrium. $BEL(x_j | t_i)$ can be calculated by temporary message passing in the same PIN as follows. We temporarily instantiate node $T_i$ through its range and for each value, $t_i$, note the value of the final $BEL(X_j = x_j)$. Although the temporary message passing scheme seems to involve a lot of overhead by instantiating $T_i$ to each of its states, this is not so. We need not perform locational matching at each composite node to determine the relevant Bayesian node. We use the pointer links between nodes, discussed earlier, for this purpose. As the reader might recall, we store the links between the nodes involved in message passing whenever we find a locational match. These links are just traced to perform the temporary updating instead of doing a full locational match.

---

1The $BEL$ notation is used to be consistent with Pearl [115]. $BEL(t_i, x_j) = P(t_i, x_j)$, the Bayesian probability.
Management

Using the method outlined above we calculate the mutual information between each pair, $T_i$ and $X_j$. This forms our information matrix $IMAT$ of size $n \times m$ where $IMAT(i, j) = I(T_i, X_j)$. We form the column sums $I_{EXP}(X_j) = \sum_{i=1}^{n} I(T_i, X_j)$, which is the total sensor mutual information. The sensor having the largest value for this is expected to provide the largest amount of the information towards resolving the ambiguity. Associated with each sensory node is a computational module. Each module has a characteristic computational expense, which may include the time of execution or amount of hardware needed. Let the computational expense to provide evidence for $X_j$ be $\tau_j$.

We have considered two resource management strategies. The first strategy assumes that we have a time constraint; we have to complete our computation within an interval, $T$. The optimal choice in this case can be decided by solving the following problem:

\[
\begin{align*}
\text{Maximize} & \quad \sum_{j=1}^{m} I_{EXP}(X_j) \delta_j \\
\text{Subject to} & \quad \sum_{j=1}^{m} \tau_j \delta_j < T
\end{align*}
\]

(7.3)

where $\delta_j$ is 1 if the resource is applied to investigate $X_j$ and is 0 otherwise. The idea to maximize the total information gain within the time constraint. This is the strategy used by Levitt, et al. [130] but with a different utility function. The problem is NP-hard and, hence, computationally expensive. Since our aim is the reduction in ambiguity, our main constraint is not time. We adopt a different strategy based on conflict resolution.

\textsuperscript{2}Note that this is different from Fisher's information matrix used in detection and estimation theory.
The second (adopted) strategy aims at applying resources until the conflicts have been resolved or we have exhausted all available resources. The sensory nodes $X_i, \ldots, X_m$ are ranked based on the expected information gain per unit computational resource, $I_{\text{EXP}}(X_j)/\tau_j$. For example, suppose we have three special purpose modules: corner, edge segment, and region analysis. Further suppose that the region analysis module takes three times as long to execute as does the corner module, and that the segment analysis module takes twice the time of the corner module. In this case, we divide the expected information $I_{\text{EXP}}$ of the sensory nodes corresponding to the corner nodes by 1, those of the segment modules by 2, and those of the region module by 3. Thus, our final ranking of the sensory nodes will be based on the expected information gain of the PIN per unit computational time. We then apply the special purpose modules, in order, only at those locations dictated by the nodes and update the PIN with the corresponding information. This updating is again computationally efficient because we use the pointer links at the nodes, as mentioned before.

7.1.3 Return Condition

Fig. 40 offers two return options. The first option is to apply the special purpose modules to find evidence for all the sensory nodes $X_i$ to resolve conflict in a particular component. The second option involves applying only a top few of the ranked special purpose modules and then recomputing the locational conflict graph. We use the first option; the second is computationally expensive in a sequential implementation. However, in a parallel implementation the second option might be appropriate.
7.1.4 Final Hypothesis Resolution

After applying the special purpose modules to reduce the component sizes we expect to be left with a sparse locational conflict graph. The actual effect on the locational conflict graph depends on the information provided by the special purpose modules and depends on the degree of "surprise" they provide. At any rate, this will represent the best we can do with the available computational resources. Other sensor modalities might be warranted if the location conflict graph is still dense.

All the features still participating in the locational conflict graph, which represent the set of strongest hypotheses, are sent to preattentive module of the next hierarchical level to detect organizations among them. In case we are at the topmost level of our desired hierarchy we resolve the conflict of the locational graph in the following manner to generate a complete description of the scene.

For each component of the final locational conflict graph we form the locational consistency graph, which is its complement. In the locational consistency graph we look for the most significant clique. The significance of the clique is not determined by the number of nodes but by the sum of the individual hypothesis probabilities. In case of a tie, the size of the clique is the determining factor; we choose the one having the larger number of nodes. This constitutes our consistent set of hypotheses for each component. The total set of largest cliques (one from each component) forms our description of the scene.
Using the same algorithm, we can construct a set of descriptions of decreasing importance from the locational conflict graph. To form such a set we rank the cliques in the locational consistency graph according to the sum of probabilities of the individual nodes. The importance of a ranked set of descriptions is enormous in the context of object recognition. Parts of objects missing in first description might be present in the next best description. This implements a form of the “Principle of Least Commitment”.

7.2 Special Purpose Modules

Before presenting the results, we present the special purpose modules used in these experiments. The call to a module provides it a location and its output is an updated location and a confidence measure. We consider three special purpose modules in this work: corner, edge segment, and region analysis modules. The algorithms represented here are very simple and may certainly be replaced by more sophisticated ones. These are used just to illustrate of the viability of the overall scheme.

7.2.1 Corner Analysis Module

The corner analysis module investigates a square region around the location of interest. The size of the square region is determined by the size of edge operator used to detect the initial edge segments. The coarser the edge operator used, the less accurately the edge points will be located. For the OZCO edge detector [21] the expected edge pixel migration is bounded by $3\gamma$ [131], where $\gamma$ is the scale of the
operator. Thus, our region is a $6\gamma \times 6\gamma$ square centered at the location of interest. At each point in the region we compute the following:

\[
\text{CORNERITY}(i,j) = |\nabla I(i,j)||\nabla(\text{Grad}_{\text{dir}}(I(i,j)))|
\]  

(7.4)

where $I(i,j)$ is the image gray level value at $(i,j)$ and $\text{Grad}_{\text{dir}}$ is the gradient direction. This is the Kitchen and Rosenfeld cornerity measure [132]. The updated location is where the measure is maximized. The mean ($\mu$), standard deviation ($\sigma$), and the maximum value ($\text{max}$) of the measure in the region are calculated. The updated confidence, $\text{BEL}(1)$, is:

\[
\text{BEL}(1) = \left\{ \begin{array}{ll}
1 - \frac{\sigma^2}{(\text{max} - \mu)^2} & \text{max} \neq \mu \\
1 & \text{max} = \mu
\end{array} \right.
\]  

(7.5)

The confidence increases for low background noise and for strong (in terms of the gray level) corners. In this case, $\sigma$ of the measure is low. Also, the measure would be high near the corner and low elsewhere and, consequently, $\mu$ is low. Thus, we have a large confidence in the corner. The behavior is intuitively satisfactory.

Since the algorithm always investigates a constant sized image, the complexity of the algorithm is $O(1)$. We consider the computational cost, $\tau_j$, to be the size of the region analyzed, $36\gamma^2$.

