

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

A Representation for Visual Information

James L. Crowley
November 1981

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Copyright © 1982 J. L. Crowley, C-MU Robotics Institute

This research was partially supported by:
National Science Foundation Grant No. APR75 - 08154
and by
Naval Electronics System Command (NELC) Grant No. N00039 - 79 - Z - 0169

SEP 3 1982

6.2 Cascaded Convolution with Expansion and Resampling	71
6.2.1 Cascaded Filtering and the $\sqrt{2}$ Expansion Operation	72
6.2.2 Cascaded Convolution with Expansion	72
6.2.3 $\sqrt{2}$ Expansion and Resampling	73
6.2.4 Frequency Domain Effects of $\sqrt{2}$ Expansion	74
6.3 The Sampled DOG Transform	77
6.3.1 Construction of a Sampled DOG Transform	79
6.3.2 Computational Complexity and Memory Requirements	82
6.3.3 Comparison of Complexity with Filtering Using FFT	82
6.3.4 The Size of Cascaded Filter Impulse Response	84
6.4 Verification of Scaling Approximation	85
6.4.1 Diagonal Method in Space Domain:	85
6.4.2 Diagonal Method in Frequency Domain:	85
6.4.3 Expansion Method:	86
6.5 The Band-Pass Filters	87
6.5.1 Size of Positive Center Radius	87
6.5.2 Relative Size of Filters and Their Transfer Functions	88
6.5.3 Filter at Band-Pass Level 0	88
6.5.4 Filter at Band-Pass Level 1	89
6.5.5 Filter at Band-Pass Level 2	94
7. A Symbolic Representation Based on the Sampled Difference of Gaussian Transform	96
7.0.1 Information Stored at Each Node	97
7.0.2 Meaning and Purpose of Peaks and Ridges	98
7.1 Phenomena in Each Band-Pass Image	99
7.1.1 The SDOG Band-Pass Impulse Response	99
7.1.2 Edges of Large Regions	100
7.1.3 Convex Protrusions: The Corner	101
7.1.4 Across a Long Thin Rectangle	102
7.1.5 At the Ends of the Rectangle	103
7.1.6 A Square Which is Smaller Than the Filter	104
7.2 Peak and Ridge Path Detection at Each Band-Pass Level	105
7.2.1 Detecting Local Peaks	105
7.2.2 Detecting Ridge Paths at a Band-Pass Level	106
7.2.3 Eliminating Small Loops	107
7.2.4 Unterminated Ridge Paths	110
7.2.5 Repairing Broken Paths	112
7.2.6 Isolated Pairs	112
7.2.7 Detecting Spurs	113
7.3 Phenomena Between Levels in the Transform Space	114
7.3.1 Connectivity of Peaks: M-Paths	115
7.3.1.1 "Spots" or Non-Elongated Forms	115
7.3.1.2 "Bar-end": The Ends of an Elongated Form	117
7.3.1.3 "Corners" and Other Protrusions	118
7.3.2 3-D Ridges: L Paths	119
7.3.2.1 Ridge Paths for a Uniform Bar	120
7.3.2.2 Bars of Changing Width	120
7.3.2.3 Edges of Regions	120

7.3.3 Connectivity of L-Paths and M-Paths	123
7.4 Connecting Peaks Between Levels	123
7.4.1 Linking M's	123
7.4.2 Detecting M*'s	124
7.4.3 Example	124
7.5 Detecting Ridge Nodes in (x, y, k) Space	124
7.5.1 Problems and Approach	125
7.5.2 Search Paths	126
7.5.3 The Modified Flag Stealing Process	128
7.5.3.1 Modified Flag Stealing	128
7.5.3.2 Resolving the T1 and T2 Flags	129
7.5.3.3 Linking L-nodes	129
7.6 Examples	129
8. Matching the Representation	149
8.0.1 Applications of Correspondence Matching	150
8.1 A Matching Procedure for Descriptions of Similar Grey Scale Forms	151
8.2 Test Data	152
8.2.1 Example of Band-Pass Images of Teapot	153
8.3 Matching M-Paths	164
8.3.1 Abstracting M-Paths from the Representation	164
8.3.1.1 Strongly Connected M-Nodes	164
8.3.1.2 Weakly Connected M-Nodes	165
8.3.1.3 Example of Abstracted M-nodes and P-Paths	165
8.4 Examples of M-node Correspondence	167
8.4.1 M-nodes for Teapot Image #1	168
8.4.2 Initial Alignment to Obtain Size and Position	170
8.4.3 Determining Further Correspondence and Orientation	172
8.4.4 Correspondence of M-nodes Under Rotation	173
8.4.5 Examples of Size Change Less than $\sqrt{2}$	175
8.4.6 Summary of Teapot Matching Examples	178
8.4.7 Stereo Matching Example	181
8.5 Matching L-Paths	189
8.5.1 Two stages of Matching	189
8.5.2 L-Path Alignment	189
8.5.2.1 L-Path Notation and Attributes	189
8.5.2.2 Alignment Parameters:	190
8.5.2.3 Alignment Function:	191
8.5.3 Similarity Measure	191
8.5.4 Examples of L-path Alignment and Matching	193
8.5.5 Summary of L-path Matching Examples	200
9. Discussion	205
9.1 Applications of the DOLP Transform	205
9.1.1 Detecting Ill-defined Pulses	205
9.1.2 Detecting Pulses of Different Durations	206
9.1.3 When Width is not known A-Priori	206
9.1.4 Automatic Focus	206
9.2 Evaluating Claims	206
9.2.1 Claims Concerning the Representation for Shape	207

List of Figures

Figure 1-1: Test Image of a Cup. Note Shape of Dark Regions.	5
Figure 1-2: A Rhomboidal Form and its Representation (Reproduced from Chapter 7, figure 7-19)	13
Figure 2-1: Masks Used in Roberts' Gradient	20
Figure 2-2: Masks Used in Sobel Operator	20
Figure 2-3: Two Discrete Approximations To the Laplacian from [Prewitt 70]	21
Figure 3-1: Example of $S_{\sqrt{2}}[p(x,y)]$ and $S_2[p(x,y)]$	33
Figure 3-2: Nyquist Boundaries for Successive Application of $\sqrt{2}$ Sampling	33
Figure 3-3: Transfer Function Constraints for a Low-Pass Filter	34
Figure 4-1: The Only Possible Symmetric 1-D Function with Two Alternations	41
Figure 4-2: Two Possible Symmetric 1-D functions with 3 Alternations	41
Figure 4-3: A Symmetric 1-D Band-Pass Function with 4 Alternations	42
Figure 4-4: A Symmetric 1-D Band-Pass Function with 5 Alternations	42
Figure 4-5: Monotonic Pass Band with Single Peak	44
Figure 4-6: The Set of Naturally Occurring Sample Distances For a Cartesian Plane	45
Figure 5-1: Direct Method for Computing a DOLP Transform	49
Figure 5-2: Difference Method for Computing a DOLP Transform	50
Figure 5-3: Permissible Alternations in Low-pass Filter	57
Figure 5-4: Permissible Alternations in Band-pass Filter	57
Figure 5-5: Transfer Function $G(\omega)$	58
Figure 5-6: Difference of Low-Pass Transfer Functions	59
Figure 5-7: Example of $S_{\sqrt{2}}[p(x,y)]$ and $S_2[p(x,y)]$ From Figure 3-1 of Chapter 3	60
Figure 5-8: Location of Peak Sample as Signal Moves to the Right	63
Figure 5-9: Uncertainty of Position of Peaks at Adjacent Levels	63
Figure 6-1: Normalized Impulse Response $g_o(x,y)$ for $R=4$, $\alpha=4.0$	70
Figure 6-2: Transfer Function $G_o(u,v)$ for $R=4$, $\alpha=4$	70
Figure 6-3: Example of mapping given by $E_{\sqrt{2}}[\cdot]$	74
Figure 6-4: Effect on Transfer Function of $E_{\sqrt{2}}$ Expansion Operator	75
Figure 6-5: Filter $G_o(u,v)$ for $R = 4.0$, $\alpha = 4.0$ Before $\sqrt{2}$ Expansion	76
Figure 6-6: Filter $G_o(u,v)$ After $\sqrt{2}$ Expansion	76
Figure 6-7: Filter $G_2(u,v)$ for $R = 4.0$, $\alpha = 4.0$ $g_2(x,y) = S_{\sqrt{2}}[g_o(x,y) * g_o(x,y)] * E_{\sqrt{2}}[g_o(x,y)]$	77
Figure 6-8: Plot of $20 \log_{10}[G_2(u,v)]$ Scale (shown at left) spans -120 dB.	78
Figure 6-9: Data Flow Graph for Sampled DOG Transform	80
Figure 6-10: Transfer Function of $E_{\sqrt{2}}\{g_o(x,y)\} - g_o(x,y) * g_o(x,y)$	87
Figure 6-11: Coefficients Along X Axis for Filters from Levels 1 Through 4	89
Figure 6-12: U Axis Of Transfer Functions for Band-Pass Filters from Levels 1 Through 4. $u = 2\pi k/64$	90
Figure 6-13: Filter for High Pass Residue, \mathfrak{B}_o	90

9.2.1.1 Invariance to Size and Rotation	207
10. Summary and Conclusions	209
10.1 Major Results of this Dissertation	209
10.2 Summary of Background Chapters	210
10.3 Measurement, Detection and Mathematical Representation of Non-Periodic Signals	210
10.4 Techniques for Fast Computation of a DOLP Transform: The DOG and Sampled DOG Transforms	211
10.4.0.1 Conclusions Concerning Signal Detection	213
10.5 Transforming the SDOG Transform of an Image into A Symbolic Description	214
10.6 Examples of Matching	216
Appendix A. Selection of Filter Parameters	218

Figure 6-14: Impulse Response of Level 1 Band-Pass Filter	91
Figure 6-15: $B_1(u,v)$, The Transfer Function of the Level 1 band-pass Filter	92
Figure 6-16: $20 \text{Log}_{10}[B_1(u,v)]$, The Transfer Function of the Level 1 Band-Pass Filter Plotted in dB Scale, shown at left in increments of -10 db, spans -40 dB	93
Figure 6-17: $B_2(u,v)$, The Transfer Function of the Level 2 band-pass Filter	94
Figure 6-18: $20 \text{Log}_{10}[B_2(u,v)]$, The Transfer Function of the Level 2 band-pass Filter Plotted in dB Scale, shown at left marks increments of -10 dB to -80 dB	95
Figure 7-1: Impulse Response of Band-Pass Filter	99
Figure 7-2: Response Across Center of a Square	100
Figure 7-3: Response at Corner of a Square	101
Figure 7-4: Response of Filter Across a Rectangle	102
Figure 7-5: Response of Filter Along a Rectangle	103
Figure 7-6: Response of Filter To a Square	104
Figure 7-7: Values at Level 2 of rod.swf	107
Figure 7-8: Pointers From First Stage of Ridge Path Detection Procedure	108
Figure 7-9: Ridge Paths After Stage 2 of Procedure	109
Figure 7-10: Classes of Small Loops	110
Figure 7-11: Ridge Path Containing Small Loop	110
Figure 7-12: Path After Removal of Small Loop	111
Figure 7-13: Example of Broken Ridge Paths Before Extension	113
Figure 7-14: Example of Repaired Ridge Paths After Extension	114
Figure 7-15: Response to an 11 x 11 Square Across Diagonal for Levels 1 Through 7	116
Figure 7-16: Examples of Bar-End M-Paths	118
Figure 7-17: Two Forms that Cause "Corner" M-Paths	119
Figure 7-18: Response to a 5 by 11 Rectangle	121
Figure 7-19: An Elongated Form That Changes Width	122
Figure 7-20: Possible Set of Indirect 2-Way Pointers for M-Paths	124
Figure 7-21: M Paths For Square of Size 11 Pixels	125
Figure 7-22: Two Configurations of Ridge Paths at Adjacent Levels	126
Figure 7-23: Two Possible Upper Neighborhoods	127
Figure 7-24: Upper Search Neighborhoods for Stage 1 and Stage 2	128
Figure 7-25: Piston Rod Image. Sampled at 256 by 256.	130
Figure 7-26a: Top Of Piston Rod at Level 10	132
Figure 7-26b: Top of Piston Rod at Level 9	133
Figure 7-26c: Top of Piston Rod at Level 8	134
Figure 7-26d: Top of Piston Rod at Level 7	135
Figure 7-26e: Top of Piston Rod at Level 6	136
Figure 7-26f: Top of Piston Rod at Level 5	137
Figure 7-26g: Top Left Corner of Piston Rod at Level 4 (Note that Sample Rate is $2\sqrt{L}$)	138
Figure 7-26h: Top Left Comer of Piston Rod at Level 3 (Note that Sample Rate is 2)	139
Figure 7-27a: Middle of Piston Rod at Level 10	141
Figure 7-27b: Middle of Piston Rod at Level 9	142
Figure 7-27c: Middle of Piston Rod at Level 8	143
Figure 7-27i: Middle of Piston Rod at Level 7	144
Figure 7-27e: Middle of Piston Rod ai Level 6	145
Figure 7-27f: Middle of Piston Rod at Level 5	146
Figure 7-27g: Middle of Piston Rod at Level 4	147
Figure 7-27h: Middle of Piston Rod at Level 3	148
Figure 8-1: Teapot #1. Size =1.0, Orientation = 0.0°	154

Figure 8-2: Teapot #2. Size = 1.14, Orientation = 0.0°	155
Figure 8-3: Teapot #3. Size = 1.36, Orientation = 0.0°	156
Figure 8-4: Teapot #4. Size = 1.0, Orientation = -15.0°	157
Figure 8-5: Teapot #5. Size = 1.14, Orientation = -15.0°	158
Figure 8-6: Format for Display of Band-Pass Levels 13 through 0	160
Figure 8-7: Band-Pass Images for Levels 13 Through 0 of Teapot #1	161
Figure 8-8: Format for Display of Zoomed Band-Pass Levels 13 through 5	162
Figure 8-9: Zoomed Band-Pass Images for Levels 13 Through 5 of Teapot #1	163
Figure 8-10: Level 7 from Teapot Image #1	166
Figure 8-11: M-nodes and P-Paths for Level 7 of Teapot #1	167
Figure 8-12: M-nodes and P-Paths for Levels 12 to 6 of Teapot #1	169
Figure 8-13: M-nodes and P-Paths for Levels 12 to 7 of Teapot #3	171
Figure 8-14: M-nodes and P-Paths for Levels 12 to 6 of Teapot #4	174
Figure 8-15: M-nodes and P-Paths for Levels 12 to 6 of Teapot #2	177
Figure 8-16: M-nodes and P-Paths for Levels 12 to 6 of Teapot #5	179
Figure 8-17: Stereo Correspondence of M-nodes for Paper Wads, Levels 13 through 9	183
Figure 8-18: Format for Paper Wad Low-Pass Images	184
Figure 8-19: Left Paper Wad and Low-Pass Images	185
Figure 8-20: Band-Pass images for Left Paper Wad	186
Figure 8-21: Right Paper Wad and Low-Pass Images	187
Figure 8-22: Band-Pass images for Right Paper Wad	188
Figure 8-23: L-path from Teapot #1	194
Figure 8-24: L-path from Teapot #3	195
Figure 8-25: L-path Correspondence: L-Path from Teapot #3 Transformed to Match L-path from Teapot #1	196
Figure 8-26: L-path from Teapot #4	198
Figure 8-27: L-path Correspondence: L-Path from Teapot #4 Transformed to Match L-path from Teapot #1	199
Figure 8-28: L-path from Teapot #2	201
Figure 8-29: L-path from Teapot #5	203

List of Tables

Table 6-1: Comparison of Filter Coefficients	85
Table 6-2: Diagonal Comparison Of Transfer Function Samples	86
Table 6-3: Values Along Line $u = v$ in Transfer Function of $E_{\sqrt{2}}\{g\} - (g * g)$	87
Table 6-4: Radii of Center Lobes As measured by Distance to Furthest Positive Coefficient	88
Table 7-1: Fields of a 64 Bit Node	97
Table 8-1: Size and Orientation of five Teapot Images	153
Table 8-2: P-Path Links for Levels 7 and 6 of Teapot #1	170
Table 8-3: P-Path Links for Levels 8 and 7 of Teapot #3	172
Table 8-4: Comparison of D and θ attributes for Teapots 1 and 3	173
Table 8-5: P-Path Links for Levels 7 and 6 of Teapot #4	175
Table 8-6: Comparison of D and θ attributes for Teapots #1 and #4	175
Table 8-7: P-Path Links for Levels 7 and 6 of Teapot #2	178
Table 8-8: Comparison of D and θ attributes for Teapots #1 and #2	178
Table 8-9: P-Path Links for Levels 7 and 6 of Teapot #5	180
Table 8-10: Comparison of D and θ attributes for Teapots #1 and #5	180
Table 8-11: Comparison of D and θ attributes for Teapots 2 and 5	180
Table 8-12: Correspondence and Distance for Transform of L-path from Teapot #3 to Match L-path from Teapot #1	197
Table 8-13: Correspondence and Distances for Transform of L-path from Teapot #1 to Match Teapot #3	197
Table 8-14: Correspondence of Transformed L-nodes from Teapot #4 to L-nodes from Teapot #1	200
Table 8-15: Correspondence of Transformed L-nodes from Teapot #1 to L-nodes from Teapot #4	200
Table 8-16: Correspondence of L-nodes and Distances for Transform of L-path from Teapot #2 to Match Teapot #1	202
Table 8-17: Transform of L-path from Teapot #1 to Match Teapot #5	204

Acknowledgements

This section gives me an opportunity to express my gratitude to a number of people whose support and encouragement made this research and dissertation possible.

At the top of this list is my thesis advisor, Dr. Alice Parker. Not only did she get me started and provide a supportive but critical sounding board for most of these ideas, but she stood behind me when the going was rough. More than once Dr. Parker fought a tough political battle to keep this research effort alive. She also carefully critiqued my early drafts of this dissertation and provided important ideas on style and organization.

Dr. Frank Quick had a very strong influence on this research in the early years. It was Frank who turned me on to the psycho-physical theories of visual perception which inspired this investigation. Many of the principles discussed in chapter 3 were initially developed in response to Frank's probing questions. It was Frank who challenged me to find a family of detection functions which were reversible, and who first raised the question of whether ridges could be detected without first measuring local directionality.

Dr. Richard Stern also deserves much credit for helping me to shape the ideas presented here. It was Rich Stern who found the obvious answer when I wrestled with the problem: what function can be convolved with itself to yield a scaled copy of itself. Rich spent many hours discussing the signal processing aspects of this dissertation. He also invested much time carefully reading and critiquing early drafts of this document.

Special credit goes to Dr. Raj Reddy, without whose support on two different occasions this work would truly not have been possible. Dr. Reddy deserves praise for shaping a research environment in which it is possible to pursue avant garde research without having to promise a short term practical application. A special thanks also for tolerating ever slipping deadlines and the drain of effort from my other Robotics Institute projects while I was preparing this manuscript.

Dr. Takeo Kanade also deserves some credit for ideas and support. While at C-MU as a visiting scientist on a one-year appointment Takeo took the time to hear my ideas and offer his comments. At an early stage he suggested that I pursue algorithms that could be implemented in parallel. Later, Dr. Kanade returned to direct the vision research at C-MU. In this capacity he provided generous access to facilities which assisted me in producing the images for this manuscript.