### 7.2.2 Segment Analysis Module

This module investigates the presence of an edge segment, straight or curved, and returns an updated location of the segment, together with a confidence measure. A strip centered at the queried location is investigated (see Fig. 41). The width
Figure 41: Segment and the area analyzed to provide evidence for it.
of the strip is $6\gamma$, based on the previous arguments (c.f. corner analysis module discussion). The strip is computed as follows.

The local tangent direction of the (parameterized) curve is estimated using:

$$
tan(\psi) = \frac{y(s) \ast \hat{g}(s)}{x(s) \ast \hat{g}(s)}
$$

(7.6)

where $(\ast)$ denotes convolution, $x(s)$ and $y(s)$ are the coordinates of the curve and $\hat{g}(s)$ is the first derivative of Gaussian $^3$. The Gaussian smoothing is needed to regularize the derivative problem. We walk perpendicular to the local tangent, $\theta$, to a distance of $3\gamma$ on either side. At each location we compute the gradient of the image function. We update the edge location to the maximum of the gradient magnitude profile. The mean ($\mu$), standard deviation ($\sigma$), and the maximum ($max$) of all the investigated locations in the strip is calculated. The confidence in the presence of the edge segment is computed as:

$$
BEL(1) = \begin{cases} 
1 - \frac{\sigma^2}{(max - \mu)^2} & max \neq \mu \\
1 & max = \mu 
\end{cases}
$$

(7.7)

The justifications for the expression are similar line to those of the corner detector. The area investigated is proportional to the length of the segment. So the complexity of the algorithm is $O(N)$. The computational cost, $\tau_j$, is the area investigated, $6\gamma L$, with $L$ the length of the segment.

$^3$The scale of Gaussian used for smoothing is 2.0. The choice of this parameter is not crucial.
Figure 42: Determination of area to be analyzed to provide evidence for a region.
7.2.3 Region Analysis Module

The region analysis module accepts as input a boundary description and analyzes it to determine the presence of a planar gray level enclosed region. The output is an updated region boundary and a confidence measure. The first step of the algorithm is to call the segment analysis module to update the boundary and give a confidence measure, $BEL_{\text{bound}}(1)$. The region inside the updated boundary is then analyzed for a planar fit to the gray levels. To determine if a point is inside the boundary, we shift the boundary coordinates so that the origin is the point under consideration and walk around the boundary to identify crossings with the positive $x'$ axis (see Fig. 42). A tangent to the $x$ axis is not counted as a crossing. If the number of crossings is odd then the point under consideration is inside the region, otherwise it is outside. Note that we need to do this for only for points inside the upright bounding rectangle of the region.

We compute the least squares planar fit to the gray levels in the region as follows. Let a plane be characterized as:

$$g_p(x_i, y_i) = \theta_1 + \theta_2 x_i + \theta_3 y_i$$

where $(x_i, y_i)$ are the coordinates of the point inside the region of interest, $g_p$ is the height of the fitted surface, and the $\theta_i$ are the parameters of the surface. Let the total number of points in the region be $n$ and the image gray level at $(x_i, y_i)$ be denoted by $g(x_i, y_i)$. The total squared error, $E(\theta)$, is:

$$E(\theta) = \sum_i (g(x_i, y_i) - (\theta_1 + \theta_2 x_i + \theta_3 y_i))^2$$

$$= (G - Z\Theta)^T(G - Z\Theta)$$

(7.8)
where \( G = [g(x_1, y_1), \ldots, g(x_i, y_i), \ldots, g(x_n, y_n)]^T \) is a vector of all the gray level values (observations), \( \Theta = [\theta_1, \theta_2, \theta_3]^T \) is a vector of the parameters, and

\[
Z = \begin{bmatrix}
1 & x_1 & y_1 \\
\vdots & \vdots & \vdots \\
1 & x_i & y_i \\
\vdots & \vdots & \vdots \\
1 & x_n & y_n \\
\end{bmatrix}
\]

is a matrix of observations on the explanatory variables. Taking the derivative of the squared error with respect to each parameter and setting each to zero, we get an expression for the estimated parameters, \( \hat{\Theta} \):

\[
\hat{\Theta} = (Z^T Z)^{-1} Z^T G \quad (7.9)
\]

\( Z^T \) is the transpose of \( Z \), \( Z^T Z \) is a 3 \times 3 matrix given by

\[
Z^T Z = \begin{bmatrix}
\sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i y_i \\
\sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i^2 y_i \\
\end{bmatrix} \quad (7.10)
\]

and

\[
Z^T G = \begin{bmatrix}
\sum_{i=1}^{n} g(x_i, y_i) \\
\sum_{i=1}^{n} x_i g(x_i, y_i) \\
\sum_{i=1}^{n} y_i g(x_i, y_i) \\
\end{bmatrix} \quad (7.11)
\]

Substituting Eq. 7.9 into Eq. 7.8, we get the minimum planar fit error, \( E_{min} \):

\[
E_{min} = G^T G - (Z^T G)^T \hat{\Theta} \quad (7.12)
\]

The smaller the fit error, greater our confidence in a homogeneous region. We update the confidence as:

\[
BEL(1) = BEL_{bound}(1)(e^{-\frac{E_{min}}{\eta_0}}); \quad (7.13)
\]

where \( BEL_{bound}(1) \) is the confidence in the boundary of the region and \( g_0 \) is the value of the significant contrast parameter. For our experiments we consider contrasts of
values greater than 25 ($g_0 = 25$) to be significant. To estimate the computational cost of region analysis we consider the following. The work to calculate the fit error is constant for each interior point. Let the length of the boundary be $N$. Then number of points inside the bounding box is $\mathcal{O}(N^2)$. Thus, the total complexity is $\mathcal{O}(N^3)$. We take the cube of the length of the boundary to be good estimate of the computational cost of region analysis.

### 7.3 Results

The visual task modules presented in the previous section were managed using the algorithm presented earlier. The Perceptual Inference Network is implemented in C++, and runs on Sun Sparc-ELC machines. The voting and graph theoretic preattentive methods provide the preliminary evidence to the PIN. This evidence is integrated by the PIN to construct composite hypotheses. For the purpose of this section we use the manually structured PIN (Fig. 21). These hypotheses are then analyzed and conflicts among them resolved using the special purpose modules in accordance with information theoretic criteria as described. The final hypotheses are selected using the algorithm based on clique detection. We present results on the aerial scene.

Pre-attentive organization, implemented as a voting module followed by a graph theoretic module, are introduced as evidence to the perceptual inference network. After the network settles to equilibrium we have various organization hypotheses with associated probabilities. The results were shown in the previous Chapter. Note how we have a number of redundant ellipse and circle hypotheses. We want
a cover of the image in terms of a non conflicting set of target hypotheses which are identified as $N_{23}$ (parallelograms), $N_{19}$ (quadrilaterals), $N_{13}$ (ribbons), $N_{6}$ (ellipses) and $N_{8}$ (circles). To this end we construct the locational conflict graph which has 32 links forming 15 connected components. We compute the information matrix using the temporary message passing discussed before and use that to rank the needed sensory evidence that the special purpose modules can supply, namely, for corners ($N_{10}, N_{11}, N_{14}$), segments ($N_{2}$), and regions ($N_{3}$). Based on the information matrix the PIN requested evidence on 70 (out of 249) corners, 18 (out of 67) regions, and 5 (out of 147) segments. Note that the special purpose modules were applied to provide evidence for about 20% of the hypothesized features $^4$. The requested locations are shown in Fig. 43. Note how the special purpose modules are used opportunistically and at selected places. For this example, we provide all the requested evidence. However, in a real situation the special purpose modules will provide evidence based on the ranking until the conflicts are resolved.

After the introduction of all the evidence the conflict graph had 29 links forming 15 components and its structure did not change greatly. However, there were some links added and some which were deleted. That the graph did not change much is not a drawback of the algorithm but a consequence of the fact that the special purpose modules did not provide much unexpected information in this particular case. In general, we would feed all the hypotheses participating in the locational conflict graph into the preattentive module again to construct the next level of the

---

$^4$The actual saving is more than this when considered against blindly applying the computational modules over the whole image.
Figure 43: Locations where special purpose modules were applied. (a) Corner module locations. (b) Segment analyzer module. (c) Region analyzer module locations.
hierarchy. However, for this particular experiment we end at the first level and demonstrate our final conflict resolution scheme. In Chapter VIII we present results where we use the results of the first level as an input to the second level.

The final resolved hypotheses at the first level are shown in Fig. 44(a). The overlay of the description on the initial edge image is shown in Fig. 44(b). The final description obtained is very reasonable. The second best description is shown in Fig. 44(c) and (d). These represent some of the less probable descriptions of the scene and would be considered (as a part a complete vision system) if the best description does not serve the purpose.
Figure 44: Final resolved hypotheses for the aerial image. (a) Best set of hypotheses. (b) Best set of hypotheses overlaid on the edge image. (c) Second best set of hypotheses. (d) Second best set of hypotheses overlaid on the edge image.
CHAPTER VIII

Evaluation

“Read not to contradict and confute,
nor to believe and take for granted,
nor to find talk and discourse,
but to weigh and consider.”

Francis Bacon

The perceptual organization system is implemented in C/C++, and runs on Sun Sparc machines. The voting and graph theoretic preattentive modules provide cues which are integrated by the PIN to construct composite hypotheses. These hypotheses are then analyzed and conflicts among them resolved based on the information from the special purpose modules. The final hypotheses at the highest hierarchical level of description are selected using an algorithm based on clique detection.

Results in perceptual organization are difficult to display, because they are in a symbolic form. A single pictorial form might not convey the full meaning. Ideally, we want to display each organization in a different image, which is not practical. As a compromise we display each type of organization in a separate image. We have also not distinguished strong organizations from the weaker ones in these images;
there has been no selection process. The reader is advised to judge the results in the light that they are strong indicators of an underlying causal agent and in no way constitute the final object description; they are just hypotheses. The reader is also cautioned against judging the results with respect to the gray level image because the starting point of our algorithm are edges. Looking at a gray level image one uses various photometric cues such as shading information to infer structures; these are not used by our algorithm.

We present results on a set of 6 images chosen from different domains: blocks world image, indoor laboratory image, indoor image from a motion sequence, satellite image of a runway and an aerial stereopair of Munich, Germany. For the first three images we construct just the first level of description (Fig. 6). For the last three images we present results upto level 2 of the hierarchy and close the loop, so as to speak, of our algorithm (Fig. 6). The timing results for all the images are summarized in Table 15. The results on the first image are shown using the manually designed pin (see Fig. 21) and the others are using the automatically structured pin (Fig. 34).

For our experiments, the edges were detected by the optimal zero crossing operator [21] and the contours were segmented into constant curvature segments using [34]. Although the algorithm is designed to be efficient with respect to the number of edge segments, restrictions arise from memory considerations. The shortest path graph routine in the preattentive part is one of the bottleneck. Besides, the parallel PIN structure implemented on a serial machine entails significant overhead. Hence, we
had to threshold the detected edges based on the edge strength (amplitude $\geq 10$ gray levels) and length ($\geq 5$ pixels) to restrict our starting set of primitives. Note that, these thresholds might be relaxed by using efficient data structures or even removed as the computing technology improves.

8.1 One Level of Organization

In these sets of experiment we present results upto the first level of the description hierarchy. The images are described in terms of organizations like ellipses, parallelograms, triangles, circles, ribbons, and quadrilaterals. In the next section we presents results on images upto the second level of description. The second level represents organizations found among the features at the first level.

8.1.1 Blocks World

The gray level image 1 is shown in Fig. 45(a). The edges were detected using the optimal zero crossing operator (OZCO) presented in [21] at a scale, $\gamma$, of 1.5. The edge contours were segmented into constant curvature segments using a modified form of Wuescher and Boyer's algorithm [34]. The constant curvature segments are shown in Fig. 45(b). Note the boundary of the top of the brighter block in the middle of the image is not segmented into four straight line segments, but into two straight lines and an arc. This is not a drawback of the segmentation algorithm, but is an example of the types of errors that will be present in all low level algorithms.

Pre-attentive organization, implemented as a voting module followed by a graph

---

1Image courtesy of Dr. A. Etemadi of the University of Surrey.
Figure 45: Organizations detected by the pre-attentive perceptual organization module. (a) Gray Level Image (b) Edge contours detected (c) Closed boundary hypotheses (d) Edge strand hypotheses (e) Parallel edge segments draw as parallelograms (f) Parallelogram hypotheses detected as intersections of parallel strips.
theoretic module, detects organizations in a purely bottom up fashion from the edge segments. Some of the organizations detected are closed boundary hypotheses, parallels, strands of edge segments, and parallelograms. The closure groupings are shown in Fig. 45(c), the open strands of edge segments in Fig. 45(d), the parallels in Fig. 45(e), and the parallelograms detected in Fig. 45(f). These are introduced as evidence to the perceptual inference network (Fig. 21). The pre-attentive evidence consists of 6 closed figures, 18 strands, 8 ribbons, 22 parallels, and 69 segments. Note that not all the parallelograms are detected, and the set of closed edge segments is not complete. The pieces of evidence are assigned probabilities based on photometric characteristics as discussed in Section 5.2.5.

After the network settles to equilibrium we have various organization hypotheses with associated probabilities. Since it is impractical to give a complete listing of the node probabilities, we show some of the organizations as images. Fig. 46 depicts a set of such organizations. Note how the set of closed boundary hypotheses is more complete than in pre-attentive organization. Some parallelograms which were not detected before are now hypothesized, especially for the top face of the bright block in the middle. The set of corners predicted is also shown overlaid on the original gray level image.

The target nodes (see Fig. 21) are identified as N23 (parallelograms), N19 (quadrilaterals), N13 (ribbons), N6 (ellipses) and N8 (circles). We seek a description of the scene in terms of these organizations. After introduction of the pre-attentive evidence the PIN hypothesized 30 parallelograms, 30 quadrilaterals, 7
Figure 46: Organizations detected using the **attentive** module on the blocks image.  
(a) New closed boundary hypotheses (b) Set of parallelograms hypothesized (c) Ellipses predicted (d) Circles (e) Ribbons (no change from pre-attentive hypothesizing) (f) Corners hypothesized
ribbons, 36 ellipses, and 41 circles. We next construct the locational conflict graph which has 32 links forming 10 connected components. We compute the information matrix using the temporary message passing discussed before and use that to rank the needed sensory evidence that the special purpose modules can supply, namely, for corners (N10, N11, N14), segments (N2), and regions (N3). Based on the information matrix the PIN requested evidence on 39 (out of 197) corners, 11 (out of 56) regions, and 6 (out of 120) segments. Note that the special purpose modules were applied to provide evidence for about 15% of the hypothesized features. The requested locations are shown in Fig. 47. Note how the special purpose modules are used opportunistically and at selected places. For this example, we provide all the requested evidence. However, in a real situation the special purpose modules will provide evidence based on the ranking until the conflicts are resolved.