Finally, I would like to thank Dr. Howard Wactlar and the staff of the engineering lab who keep the cycles turning. These guys are a corner stone of the Computer Science Dept. and the Robotics Institute.

Abstract

This dissertation presents a new technique for representing digital pictures. The principal benefit of this representation is that it greatly simplifies the problem of finding the correspondence between components in the description of two pictures.

This representation technique is based on a new class of reversible transforms (the Difference of Low Pass or DOLP transform). A DOLP transform separates a signal into a set of band-pass components. The set of band-pass filters used in a DOLP transform are defined by subtracting adjacent members of a sequence of low-pass filters. This sequence of low-pass filters is formed by scaling a low-pass filter in size by an exponential set of scale factors. The result of these subtractions is a set of band-pass filters which are all scaled copies of a smallest band-pass filter.

Several techniques are presented for reducing the complexity of computing a DOLP transform. It is shown that as the each band-pass image can be resampled at a sample rate proportional to the scale of the band-pass image. This is called a Sampled DOLP transform. Resampling reduces the cost of computing a DOLP transform from $O(N^2)$ multiplies¹ to $O(J_0 \log N)$ multiplies and reduces the memory requirements from $O(A^7 \log N)$ storage elements to $^3 N$ storage elements.

A fast algorithm for computing the **DOLP** transform is then presented. This algorithm, called "cascade convolution with expansion" is based on the auto-convolution scaling property of Gaussian functions. Cascaded convolution with expansion also reduces the cost of computing a **DOLP** transform to $O(N \log N)$ multiplies. When combined with resampling, this fast algorithm can compute a Sampled DOLP transform in $3 X_0 N$ multiplies.²

Techniques are then described for constructing a structural description of an image from its Sampled DOLP transform. The symbols in this description are detected by detecting local peaks and ridges in each band-pass image, and among all of the band-pass image. This description has the form of a tree of peaks, with the peaks interconnected by chains of symbols from the ridges. The tree of peaks has a structure which can be matched despite changes in size, orientation, or position of the gray scale shape that is described.

The tree of peaks permits the global shape of a gray-scale form to be matched independently of the

¹ N is the number of sample points in an image or signal

² J_0 is the number of coefficients in the smallest low-pass filter.

high resolution details of the form. Thus it can be used for rapidly searching through a data base of prototype descriptions for potential matches. This representation is very efficient for finding the correspondence of components of forms from two images. In such matching the peaks serve as the tokens for which correspondence is determined. The correspondence of peaks at each band-pass level constrain the possible matches at the next, higher resolution image. This representation can also be used to describe forms which are textured or have blurry boundaries. Examples are presented in which the descriptions of images of the same object are matched despite changes in the size and image plane orientation of the object.

Chapter 1

Introduction

This dissertation describes a representation for visual information. This representation is not specific to a particular visual domain; it can be applied to any problem in which a two dimensional sampled function must be represented with symbols. It is particularly appropriate for images where the picture elements have many values, where the objects represented in the picture have blurred or fuzzy boundaries, or have textured surfaces, and where objects occur at unknown sizes and orientations.

Interpreting an image requires assertions about regions of the image whose sizes may span the range from a few picture elements to the entire image. The representation developed below provides visual primitives which span this range of sizes. The position of these primitives are encoded *as* nodes in a graph, the result is a data structure which is relatively invariant to the actual size, orientation and position of the gray scale form in the image.

1.1 The Problem Context: Machine Vision

This Section describes the general vision problem and how this dissertation relates to it

This thesis addresses the problem of representing two dimensional (2-D) visual information. The visual world in which humans function is a three dimensional (3-D) world. Understanding this 3-D visual world requires representation of the 3-D form of objects. The representation described in this thesis does not, by itself, provide this capability; it is inherently 2-D.

The human visual system receives as raw data a stereo pair of 2-D images. Each of these images must be represented as a 2-D signal and the pair matched against each other to receive 3-D information. The representation described here is well suited for the analysis of stereo pairs. It is also well suited for the interpretation of images from some domains which are inherently two dimensional such as many classes of biomedical images, aerial and satellite photography, and also terrain data (where depth is represented as intensity).

Test data for this research has been acquired from diverse domains. Many of the images were digitized from photographs of 3-D objects, such as the cup image shown as figure 1-1 below. The cup image is placed here to illustrate a point about 2-D images of 3-D objects. Careful viewing of a 2-D image of a 3-D object will usually show that, the light and dark regions in the image do not directly correspond to our ideas of the object's shape.

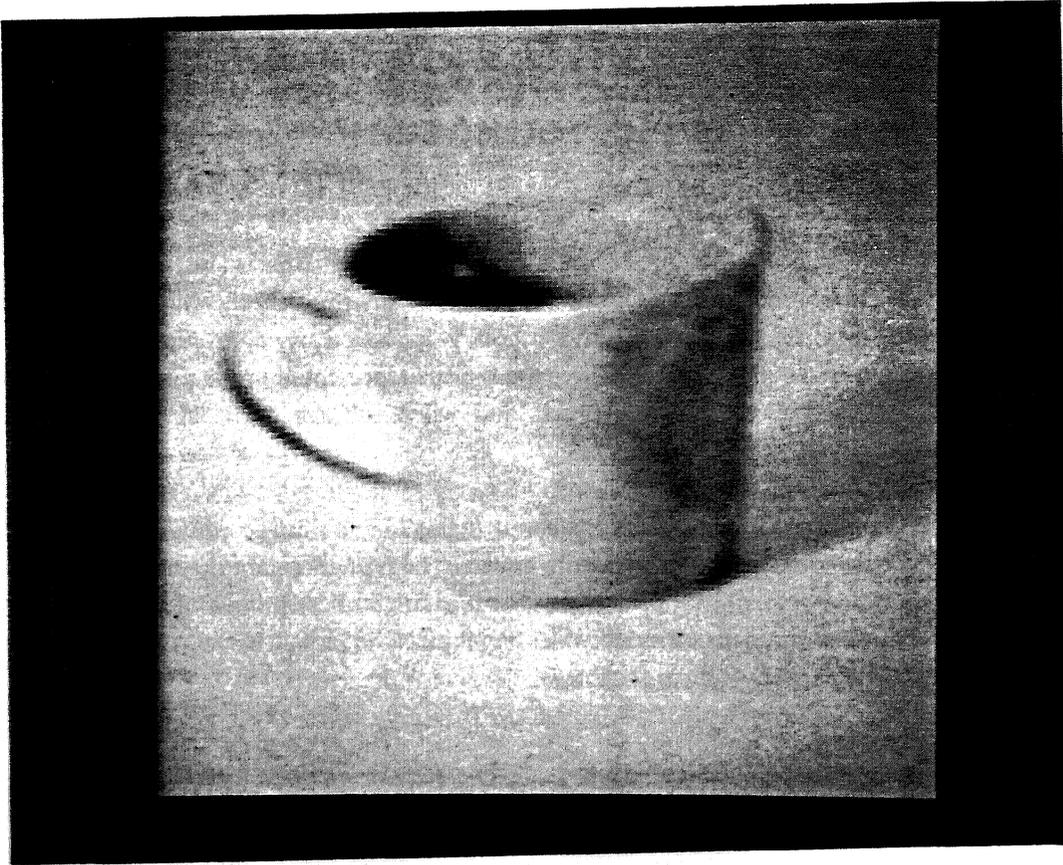


Figure 1-1: Test Image of a Cup. Note Shape of Dark Regions.

Note the shape of the dark regions of the cup. There is a dark handle which one might expect. There is also a dark region at the top where the cup is open, and there is a dark region on the right side. The shape of these regions are not at all like what an untrained person would draw if asked to draw a cup. The human visual system takes the shading, highlights, and textural information, from such an image and uses them to reconstruct or recall a model of a 3-D object. This process is unconscious, and these visual cues are often not noticed by an untrained observer unless they are explicitly looked for. Although interpreting shading, highlights and texture is an important and timely problem in machine vision, it is not the problem addressed by this thesis. Rather, this research will provide a new foundation for such interpretation.

Figure 1-1 also provides an opportunity to define an important term. The dark regions in the cup

image are examples of "gray scale forms". The representation describes the shape of both individual forms and the shape produced by a configuration of forms. The word "form" is borrowed from the art community. It refers to a pattern of any shape which is not necessarily uniform in intensity. It is used in place of image object, because image objects could be confused with real world objects. The words shape and blob were avoided because they carry connotations of uniform-intensity connected patterns.

1.1.1 Role of Representation in 2-D Visual Domains

In a 2-D visual domain, such as aerial photography, many assembly and inspection applications, some classes of biomedical images, or terrain data, recognition of objects requires the following components:

1. A representation technique which compresses the information and expresses it in a useful and efficient form for recognition;
2. A set of object models (or perhaps in the case of terrain data a model of the terrain of a very large region). These models should be expressed in a representation which can be processed efficiently for recognition, or any representation which is easily converted to such a representation.
3. A matching procedure which compares observed data to stored models, gives some measure of similarity, and, if desired, a description of where the observed data matches and does not match a specific object model.

Interpretation is then a matter of encoding the observed data and applying the matching procedure between it and the object models (or regions of the terrain data base). This sounds simple enough, but in fact finding an efficient procedure for such matching can be very difficult. A crucial aspect of the matching problem is finding the correct representation for both the observed data and the object models. The main contribution of this thesis is the development of such a representation.

In statistical pattern recognition, a pattern is represented by a set of measurements called features. The set of features comprise a multi-dimensional space called a "feature space". The features are chosen so that each class of pattern produces a vectors of features that reside in a unique region of the feature space. A pattern is assigned to the class which occupies the region of the feature space into which its vector of feature measurements falls.

Recently there has been interest in a different approach to recognizing 2-D patterns: so called "structural pattern recognition". A structural pattern recognition algorithm employs a proto-type representation for each pattern class. This prototype consists of symbols for certain structural elements, such as edges or corners, which are linked together into a spatial relationship. A pattern is classified by constructing a correspondence between elements of the pattern and elements of the prototypes. A 2-D pattern is assigned the class label for the prototype whose elements most closely correspond to those of the pattern. The representation developed below may be used for structural pattern recognition, although this is not the only application to which it may be applied.

1.1.2 Representation in 3-D Visual Domains

In a 3-D visual world in which input data consists of stereo pairs of 2-D images, interpretation requires the following components:

1. A representation for the 2-D images which may be efficiently used for depth detection by stereo matching.
2. A procedure for obtaining depth information by detecting corresponding objects in the two images and observing their relative shift. This procedure should also make use of information in shading, highlights, texture, and other visual cues.
3. A representation for the 3-D form of objects.
4. A repertoire of models for the 3-D form of objects.
5. A matching procedure to identify which 3-D object model(s) correspond to the observed 3-D input data.

Although this dissertation is primarily concerned with 2-D representation, some suggestions will be made as to how this representation may be used for interpretation of stereo pairs. The other components remain as timely and important research topics.

1.2 Thesis Summary and Background

This Section presents the thesis of this dissertation, describes the methodology for demonstrating this thesis, and reviews the major results of the research.

1.2.1 The Thesis

This research began as an investigation of the use of a set of band-pass spatial frequency channels for representing visual information. This topic was inspired by psycho-physical theories of human visual perception that hypothesize a set of "spatial frequency channels" in the human visual system [Campbell 68]. These theories are summarized in an appendix to [Crowley 76].

Early in this research principles (referred to as postulates) were formed to guide and constrain the design of band-pass filters for representing images. These principles were refined in the course of experiments in which filters were designed and convolved with test patterns. Some of the results from these experiments are described in [Crowley 78a] and [Crowley 78b]. A refined version of these principles is given in Section 4.2 below.

These principles and experiments led to the development of the reversible Difference of Low-Pass (DOLP) Transform. The DOLP transform is based on a set of scaled copies of a circularly symmetric low pass filter. The scale factors for these filters form an exponential sequence. Each low-pass filter is

subtracted from the previous low-pass filter to form an exponential sequence of band-pass filters. These band-pass filters may be convolved with the image to form a set of band-pass images. The set of band-pass images is very similar to the images which would be produced by the set of spatial frequency channels which have been hypothesized to exist in the human visual system.

The set of band-pass filters and the largest low-pass filter sum to form a single coefficient whose value is 1. Another way to say this is that the sum of all of the band-pass images and the low-pass image produced by filtering with the largest low-pass filter can be added together to form the original image. This property demonstrates that no information is lost by the DOLP transform.

The low-pass filters are each a scaled (in size) copy of the same function. Thus the band-pass filters formed from their difference are also scaled (in size) copies of the same function. This gives the property that scaling a 2-D pattern shifts the pattern in each band-pass image to a new band-pass image. Thus a representation based on peaks and ridges in the band-pass images is invariant to changes of scale of the pattern. The scale information is preserved by noting which band-pass image the peaks and ridges actually exist at. It is the network of symbols which is not changed by scaling the 2-D image. Note that in fact there are small cyclic distortions that occur during scaling, but these can be obviated during matching.

A straightforward implementation of a DOLP transform for an N point signal requires $O(N^2)$ multiplies and produces $O(N \log(N))$ samples. This can be quite expensive on a general purpose computer. In an effort to reduce this complexity the concept of re-sampling each band-pass image was investigated. Re-sampling at a rate proportional to the scale of the band-pass filter provides the benefits of:

- making the representation size invariant,
- reducing the computational complexity, and
- reducing the storage requirements

for the DOLP transform. Re-sampling creates a class of DOLP transforms referred to as "the Sampled DOLP transform". The re-sampling operation is described in Section 3.3 and the re-sampled DOLP transform is defined in Section 5.5.

Seeking to further reduce the computational complexity of the DOLP transform we investigated the use of repeatedly convolving an image with a Gaussian low-pass filter and re-sampling. This algorithm, referred to as cascaded filtering with sampling, produces a set of low-pass images with impulse responses which are scaled in standard deviation by a factor of $\sqrt{2}$ for each convolution. Subtracting each low-pass image from the previous low-pass image gives a set of band-pass images.

Cascaded convolution with Gaussian filters can produce a set of low-pass images whose impulse responses are scaled exponentially in standard deviation. This is a consequence of the Gaussian Scaling property, discussed in Section 6.1. The Gaussian scaling property shows that convolving a Gaussian function with itself produces a new Gaussian function which is larger in standard deviation by a factor of $\sqrt{2}$. Cascaded Convolution with sampling using a Gaussian filter may be used to

compute a subclass of the Sampled DOLP transform called the "Sampled Difference of Gaussian" (SDOG) Transform. Storage efficiency and size invariance result from re-sampling, while the computational efficiency is the result of both re-sampling and an auto-convolution scaling property of Gaussian functions.

Both the DOLP transform and the SDOG transform expand a 2-D (x,y) image into a 3-D discrete space (x,y,k). The new dimension of this space is k, the filter index. For an N point image, the SDOG transform has $3N^2$ samples and requires $3N X_0$ multiplies, where X_0 is the number of coefficients in the smallest low-pass filter. This computational complexity, derived in Section 6.3, is less than that of an *FFT* for most signals.

Because the filters implemented by the SDOG transform satisfy the criteria established in Chapter 4 it is possible to construct a structural representation of an image which has certain desirable properties for matching object descriptions. This representation is created by detecting peaks and ridges in the (x,y,k) space given by the SDOG transform.

Let us elaborate on the terms "peak" and "ridge" and on the role of peaks and ridges in this structural representation. At each band-pass image, or level, of the SDOG Transform, there are points where the band-pass impulse response is a "best match" to one of the gray scale forms in the picture. At these points, the filtered picture has a local positive maximum or negative minimum; such points are called peaks. Because the filter size at any level, k, is $\sqrt{2}$ larger than the filter at level k-1, there is a connectivity between peaks at adjacent levels. Connecting adjacent peaks between all of the levels gives a tree (or set of trees under some conditions) in which the path of the branches describes the location, size, orientation and shape of objects in the picture. In fact, it is necessary to compare the values along each branch to detect local maxima *along the branch*. These points serve as landmarks for determining the size, position, and orientation of gray-scale forms.

When an object has an elongated shape, it will give rise to a path of values which are larger than any adjacent values, that is, a "ridge". Ridges tend to begin and end at branches in the tree, and follow a path which can travel both between and along, a level. The paths of the ridges gives further information about the shape of objects in the image.

Figure 1-2 shows an example of a graph composed of peaks (Vt's)³ and ridges (Us) which represents a rhomboid form. This figure is taken from Chapter 7 where it illustrates the sequence of ridge points that represent an elongated form which changes width.

This tree and its ridges describes a gray scale form with symbols which represent circular regions*. The size of these regions span the range from radius = 4 to the size of the image. The tree and graphs for a particular gray scale form will have the same structure regardless of the gray scale form's size, position, or orientation, because this representation spans from global to local it may be used to align the representations of a pair of forms which are to be matched, even if they are of different

³Four types of symbols are used in the representation. These symbols are labeled with the letters { M, U, P }. These symbols are defined in section 1.3, and discussed at length in chapter 7 and ft.

sizes. The correct scale, orientation, and position of one form to the other may be determined by making a correspondence between the few "distinguished nodes" in the tree. Similarity in shape between two forms is readily apparent from the few symbols at the most global level. Thus if the identity of a form requires matching to a large set of prototypes, the search may be pruned based on the few most global symbols in the representation.

The representation produced by linking peaks and ridges in the 3-space function given by a SDOG Transform of an image:

1. is invariant (except for the effects of a discrete space) to changes in the size or position of a gray scale form (the effects of 2-D orientation can be easily compensated for);
2. provides a structure which may be used to determine the relative size, orientation, and position of two gray scale forms from two images;
3. permits the global shape of two gray scale forms to be compared without the cost of comparing details;
4. is not seriously degraded by textured regions, and degrades gracefully with image noise, or blurry edges.

The invariance to changes in size and position is qualified because there are small cyclic distortions which occur when an object is moved or scaled in size. These distortions are the result of the discrete nature of the 3-D space given by the SDOG transform.

1.2.2 Demonstrating the Properties of the Representation

The validity of the claims made above should become apparent as the reader absorbs the material presented in Chapters 3 through 8. These claims have been verified by experiments and are demonstrated with examples. Test images were taken from local data bases, in particular, from a copy of test images from GM for the "bin of parts" problem [Baird 77], and from a terrain data base of the Washington DC area. Six test images were digitized from 35 mm Black and white negatives by SRI International. In the last year, the CMU image understanding group has permitted access to the image digitizer on its Grinnell Display system. This has been used to make stereo pair images of a paper wad and a paint stirrer.

The *partial invariance to size* of the representation is illustrated by the representations from five teapot images. These images were formed from photographs of a teapot taken at three distances with two orientations at each distance. The change in size from the smallest teapot to the largest teapot spans a factor of approximately $\sqrt{2}$. The distortion of the representation from changes in scale is cyclic as scale changes by a factor of $\sqrt{2}$. The effects of this distortion are illustrate with the teapot images in chapter 8.

The effects of orientation are cyclic over a rotation of 90° . Rotating an object has only minor effects on the tree of peaks. The major effect of rotation is to change the density of the symbols along

a ridge path. This effect can also be compensated for in a matching rule. This effect is illustrated by two teapot orientations that differ in orientation by approximately 30°.

The use of the representation to determine the relative size and orientation of two images of an object is illustrated with the teapot images. It has also been demonstrated with the stereo pair of images of the paint stirrer.