After the introduction of all the evidence the conflict graph had 32 links forming 11 components and its structure did not change greatly. However, there were some links added and some which were deleted. That the graph did not change much is not a drawback of the algorithm but a consequence of the fact that the special purpose modules did not provide much unexpected information in this particular case. As we shall see, for other images the conflict resolution graph underwent a significant change. The final resolved hypotheses are shown in Fig. 48(a). The overlay of the description on the initial edge image is shown in Fig. 48(b). The final description obtained is very reasonable. The top horizontal bar is extracted as a

\[2^{\text{The actual saving is more than this when considered against blindly applying the computational modules over the whole image.}}\]
Figure 47: Locations where special purpose modules were applied. (a) Corner module locations. (b) Segment analyzer module. (c) Region analyzer module locations.
Figure 48: Final resolved hypotheses for the blocks image. (a) Best set of hypotheses. (b) Best set of hypotheses overlaid on the edge image. (c) Second best set of hypotheses. (d) Second best set of hypotheses overlaid on the edge image.
parallelogram as are the dark areas on top. The top of the dark block in the middle is described as a ribbon. The second best description is shown in Fig. 48(a) and (b). These represent some of the less probable descriptions of the scene and would be considered (as a part a complete vision system) if the best description does not serve the purpose. These descriptions provide a basis for constructing complex scene descriptions.

8.1.2 The laboratory image

The next set of results use an indoor laboratory image. The same edge detector and constant curvature segmentation algorithm was used. The preattentive organizations are shown in Fig. 49. There are 34 cycles, 31 strands, 1 curved ribbon, 43 parallels, and 147 constant curvature segments.

Introducing the preattentive evidence into the automatically structured PIN (the version in Fig. 34) and letting it settle yields the organizations depicted in Fig. 50. As a final count we have 40 parallelograms, 5 ellipses, 3 triangles, 25 circles, 44 quadrilaterals and 7 ribbons. Note that we now have a number of parallelogram hypotheses. The structures like circles and ellipses were not detected in the preattentive phase. The conflict graph of these hypothesized target features has 208 links and 19 components. Out of a total of 342 corners, 215 edge strands, and 52 closed regions hypothesized the special purpose modules were used to verify 25 corners, 35 strands and 27 closed regions, respectively. The total execution time for the special purpose modules was 75 secs. In light of the new evidence, the locational conflict graph was simplified to have 199 links and 41 components, a number larger than
our previous 19 components. The final resolved description of the scene in terms of the target hypotheses are shown in Fig. 51.

### 8.1.3 Room Image

Next we test out algorithm on an indoor image which is a part of an image sequence (courtesy University of Massachusetts), Fig. 52(a). The total number of constant curvature segments of the detected edges is 277 (Fig. 52(b)). The preattentive algorithm found 69 parallel symmetries, 31 closed cycles of four segments and 7 strands of three edge segments as shown in Fig. 52(c)-(f). Based on these preattentive cues the automatically structured PIN (Fig. 34) inferred 16 ellipses, 4 triangles, 72 circles, 77 quadrilaterals, 30 ribbons, and 66 parallelograms as shown in Fig. 53. These hypotheses formed a locational conflict graph of 467 links and 19 components. The special purpose modules were used to find evidence for 25 (out of 571) corners, 54 (out of 364) strands, and 55 (out of 93) hypothesized closed region. Based on the new evidence, the locational conflict graph had 7 links and 258 components and the final best description is shown in Fig. 54. Note the very significant change in the locational conflict graph after the judicious application of the computational resources.

### 8.2 Two Levels of Organization Hierarchy

The above results constructed just one level of the description hierarchy. We did not attempt to construct the next level of description in terms of organizations among the features found in the first level because it was not meaningful. The images were
Figure 49: Organizations detected by the preattentive perceptual organization module. (a) Gray Level Image (b) Edge contours detected (c) Constant curvature segments (d) Parallel Symmetries (e) Closed boundary hypotheses (d) Edge strand hypotheses
Figure 50: Organizations detected using the attentive module on the lab image. (a) Ellipses (b) Triangles (c) Circles (d) Quadrilaterals (e) Ribbons (f) Parallelograms
small compared to the features and there was insignificant amount of second level organization. In this section we show results images where we use the first level features and create a second level of description.

8.2.1 Runway Image

The gray level image is shown in Fig. 55(a) and the constant curvature segments are in Fig. 55(b). The preattentive module of the first level found 70 straight parallel symmetries, 6 curved symmetries, 1 closed ring of 4 edge segments and 2 strands of edges (Fig. 55). Based on these cues, the attentive module (the automatically structured PIN) inferred 21 circles, 49 quadrilaterals, 7 ribbons, 53 parallelograms, 1 triangle and no ellipses. The major features are shown in Fig. 56. The special purpose modules were used for about 55 secs to verify 42 (out of 341) corners, 46 (out of 217) edge strands, and 45 (out of 55) closed regions. Based on the new
Figure 52: (a) Gray level image of a room. (b) Segmented edge contours. (c) Curved linear symmetries (lighter) for the edge segments (darker). (d) Straight linear symmetries. The symmetry axes are in lighter shade. (e) Strands of three segments. (f) Closed ring of four segments. The constituting edges are in lighter shade.
Figure 53: Organizations detected using the attentive module on the room image. (a) Ellipses (b) Triangles (c) Circles (d) Quadrilaterals (e) Ribbons (f) Parallelograms
evidence, the number of links in the locational conflict graph increased slightly from 387 to 395 but the number of components increased from 16 to 23. This means that the average locational conflict size decreased; larger conflicts were broken into smaller ones by adding new links and breaking old ones.

All the hypotheses participating in the locational conflict graph (Fig. 57(a)) are used to construct the next level of the hierarchy. There were 53 final parallelograms, 5 ribbons, 47 quadrilaterals, 18 circles and one triangle. As mentioned in Chapter 2, each of the hypotheses are represented by a line and point tokens associated with attribute of the hypotheses. For our experiments we use the symmetry axes of the hypotheses, if they exist, as the line tokens and the centroid of the hypotheses as the point tokens. The line tokens (Fig. 57(b)) are described by the cross sectional width of the symmetry. Note that we do not make distinction between the types
of the hypotheses. We allow features of different types but with roughly the same gross shape to be grouped together. Thus, we can find a linear arrangement of a combination of parallelogram and ellipses. The point tokens (Fig. 57(c)) are described by the area, elongatedness, and convexity of the feature they represents.

The line and the point representatives of the first level of the hierarchy form the input to the preattentive algorithm of the second level. As discussed in Section 4.1 of Chapter IV, the essential nature of the preattentive algorithm of the second level is the same as that of the first. Except for changes needed for point token clustering, we use the same code as that for the first level. The changes are in the interface algorithm to call the various graph theoretic algorithms. Our sets of Gestalt graphs remain the same.

The preattentive output of the second level are shown in Fig. 58. Note how the parallelism of the symmetry axes to the right and to the top of the image are made explicit. Fig. 58(b) shows the axes which are continuous and Fig. 58(c) shows the clusters of objects based on proximity, similarity in size, elongatedness and convexity. The centroids of similar hypotheses are joined by a straight line.
Figure 55: (a) Gray level image of a runway. (b) Segmented edge contours. (c) Curved linear symmetries. (d) Straight linear symmetries. The symmetry axes are in lighter shade. (e) Strands of three segments. (f) Closed ring of four segments. The constituting edges are in lighter shade.
Figure 56: Organizations detected using the **attentive** module on the runway image.
(a) Circles (b) Quadrilaterals (c) Ribbons (d) Parallelograms
Figure 57: Strong hypotheses of the first level which form the input to the preattentive algorithm of the second level. (a) The hypotheses (b) Line abstraction of the hypotheses (Symmetry Axes) (c) Point tokens (Centroid).
Figure 58: Preattentive cues of the second level of the description hierarchy. (a) Second Level parallel symmetries (b) Continuous parallel symmetries (c) Clusters of the centroids of the features at the first level.
The pre-attentive cues of the second hierarchical level can then be used as evidence to a PIN appropriate for the second level of description. The structure of the PIN will depend on the target hypotheses constituting our second level of description. We have not implemented rest of the second level construction algorithm; conceptually, it involves the same processes as those in the first level. Future work might involve defining higher levels of the description hierarchy and structuring appropriate PINs using the algorithm of Chapter VI. However, we did demonstrate our claim that our perceptual organizational framework can be used to construct multiple levels of description by "closing the loop" of the algorithm.

8.2.2 Aerial Stereopair

In this section we present results on large 482 \times 482 aerial stereopairs. The left image is shown in Fig. 59 and the right is in Fig. 60. Because of its big size we have a large number of edge tokens to work with at the first level and we run into memory shortage in the PIN. So, to construct the first level of the description hierarchy we break up each image into four overlapping 250 \times 250 parts. Each part is processed independently by the preattentive and attentive PIN of the first level and all the results are grouped together and sent to the second level to find preattentive organizational cues.

Left Image

The preattentive organizations of the first level are shown in Fig. 61-64. The number of features in each parts are also indicated in the corresponding figures. These
Figure 59: Aerial Image (Left) used (482 × 482)
Figure 60: Aerial Image (Right) used (482 x 482)
preattentive organizations are independently processed by the attentive module: PIN and the special purpose algorithms. The resulting organizations of the four parts are shown in Fig. 65. Note how the highway structure in the middle of the image is approximated by parallelograms and the other roads are represented by curved ribbons. Some of the alleys also show up a parallelograms. The hypotheses participating the locational conflict graph form Level 1 of the description hierarchy.

All the linear and point representations of the first level features are sent to the second level preattentive module to look for more complex organizational cues. We do not divide the image for this level. The linear and the point representations (Fig. 66) are the symmetry axes and the centroid of the features, respectively. The preattentive organizations detected among the total of 388 point tokens and 324 line tokens are shown in Fig. 67. The highly organized road structure in the middle of the image shows up as parallel structures. Parts of the road network show up as strands and continuous segments. The points cluster are depicted by straight lines joining the nearby, similar features (Fig. 67(d)).

**Right Image**

The preattentive organizations of the first level for the right aerial image are shown in Fig. 68-71. The number of features in each parts are indicated in the corresponding figures. These preattentive organizations are independently processed by the attentive module: PIN and the special purpose algorithms. The resulting organizations of the four parts are shown in Fig. 72. Note how part of the highway
Figure 61: Left Image (a) First section of the aerial image ($250 \times 250$). (b) 428 constant curvature edge segments. (c) 11 curved linear symmetries. (d) 67 straight linear symmetries. The symmetry axes are in lighter shade. (e) 22 strands of three segments. (f) One closed ring of four segments. The constituting edges are in lighter shade.
Figure 62: Left Image (a) Second section of the aerial image \((242 \times 250)\). (b) 364 constant curvature edge segments. (c) 16 curved linear symmetries. (d) 29 straight linear symmetries. The symmetry axes are in lighter shade. (e) 5 strands of three segments. (f) 5 closed ring of four segments. The constituting edges are in lighter shade.
Figure 63: Left Image (a) Third section of the aerial image (250 x 242). (b) 370 constant curvature edge segments. (c) 10 curved linear symmetries. (d) 48 straight linear symmetries. The symmetry axes are in lighter shade. (e) 16 strands of three segments. (f) 5 closed ring of four segments. The constituting edges are in lighter shade.
Figure 64: Left Image (a) Fourth section of the aerial image (242 × 242). (b) 339 constant curvature edge segments. (c) 14 curved linear symmetries. (d) 27 straight linear symmetries. The symmetry axes are in lighter shade. (e) 13 strands of three segments. (f) One closed ring of four segments. The constituting edges are in lighter shade.
Figure 65: Left Image The first level description of the aerial image sections in terms of ellipses, circles, triangles, quadrilaterals, parallelograms, and ribbons. For (a) First section (250 × 250). (b) Second section (250 × 242) (c) Third section (242 × 250) (d) Fourth section (242 × 242).
Figure 66: Left Image. The (a) 324 linear and (b) 388 point tokens used for the construction of the second level.
Figure 67: The second level preattentive organization for the left image. (a) Parallel symmetries (b) Continuous (symmetry) lines (c) Strands of symmetry axes (d) Clusters of similar features. Each feature is represented as a dot and lines are drawn between similar features. (Note that the images have been reduced for display.)
structure to the left corner of the image is again approximated by parallelograms and the other roads are represented by curved ribbons. As per our algorithm, the hypotheses participating the locational conflict graph form Level 1 of the description hierarchy.

The linear and the point representations at the second level (Fig. 73) are the symmetry axes and the centroid of the first level features, respectively. The preattentive organizations detected among the total of 432 point tokens and 291 line tokens are shown in Fig. 67. As in the left image, the highly organized road structure in the image shows up as parallel structures. Parts of the road network also manifest as strands and continuous parallel structures. The points cluster are depicted by straight lines joining the nearby, similar features (Fig. 67(d)).

Observe the qualitative similarity between the organizations extracted from the left and the right stereo images. From the knowledge of the correspondence between the features of the two images, we can derive depth estimates over the image. The use of high level features for stereo matching tends to make the solution of the correspondence problem robust with respect to noise and occlusion. Chung and Nevatia [45] match two hierarchical descriptions of a stereo scene by propogating match constraints from higher levels to lower levels. Boyer and Kak [51] provide a more comprehensive information theoretic framework for matching two high level structural description of a stereopair. It would be interesting to adapt Boyer and Kak’s framework to use the hierarchcal descriptions created by the perceptual organization algorithm.
Figure 68: Right Image (a) First section of the aerial image (250 x 250). (b) 361 constant curvature edge segments. (c) 13 curved linear symmetries. (d) 62 straight linear symmetries. The symmetry axes are in lighter shade. (e) 13 strands of three segments. (f) 2 closed rings of four segments. The constituting edges are in lighter shade.
Figure 69: Right Image (a) Second section of the aerial image (242 x 250). (b) 409 constant curvature edge segments. (c) 6 curved linear symmetries. (d) 20 straight linear symmetries. The symmetry axes are in lighter shade. (e) 3 strands of three segments. (f) 4 closed ring of four segments. The constituting edges are in lighter shade.
Figure 70: Right Image (a) Third section of the aerial image (250 × 242). (b) 316 constant curvature edge segments. (c) 11 curved linear symmetries. (d) 44 straight linear symmetries. The symmetry axes are in lighter shade. (e) 24 strands of three segments. (f) 2 closed ring of four segments. The constituting edges are in lighter shade.
Figure 71: Right Image (a) Fourth section of the aerial image (242 x 242). (b) 440 constant curvature edge segments. (c) 5 curved linear symmetries. (d) 27 straight linear symmetries. The symmetry axes are in lighter shade. (e) 22 strands of three segments. (f) One closed ring of four segments. The constituting edges are in lighter shade.
Figure 72: The first level description of the aerial image sections in terms of ellipses, circles, triangles, quadrilaterals, parallelograms, and ribbons. For (a) First section (250 x 250). (b) Second section (250 x 242) (c) Third section (242 x 250) (d) Fourth section (242 x 242).
Figure 73: Right Image. The (a) 291 linear and (b) 432 point tokens used for the construction of the second level.
Figure 74: The second level preattentive organizations for the right image. (a) Parallel symmetries (b) Continuous (symmetry) lines (c) Strands of symmetry axes (d) Clusters of similar features. Each feature is represented as a dot and lines are drawn between similar features. (Note that the images have been reduced for display.)
Table 15: Execution times. The timing for Level 1 of the aerial image is the sum of the timings for the four parts of the image. The last column refers to the total execution CPU time including the time taken by the parts not itemized.