Graceful degradation of the representation with noise, and the ability to represent both surface texture and the shape of a textured object have been demonstrated with the stereo pair of images of a paper wad. A portion of one of the paper wad images was degraded by substantial high frequency noise during digitization, this high frequency noise is almost entirely confined to the most local level of the representation. The paper wads also have surface texture which is represented in the lower (more local) levels of the representation while the shape of the paper wads is represented in the higher (more global) levels.

A simple explanation can obviate concern about blurred edges. A blur is the result of a convolution with a low-pass "blurring function" which occurs optically in the imaging system, usually from poor objective lenses, or motion. Only the highest frequency filters used in the representation are sensitive to such a distortion. Thus binning affects only the most local levels of the representation, the same can be said for a true high frequency noise, and for textured surfaces.

1.2.3 Research Methodology

The nature of image signals and the representation are used to describe a set of constraints for the filter. A more important observation is made to demonstrate that the filter is able to distinguish sequences of frequencies or phases. Several mathematical methods are also used to describe the filter (e.g., LP transform using Gaborian filters [The sampled DOG transform]).

Of course, the filter has the characteristics and the rules for describing the filter. The experimental tasks were designed to demonstrate that the filter could be applied to the problem of representing visual information whose structure must be compared to other visual information (as in stereo matching) or prototype representation of classes of visual objects (as in structural pattern matching). Of course, the empirical stage of the investigation yielded important principles and techniques for describing visual information with band-pass filters.

This empirical investigation yielded important principles and techniques for describing visual information with band-pass filters. The filter had the properties which they were derived to have, and that they could be applied to the problem of representing visual information whose structure must be compared to other visual information (as in stereo matching) or prototype representation of classes of visual objects (as in structural pattern matching). Of course, the empirical stage of the investigation yielded important principles and techniques for describing visual information with band-pass filters.

1.3 Results

This Section describes the major innovations developed in this research. New techniques were developed in three related problem domains:

1. The detection and measurement of gray scale forms in 2-D images;
2. Computational techniques for such measurement; and,
3. The representation of 2-D gray scale information.

The following three Sections summarize the results in each of these problem domains. The first of these Sections describes the new representation. In particular it describes the set of symbols used in this representation, the meaning of these symbols, and how they are interconnected. Some of the novel and important properties of this representation are also described. The second Section describes the measurements on which this representation is based. The final Subsection describes new computational techniques which were developed to reduce the time required to compute these measurements.

1.3.1 The Representation

This research produced a representation for two dimensional gray-scale signals. The representation is composed of a tree-like network of symbols which may exist at discrete locations in the three space (x,y,k) . The x and y dimensions of this space represents spatial position, while the k variable references a spatial frequency band.

This representation may be used for 2-D object class prototypes as well as image data. A representation computed from image data may be matched to a prototype despite changes in size, orientation or position. This matching may proceed from a few symbols which describe global form to more detailed local form. In this process, the matching process may be terminated if the global form is a poor match. Also, when matching stereo pairs, the correspondence between points in the two images may be easily determined by tracking through the representation.

There are four types of symbols in the representation:

- **M***: Peak points (positive maxima and negative minima) in the 3-space
- **L**: Ridge points in the 3-space
- **M**: Peak points at a given k (frequency band)
- **P**: Ridge points at a given k .

Each point in the 3-space, (x,y,k) , contains the inner product of a neighborhood of the image centered at (x,y) and a circularly symmetric filter impulse response of a radius selected by k . Peak and

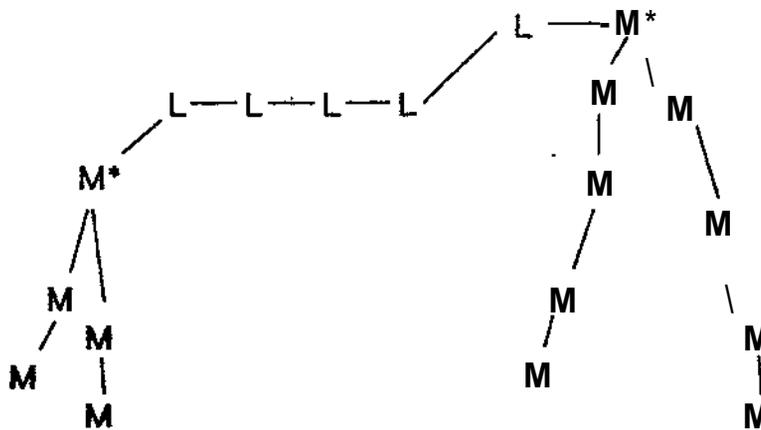
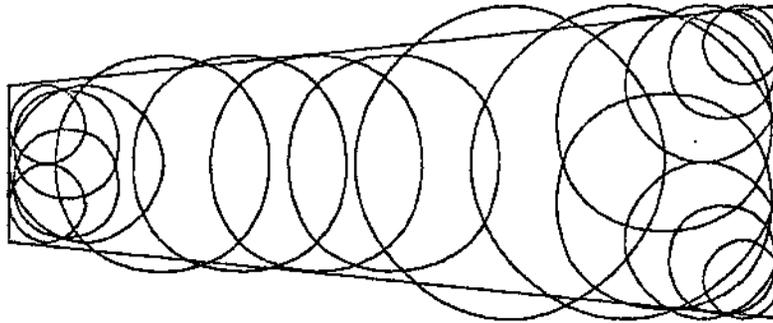


Figure 1-2: A Rhomboidal Form and its Representation
(Reproduced from Chapter 7, figure 7-19)

ridge points (M^* 's and U 's) in the 3-space mark the best fit of the primitive over a range of scales to a local set of image neighborhoods. Peak and ridge points (M 's and F 's) at a particular level (or band-pass image), mark the best fit of a particular fixed scale version of the primitive to a local set of image neighborhoods.

M^n points are particularly significant. These mark distinct visual landmarks or regions. The level, n , of an M^n symbol gives an estimate of the size of the visual landmark. More detailed information about the shape of the landmark is given by the linked paths of L 's (L-paths) and M 's (M-paths) that are connected to the M^n . The filters adhere to smoothness constraints which provide a continuity to the L 's to the M 's and between the L 's and M 's. The continuity permits paths in the 3-space to be formed by connecting adjacent L 's and adjacent M 's.

The shape of a form is represented by the network of L-paths and M-paths which result from it. If the form increases in size, the entire network moves in the k direction in the 3-space, but maintains its

connectivity and structure. Note, however, that since the components of the networks exist at discrete points in the 3-space, the motion occurs as discrete jumps of pieces of the network. Similarly, if the shape rotates, its network rotates, and if the shape moves, its network moves. The scale, orientation, and position quasi-invariance that is spoken of in this dissertation refers to the network. The size, orientation, and position information is available from the position (and orientation) of the network in the 3-space. The modifier "quasi-" is used because the individual symbols may only exist at discrete points, and make discrete jumps as the form changes smoothly in size, orientation, or position.

Figure 1-2 shows an example of the use of peaks and ridges for representing the shape of a gray-scale form. This figure, which appears in Chapter 7, shows a rhomboid shape. Circles over this form illustrate the position and radii of band-pass filters whose positive center lobes best fit the rhomboid. Below the rhomboid is part of the graph which is produced by detecting and linking peaks and ridges in the SDOG transform. The meaning of these symbols is described in Chapter 7.

1.3.2 Measurement Technique

This research produced two results which pertain to the problem of sensing (or measuring) the presence of gray scale forms in two dimensional data;

1. Design criteria for band-pass filters required to describe non-periodic data by means of peak and ridge detection.
2. A reversible transform (The OOLP Transform) that separates image signals into spatial frequency channels that meet the criteria for describing non-periodic data with peak and ridge detection.

The DOLP transform provides an ordered sequence of band-pass filtered versions of the input image. The impulse response of each band-pass image is a finite circularly symmetric function formed from the difference of two low-pass filters. The radii of the Impulse responses form an exponential sequence of the form:

$$R_n = S^n$$

where R_n is an Initial radius (typically 4.0 X S is a scale factor (typically V)* m is an index that ranges from 0 to K K is 16 for a 256 by 256 image).

One of the principal characteristics of the OOLP transform is that it is reversible. The Impulse responses may be thought of as a set of primitive functions from which pictures may be constructed. This primitive looks like a fuzzy disk on an inversely shaded background. The two dimensional convolution of the picture with each impulse response is equivalent to a sequence of inner products (see Section 3.1.3). This result facilitates an intuitive understanding of the filtering process. Each sample from the convolution indicates the proportion of signal energy within the neighborhood overlapped by the impulse response which is identical to the impulse response. In other words It is a measure of similarity between the impulse response and the image signal centered at that sample point.

Because these primitive functions are band-pass, they are sensitive to patterns over a narrow range of sizes. Thus for a textured region, the shape of the texture elements is described by a configuration of high frequency (smaller) impulse responses, while the shape of the entire region is described by a separate configuration of lower frequency (larger) impulse responses.

1.3.3 Computational Techniques

There are two computational techniques which resulted from this research:

1. The use of re-sampling in computing the Difference of Low Pass transform, and
2. A fast $O(n)$ implementation of the transform (the Sampled Difference of Gaussian Transform) that uses a novel technique: "Cascade filtering with re-sampling"

A consequence of the use of band-pass impulse responses is that the the cost of the convolution can be reduced by computing only at sample points. The distance between re-sample points has a lower bound which is a proportional to the size of the impulse response. Thus as the impulse response grows in size, the number of points at which the convolution must be computed decreases. If the convolution is done in the usual manner the increase in size of the impulse response is exactly balanced by the decrease (due to sampling) in the number of points at which the convolution is computed [Crowley 78a]. In addition to reducing the complexity and storage requirements of the filtering operation, re-sampling also contributes to the size invariance of the representation.

The Sampled DOG Transform, described in Chapter 6, is based on a property of Gaussian functions. Whereas, with re-sampling, a DOLP transform of an $N \times N$ image requires $O(N \log N)$ steps, the Sampled DOG Transform produces the same result in $O(N)$ steps. A step may be a multiply or an inner product.⁴

1.4 Organization of this Dissertation

This dissertation may be divided into the following sections:

- Background Material (Chapters 1, 2 and 3);
- Measurement, detection and mathematical representation of nonperiodic signals (Chapters 4 and 5);
- Fast computation techniques for the DOLP transform (Chapter 6);
- Converting the mathematical representation to a symbolic representation which describes gray-scale shape heirarchically by spatial frequency (Chapter 7);

⁴The symbol " $O(.)$ ", is pronounced order and used to indicate that the number of steps in the process under discussion is less than or equal to (bounded by) a linear function of the argument.

- Examples of the representation and its use for matching, including demonstrations of the invariance of the structure of a description to the size and orientation of the pattern (Chapter 8).

Chapter 2 describes related work by other researchers in sensing and representing forms in 2-D grey scale images. Chapter three provides a quick review of signal processing techniques and terms which were appear in this dissertation.

In Chapter 4, a set of criteria for designing band-pass filters for detecting and describing non-periodic signals is described. The criteria described in this Chapter defines a broad class of filters which may be used for detecting the presence of non-resonant signals of particular sizes (durations).

In Chapter 5, a reversible transform is defined which separates a signal into a set of short duration spatial frequency channels. The filters used in this transform satisfy the criteria established in Chapter 4. This transform employs a sequence of low-pass filters which are scale copies of a single function. The subtraction of adjacent low-pass filters gives a sequence of band-pass filters. These band-pass filters and the lowest frequency low-pass filter define the reversible DOLP transform. When an image has been convolved with these filters, the band-pass images may be added together to recover the original signal. The DOLP transform is shown to require SN^2 multiplies and $N \log_S(N/X_o) + N$ storage cells for an image with N sample points, a base filter of X_o coefficients, and a scale factor between filters of S . The technique of computing the convolutions at re-sample points spaced proportionally to the scale of the filters is then introduced. The re-sampled DOLP transform is shown to require $S X_o N \log_S(N/X_o) + X_o N$ multiplies and require $\approx 3N$ storage cells.

In Chapter 6 a fast version of this transform is defined which employs re-sampling and Gaussian filters to reduce the computational complexity to $3 X_o N$ multiplies. This fast transform employs repeated convolution with a small filter, and yet gives measurements which span the range of neighborhood sizes from a pixel to the size of the image.

In Chapter 7, techniques are described for detecting peaks and ridges within this three-dimensional transform space, and connecting these to form the representation. The structure of this tree represents a gray scale shape independent of its size, position or orientation.

Chapter 8 provides examples of the usefulness of the representation for matching as well as examples of the size, rotation and position quasi-invariance of the representation. This chapter describes the matching (or correspondence) problem in the domains of structural pattern recognition and stereo image interpretation. Examples are then presented in which the tree of peaks from the teapot images are matched despite changes of size and image plane orientation. A alignment procedure and similarity measure is then presented for ridge paths in the 3-space.

Chapter 2

Background: Related Techniques

This chapter reviews existing techniques for detecting and representing gray-scale forms in 2-D images. The first section discusses detecting and representing forms by their boundaries or as regions. Both region shape and boundaries are encoded in the representation developed in this research.

The second section covers popular techniques for detecting the presence of uniform regions using some form of linear detection function followed by a nonlinear decision rule. These techniques attempt to find edges which are then used to locate the boundaries of a region. The techniques described in this section range from very local edge detectors, such as Roberts' gradient [Roberts 65], to detectors which cover large areas, such as David Marr's Laplacian of Gaussians [Marr 79a].

The third section describes representation techniques. The problem here is to develop a representation for gray-scale forms or uniform regions which permits a fast search, alignment, and similarity measure. Techniques in this section include representations that are produced by segmentation programs, Blum's medial axis transform [Blum 67], and Marr's primal sketch.

2.1 Boundaries vs. Regions

At present there are two popular approaches to image representation: boundary representation and region representation. Pioneering work with the boundary description approach was done by Roberts' [Roberts 65]. The literature is full of recent work with this approach. Notable examples are [McKee 77] and [Perkins 78]. Estimates of the boundary position are usually obtained by convolving the picture with one or more small local edge detector followed by a non-linear decision function such as Roberts' gradient, the Sobel operator [Duda 73], or the Hueckel operator [Hueckel 71], [Hueckel 73]. See [Crowley 78b] for a list of many popular small edge detection functions and their transfer function. Some further encoding of boundary points is usually made to yield a representation which may be matched against stored models. McKee's paper [McKee 77] is a good example of this approach.

The primary advantage of most boundary detection schemes is that the description may be computed by a small, fast operator. However, a small operator can be a disadvantage, since the boundaries that are to be detected can be much larger (in width) than the operator. Also, a small operators tend to be sensitive to image noise, which is small and high frequency. Also, such a

description is expressed as many symbols which stand for very local events. It can be more efficient to represent the image as fewer symbols which represent more global (larger) events.

Region description is based on detecting regions of uniform intensity or color. This step is often referred to as segmentation. The usual approach is to compute a histogram of image intensities or histograms of color features which is(are) then scanned for well defined valleys. A threshold is set at the value in the valley. This technique can separate object from background nicely under proper lighting conditions. Regions are then represented by a binary bit map, or by measuring a set of features about the binary shape. This approach was pioneered by Prewitt [Prewitt 66], and Rosenfeld [Rosenfeld 69]. A good example of applying this approach to color features is described in Ohlander's Thesis [Ohlander 75].

Neither of these approaches are sufficient for an image which contains surface texture or weak and blurry boundaries. With both approaches there are problems in how the image structure is measured and in how the representation presents the information to later recognition processes.

2.1.1 Measurement Problems

Consider an image containing gradual intensity transitions. Such an image could be said to have blurry edges. If a local edge detector is used it will respond weakly over the entire large transition regions and the response will be so weak in some places that it will be lost. Increasing the gain will increase the sensitivity to noise. Similarly a region detection process will run into problems defining where such a region stops and starts. In such regions it is difficult to even define what is meant by an edge or a uniform region.

In images of real-world scenes, some boundaries between genuine objects are very weak. In a boundary description produced from local edge detectors, this usually results in missing boundaries and/or a failure of boundaries to form a closed loop.

In a threshold-based region segmented regions which should be distinct turn up joined. Also, unless a region has sharp boundaries and its intensities are distinct from those of the background* the 2-D shape of a region will be very dependent on the threshold.

One of the biggest trouble areas for both of these approaches is Image texture. Texture here refers in regions of an image containing many small forms which have random gray level shapes. Often in natural textures these small gray level forms are not uniform in intensity. Such textures may appear as many small hills and valleys in a terrain map. If the slope of these hills is approximately uniform across the object, the way in which the size varies in the image may be used to infer information about the depth of the object surface (Kistler 80).

A texture composed of randomly shaped nonuniform elements will swamp a threshold-based region segmenter with many small randomly shaped regions. The shape of an element can depend on the threshold. This region segmenter will spend a large amount of time and memory **representing** each element when what is needed is the shape of the whole textured region. Rosenfeld

[Rosenfeld 69] has noted that successively blurring such regions until the elements merge can be used to segment adjacent regions of different textures. This technique is based on the same principle as the representation developed in this dissertation.

With a natural texture, a local edge detector will respond sporadically over a large area with the result that there is no clear boundary. However, local edge detectors have been used to detect textured regions for region segmenters [Ohlander 75],

2.1.2 Representation Problems

A boundary description attempts to draw a closed boundary around regions which correspond to unique objects. Encoding the boundary with a chain code [Freeman 61], [McKee 77], for example, provides a representation which can be matched to a prototype to identify each closed region. There is a problem if the boundary does not close. In this case the interpretation program will not know which set of boundaries to attempt to identify. If there are many adjacent closed boundaries, there can be a problem knowing which corresponds to a genuine object, and which are artifacts. Also the entire boundary must be matched to identify an object. That is, if half of the outline of a region corresponds roughly to a prototype, but the other half is grossly different, the matching program may not discover the problem until it has attempted to match most of the boundary. The main problem is that in many situations edge detectors will report boundaries that do not correspond to an object's actual shape.

In a similar manner a region segmenter may produce erroneous data because of measurement problems, particularly when applied to images with weak or blurry boundaries.

Finally, with both techniques the resulting representation is dependent on the specific size of the objects in the image when what is desired is to recognize a shape independent of its size. Furthermore, a good representation should make available both the global shape of a form as well as local details. In this way a 2-D matching procedure can begin by matching the global form, and proceed to finer detail only if necessary.

2.2 Edge Detection Techniques for Boundary Representation

In this section we will review several measurement techniques which are related to the techniques described in this dissertation. The techniques described in this section have in common the goal of detecting edge segments for use as primitive symbols in a boundary representation of the forms in an image. As with the representation developed in this dissertation, most of these techniques are based on some linear measurement of image intensity, and seek to provide a description of the 2-D shapes in m image.