<table>
<thead>
<tr>
<th>Image</th>
<th>Sparc</th>
<th>Preattentive</th>
<th>Attentive</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocks (255 x 255)</td>
<td>ELC</td>
<td>27 s</td>
<td>210 s</td>
<td>272 s</td>
</tr>
<tr>
<td>Lab. (250 x 241)</td>
<td>IPX</td>
<td>146 s</td>
<td>74 s</td>
<td>79 s</td>
</tr>
<tr>
<td>Room (255 x 255)</td>
<td>IPX</td>
<td>136 s</td>
<td>150 s</td>
<td>160 s</td>
</tr>
<tr>
<td>Runway (Level 1) (252 x 222) (Level 2)</td>
<td>IPX</td>
<td>95 s</td>
<td>87 s</td>
<td>56 s</td>
</tr>
<tr>
<td>Lake (left, Level 1) (482 x 482) (Level 2)</td>
<td>IPX</td>
<td>386 s</td>
<td>396 s</td>
<td>261 s</td>
</tr>
<tr>
<td>Lake (right, Level 1) (482 x 482) (Level 2)</td>
<td>IPX</td>
<td>59 s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.3 Timing Considerations

Before presenting the timing results we would like to add a disclaimer that our implementation is not optimal and there are ample opportunities for performance improvement by efficient coding. However, to give the reader an idea of the computations involved we provide the performance we have achieved to date for the test images. The timings are depicted in Table 15.

The execution times refer to the CPU time on the Sun-Sparc type mentioned in the second column. We display the times required for each of the major components of our algorithm: the preattentive voting and graph theoretic operations, the perceptual inference network and the special purpose module. The fourth column lists the time needed for the PIN to settle to an equilibrium after the introduction of all
the preattentive evidence. The last column lists the total CPU time of the whole algorithm: from edges to final description. Note that the average execution time for an medium sized image (250 × 250) is of the order of 5-7 mins. For the large aerial stereopairs the total execution time is between 20-25 mins.

Of course, the execution time can be improved significantly in a parallel implementations. But, even on a serial machine there are ample opportunities for improvement by using more efficient graph theoretic algorithms (particularly, a better all path enumeration algorithm) in our preattentive part and using efficient data structures in our PIN so as to break the memory bottleneck. However, the achieved timing are certainly very encouraging.
We presented a theoretical framework to compute perceptual organization in the context of computer vision. It is built around voting methods, graph enumeration, probabilistic Bayesian Networks, and information theoretic resource management. It incorporates a number of concepts from human visual analysis especially the Gestalt laws of organization and the notion of preattentive and attentive vision. We test our algorithm on a set of images from various domains. The methods is also computationally efficient and is potentially highly parallelizable.

Among the system improvements possible is the use of more efficient (or approximate) graph algorithms for preattentive vision. The shortest path enumeration is the computational bottleneck. Another improvement is the design of a comprehensive set of visual modules which can provide information about more complex features instead of the very simple ones we have used. Work in the concept of visual routines [133] might be useful in this regard.

Possibilities for future work are numerous. Although we presented results on 2D structural level perceptual organization (see Chapter II for nomenclature), the formalism has potential of being extended to other forms of perceptual organization as suggested by the classificatory structure of perceptual organization work of
Chapter II. In the classification table (Fig. 3) we see that the blocks towards the upper right corner are "empty" as opposed to those to the lower left. More work needs to be done in these "empty" areas. Research in range image segmentation has been mainly in the area of signal level and primitive level organization. Although there is some work in structural and assembly level organization, more needs to be done. The structural level will involve finding useful surface combinations like parallel surfaces, vertices, and convex junctions. The assembly level will organize these surface clusters into meaningful parts based on the Gestalt principles.

Research on perceptual organization in motion sequences has been minimal. Except for some work in signal level organization, work at the primitive, structural, or assembly level is lacking. Extracting organizations in the optic flow field will help in fast navigation without explicitly solving for the 3D structure of the world. An organization which persists in time is very significant and suggests a common cause. Primitive level organization could include 2D patterns in the flow field like swirls, vortices, sinks, and sources. The structural level would involve finding organizations among these. This would help in segmenting the field into regions of interest. We speculate that the assembly level will involve organizing the regions of interest into clusters which would roughly correspond to objects (consistent dynamics).

We could not find work related to extracting organizations in a 3D world over time. Organizations in dynamic 3D scenes are the most significant because of the dimensionality of the space. Work in active vision and non-rigid motion might benefit from this. As to the nature of future work in this area we can be only highly
speculative. Signal level work might involve extracting 3D optic flow vectors. The primitive level will organize them into stream lines or co-parametric surfaces. At the structural level we can look for surfaces which exhibit regularity in 3D shape and motion. Assembly level organization might group objects exhibiting similar motion.

Besides the extension of the formalism to other domains of perceptual organization, the PIN itself might be useful for higher level visual tasks. The PIN which is an efficient and compact probabilistic knowledge representation scheme might prove useful for complex objects. The PIN structuring algorithm has potential of being used to construct a network for higher level objects, thus, bringing the possibility of parallelization to high level vision.
Appendix A

Tutorial for Bayesian Networks

"Probable impossibilities are to be preferred to improbable possibilities."

Aristotle

Bayesian networks, also known as belief networks, influence networks, or causal networks, are directed acyclic graphs with nodes representing propositions (or random variables) and arcs signifying direct dependencies as quantified by conditional probabilities. The generic knowledge of a domain expert can be easily represented in a network by storing factual knowledge in its topology and facilitating knowledge manipulation by communication along the arcs. Networks have the desirable property that the structure of dependency among variables is made explicit. In the following section we first look into the motivation for choosing Bayesian networks. The representation of three place dependencies is discussed next, followed by fusing and propagation issues in singly connected networks.

A.0.1 Bayesian network dependencies

Fundamental to the construction of a Bayesian network is the concept of representing dependencies using its topology. Consider the random variables $X_a, X_b, X_c$ where
$X_a$ is connected to $X_c$ via $X_b$. The two links can have the following combination of directions:

1. tail to tail, $X_a \leftarrow X_b \rightarrow X_c$: $X_a$ and $X_c$ are conditionally independent given $X_b$, i.e. $P(x_a, x_c|x_b) = P(x_a|x_b)P(x_c|x_b)$.

2. head to tail, $X_a \rightarrow X_b \rightarrow X_c$ or $X_a \leftarrow X_b \leftarrow X_c$: also denotes $P(x_a, x_c|x_b) = P(x_a|x_b)P(x_c|x_b)$.

3. head to head, $X_a \rightarrow X_b \leftarrow X_c$: $X_a$ and $X_c$ are marginally independent, i.e., $P(x_c|x_a) = P(x_c)$ and $P(x_a|x_c) = P(x_a)$, but may become dependent given the value of $X_b$. Moreover, if $X_b$ has descendants then $X_a$ and $X_c$ will also be dependent if any one of the descendant variables is instantiated.

To map conditional independencies among variables to the separation of nodes in the DAG (directed acyclic graph) we consider the following definition of separability based on the above discussion.

**Definition 1:** Two links are said to be blocked by a subset of variables $S_e$ if the two links meet head to tail or tail to tail at node $X$ and $X$ is in $S_e$, or if the two links meet head to head at $X$ and neither $X$ nor any of its descendants is in $S_e$.

**Definition 2:** A path $P$ is said to be separated by a subset of variables $S_e$ if at least one pair of successive links along $P$ is blocked by $S_e$.

**Definition 3:** A subset of variables $S_e$ is said to separate $X_i$ from $X_j$ if all paths between $X_i$ and $X_j$ are blocked by $S_e$.

---

1 $P(z_j|x_i)$ actually denotes $P(X_j = z_j|X_i = x_i)$, the conditional probability of the event $\{X_j = z_j\}$ given the event $\{X_i = x_i\}$.
Definition 4: A screening neighborhood, $S$, for any given node is a set of variables that renders the given variable independent of every variable not in $S$. For a Bayesian network this consists of direct parents, direct successors and all direct parents of the latter.

A.0.2 Belief updating in networks

After representing knowledge using a Bayesian net, we can consult the net to reason about, or interpret, the input data. The interpretation involves instantiating the nodes representing the input data and then computing the change in probability density functions of the random variables represented by the nodes, because of the new evidence. One way to achieve this is the use of an external interpreter taking advantage of the network topology to schedule its computational facilities. An elegant solution, however, involves reasoning at the knowledge level itself. The nodes are each associated with a small amount of computational power and storage space and the graphical links are also the communication links between the processors representing various variables. The communication links are open continuously. On the arrival of evidence the corresponding processor is activated, local probabilities are updated, messages are passed to its parents and children for them to update their own probabilities, and so on. A multidirectional propagation is setup which stops when equilibrium is reached. The stability of the local propagation process can be guaranteed by proper bookkeeping. The distributed message passing also

---

2The messages are passed among the nodes, irrespective of the graphical link direction. The graphical link direction just determines the type of message. We will say more on message passing later.
enables one to give semantic interpretation to the individual steps. The parallel propagation scheme can be very easily implemented in serial machines using blackboard architectures, or in object oriented languages. Another advantage of a network representation is that the stored knowledge will be completely separated from the control mechanism, which may be a predetermined schedule, or entirely random.

The following conventions are adopted:

- The set of possible values taken by a node, say $X$, representing a random variable, is denoted by its lower case, $x$. Each directed link, $X \rightarrow Y_j$ is quantified by a fixed conditional probability matrix $M_{Y_j|X} = [P(y_j|x)]$, with rows corresponding to the various values $X$ can take and the columns corresponding to the values of $Y_j$.

- Incoming information is of two types: specific and virtual. Specific evidence corresponds to direct observations which validate with certainty the values of some variables in the network. Virtual evidence corresponds to probabilistic judgments based on undisclosed observations which affect the belief in some variables in the network. These undisclosed observations are represented by dummy nodes, $Z$, connected by (dummy) links to the variable(s), say $X$, affected by them. The dummy links are characterized by the conditional probability $P(Z = z|X = x)$ and carry information only from the evidence to the variables and not vice-versa. All incoming evidence will be denoted by $e$ and will be regarded as coming from a set $E$ of instantiated variables, i.e. variables whose values are known.
- The updated node probabilities, or dynamic probabilities, will be denoted by $BEL(x) = P(x|e)^3$, representing the overall belief accorded to proposition $\{X = x\}$ by the evidence, $e$, received so far. The functions $BEL(x), P(x), \lambda(x), \pi(x)$ ($\lambda$ and $\pi$ are defined later) are functions of $x$ if $X$ is a continuous random variable, or a vector (list) if $X$ is a discrete random variable, with each vector component corresponding to a different value of $X$. The product $f(x)g(x)$ of two such vectors will stand for term-by-term multiplication.

- $\alpha$ will denote a normalizing constant.

Before considering a general tree, let us analyze the simplest tree consisting of two nodes and a link, $X \rightarrow Y$. If evidence $e = \{Y = y\}$ is observed, then from Bayes' Rule, the belief distribution of $X$ is given by

$$BEL(x) = P(x|e) = \alpha P(x)\lambda(x)$$  \hspace{1cm} (A.1)

where $\alpha = [P(e)]^{-1}$, $P(x)$ is the prior probability of $X$, and $\lambda(x)$ is the likelihood vector

$$\lambda(x) = P(e|x) = P(Y = y|x)$$  \hspace{1cm} (A.2)

So $\lambda(x)$ is one column of the link matrix $M_{Y|X}$ stored at node $Y$. It can be computed at $Y$ and transmitted as a message to $X$, enabling $X$ to compute its belief distribution, $BEL(x)$.

3The is different from the Bel function of DS theory.
Figure 75: A segment of a tree
Suppose now that $Y$ is not observed directly, but is supported by an indirect observation $e = \{Z = z\}$ of some descendent, $Z$, of $Y$. Then we have the chain $X \rightarrow Y \rightarrow Z$, and we can still write

$$\text{BEL}(x) = P(x|e) = \alpha P(x)\lambda(x) \quad (A.3)$$

Where the likelihood vector, $\lambda(x)$, is now calculated as follows

$$\lambda(x) = P(e|x) = \sum_{y} P(e|y, x)P(y|x) = \sum_{y} P(e|y)P(y|x) \quad (A.4)$$

using the fact that $Y$ separates $X$ from $Z$. Thus we see that $\lambda(x)$ can be calculated at $Y$ using the conditional matrix $M_{Y|X}$ and the likelihood message, $P(e|y)$, sent by $Z$. Using $\lambda(x)$, $X$ updates its own belief. Thus belief updating can be accomplished by local propagation mechanisms.