2.2.1 Local Edge Detectors

Many local operators have been proposed for detecting edges elements. A survey of such operators is included in [Crowley 80] along with the formula and plots of their transfer functions. The earliest such operator is Roberts' Gradient [Roberts 65]. This operator consists of a pair of first difference masks oriented at $\pm 45^\circ$. These masks are shown below in figure 2-1.⁵ Let the output of the convolution of the two masks at point (x,y) in the image be defined as $c_1(x,y)$ and $c_2(x,y)$. The estimate of the boundary at point x,y , denoted $e(x,y)$, is then formed as the square root of the sum of the squares, as shown in the following equation.

$$e(x,y) = \sqrt{c_1(x,y)^2 + c_2(x,y)^2} \quad (2.1)$$

Since Roberts' first defined this operator many researchers have observed that equation (2.1) may be approximated by the maximum of the absolute values or the sum of the absolute values as shown in equations (2.2) and equation (2.3).

$$e(x,y) = \text{Max}(|c_1(x,y)| + |c_2(x,y)|) \quad (2.2)$$

$$e(x,y) = |c_1(x,y)| + |c_2(x,y)| \quad (2.3)$$

$$\begin{array}{cc} 0 & 1 \\ -1 & 0 \end{array} \qquad \begin{array}{cc} -1 & 0 \\ 0 & 1 \end{array}$$

Figure 2-1: Masks Used in Roberts' Gradient

Probably the most popular local edge detector has been the Sobel operator [Duda 73]. Like Roberts' gradient, the Sobel operator consist of two small masks that are 90° orientations from each other. These masks are shown in figure 2-2.

$$\begin{array}{ccc} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array} \qquad \begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{array}$$

Figure 2-2: Masks Used in Sobel Operator

As with Roberts' Gradient, the results of the convolution may be combined by either equation (2.1), (2.2), or (2.3).

The Laplacian operator, $\nabla^2 p(x,y)$, has often been suggested as an ideal edge operator. The Laplacian, and its Fourier transform, are given in the following equations.

$$\nabla^2 p(x,y) = \frac{\partial^2 p(x,y)}{\partial x^2} + \frac{\partial^2 p(x,y)}{\partial y^2}$$

⁵Figures 2-1 through 2-3 show the masks for local edge detectors. These masks are shown as an array of coefficients which are convolved with an image.

$$\mathcal{F}\{\nabla^2 p(x,y)\} = -(u^2 + v^2)\mathcal{F}\{p(x,y)\}$$

where u and v are the spatial frequency variables and $\mathcal{F}\{\}$ is the Fourier Transform Operator.

Prewitt [Prewitt 70] designed two different two-dimensional difference equations which approximate the Laplacian operator. These masks are shown in figure 2-3 below.

$$\begin{array}{ccc} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{array} \qquad \begin{array}{ccc} -1 & -1 & -1 \\ - & 1 & 8 & -1 \\ -1 & -1 & -1 \end{array}$$

Figure 2-3: Two Discrete Approximations To the Laplacian from [Prewitt 70]

As with the Roberts' Gradient Edge Detector, these masks are convolved with an image. The result of the convolutions are then combined using equations 2.1, 2.2, or 2.3 to produce a map of edges in an image.

2.2.2 The Hueckel Edge and Bar Detector

Hueckel developed a function for detecting edges and bars that partially compensates for the fact that edges are not always very local discontinuities in an image. The Hueckel edge and bar detector [Hueckel 71] and [Hueckel 73] is based on a model of an edge as a step function, F , within a circular neighborhood. This step function has a number of parameters as shown in the following equation.

$$F(x,y;C,S,p,b,d) = \begin{cases} b & \text{for } Cx + Sy \leq p \\ b + d & \text{for } Cx + Sy > p \end{cases}$$

The parameters C , S , and p describe the direction of an edge or line. The parameters b and d describe the average grey level on either side of the edge. The Hueckel operator approximates the pixel values within a circular neighborhood,⁶ $E(x,y)$, by finding the parameters for which F is a **minimum** distance from E as shown in the following equation.

$$\iint [E(x,y) - F(x,y;C,S,p,b,d)]^2 dx dy$$

The Hueckel operator **solves** this minimization problem by multiplying the neighborhood, $E(x,y)$, and the ideal step, F , by a set of eight basis **functions**, $H_i(x,y)$ for $i = \{0, 1, 2, 3, \dots, 7\}$, as shown in the equations below. These basis **functions**, which are 'separable into a product of angular and radial **components**, are **referred** to as Hilbert functions. The interested reader should see [Hueckel 71] for a discussion, **definition**, and drawings of the zero crossings of these basis functions.

$$a_i = \iint H_i(x,y)E(x,y)dx dy$$

⁶ Although Hueckel defines these functions using integrals they are evaluated as a discrete summation over a circular neighborhood.

$$s_i = \iint H_i(x,y) F(x,y,C,S,\rho,b,d) dx dy$$

In these equations, the s_i 's are variables and the a_i 's are constants. Finding the parameters of F then becomes a matter of minimizing the following equation.

$$\sum_{i=0}^7 (a_i - s_i)^2$$

This minimization produces the parameters for the closest fit of an edge and an estimate of the likelihood that an edge is present.

All of the techniques described above detect and encode small sharp discontinuities in image intensity. As we discussed in section 2.1, such a representation does not capture all of the information in an image that is needed for matching to an object model. Such a representation is also inherently inefficient because it describes only very local detail and does not describe the global shape of regions.

2.2.3 Kelly's Use of Planning

One of the first researchers who attempted to use information from more than the most local resolution for finding boundaries was Kelly [Kelly 71]. Kelly called his technique "planning". Planning is a problem-solving technique for reducing the search space for a possible solution. Planning is the use of the solution to a simplified version of a problem as a guide to the solution of the original (more complex) problem [Minsky 63]. Planning was first employed by Newell, Shaw and Simon in the General Problem Solver [Newell 59].

Planning was applied to boundary detection by Kelly as part of his system for classifying images of faces [Kelly 71]. In this form of planning, edges are first detected in a reduced resolution version of an image. These edges are then used to guide the detection of edges in the original image.

Kelly's system operated on images composed of 250 by 330 picture elements. A 28 by 40 plan was prepared by dividing the image into *disjoint* 8 by 8 segments and calculating the average intensity within each segment. This operation is equivalent to a form of low-pass filtering followed by re-sampling. The low-pass filter for this application is an 8 by 8 array of coefficients of value 1/64. The re-sample distance is 8 picture elements. Serious aliasing can occur when the sample rate is the same size as the window. This can be seen by deriving the transfer function of the uniform square low-pass window [Crowley 78a]. (The transfer function is defined in section 3.3.)

2.2.4 Cones and Pyramids

In this section we will describe several recent research efforts which employ multiple-resolution versions of an image.

2.2.4.1 Uhr's Recognition Cones

Uhr has investigated the use of "recognition cones" for the low level processes of a machine vision system [Uhr 72], [Uhr 78]. A recognition cone is a multilayer array of micro-processors which execute the same instructions in "lock-step" fashion. Each processor in the lowest layer covers and operates on a disjoint region of an image. Successive layers of the cone see the output of the processors directly below. With each layer, the size of the image is reduced by averaging disjoint regions so that the cone converges to a single processor at the apex. Uhr has investigated the use of averaging and differencing on such a processor structure. He also suggests that such a structure may be used to assign symbols to regions of the image.

2.2.4.2 Hanson and Riseman's Preprocessing Cones

Hanson and Riseman have also investigated segmentation procedures which may be implemented on a recognition cone [Hanson and Riseman 74] and [Hanson and Riseman 78]. However, they prefer the term "pre-processing cone" rather than "recognition cone" because the processes performed are pre-recognition. In their system, the pre-processing cone serves as the front end of a general purpose color vision system. The system builds a structural description of a scene using multiple knowledge sources and threshold based segmentation.

Hanson and Riseman have categorized the operations which may be computed on a pre-processing cone into the following classes:

- **Data Reduction:** Operations such as averaging which pass information up to the next higher level.
- **Data Projection:** Operations in which image data and interpretations are passed down to lower levels.
- **Iterative (or Lateral):** Operations which are based solely on the neighboring processors at the same level.

2.2.4.3 Pyramid Data Structures

A recognition or pre-processing cone is a form of parallel Single Instruction-Multiple Data (SIMD) Processor. The data structure which it contains is sometimes referred to as a "pyramid data structure". The low-pass images on which the DOLP transform is based can be considered as a form of pyramid data structure. While some researchers lump together the characteristics of the processor and the data structure it builds, others have made a distinction in order to study the properties of the data structure.

Tanimoto has defined a pyramid data structure as "a series of digitizations of the same image at increasingly higher degrees of spatial resolution" [Tanimoto 78]. A standard relationship between a given level of a pyramid and the level under it is that a local property (such as edge intensity, color, or intensity) at the given level is obtained by averaging the local property over some neighborhood in the level under it. In virtually every system these averages are formed over disjoint regions, which can cause a randomness due to aliasing [Crowley 78a] as noted above in the description of Kelly's planning technique.

Tanimoto has suggested that the sequence of reduced resolution images need not be obtained by averaging nor even based on powers of 2, but can be obtained by a specially designed digitizer and computer controlled optics capable of providing magnification of the image over a continuous range.

Levine [Levine and Leemet 76] has investigated a system in which a pyramid data structure is used for bottom-up and top-down segmentation. His algorithm constructs five pyramids from the original image; one for each of the following local properties: intensity, a texture measure, hue, saturation, and edges. These pyramids contain outlines of segmented regions which are then passed to an intermediate level process for interpretation.

2.2.5 Other Work with Multiple Resolution Representations

Kelly is most frequently cited in the image processing literature for pioneering the use of multiple resolution versions of an image. However, similar ideas appeared in other literature at about the same time.

The use of a reduced resolution "plan" for space planning (i.e. arranging 2-D shapes in an area) is discussed in a 1970 paper by Eastman [Eastman 70]. Eastman credits work conducted at SRI on trajectory planning and on reconnaissance for the idea [Nilsson 69] and [Rosen and Nilsson 69]. Eastman referred to this data structure as a "Hierarchical Data Structure" but it has since come to be known as a quad tree [Klinger and Dyer 76], [Horowitz 76]. Quad trees represent binary shapes in an image by recursively dividing the picture into a 2 x 2 set of sub pictures. If any subpicture is completely filled or completely empty, it is marked as such and not divided further. If a subpicture is only partially filled it is further divided. This process continues until either all the subpictures are uniform or the individual pixels are reached. The result is a tree which can be traced to determine if any point in the picture is filled or empty. This algorithm can be very efficient in terms of the storage required for pictures that have large uniform regions. However, the description of a region which this representation gives can vary drastically in its structure if the region is translated in position or rotated.

Warnock [Warnock 67] devised a similar algorithm for computing the hidden surfaces in two-dimensional views of three-dimensional polyhedra. In Warnock's algorithm, a two dimensional picture or subpicture is recursively divided into four squares if it contains a boundary between two faces of polyhedra or a boundary between a face and the background.

A pyramid data structure has been used by to speed up correlation template matching of aerial

imagery using hierarchical search [Hall et. al. 76]. Two-stage hierarchical template matching has also been reported for image feature detection [Rosenfeld and Vanderbrug 77].

2.2.6 Marr's Laplacian of Gaussians

Probably the work most similar to that described in this dissertation is that of David Marr. Marr sought to understand the information processing problems inherent in vision. He was interested in both the mechanisms to visual stimuli in the human visual system and in the computational problems of implementing such processes in machines.

[Marr 79a] presents a theory of edge detection which recognizes that the information in visual stimuli occurs at many scales (or resolutions). To detect these stimuli at different scales he employs band-pass filters which are formed from a Laplacian of Gaussian low-pass filters ($\nabla^2 g(x,y)$). Marr forms these filters using a difference of Gaussian low-pass filters whose standard deviations have a ratio of 1.6. He uses an informal argument to show that such a ratio gives an optimum narrow band width. (The implementation described in this dissertation employs a ratio of $\sqrt{2}$ arrived at by a very different line of reasoning.)

A set of such filters (4 in [Marr 79a]) are convolved with an image. The results are encoded by detecting the presence of zero crossing segments and the directional derivative perpendicular to the zero crossing at each segment (called the amplitude of the segment). This set of zero crossing images is referred to as the "raw primal sketch". Marr speculated that if filters were used at a sufficient number of scales, the raw primal sketch would be reversible. That is, the original image could be recovered from the raw primal sketch.

Zero crossing elements from several scales are collapsed into a single boundary estimate called the "primal sketch". This is done by comparing zero crossing segments from adjacent spatial frequency levels, to test for similar directions and amplitudes. The zero crossing segment from the highest resolution raw primal sketch is encoded in the primal sketch. Closed boundaries are labeled as blobs and assigned attributes of length, orientation, and average contrast. Terminations are assigned a position and orientation. We shall have more to say about Marr's work in the section on representation below.

2.3 Representation Techniques

2.3.1 Blum's Medial Axis Transform

Blum developed a representation for binary shapes called the "Medial Axis Transform" (MAT) [Blum 67]. This representation is interesting because it is object centered; that is, components of a shape are defined relative to a central (or medial) axis. This region representation bears some similarity to the representation developed in this dissertation.

The medial axis transform produces a form of skeleton for a binary shape defined on a continuous medium. The MAT may be defined by the following process. Each point on the boundary of a binary region transmits a circular wavefront on both sides of the boundary. These wavefronts propagate until they reach another boundary point or until they meet a wavefront traveling in exactly the opposite direction. When two wave fronts meet traveling in opposite directions, they cancel each other, and the point where they meet is marked as belonging to the medial axis. Such points correspond to the center of circles which are fit tangent to two or more points on the boundary of the shape.

The collection of medial axis points defines a set of connected spines (or center axes) describing the form of the shape. Where a shape contains a concavity, spines occur outside the binary shape as well. Similarly, spines occur for the space between shapes. (This is the negative shape which occurs between two positive shapes.) Spine points can be encoded with the distance to the boundary from which they propagated. This gives a reversible representation of the binary shape as these distances correspond to the radii of discs that must be placed overlapping on the spine to reconstruct the binary shape.

Unfortunately there are several problems with the medial axis transform. For one thing, the transform operates only on binary shapes which introduces all of the problems attendant to thresholding techniques. Also the transform is only defined for a continuous medium. Propagating circular wavefronts on a discrete grid is a difficult and costly process. Perhaps most troublesome is that the structure of the medial axes are altered drastically by minor nicks and protrusions on the boundary of the shape.

There is some similarity between the MAT and the representation described in this dissertation. The path of the spines for a simple object resemble the paths of peaks and ridges from our representation projected onto the original picture. Our representation also produces a description of the negative shapes outside a gray scale form when there is a concavity and when two shapes are nearby. However, nicks or protrusions narrower than half the width of the gray scale form do not affect the overall path of ridges and peaks. The biggest difference is that our representation is computed for discrete gray scale forms, while the MAT is defined for continuous binary forms.

2.3.2 Marr's Three Levels

David Marr has developed a framework for visual information processing that includes representations at three levels [Marr 78]. The first such representation is the primal sketch which is described above. The primal sketch encodes information about the boundaries of forms in an image from different resolutions.

The second representation is referred to as the 2 1/2-D sketch [Marr 79a]. This is a form of depth map of surfaces as seen by the viewer. Various processes that interpret depth cues from such phenomena as texture, shading, and stereo perception contribute information to form the 2 1/2 D sketch.

Marr asserts that an object centered representation is also required for general purpose vision and that this 3-D representation should include shape primitives from many resolutions. Furthermore he asserts that this representation should take advantage of axes of symmetry which are intrinsic to the object. He cites the generalized cylinder representation [Agin and Binford 73], [Nevatia and Binford 74] and the Medial Axis Transform [Blum 67] as examples of representations that have these properties.

Chapter 3

Signal Processing Background

Digital signal processing is an engineering discipline which, like image understanding, has been made possible by the widespread use of digital computers since the early 1960's. Its theoretical foundation is linear systems theory, a body of continuous mathematics which is fundamental to electrical engineering.

Since many persons interested in image understanding lack training in digital signal processing, this chapter provides some definitions and intuitive explanations for techniques from digital signal processing which are necessary in later chapters. Most of the material in sections 3.1, 3.2 and 3.4 is available in widely used references. The text [Oppenheim 75] is particularly relevant. A very readable introduction to digital signal processing for non-electrical engineers is [Hamming 77]. The transfer function derivation given in section 3.2 is from this book.

3.1 Convolution, Correlation, and Inner Products

This section provides the formulae for the 2-D convolution and 2-D cross-correlation of a finite 2-D filter with a 2-D signal. These formulae are shown to be identical for filters which are symmetric about both axes, as is the case with the circular symmetric filters discussed in chapters 5 and 6. The 2-D cross-correlation is then shown to be equivalent to a 2-D sequence (or array) of inner products. This equivalence gives a heuristic for interpreting the results of the cross-correlation. This heuristic leads to the use of peak and ridge detection for converting the filtered signals into symbols, as described in chapter 7.

This research has concentrated on the use of non-recursive finite impulse response (FIR) filters; we have avoided the design problems involved in 2-D recursive filters. It is impossible for a causal recursive filter to have zero or linear phase. Furthermore, there is no known design procedure for generating a stable 2-D recursive filter which would satisfy the constraints developed below.

3.1.1 Convolution

A 2-D finite impulse response digital filter may be defined by specifying its impulse response. For discussion, let us define a 2-D discrete impulse response:

$$g(x,y) \text{ for } |x| \leq X_g \text{ and } |y| \leq Y_g$$

The variables x and y are, of course, integers.

The filtering operation is usually expressed as a convolution, denoted M^* . Let us also define a 2-D discrete input signal:

$$p(x,y) \text{ for } |x| \leq X_p \text{ and } |y| \leq Y_p$$

The convolution of $g(x,y)$ with $p(x,y)$ is given by the formula:

$$g(x,y) * p(x,y) = \sum_{k=-X_g}^{X_g} \sum_{l=-Y_g}^{Y_g} p(x-k, y-l) g(k,l)$$

3.1.2 Correlation

In this work we have preferred to express the filtering operation as a cross-correlation. The reason will be explained below. We shall denote cross correlation with the symbol "*" for lack of a better symbol. The formula for a 2-D cross-correlation is:

$$g(x,y) * p(x,y) = \sum_{k=-X_g}^{X_g} \sum_{l=-Y_g}^{Y_g} p(x+k, y+l) g(k,l)$$

The difference between correlation and convolution is the presence of the minus sign in the term $p(x-k, y-l)$. These minus signs have the effect of rotating the impulse response about both axes. This rotation describes the behavior of a continuous linear filter, as implemented, for example, in a circuit. If the impulse response is symmetric about both axes, as in the case of the circularly symmetric filters described below, there is no difference.

3.1.3 Inner Products

In this foetid* we are interested in expressing an image as a configuration of primitive signals. These primitives were referred to as a family of "detection functions" in our early work, [Crowley 78a]. We have since developed a class of families of detection functions such that an image signal can be expressed uniquely as a weighted, displaced sum of detection functions. A method for computing the weights* which is reversible, has come to be known as the DOLP transform, and is defined in chapters.

The weight tells how strongly the primitive matches the image signal at a particular point. This weight may be determined by computing an inner product of the primitive (which is an impulse response) and the signal within a finite neighborhood centered at the sample point. The size of the neighborhood is the same as the size of the primitive.

An inner product at some sample point x_0 or y_0 is given by the formula:

$$\langle S, p(x_0, y_0) \rangle = \sum_{k=-X_g}^{X_g} \sum_{l=-Y_g}^{Y_g} p(x_0+k, y_0+l) g(k,l)$$

This formula is identical with the formula for each point in the cross-correlation*

The point here is that the filtering operation, or cross-correlation is a sequence of inner products.

This notion of the filtering operation as a sequence of inner-products leads to an important heuristic for converting the filtered signal into a network of symbols. Those points at which the correlation of a particular filter and the input signal are at a 2-D local positive maximum or negative minimum are the points at which that filter most strongly resembles the input signal. If the inner-product at that point is also larger than inner-products from filters which are similar in size, then that filter at that point is the best approximation of the image signal centered at that point. Such points form an important class of symbols in our representation. They are labeled M^* and serve as landmarks in the representation, as well as the root for subgraphs.

In summary, the view of the filtering operation as a sequence of inner-products leads to the use of peaks (and ridges) in the filtered signals to construct the representation of the image. This is in contrast to the more popular approach of using zero-crossings as pursued by Marr in his related work [Marr 78].

3.1.4 Boundary Values

The DOLP transform employs circularly symmetric low pass filters whose radii range from 4 pixels to the size of the image. In each correlation there is a strip along the border of the filtered image whose width is the same as the filter's, along which the filtered signal is corrupted because the filter only partially overlapped the image. These points could be discarded, but this would lead to an inability to detect any object closer than its own width to the border of the image. Our solution was to provide a default border value, given by the mean of the image pixel values. This has the desirable effects of allowing description of objects near the border of the image, and keeping the filtered image sizes as powers of 2. It has the undesirable affect of causing a ripple along the border whenever the pixels at the border are not close in value to the mean.

3.2 The Transfer Function

The transfer function is an important tool for the design and analysis of discrete linear functions. In this section we will define the transfer function for the case of a two dimensional discrete linear function. We will then show that any discrete 2-D function has a transfer function which is continuous and periodic in two dimensions. The boundary of the region over which the transfer function is unique is called the Nyquist Boundary. The shape and size of this boundary is determined by the pattern of sample points used in filtering. The Nyquist Boundary is the primary tool for selecting the density of sample points for a filter or designing a filter for a given sampling density.

3.2.1 Eigenfunctions

One of the properties which make linear systems so mathematically tractable is the existence of a class of well behaved eigenfunctions (also known as characteristic functions). The eigenfunctions of a discrete 2-D linear system are the set of sampled 2-D exponentials given in equation (3.1)

$$e^{\pm j(xu+yv)} = \text{Cos}(xu + yv) \pm j\text{Sin}(xu+yv) \quad (3.1)$$

The variables u and v are continuous and often referred to as spatial frequencies. The eigenfunctions for a given discrete 2-D linear system are those complex exponentials for which u and v fall within a bounded region in the center of the u,v plane. The boundary of this region is known as the Nyquist Boundary. Its shape is determined by the pattern of sample points used in the filter operation. We shall return to the Nyquist boundary in the next section.

3.2.2 Derivation of the Transfer Function

When a linear function is convolved with an eigenfunction the result is the same eigenfunction shifted in space (or phase) and scaled in amplitude. The phase shift, $\Phi(u,v)$, and the amplitude attenuation, $A(u,v)$, are position invariant. They are a function of only the spatial frequencies of the eigenfunction.

We can express this phase shift and amplitude attenuation as a complex function, $H(u,v)$, known as the transfer function. Its relation to $\Phi(u,v)$ and $A(u,v)$ is given by the following equations:

$$A(u,v) = |H(u,v)|$$

$$\Phi(u,v) = \text{ArcTan}[\text{Im}\{H(u,v)\}/\text{Re}\{H(u,v)\}]$$

$$H(u,v) = A(u,v) e^{j\Phi(u,v)}$$

Where $\text{Im}\{.\}$ gives the imaginary part of a complex function and $\text{Re}\{.\}$ gives the real part

The effect of convolving a discrete 2-D finite impulse response filter,

$$W(x,y) \text{ for } |x| \leq X_h \text{ and } |y| \leq Y_h$$

with an eigenfunction may be expressed as a multiplication with the transfer function in the spatial frequency plane as shown in equation (3.2).

$$H(u,v) e^{j(u^*x + v^*y)} = \sum_{k=-X_h}^{X_h} \sum_{l=-Y_h}^{Y_h} h(k,l) e^{j(k^*x + l^*y + u^*x + v^*y)} \quad (3.2)$$

We can easily derive the formula for computing the transfer function from the impulse response by factoring out the eigenfunction from both sides of equation (3.2). This formula is given in equation (3.3).

$$H(u,v) = \sum_{k=-X_h}^{X_h} \sum_{l=-Y_h}^{Y_h} h(k,l) e^{j(ku + lv)} \quad (3.3)$$

$$k = -X_h \quad l = -Y_h$$

3.3 Two Dimensional Re-Sampling

In this section we examine in more detail what the Nyquist Boundary tells us about the pattern of sample points. In this discussion it is assumed that the input image and the impulse response are given as discrete 2-D sequences. We are concerned with reducing the number of sample points. We use the term "re-sampling" to distinguish this from the related problem of sampling a continuous function to produce a discrete sequence. Sampling a continuous function is amply treated in many digital signal processing texts. We recommend [Oppenheim 75] which has come to be recognized as the classic text book for digital signal processing. Re-sampling a 1-D sequence will be discussed first and then the results extended to 2-D.

3.3.1 Re-Sampling a One Dimensional Filtered Sequence

For a one dimensional linear function, the eigen-functions are the complex exponentials, $e^{j\omega x}$ for which the continuous frequency variable, ω , is within the bounded region $|\omega| \leq \pi/S_R$, where S_R is the distance between samples, and must be an integer. Complex exponentials for which ω is outside this range are aliased by the sampling. That is, they appear in the sampled sequence as one of the complex exponentials from within the interval. Complex exponentials from outside the Nyquist boundary are, in effect, rotated about the interval boundary.

3.3.2 Two-Dimensional Nyquist Boundary

The extension to two dimensions is straight-forward if the samples are taken at points along axes which are aligned with the original sample axes. That is, if every S_x th point in the x direction on every S_y th row in the y direction are chosen as sample points, then the transfer function of the sampled sequence will be defined within the rectangular boundary:

$$|u| \leq \pi/S_x \quad \text{and} \quad |v| \leq \pi/S_y.$$

In the techniques developed in chapter 5 we employ a type of sampling in which the samples are along the diagonals, $\pm 45^\circ$. We refer to this form of sampling as $\sqrt{2}$ resampling, because this is the minimum distance between sample points. The $\sqrt{2}$ resampling operation, $S_{\sqrt{2}}$ may be defined as:

$$S_{\sqrt{2}}[p(x,y)] = \begin{cases} p(x,y) & \text{for } x \bmod 2 = y \bmod 2 \\ \text{undefined} & \text{otherwise} \end{cases}$$

When applied to a cartesian grid with axes at 0° and 90° it yields a new grid where the unit sampling distance axes are at $\pm 45^\circ$ as shown by the circles in the Figure 3-1 below. When applied to a grid where the axes are at $\pm 45^\circ$ it produces a new sampling grid with a unit distance of 2 and unit distance axes at 0° and 90° as shown by the squares in figure 3-L

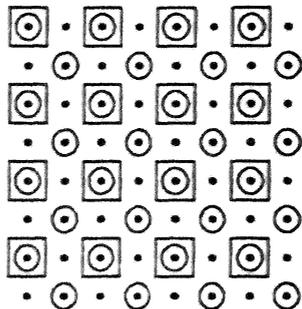


Figure 3-1: Example of $S_{\sqrt{2}}[p(x,y)]$ and $S_2[p(x,y)]$

In the frequency domain, each application of $\sqrt{2}$ sampling introduces a new Nyquist boundary which is skewed by 45° from the previous Nyquist boundary, and just fits inside it, as shown in figure 3-2.

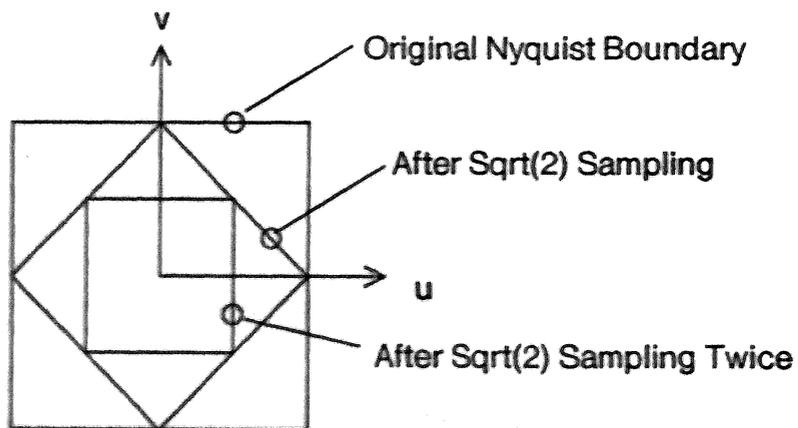


Figure 3-2: Nyquist Boundaries for Successive Application of $\sqrt{2}$ Sampling

Aliasing is minimized by designing the filters so that there is a large attenuation for all points outside of the new Nyquist boundary.

3.4 Design Parameters for Digital Filters

In this section we will define some of the terms that are commonly used in the design of finite impulse response digital filters. There is nothing original in this section. It is included so that when these terms are used in later sections and chapters the reader will know what they mean.

Digital filter design is an optimization problem. Digital filters are generally designed by specifying a set of constraints on the transfer function and then allowing a linear optimization program, such as the Parks-McClellan algorithm [Parks 72] to find the coefficients for the best solution. The constraints that are commonly used for designing a low pass filter are illustrated below in figure 3-3.

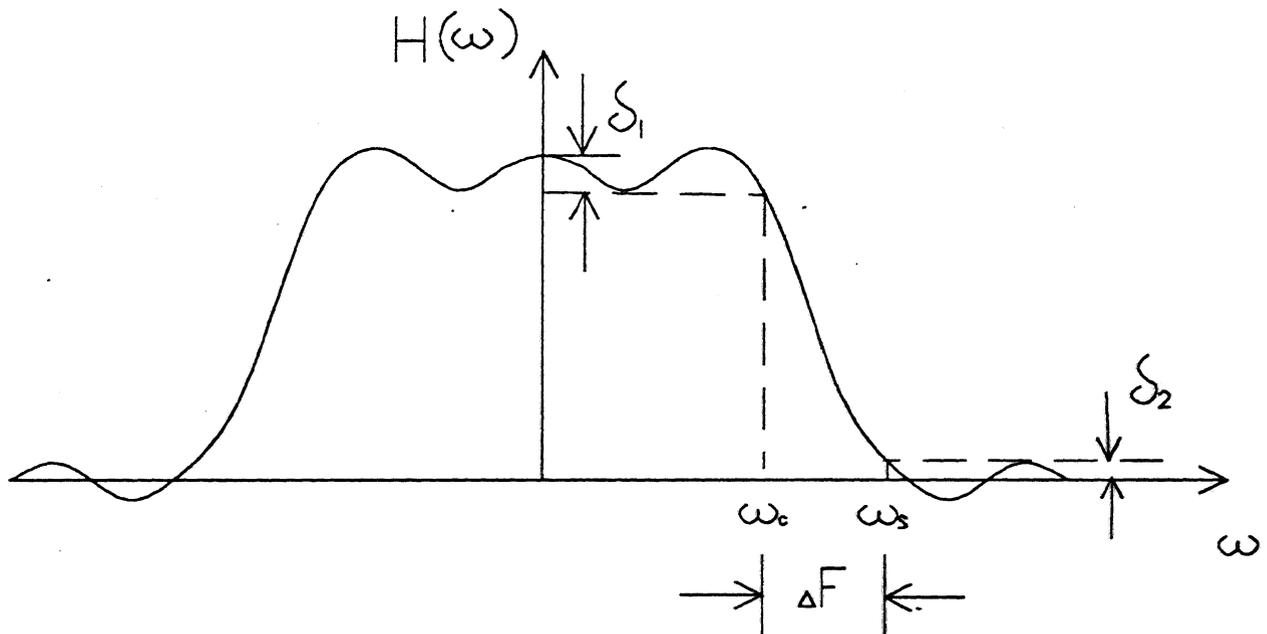


Figure 3-3: Transfer Function Constraints for a Low-Pass Filter

The symbols for the constraints are:

δ_1 : The pass band ripple peak amplitude

δ_2 : The stop band ripple peak amplitude

ω_c : The pass-band cut-off frequency where response falls below $1-\delta_1$.

ω_s : The stop-band frequency edge where response falls below δ_2

AF: The transition width, or width of the transition region, given by $\omega_s - \omega_p$

ω_p : the frequency where response falls below 1/2 (-3dB).

The usual goal is to find the shortest filter which has a sufficiently flat pass and stop band and a sufficiently narrow transition width. δ_1 and δ_2 can be traded off against each other. Their product can be traded off against AF. The product of all three can be traded off against the number of coefficients.

Chapter 4

Criteria for the Design of Band-Pass Filters for Detecting Non-periodic Signals

In this chapter we develop several ideas which are fundamental to the results described in later chapters. Section 4.1 describes the concept of a family of detection functions which are scaled copies of a single prototype function. This concept leads to a reversible transform based on the difference of size scaled copies of a low-pass filter, which is described in the next chapter. Such a family of detection functions are convolved with a signal or image to separate the information into spatial frequency channels. This provides an ability to discriminate the size of a gray-scale form by detecting the frequency at which the maximum response occurs. This transform also provides the basis for the representation described in chapters 7.

Section 4.2 establishes a set of design criteria for band pass filters that are to be used with peak (and ridge) detection to construct a scale invariant representation of non-periodic signals. These criteria are general; there are many methods by which a band-pass filter may be designed to meet them. Our early work with this criteria used filters which were designed by a quite different technique than the difference of low-pass filters that is described in chapters 5 and 6 [Crowley 78a], [Crowley 78b].

In section 4.3 we consider the problem of selecting the set of scale factors for a family of detection functions. We show that the criteria of size invariance constrains the filter radii to be members of an exponential sequence. Size invariance also dictates re-sampling at a rate proportional to the radius of each filter. Unless we interpolate and then decimate, the resampling distances must be members of the set of distances that occur between points on the sample grid on which the picture (or signal) has been digitized. The smallest base for such a sequence which occurs on the 2-D cartesian sample grid is **VI**.

4.1 Family of Detection Functions

In this section we define the term "detection function" and then introduce the concept of a parameterized family of detection functions. Some of the possible approaches for designing a family of detection functions are then examined.

4.1.1 Detection Functions

The term "detection function" was coined early in this research. A detection function is a linear function (impulse response) followed by some non-linear decision rule. Most of the edge detectors described in section 3.2 are examples of detection functions.

The techniques developed below extend the concept of a detection function beyond the detection of local sharp transitions in gray level.

The linear function part of a detection function is typically designed as a matched filter for the pattern which it is to detect. See [Wozencraft 65] for a discussion of matched filter design. The obvious example are the plethora of edge detectors in the literature, but there are other examples such as the GM system for IC chip alignment in which corners are detected. In some systems, such as the GM system, the image domain can be sufficiently constrained and the problem structured so that a specialized detection function is quite reliable. However for general purpose vision, where there are few constraints on image quality or content there are serious problems. For example, what pattern should be detected? We have already discussed in section 2.1 some of the problems with detecting edges and interpreting them as boundaries. Another problem is that patterns can occur over a range of neighborhood sizes. If the pattern is blurred or noisy or the contrast is low, a larger neighborhood must be examined. But then it becomes easy to miss the edges of small patterns. Textured regions are particularly troublesome because it may be desirable to detect information at many neighborhood sizes. In the following sections we shall describe a solution that employs a set of functions whose sizes range from very local to global.

4.1.2 A Family of Detection Functions Which Provide Spatial Frequency Channels

This research began as an effort to demonstrate the following idea [Crowley 78b]:

A robust (in the sense of able to handle blurry or textured images) and efficient (in the sense of representing global shape of an object in a few symbols) structural description of an image can be formed by filtering the image into a set of spatial frequency channels and then representing peak points and ridge points with symbols.

A principle on which much of this work is based is that a class of band pass filters can be defined such that each filter is sensitive to signals of a particular range of widths. Furthermore the width of a signal can be determined, within some tolerance, by determining which filter gives the largest peak response. In section 4.2 we develop a set of constraints for designing detection functions for this purpose.

Investigating the design of the spatial frequency channels led to the concept of a parameterized "family of detection functions". A family of detection functions is defined by a closed form expression which includes one or more independent parameters. The independent parameters determine the coefficients of the linear part of a particular detection function. Initial experiments were conducted with a family of detection functions formed by the product of a circularly symmetric low-pass window and a 1-D cosine [Crowley 78a]. The independent parameters were the frequency and orientation of the cosine.

Ideally we would like to convolve the image with a continuum of filters such that if a test pattern (say a solid disc) of a particular size is the input signal, one filter from the continuum will have a peak response which is larger than all of the others. Furthermore, it should be possible to determine the size of the test pattern (within some tolerance) from the identity of the filter with the largest peak response.

A number of experiments were reported in the proposal for this dissertation in which band-pass detection functions were convolved with uniform intensity circles and squares of different sizes and with uniform intensity bars of different widths and orientations. These experiments demonstrated that the size of the circles and squares, and the width and orientation of the bars could be determined by observing which detection function produced the largest peak in the convolution. We also observed that certain structural elements such as edges and corners resulted in easily detected patterns of peaks and/or ridges when convolved with each of the detection functions smaller than the object. Thus it is possible to detect these structural elements at many neighborhood sizes and sampling densities. Also it was noted that a configuration of test patterns forms a shape which is independent of the test patterns (a textured shape). The size and structural features of this textured shape are apparent in the convolution with detection functions which are larger than the individual test patterns.

4.1.3 The Goal of Size Invariance

The three dimensional shape of an object is intrinsic to the object. The two dimensional image of an object should depend only on the objects 3-D shape, the viewing angle, and the lighting conditions. A description of the 2-D gray scale shape of an object should not depend on the size at which the object is imaged.

Early in this research we decided to pursue a representation for 2-D form that has the property of being independent of the scale at which the object is imaged. That is, suppose an object is in the field of view of a television camera, and a representation is constantly being constructed of how the object appears in a sampled, digitized image from the camera. If the object is moved toward the camera, the representation should shift in size but retain its structure. Also, as additional information about the object's surface texture and edges becomes available it should be appended to the representation, but this should not alter the part of the representation that denotes the global shape of the object. In this research we pursued the goal of producing a size invariant representation using detection functions that are size scaled copies of the same function.

4.2 Linear Functions for Describing Non-Periodic Signals with Peak and Ridge Detection

In this section we develop a set of constraints for the space domain coefficients and the frequency domain (transfer function) for the design of a set of 2-D linear functions. These functions are to be used with peak and ridge detection to construct a representation for die non-periodic signals which occur in images. We are not able to provide a rigorous proof that all of these constraints are necessary. We only make the claim that these constraints are sufficient

The following subsection will develop the reason why the detection functions are constrained to be:

1. Zero Phase
2. Finite Impulse Response,
3. Circularly Symmetric, and
4. Band Pass Filters,

We will then develop the more complex criteria that the functions:

1. Must have 3 peaks (5 alternations) in the coefficients, and
2. Must have a pass band which rises monotonically to a single peak.

4.2.1 Zero Phase

The transfer function of the linear function must be zero or linear phase. A non-zero phase will shift the position of the response. If the phase is linear the shift is the same for all frequencies. If the phase is non-linear, the shift will vary with spatial frequency. The position of the signal is important to the structure of the representation. We cannot permit unpredictable shifts in the reported position of a signal because of a slight uncertainty in its width (frequency content).