Now let us generalize the above results by considering a tree fragment as shown in Fig. 75. We now outline the message propagation scheme in singly connected networks, or causal polytrees, where a node $X$ has $n$ parents, $U_1, U_2, \ldots, U_n$, and $m$ children, $Y_1, Y_2, \ldots, Y_m$. Each node may represent a multivalued hypothesis. We focus on node $X$ and calculate the messages it receives and sends. Let $e_X^-$ be the evidence contained in the tree rooted at $X$ and $e_X^+$ be the evidence contained in the subgraph connected to $X$ through its parents. $e_X^-$ can be further decomposed

---

4 A graph with a maximum of one link between each pair of node is called a singly connected graph.

5 Causal Polytree: A singly connected tree with multiple parents which has been configured using causality between random variables.
into $\bigcup e_{XY}^-$, corresponding to the evidence in the subgraph on the head of the link $X \rightarrow Y_j$. $e_X^+$ consists of $\bigcup e_{i,X}^+$; the evidence in the part of the subgraph defined by the links $U_iX$, as shown in the Fig. 75. We wish to find the belief at $X$ induced by the current evidence, $e = e_X^- \cup e_X^+$. Using Bayes rule we have:

$$BEL(x) = \frac{P(x|e_X^-, e_X^+)}{P(e_X^-\cdot e_X^+)}$$

Using Bayes Rule

$$= \frac{P(e_X^-|x, e_X^+)P(x|e_X^+)}{P(e_X^-\cdot e_X^+)} \frac{P(e_X^+)}{P(e_X^-\cdot e_X^+)}$$

Since $x$ separates $e_X^+$ and $e_X^-$

$$= \alpha P(e_X^-|x)P(x|e_X^+$$

where $\alpha = \frac{P(e_X^+)}{P(e_X^-\cdot e_X^+)}$ is a normalizing constant. It is in some sense a product of predictive and explanatory factors of the node $X$. Let $\pi(x)$ denote the causal or anticipatory support attributed to $\{X = x\}$ by the evidence in the ancestors of $X$, $P(x|e_X^+)$, and $\lambda(x)$ represent the diagnostic or retrospective support $\{X = x\}$ receives from $X$'s descendants, $P(e_X^-|x)$. The total belief in $\{X = x\}$ is the product of these two factors, which we will write as:

$$BEL(x) = \alpha \lambda(x) \pi(x)$$

Now let us see how $\lambda(x)$ is calculated at $X$ from the messages it receives from its descendants. Evidence $e_X^-$ can be partitioned into disjoint subsets, $e_X^{-Y_1}, e_X^{-Y_2}, \ldots, e_X^{-Y_m}$.

---

*For notational purposes: $e_{AB}^+$ is the evidence in the part of the network connected to $A$ the tail of the link $A \rightarrow B$, and $e_{AB}^-$ is the evidence in the network rooted at $B$.

*We will use $\alpha$ to denote any normalizing constant we need.*
one for each subtree emanating from (the \( m \) children of) \( X \). Since \( X \) separates these subtrees, conditional independence holds. Thus

\[
\lambda(x) = P(e_X^- | x) = \prod_k P(e_{XY}^- | x) \tag{A.7}
\]

So \( \lambda(x) \) is calculated by multiplying the messages, \( \lambda_Y(x) = P(e_{XY}^- | x) \) \( \forall k \), that each of the children send \( X \), their parent. \( \lambda_Y(x) \) is the message from node \( Y \) to node \( x \) concerning its possible value \( x \).

\( X \) then sends similar messages, \( \lambda_X(u_i) = P(e_{U_i,X}^- | u_i) \) \( \forall i \), to each of its parents. Note that \( e_{U_i,X}^- \) refers to the evidence in the part of the network at the head of the link \( U_i \rightarrow X \) analogous to \( e_{X Y}^- \). The message \( X \) sends to its parents is:

\[
\lambda_X(u_i) = P(e_{U_i,X}^- | u_i) = P(\bigcup_{k \neq i} e_{U_k,X}^+, e_{X}^-, x | u_i) \tag{A.8}
\]

where evidence \( e_{U_i,X}^- \) can be decomposed into evidence in the rest of the parents of \( X \), the total evidence in the subgraph induced by the children of \( X \), and the value of \( X \) itself. Conditioning on the rest of the parents, considering separability factors, and using Bayes rule we have:
\[ \lambda_X(u_i) = P(\bigcup_{k \neq i} e_{uk}^+, e_{X}^- | x, u_i) \]

Conditioning on \( X, U_1, \ldots, U_{i-1}, U_{i+1}, \ldots, U_n \), we have
\[
= \sum_x \sum_{u_k : k \neq i} P(\bigcup_{k \neq i} e_{uk}^+, e_{X}^- | u_i, x, u_1, \ldots, u_{i-1}, u_{i+1}, U_n)P(x, u_1, \ldots, u_{i-1}, u_{i+1}, U_n | u_i)
\]

Using conditional independence, we get
\[
= \sum_x \sum_{u_k : k \neq i} P(e_{X}^- | x)P(\bigcup_{k \neq i} e_{uk}^+ | u_1, \ldots, u_{i-1}, u_{i+1}, U_n)P(x, u_1, \ldots, u_{i-1}, u_{i+1}, U_n | u_i)
\]

By Bayes rule
\[
= \alpha \sum_x \sum_{u_k : k \neq i} P(e_{X}^- | x)P(u_1, \ldots, u_{i-1}, u_{i+1}, U_n | \bigcup_{k \neq i} e_{uk}^+)P(x | u_1, \ldots, U_n)
\]

Using conditional independence, again
\[
= \alpha \sum_x P(e_{X}^- | x) \sum_{u_k : k \neq i} \left( P(x | u_1, \ldots, U_n) \prod_{k \neq i} P(u_k | e_{uk}^+) \right)
\]
\[
= \alpha \sum_x \lambda(x) \sum_{u_k : k \neq i} \left( P(x | u_1, \ldots, U_n) \prod_{k \neq i} \pi_X(u_k) \right)
\]

\[ (A.9) \]

\( P(x | u_1, \ldots, U_n) \) quantifies the links \(^8\) from the parents \( U_i \) to \( X \). Thus the message going to the parents \( U_i \), can be calculated from the messages from \( X \)'s parents, \( \pi_X(u_i) \), and the stored conditional probability matrix. The next component needed to calculate \( BEL(x) \) (Eq. A.6) is \( \pi(x) \) given by:

\(^8\)In a sense, Bayesian networks are hypergraphs because to describe the dependency of a given node on its \( m \) parents requires a function, \( P(x | u_1, \ldots, u_n) \), of \( m + 1 \) arguments which, in general, could not be specified by \( m \) two-place functions on the individual links.
\[ \pi(x) = P(x|e^+_X) \]

Conditioning on the parents and summing
\[ = \sum_{u_1, \ldots, u_n} P(x|u_1, \ldots, u_n, e^+_X)P(u_1, \ldots, u_n|e^+_X) \]

Expanding \( e^+_X \)
\[ = \sum_{u_1, \ldots, u_n} P(x|u_1, \ldots, u_n)P(u_1, \ldots, u_n|e^+_{U_1}X, \ldots, e^+_{U_n}X) \quad (A.10) \]

Again, using conditional independence
\[ = \sum_{u_1, \ldots, u_n} P(x|u_1, \ldots, u_n)P(u_1|e^+_{U_1}X) \cdots P(u_n|e^+_{U_n}X) \]
\[ = \sum_{u_1, \ldots, u_n} P(x|u_1, \ldots, u_n) \prod_{i=1} \pi_X(u_i) \]

where \( P(u_i|e^+_{U_i}X) \) is the message, \( \pi_X(u_i) \), sent by \( U_i \) to \( X \). \( X \) should send similar messages to its children, the \( Y_j \)'s, given by \( \pi_{Y_j}(x) \ \forall j \).

\[ \pi_{Y_j}(x) = P(x|e^+_{X_{Y_j}}) \]
\[ = P(x|e \cap (e^-_{X_{Y_j}})^c) \]
\[ = BEL(x) \text{ when the evidence } e^-_{X_{Y_j}} \text{ is suppressed} \quad (A.11) \]
\[ = \alpha^{BEL(x)}_{\lambda_{Y_j}(x)} \]
REFERENCES


276