4.2.2 Finite Impulse Response

The impulse response must be finite. The reason is that infinite impulse response filters can only be implemented by recursive filters. There is no design process for a 2-D recursive filter that will guarantee a zero or linear phase. There are also problems with designing 2-D recursive filters which are stable. We have limited our inquiry to finite impulse response filters to avoid these problems.

4.2.3 Circular Symmetric

The impulse response must be circularly symmetric. This is because the representation should be as invariant to orientation as possible. We cannot allow the detected size and position of a peak to be affected by the orientation of a signal.

4.2.4 Band Pass

The impulse response coefficients must sum to zero. This will assure that if the function is convolved with a uniform signal, the response will be zero. Another way to say this is that the DC response must be zero.

The transfer function must also have a high frequency stop band. This will allow the convolution to be computed at re-sample points without aliasing. The net effect of these two constraints is that the function will be a band pass filter.

4.2.5 Constraining Alternation (Peaks) in the Space Domain Coefficients

In this section we will show that the linear function must have 3 peaks (5 alternations) in its coefficients. This constraint is necessary when the detection functions are to be used with peak and ridge detection (detecting local positive maxima and negative minima). Without this constraint, other constraints such as the need for a narrow pass-band and sharp transition band would drive the design to a function which had many ripples (alternations) in its impulse response. To see why this is a problem, consider the case where a detection function is convolved with a bar which is smaller than half the width of the detection function. Each peak in the detection function coefficients will result in a peak in the convolution output. Since the presence and shape of the bar is to be encoded from the peaks and ridges in the convolution, the result will appear to be many bars.

We can determine the smallest number of peaks which the detection functions can have by enumerating the possibilities and examining the function which results from each. For convenience this discussion will consider 1-D functions. The results must apply to 2-D circularly symmetric functions. The results will only apply to a circularly symmetric function if the 1-D function is symmetric, i.e. if $g(x) = g(-x)$. Thus the 1-D functions discussed below are constrained to be symmetric. Also, we are only interested in finite zero-phase functions for the reasons explained above.

Let us define the term "alternation" to refer to a change in sign in the first difference, $d[g(x)]$ of the function, where first difference of a discrete function $g(x)$ is defined by:

$$d[g(x)] \triangleq g(x) - g(x-1)$$

Let us make the arbitrary definition that when the first difference is zero, its sign is the same as the point to the right. With this definition functions which have a constant interval can be considered in this discussion. Also, to keep things tidy, let us define the boundaries of the support for a finite

discrete function to be alternations. Thus all finite 1-D functions automatically have at least two alternations.

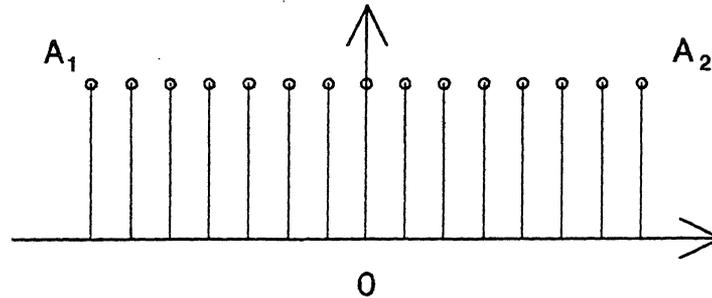


Figure 4-1: The Only Possible Symmetric 1-D Function with Two Alternations

Two Alternations: (see figure 4-1 above.) In order to be symmetric such a function must be constant. It is thus a low pass function.

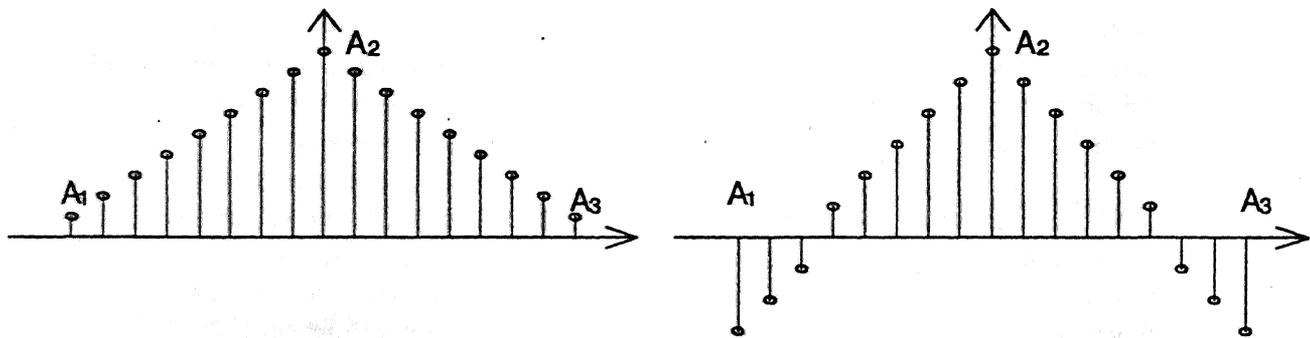


Figure 4-2: Two Possible Symmetric 1-D functions with 3 Alternations

Three Alternations: The third alternation must be in the center for the function to be symmetric. There are two cases (see figure 4-2): The coefficients can be all of the same sign, or of different signs. If the coefficients are all of the same sign, then the filter will have a non-zero DC response (sum of the coefficients) and will not be band-pass. If the coefficients are of both signs and sum to zero, then the function can be band pass. However, if it is band-pass, the negative side-lobes will be monotonically decreasing. This results in sharp discontinuities at the boundaries. These discontinuities cause large ripples in the high-frequency response which makes the function unsuitable for use with re-sampling.

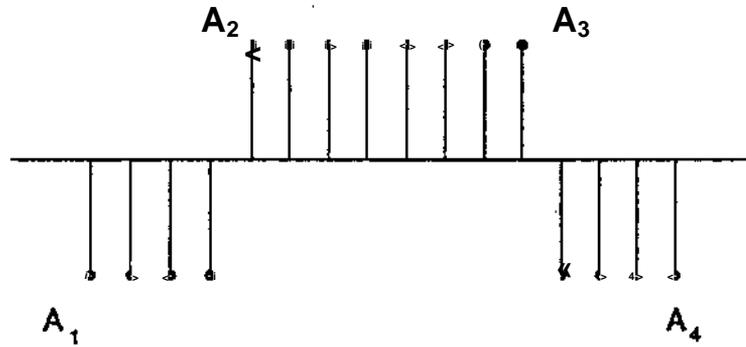


Figure 4-3: A Symmetric 1-D Band-Pass Function with 4 Alternations

Four Alternations: If the function is finite, then two alternations are at the support boundaries. The remaining two alternations must be placed symmetrically for the function to be symmetric. Since there can be no alternation at the origin, in order to be symmetric the function must be constant between the two inner alternations. In order for our function to be band-pass, its coefficients must sum to zero. The function shown in Figure 4-3 is such a function. This particular function is the difference of two constant windows. For 2-D images, convolution with this function can be implemented as a difference of square uniform windows, for which there is a fast convolution algorithm [Price 76]. However, the sharp transitions cause large ripples in the stop band which can cause aliasing when used with re-sampling.

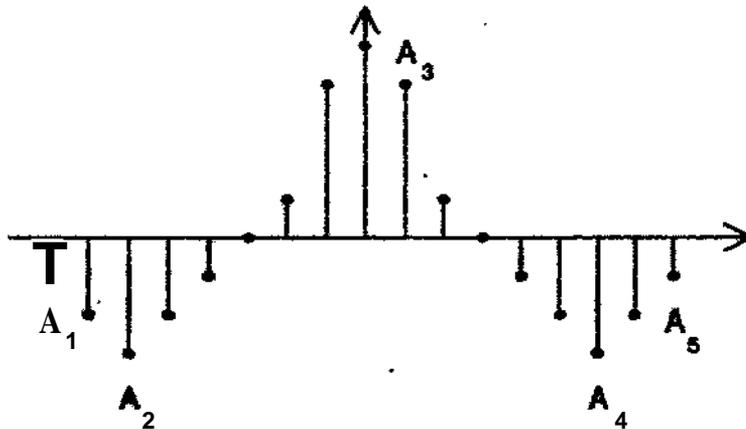


Figure 4-4: A Symmetric 1-D Band-Pass Function with 5 Alternations

Five Alternations*: [See figure 4-4] the alternation!; is the minimum which a symmetric band pass

function with a well behaved stop-band can have. This is one of the constraints which is used in the detection function design. Note that the coefficients must sum to zero in order for the function to have a zero DC response. Note also that the coefficients must taper to zero at the boundaries in order for the stop-band ripples to be small.

4.2.6 Monotonic Pass Band with a Single Peak in the Transfer Function

The constraint of five alternations in the detection function coefficients severely limits the form of the transfer function. In particular, it limits the flatness of the pass band and the width of the transition region.

The ideal situation would be to have a family of filters in which the peak frequencies give a continuum. However, this would require an infinite set of convolutions, and so we are forced to choose a finite set of filters, with the peaks staggered throughout the frequency domain. This is, in effect, sampling in frequency. For detection functions which are size scaled copies of a closed form expression, the peak frequency for a given family of detection functions may be determined by the radius of the function. For reasons explained below, we end up constraining the filter radii to be members of an exponential sequence:

$$R \in \{R_0, R_0 S, R_0 S^2, \dots, R_0 S^k\}$$

This gives an a sequence of pass bands whose center frequencies are an exponential sequence of the form $\omega \sim S^k$.

Let us define a 3 space, (x,y,k) , such that each point contains the value of the inner product of the filter of radius $R_0 S^k$ with the image neighborhood centered at x,y . Furthermore, let us specify that for each increment in k , the points in the image are resampled so that the minimum distance between samples will increase by a scale factor, S . A representation can be constructed by detecting peak and ridge points in this three space and linking them together to form a graph. In order for the structure of this graph to be invariant to the size of a grey-scale form we must constrain the transfer function of the filters to rise monotonically to a peak and then fall monotonically as spatial frequency increases. To see why this is so, consider the following situation.

Suppose we have a test pattern which is a uniform intensity square. It will result in a distinct interconnection of peak and ridge points. An example of such a graph is shown as figure 7-21 in chapter 7. A uniform intensity rectangle with an aspect ratio between 2 and 1/2 will result in a peak at the top of this graph whose value is significantly larger than any other peak in the graph. This peak is labeled as an M^* and forms the root of the graph which describes the square. It should be possible to determine the size of the square from the level, k , at which this root peak occurs.

If the test pattern is gradually increased in size the graph which represents it must move upward (in the k dimension). This movement must be monotonic with size in order for the size invariance of the description to hold. As a sufficient condition for this movement in the k direction to be monotonic we make the following constraint on the transfer function of the detection functions.

Transfer Function Constraint

The transfer function must rise monotonically from a response of zero at DC to a peak response at some frequency. It must then fall monotonically until it has entered the stop band. Within the stop band it is permitted to ripple with a magnitude less than or equal to some value δ .

This constraint is illustrated by figure 4-5.

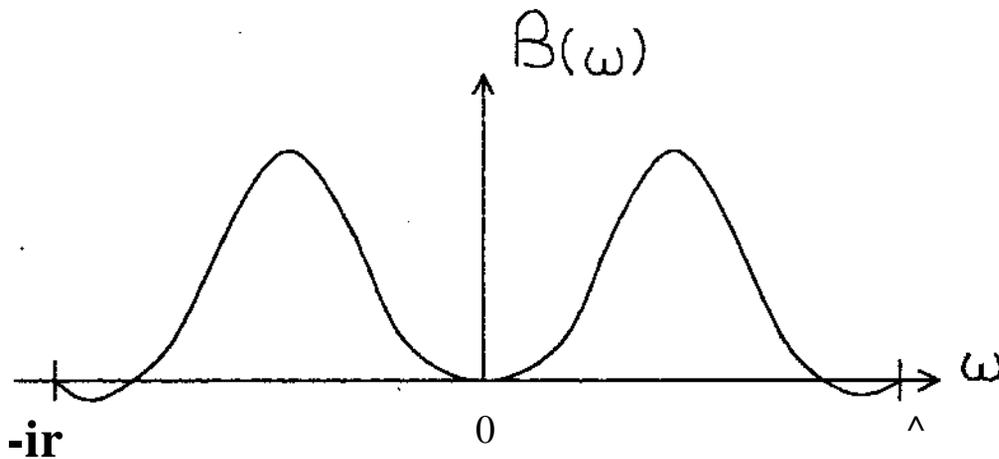


Figure 4-5: MOEIODIC Pass Band with Single Peak

4.3 Selecting the Sequence of Radii and Re-Sample Distances

In this section we will address the problem of choosing the sequence of radii which the family of detection functions should have. We also address the problem of choosing the set of re-sampling distances, like two problems are intimately related because the representation can only be **quasi-size** invariant if the re-sample distance is the same fraction of the filter radius for all of the filters.

4*3*1 Filter Radius

Scaling the size of a gray scale form k a multiplicative operation. That is if a filter is scaled in size by some factor, F , all of its dimensions are multiplied by F . The ideal situation would be to have a sequence of radii and re-sampling distances which includes all possible scaling factors. This is

impossible, because the set of such factors that can occur is infinite. It is the set of real numbers, which even over a closed interval is infinite. Thus we must choose a sequence which gives a reasonable approximation.

Suppose there are two instances of a form such that the second is a copy of the first scaled in size by F . For size invariance, we require that the representation of both forms be composed of the same interconnection of symbols, albeit from different size detection functions. Each structural component of the form must be shifted in the size dimension (k in our earlier discussion) by the same amount. Also the sampling distance (measured in terms of pixels in the original image) must be scaled by the same amount as the filter radius. That is, a configuration of peak and ridge points from the filters of radius 8 must correspond to a configuration of peak and ridge points at radius $8F$ in the second image. Similarly, a configuration from radius 4 in the first image must match a configuration at $4F$ in the second.

if we employed a non-exponential sequence such as the fibonacci sequence, $s_{i+1} = s_i + s_{i-1}$ or the set of integers, the number of detection functions between radius 8 and radius $8F$ would be different from the number of functions between radius 4 and radius $4F$. As a consequence, the representation of the scaled form would not contain the same configuration of symbols as the original. An exponential sequence allows us to approximate the scale change, F_4 by some factor of the form S^k , where S is the base scale factor, and k is an index. Scaling by S^k then shifts all configurations of peak and ridges by k levels in the representation, thus preserving the interconnection of the symbols in the representation. **It is also necessary** to have re-sampled the image by the same factor, S^k , so that the density of symbols is the same.

4.3.2 Re-Sampling Distances

The accuracy of the size invariance is determined by how closely the change in scale, F , can be approximated by S^k . If not constrained by sampling, the value of S would provide a trade off between the accuracy of the size invariance and the cost in terms of computation and storage. However, S is constrained by the requirement that the sample distance be a fixed proportion of the filter radius. There is only a small finite set of re-sampling distances that can be used without interpolating the image sample points. If we are to avoid the great increase in processing cost which would come from interpolation we must use one of the naturally occurring sample distances as the scale factor, S . The set of distances to neighboring points for a cartesian grid is shown in figure 4-6. Each number in this figure is the cartesian distance to the point on the lower left of the figure.

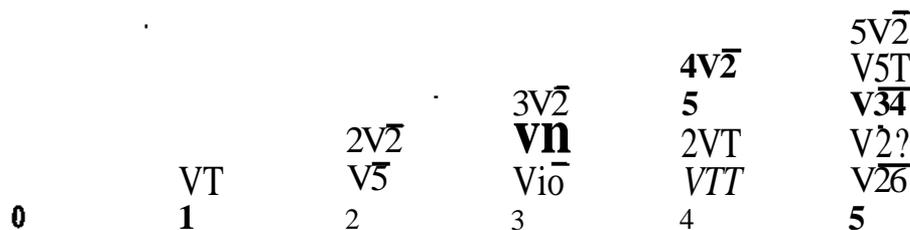


Figure 4-6: The Set of Naturally Occurring Sample Distances
For a Cartesian Plane

Let us define the set of distances between points on any grid as the set of "natural re-sample distances". Within this set we can choose subsets which are members of exponential sequences, i.e. have the form S^k . In fact, each natural re-sample distance provides the base, S , for such a subset

In the following chapters we will define a process in which the image is repeatedly filtered and then re-sampled at some base distance, S . The smallest such S which naturally occurs on a cartesian grid (greater than 1, of course) is the value $\sqrt{2}$. This is the base value which is used for scaling both the re-sampling distance and the filter size.

In summary for reasons of size invariance a family of detection functions whose radii are an exponential sequence must be used to filter the image. The set of re-sample distances must also be from the same exponential sequence, although smaller by a constant fraction. A great savings in computational cost is possible if the base number of the exponential sequence is a natural re-sample distance. Thus the experimental implementation is constructed using the smallest such resample distance for a cartesian grid, $\sqrt{2}$.

Chapter 5

A Reversible DOLP Transform Which Resolves Non-Periodic Data into Short-term Frequency Components

This chapter introduces the Difference of Low Pass (DOLP) transform which is designed to separate a signal *into* short-term frequency components. This transform was devised to be used with peak detection to represent non-periodic 2-D signals as a first step in stereo matching or determining object identity. The DOLP transform is reversible and thus preserves the information in a signal.

The DOLP transform is defined in the first section of this chapter so that the reader is aware of the motivation for the problems addressed in later sections. After the transform has been defined and its reversibility demonstrated* the form of the band-pass impulse response that results at many sizes will be described. The computational requirements of the DOLP transform will then be examined. The DOLP transform is shown to require $O(N^2)$ multiplies for an N point signal of one or two dimensions md produces $O(N \log(N))$ result data points. It is then shown that the DOLP transform can be computed using resampling with a reduction to $O(N \log(N))$ multiplies and $O(N)$ result data points. This is followed by a discussion of the degradations in frequency and position resolution that result from such resampling* Chapter 6 will present the sampled Difference of Gaussian (DOG) transform, a two dimensional implementation of the DOLP transform that exploits a property of Gaussian functions to produce a form of sampled DOLP transform in $O(n)$ computations.

Notation:

The set of symbols which are defined below are used extensively in the next two chapters* Filters have an *index* variable, L . The filter's radius is determined by the product of the smallest radius* R_m multiplied by a scale factor, S , raised to the k^{th} power. Thus the radius of the k^{th} filter R_k is given by

$$R_k = R_m S^k$$

The L filter with which the filter with which the k^{th} filter is compared is referred to as the k^{th} filter.

The L filter is defined by the k^{th} filter to L filter. The L filter is defined by the k^{th} filter to L filter. The L filter is defined by the k^{th} filter to L filter. This simplification also illustrates the point that this transform is not specific to signals of a particular dimensionality.

Table of Contents

Acknowledgements	1
Abstract	2
1. Introduction	4
1.1 The Problem Context: Machine Vision	4
1.1.1 Role of Representation in 2-D Visual Domains	6
1.1.2 Representation in 3-D Visual Domains	7
1.2 Thesis Summary and Background	7
1.2.1 The Thesis	7
1.2.2 Demonstrating the Properties of the Representation	10
1.2.3 Research Methodology	11
1.3 Results	12
1.3.1 The Representation	12
1.3.2 Measurement Technique	14
1.3.3 Computational Techniques	15
1.4 Organization of this Dissertation	15
2. Background: Related Techniques	17
2.1 Boundaries vs. Regions	17
2.1.1 Measurement Problems	18
2.1.2 Representation Problems	19
2.2 Edge Detection Techniques for Boundary Representation	19
2.2.1 Local Edge Detectors	20
2.2.2 The Hueckel Edge and Bar Detector	21
2.2.3 Kelly's Use of Planning	22
2.2.4 Cones and Pyramids	23
2.2.4.1 Uhr's Recognition Cones	23
2.2.4.2 Hanson and Riseman's Preprocessing Cones	23
2.2.4.3 Pyramid Data Structures	23
2.2.5 Other Work with Multiple Resolution Representations	24
2.2.6 Marr's Laplacian of Gaussians	25
2.3 Representation Techniques	25
2.3.1 Blum's Medial Axis Transform	25
2.3.2 Marr's Three Levels	26
3. Signal Processing Background	28
3.1 Convolution, Correlation, and Inner Products	28
3.1.1 Convolution	28
3.1.2 Correlation	29

3.1.3 Inner Products	29
3.1.4 Boundary Values	30
3.2 The Transfer Function	30
3.2.1 Eigenfunctions	31
3.2.2 Derivation of the Transfer Function	31
3.3 Two Dimensional Re-Sampling	32
3.3.1 Re-Sampling a One Dimensional Filtered Sequence	32
3.3.2 Two-Dimensional Nyquist Boundary	32
3.4 Design Parameters for Digital Filters	34
4. Criteria for the Design of Band-Pass Filters for Detecting Non-periodic Signals	36
4.1 Family of Detection Functions	36
4.1.1 Detection Functions	37
4.1.2 A Family of Detection Functions Which Provide Spatial Frequency Channels	37
4.1.3 The Goal of Size Invariance	38
4.2 Linear Functions for Describing Non-Periodic Signals with Peak and Ridge Detection	39
4.2.1 Zero Phase	39
4.2.2 Finite Impulse Response	39
4.2.3 Circular Symmetric	40
4.2.4 Band Pass	40
4.2.5 Constraining Alternation (Peaks) in the Space Domain Coefficients	40
4.2.6 Monotonic Pass Band with a Single Peak in the Transfer Function	43
4.3 Selecting the Sequence of Radii and Re-Sample Distances	44
4.3.1 Filter Radius	44
4.3.2 Re-Sampling Distances	45
5. A Reversible DOLP Transform Which Resolves Non-Periodic Data into Short-term Frequency Components	47
5.1 The DOLP Transform	48
5.2 The DOLP Transform Parameters	52
5.3 Complexity of the DOLP Transform	53
5.3.1 Number of Coefficients for Each Filter	53
5.3.1.1 One Dimensional DOLP Transform	53
5.3.1.2 Two Dimensional DOLP Transform	54
5.3.2 Computational Complexity	54
5.4 The Form of the Band-Pass Filters	56
5.4.1 Space Domain Constraints	56
5.4.2 Transfer Function Constraints	57
5.5 The Re-Sampled DOLP Transform	59
5.5.1 Re-Sampling	60
5.5.2 Complexity of the Sampled DOLP Transform	60
5.5.3 The Effects of Re-sampling on the Representation	62
5.5.4 Sampling in Frequency	63
6. The Sampled Difference of Gaussian Transform	65
6.1 Gaussian Functions	66
6.1.1 Scaling by Auto-Convolution	67
6.1.2 Discrete Gaussian Filter	68
6.1.3 Two Dimensional Digital Gaussian Filter	68

Let us start with the definitions:

$p(x,y)$: The input signal defined for $0 < x \leq N$, $0 < y \leq M$. In all the examples below $N = M$.

$g_k(x,y)$: A finite low-pass filter of radius R_k , which has been normalized so that the sum of its coefficients is 1.0. For a 1-D filter, radius is the half width.

R_o : The radius of the smallest filter with a useful frequency response, $g_o(x,y)$.

S : A Scaling Factor; typically $\sqrt{2}$ or 2.

$L_k(x,y)$: low-pass signal at level k .

$B_k(x,y)$: band-pass signal at level k .

$b_k(x,y)$: The band-pass impulse response (filter) of radius R_k .

X_k : The number of coefficients in the k^{th} band-pass filter.

K : The level at which the size of $b_K(x,y)$ exceeds the size of $p(x,y)$. ($X_K \geq N^2$ for two dimensions)

Size Scaling:

The DOLP transform is based on a set of filters which are size scaled copies of a discrete function. For purposes of the following discussion, assume that the low-pass filter is defined by a continuous function that has infinite duration and approaches zero asymptotically. Furthermore, assume that this function is sampled over a fixed interval of its range. Thus the radius of each scaled copy, R_k , actually defines the number of discrete samples which are obtained over the finite interval. This permits us to discuss the scale of a filter in terms of the filters' radius.

5.1 The DOLP Transform

This section defines the DOLP transform. The DOLP transform separates a signal into a set of band-pass components with exponentially spaced center frequencies. These band-pass components may be formed by convolving the signal with a set of band-pass filters which are size scaled copies of a single prototype filter. These filters are all formed by subtracting a low-pass filter from a copy of itself which is smaller in size by a factor of S .

The operations of convolution and subtraction are commutative. Because each band-pass filter is a difference of two low-pass filters, there are two obvious equivalent methods for computing a DOLP transform:

1. **(The Direct Method)** Form the set of band-pass filters by subtracting each pair of low-pass filters, and then convolve each of these band-pass filters with the signal. This method

is illustrated in figure 5-1 below. If reversibility is desired the signal must also be convolved with the largest low-pass filter.

- (The Difference Method) Convolve the signal with each low-pass filter, and then subtract each low-pass filtered signal from the low-pass signal formed from the next larger low-pass filter. This technique is illustrated in figure 5-2.

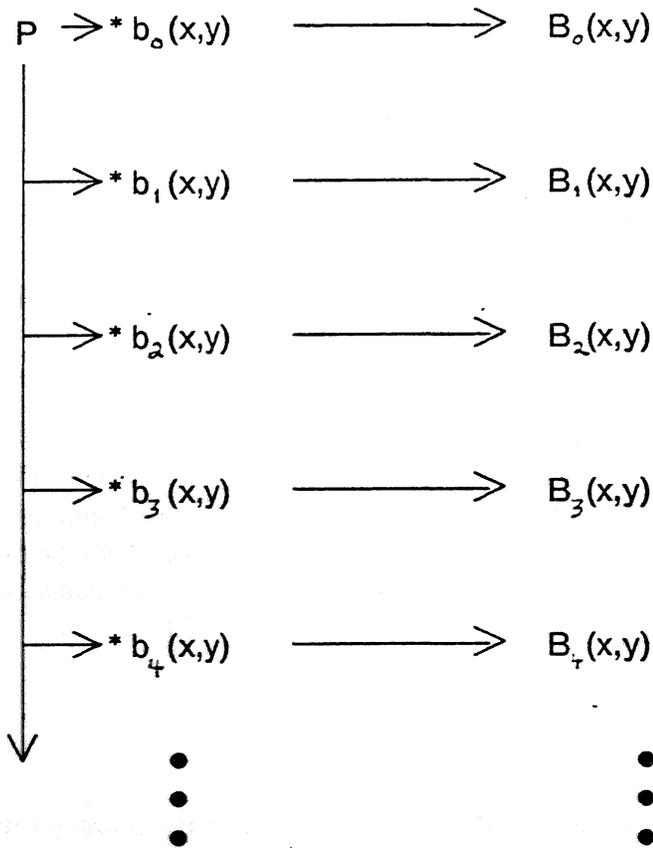


Figure 5-1: Direct Method for Computing a DOLP Transform

The direct method is the simplest to describe. For the DOLP transform as described in this section it is also the most efficient to compute, as it avoids the subtraction step required by the difference method. With the difference method, however, it is easier to illustrate the reversibility of the DOLP transform. Furthermore, in the next section we describe a fast algorithm for computing the convolution with the sequence of low-pass signals. The following is a definition "by construction" of the DOLP transform. For each level, we define the band-pass filter, describe the direct method, and then define the difference method. Reversibility is shown at each level using the low-pass signals.

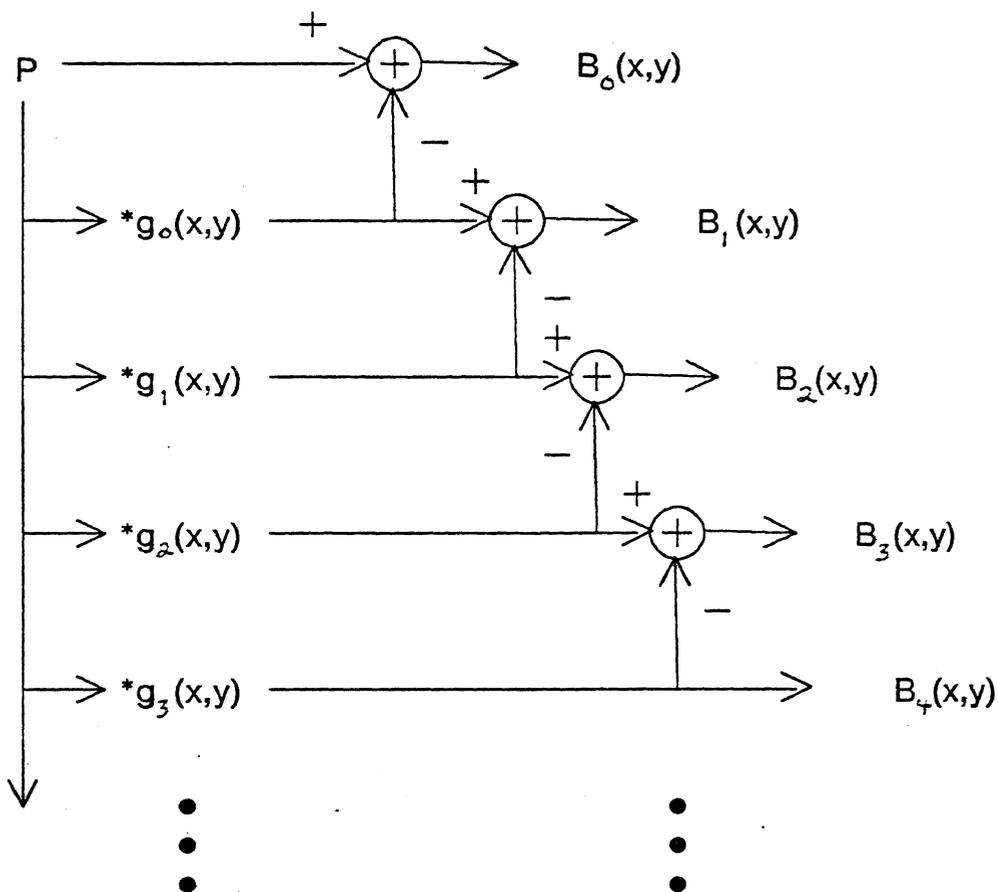


Figure 5-2: Difference Method for Computing a DOLP Transform

Level 0

The impulse response (coefficient array) for the level 0 low pass filter is g_0 by definition. The level 0 band pass filter, b_0 , has an impulse response of

$$b_0 = 1 - g_0$$

The level 0 band-pass signal, \mathfrak{B}_0 , also known as the high-pass residue, is computed by the convolution⁷

$$\mathfrak{B}_0 = p * b_0$$

With the difference method, the level 0 low-pass signal, \mathcal{L}_0 , is computed by

⁷In this and all subsequent convolutions we assume that some boundary value is supplied so that every \mathcal{L}_k and \mathfrak{B}_k will have the same duration as p .

$$\mathcal{L}_0 \triangleq p * g_0$$

The level 0 band-pass signal, \mathcal{B}_0 , is then formed by the subtraction

$$\mathcal{B}_0 \triangleq p - \mathcal{L}_0 = p - (p * g_0) = (1 - g_0) * p$$

Note that p may be recovered from \mathcal{B}_0 and \mathcal{L}_0 by

$$p = \mathcal{B}_0 + \mathcal{L}_0 = p - (p * g_0) + (p * g_0)$$

Some readers may note that for two dimensional signals, the operation producing the high pass residue is known as unsharp masking, and is sometimes used for edge detection.

Level 1

The level 1 low-pass signal, \mathcal{L}_1 , is obtained by convolving low-pass filter g_1 with p . The low-pass filter g_1 is defined as a copy of filter g_0 , scaled larger in size by a factor of S .

The impulse response for the level 1 band-pass filter, b_1 , is

$$b_1 = g_0 - g_1$$

In the direct method, the level 1 band pass signal, \mathcal{B}_1 , is formed by the convolution

$$\mathcal{B}_1 = p * b_1$$

The difference method requires computing the level 1 low-pass signal, \mathcal{L}_1 .

$$\mathcal{L}_1 \triangleq p * g_1$$

The level 1 band-pass signal may then be formed by subtracting the level 1 low-pass signal from the level 0 low-pass signal.

$$\mathcal{B}_1 \triangleq \mathcal{L}_0 - \mathcal{L}_1$$

Note that the original signal may still be recovered by

$$\begin{aligned} p &= \mathcal{B}_0 + \mathcal{B}_1 + \mathcal{L}_1 \\ &= p - (p * g_0) + (p * g_0) - (p * g_1) + (p * g_1) \end{aligned}$$

Levels 2 Through K

The low-pass filter at any level, k , is a copy of the level 0 low pass filter, g_0 , scaled larger by a factor of $\sqrt{2}^k$. As with level 1, the band-pass filter for level k is the difference of two low-pass filters

$$b_k = g_{k-1} - g_k$$

Thus for any level, k , the band-pass signal, \mathcal{B}_k , may be computed by

$$\mathcal{B}_k = p * b_k$$

With the difference method, low-pass and band-pass signals at level k may be formed by

$$\mathcal{L}_k = p * g_k \quad (5.1)$$

and

$$\mathcal{B}_k = \mathcal{L}_{k-1} - \mathcal{L}_k \quad (5.2)$$

As with level 1, for any K the original signal may be recovered by

$$p = \mathcal{L}_K + \sum_{k=0}^K \mathcal{B}_k \quad (5.3)$$

At some level (value of k) the size of the low-pass filter will exceed the size of the finite signal. Beyond this value of k the band-pass signals contain no new information about the signal. This level, K , is thus chosen as the level at which the transform is halted. Thus the DOLP transform produces:

\mathcal{B}_0 : The high pass residue.

\mathcal{B}_k for $1 \leq k \leq K$: The band-pass signals

and

\mathcal{L}_K : A low-pass residue.

Reversibility proves that no information is lost by the DOLP transform.

5.2 The DOLP Transform Parameters

Implementation of this transform requires choosing:

$g(x,y)$: The low-pass filter and its parameters

R_0 : The radius for the smallest filter, $g_0(x,y)$; and

S : The scale factor.

The low-pass filter $g(x,y)$ and its initial radius R_0 must be chosen with regard to how well the band-pass filters, $b_k \triangleq g_{k-1} - g_k$ meet the requirements for describing non-periodic signals, described in chapter 4. If re-sampling is used in the DOLP transform, the low pass filter and its parameters must also be chosen so that a minimum of aliasing results from the re-sampling. This generally involves trading off transition width (ΔF) and stop band ripple (δ) against processing time.

The scale factor, S , governs the bandwidth of $b_k(x,y)$ and the frequency resolution of the

transform. Since maximizing the frequency resolution also minimizes the degradations to the size invariance (see section 4.3), the choice of S governs the trade-off between degradations to size invariance and the cost in terms of processing steps and memory. However, if re-sampling is used, S must be one of the naturally occurring re-sample distances on the original sample grid, as was described in section 4.3.

5.3 Complexity of the DOLP Transform

In this section we examine the computational complexity of computing a DOLP transform with the direct method. This analysis shows that the direct method requires $2N^2$ multiplies and adds to produce the $N \log_s(N/X_0) + N$ samples in the DOLP transform.

The DOLP transform is based on a set of size scaled copies of a low-pass filter, $g_k(x)$ (or in the 2-D case $g_k(x,y)$). The scaling relationship between the filters is defined by an exponential relationship for the radii, R^k .

$$R_k = R_0 S^k \quad (5.4)$$

where R_0 is the radius of the smallest low-pass filter. This relationship may also be expressed recursively as:

$$R_k = R_{k-1} S \quad (5.5)$$

The band-pass filters, $b_k(x)$ or $b_k(x,y)$, are defined by the difference of two low pass filters.

$$b_k(x) = g^{k+1}(x) - g^k(x) \quad \text{for } k \in \{0, 1, 2, \dots, K\}$$

where $g_0(x) = 1$

Thus the radius for each band-pass filter is given by equation (5.4) or equation (5.5).

5.3.1 Number of Coefficients for Each Filter

As the first step of complexity analysis, let us examine the number of coefficients in the band-pass filters used in a 1-D DOLP transform and in a 2-D DOLP transform.

5.3.1.1 One Dimensional DOLP Transform

Let S_1 be the scale factor used in a 1-D DOLP transform. A typical value for S_1 would be 2. The number of coefficients, X_k , for the k^{th} bandpass filter is given by:

$$X_k = 2R_k + 1 \quad (5.6)$$

By substituting equation (5.4) into equation (5.6) we get the exponential relationship:

$$X_k = 2R_0 S_1^k + 1 \quad (5.7)$$

This sequence can be solved to arrive at the relationship:

$$X_k = (X_o - 1) S_1^k + 1 \quad (5.8)$$

For all k such that $S_1^k > X_o$ we can simplify the mathematics by replacing equation (5.8) with the approximation:

$$X_k \approx X_o S_1^k \quad (5.9)$$

5.3.1.2 Two Dimensional DOLP Transform

Let us denote the scale factor for a two dimensional DOLP transform by S_2 . When resampling is used a typical value is $S_2 = \sqrt{2}$ (See section 4.3).

As with the 1-D filters, the 2-D filters are defined to have the relationship between radii given by equations (5.4) and (5.5).

The 2-D band-pass filter, $b_k(x,y)$, is defined to have non-zero coefficients over the disc:

$$x^2 + y^2 \leq R_k^2$$

This disc is bounded by a square of sides $2 R_k + 1$. The number of non-zero coefficients, X_k , may be approximated by

$$X_k = \pi R_k^2 \quad (5.10)$$

Plugging equation (5.4) into equation (5.10) gives:

$$X_k = \pi R_o^2 S_2^{2k} \quad (5.11)$$

This can be solved to yield:

$$X_k = X_o S_2^{2k} \quad (5.12)$$

Thus for each increment in k , the number of coefficients of the filter increases by a factor of S_1 for a one dimensional filter or a factor of S_2^2 for a two dimensional filter.

5.3.2 Computational Complexity

This analysis of computational complexity and memory requirements applies to both the 1-D and 2-D DOLP transforms. In the 1-d case, let:

$$S = S_1 \text{ and } X_o = 2 R_o + 1$$

For the 2-D case let:

$$S = S_2^2 \text{ and } X_o = \pi R_o^2$$

Assume that we have a signal with N samples, (1-D or 2-D) and that one convolution inner-product step is to be computed for the filter centered over each of the N samples. This assumes that a default boundary value is supplied when the filter coefficients fall over the edge of the signal. Thus

each convolution produces N sample values as its result. Also, assume that the smallest low pass filter with a reasonable stop band has X_0 coefficients.

The first filter, which produces the level 0 or high pass residue has X_0 coefficients. Thus there are N inner product steps, with each requiring X_0 multiplies, for a total of $X_0 N$ multiplies.

For each level, k , from 0 through K , the filter has: $X_0 S^k$ coefficients. Thus the total number of multiplies, denoted C (for cost), is given by:

$$\begin{aligned} C &= X_0 N (1 + S + S^2 + \dots + S^K) \\ &= X_0 N \left(\sum_{k=0}^K S^k \right) \\ &= X_0 N (S^{K+1} - 1) / (S - 1) \end{aligned}$$

For the typical values of $S_1 = 2$ and $S_2 = \sqrt{2}$, S will have a value of 2.

For $S=2$, we can make the approximation:

$$\frac{S^{K+1} - 1}{S - 1} \approx S^{K+1}$$

Thus our cost becomes:

$$C \approx X_0 N S^{K+1} \quad (5.13)$$

The largest filter in this sequence has an index, K , chosen such that it is the smallest integer for which:

$$X_0 S^K \geq N$$

Plugging this into our cost formula for $S=2$ gives:

$$C \approx S N^2$$

Since there are $K+1$ filters and each filter produces N sample values, the total memory requirement, M , is:

$$M = (K + 1) N$$

Since $X_0 S^K \approx N$ then the number of levels, K , is:

$$K \approx \text{Log}_S(N/X_0)$$

Thus our total memory cost is:

$$M \approx N \text{Log}_S(N/X_0) + N \quad (5.14)$$

5.4 The Form of the Band-Pass Filters

Section 5.1.1 described forming band-pass signals by subtraction of two low-pass signals. Because convolution and subtraction are both linear operations, they are associative. Thus in the case of the band-pass images:

$$(p * g_{k-1}) - (p * g_k) = p * (g_{k-1} - g_k)$$

Thus the DOLP transform may be computed as either a difference of low pass images as described above, or by precomputing the coefficients of each band-pass filter and then convolving each band-pass filter with the signal. In fact, the latter process saves the subtraction step, and so is less expensive. However in chapter 6 we describe a fast version of the DOLP transform in which the computational complexity is reduced by using each low pass signal L^k to produce the next low pass signal L_{k+1} .

In chapter 7 a description technique which uses peak detection will be described. The use of peak detection for describing band-pass signals requires a constraint on the smoothness of the band-pass impulse response (as described in section 4.2) as well as on its transfer function. In this section we show how the low-pass filter employed by the DOLP transform must be constrained to produce a band-pass filter which meets the constraints described in section 4.2.

This discussion is illustrated with one dimensional filters: $b(x)$ and $g(x)$. For two dimensions, the filters should be circularly symmetric, so that response is not dependent on orientation. The variable x may then be replaced by a radial distance to the center, r , at any orientation. The transfer functions of the filters are denoted as:

$$B(\omega) \triangleq \mathcal{F}\{b(x)\} \quad \text{and}$$

$$G(\omega) \triangleq \mathcal{F}\{g(x)\}.$$

5.4.1 Space Domain Constraints

The smoothness of the band-pass impulse response is obtained by constraining the low-pass impulse response to three alternations, or changes in sign of its first difference*. The reasons for this constraint are described in section 4.2.5. These alternations should occur only at the boundaries of the low-pass impulse response and at its center as shown in the following figure.

The band-pass impulse response*

$$b_{k+1}(x) \triangleq g_k(x) - g_{k+1}(x)$$

which has a radius of $R_{k+1} = R^k S = R_0 S^{k+1}$, will then have 5 alternations as shown below. Two of these are at the outer edges* $x \approx R^k$ labeled A_1^* and A_5 . Two alternations, A_2 and A_4 , will be at approximately $x = J_{k+1}$ where the first difference

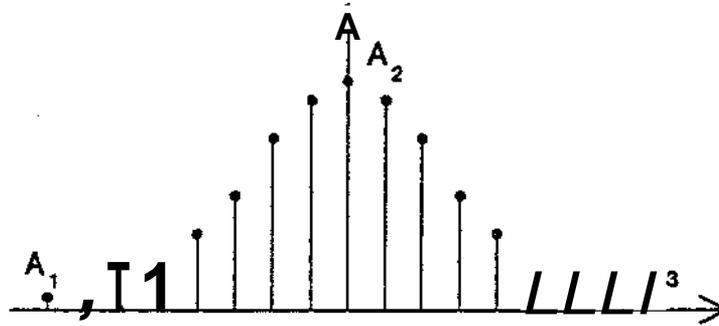


Figure 5-3: Permissible Alternations in Low-pass Filter

$$g_k(x_p) - g_k(x_l - 1)$$

first becomes larger than

$$g_{k+1}(x_p) - g_{k+1}(x_l - 1)$$

and of course, one at the center, A_2 , where $x=0$.

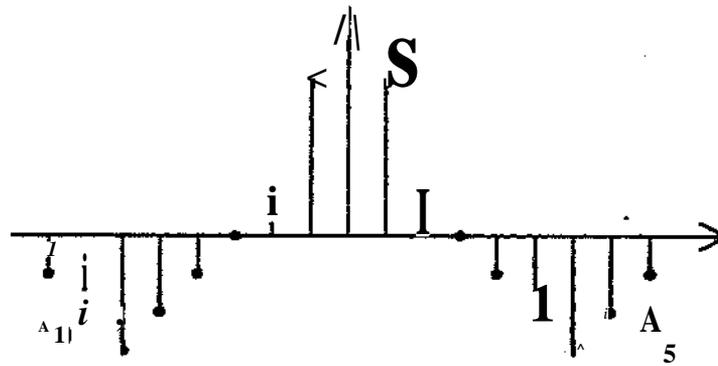


figure 5-4: Permissible Alternations in Band-pass Filter

5.4.2 Transfer function Constraints

The size invariance of the final description requires that as a gray scale form (or signal) increases its size, the position of the signals in the transform move up through the low's $\Delta mmMy$. This requires that the pass region of the transfer function of the band-pass filter has a Δ and be monotonic on either side of that peak.

Both low-pass filters are normalized so that they have a gain of 1.0 at DC ($\omega=0$). Since subtraction and the transfer function are both linear operations, they are associative. That is:

$$\mathcal{F}\{h\} - \mathcal{F}\{g\} = \mathcal{F}\{h - g\}$$

Thus the difference of such normalized filters will have a DC response of 0. This will guarantee that there is no reponse by a filter when it covers a region which is entirely uniform. Both low-pass filters should have a single peak at DC and monotonically falling pass and transition regions, as shown below in figure 5-5.

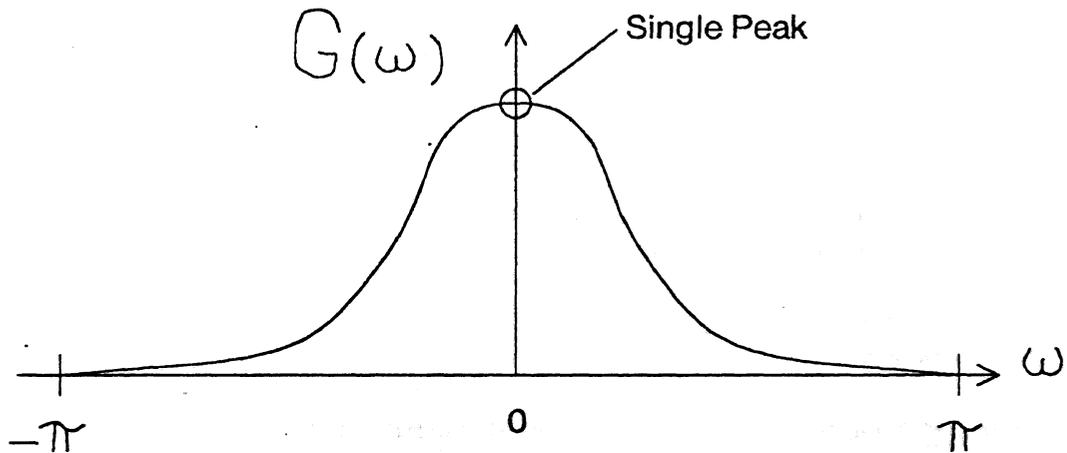


Figure 5-5: Transfer Function $G(\omega)$

This will guarantee that the low-frequency side of the band-pass filter transfer-function pass band is monotonically increasing. The peak frequency of the pass band, ω_o , will occur somewhere before the negative minimum of the first ripple of the larger low-pass filter's transfer function. It occurs at this minimum for large values of S ($S > 2$) and at lower frequencies for smaller S . Since this should be the first alternation in either low-pass transfer function (after the DC alternation) there should be no problem maintaining monotonically increasing response on the low frequency side of the peak frequency.

A local peak will occur in $B_{k+1}(\omega)$ for each interval in which

$$\frac{\partial G_{k+1}(\omega)}{\partial \omega} > \frac{\partial G_k(\omega)}{\partial \omega}$$

This is the source of the peak response of $B_{k+1}(\omega)$ at ω_o . However such a peak must not be permitted any where else in the pass or transition regions of $B_{k+1}(\omega)$. Otherwise, the size invariance of the description will be corrupted as a result of the filter having more than one peak response as the size of an object increases. The regions where this could happen are where the ripples in $G_{k+1}(\omega)$ go through a zero crossing from positive to negative. Thus we must guarantee either:

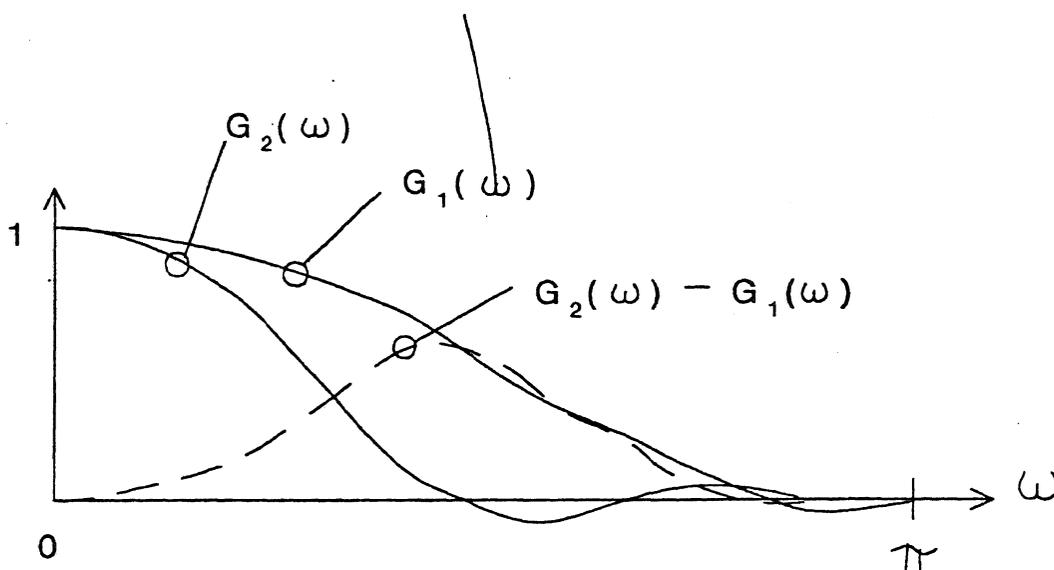


Figure 5-6: Difference of Low-Pass Transfer Functions

- That the second zero crossing from positive to negative at $G_{k+1}(\omega)$ occurs outside the transition region of $B_{k+1}(\omega)$ or,
- That the derivative $\partial G_{k+1}(\omega)/\partial\omega$ near this zero crossing is smaller than $\partial G_k(\omega)/\partial\omega$ at the same ω .

For $S \leq 2$, the first criterion is met for most low-pass filters that meet the space domain criteria. For larger values of S , if the first criterion is not met, the second may be achieved by adjusting the stop band ripple magnitude, δ .

5.5 The Re-Sampled DOLP Transform

In this section we describe the re-sampled DOLP transform. In this version of the DOLP transform the convolution "inner product steps" are computed at a set of re-sample points.⁸ The distance between these re-sample points is a fixed fraction of the filter impulse response.

In this section we show that such re-sampling cancels the growth in computational cost that occurs in the DOLP transform as a result of the exponential growth of the number of filter coefficients as k increases. This occurs because the distance between samples grows by the same scale factor as the impulse response size. The result is a form of DOLP transform which may be computed in $O(N \log_s(N))$ multiplies. We also show that the storage cost is reduced by re-sampling to $O(N)$ (For $S_2 = \sqrt{2}$, $M = 3N$).

⁸This is equivalent to resampling the filtered image that results from each convolution.

5.5.1 Re-Sampling

The family of band-pass functions employed in the DOLP transform have a high frequency stop band. For each increment in the filter index, k , the low frequency edge of the stop band moves lower in frequency by a factor S_1 for a 1-D signal or S_2 for a 2-D signal.

Because each filter has a high-frequency stop band it is possible to save a significant amount of storage and processing cost by computing each convolution at a set of resample points. That is, when computing the convolution

$$\mathfrak{B}_2(n,m) = b_2(x,y) * p(n,m)$$

the inner product step of the convolution need only be computed for the filter centered over the points along every other diagonal as shown by the boxes in figure 5-7 which is a reproduction of figure 3-1 of chapter 3. A two dimensional form of the Nyquist sampling theorem can be used to show that virtually no information is lost; The value of the convolution at the omitted sample points can be recovered by interpolation.

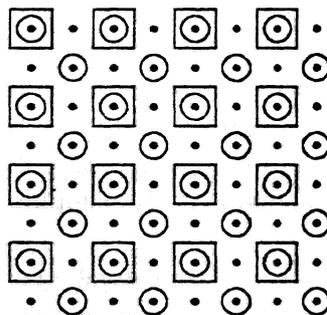


Figure 5-7: Example of $S_{\sqrt{2}}[p(x,y)]$ and $S_2[p(x,y)]$
From Figure 3-1 of Chapter 3

In addition to the savings in computational cost and storage, the re-sampling used in the DOLP transform is fundamental to the quasi-size invariance of the representation for images based on the Sampled DOLP transform described in chapter 7.

5.5.2 Complexity of the Sampled DOLP Transform

In this subsection we describe the re-sampling in the sampled DOLP transform, and derive its computational cost and memory requirements.

As before, assume that we have a one or two dimensional signal composed of N samples, and that default boundary value is provided for the case when the filter coefficients fall over the edge of the

signal Also, assume that the smallest band-pass filter has X_0 coefficients and that the filter sizes are related by a scaling factor, S , by:

$$X_k = S X_{k-1} = S^k X_0$$

As in section 5.4, this analysis of computational complexity and memory requirements applies to both the 1-D and 2-D DOLP transforms. In the 1-d case, let:

$$S = S_1 \text{ and } X_0 = 2R_0 + 1$$

For the 2-D case let

$$S = \begin{matrix} S_x \\ S_y \end{matrix} \text{ and } X_0 = w R_x R_y$$

The filter for $k = 0$, $b_0(x)$ or $b_0(x,y)$, is a high-pass filter. Convolution with this filter can not be resampled. This filter has X_0 coefficients and so requires $X_0 N$ multiplies and produces N result sample points,

The filter for $k = 1$ is a band-pass filter, its pass band is contained in the original Nyquist boundary of the signal and so its convolution with the image also cannot be resampled without causing distortion due to aliasing. This filter has SX_0 coefficients so its convolution requires $SX_0 N$ multiplies and produces N result sample points

The filter for $k = 2$ is a scaled copy of the filter for $k = 1$. Its pass-band h within a new Nyquist boundary is scaled lower frequency by a factor of S_x or S_y . The convolution of this filter with the image can be resampled at points separated by a distance of S_x or S_y . In the 2-D case, re-sampling at a distance of S reduces the number of samples by a factor of $S = S_x S_y$. There are thus N/S points at which the convolution inner product must be computed. Since this filter has $S^2 X_0$ coefficients, the convolution requires $SX_0 N$ multiplies and produces N/S sample values.

As described in section 4.3, the smallest naturally occurring resample distance for a 2-D cartesian grid is $\sqrt{2}$. Unless the signal is interpolated before the convolution, S is constrained to be one of the naturally occurring resample distances, limits in the absence of interpolation, the smallest possible S for a 2-D sampling is $\sqrt{2}$. For $S_n \ll \sqrt{2}$, this resampling consists of computing the convolution inner products with the filter centered at (m, n) along every other diagonal as shown by the squares in figure 5*5.

For $k = 3$, $S^3 X_0$ coefficients and is a copy of the filter for $k = 1$ scaled lower frequency by a factor of S_x^2 or S_y^2 . If the original image and this filter may be convolved at a distance of S_x or S_y . This allows resampled convolution results to be computed at N/S^2 locations.

At the 24th step, if $S_x = \sqrt{2}$, this requires computing an inner product convolution step at every other column of every other row.

In general, for each filter, k , its pass-band is offset by the increase in distance S^k . This is thus the

same for every band-pass filter for $k \in \{1, 2, 3, \dots, K\}$. Given that there are $K = \text{Log}_S(N/X_o)$ band-pass filters that require SX_o multiplies, and one high pass level, $k=0$, that requires X_oN multiplies, the total cost, C , of the Sampled DOLP transform is:

$$C = S X_o N \text{Log}_S(N/X_o) + X_o N$$

The number of sample points produced by each convolution decreases by a factor of S for each increment of k from $k=1$ to $k=K$. Thus the storage requirement, M , for the Sampled DOLP transform is:

$$\begin{aligned} M &= N(1 + 1 + 1/S + 1/S^2 + 1/S^3 + \dots + 1/S^K) \\ &= N\left(1 + \frac{(1 - S^{-K-1})}{(1 - S^{-1})}\right) \text{ Storage elements.} \end{aligned}$$

Note that for $S = 2$,

$$\begin{aligned} M &\approx N + N \frac{1 - 0}{1 - 1/2} \\ &\approx N(1 + 2) \\ &\approx 3N \text{ storage elements.} \end{aligned}$$

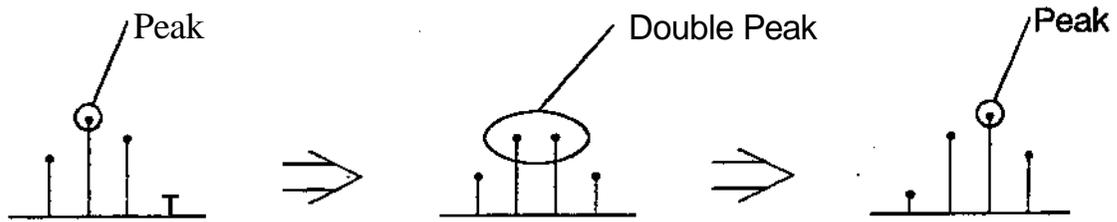
5.5.3 The Effects of Re-sampling on the Representation

As described in section 3.3, the distortion from re-sampling (and subsequent loss of information in the description) may be minimized by minimizing the signal energy outside of the nyquist boundary defined by $|u, v| \leq \pi/S_R$, where u and v are the spatial frequency variables and S_R is the distance in pixels between the new sample points. This analysis tells what information could be recovered by interpolation. However, a peak detection algorithm will be employed to describe the transform. Re-sampling introduces an uncertainty in the location of peak. That is, when a peak is detected in a re-sampled signal it may actually have occurred anywhere in the interval bounded by $(x \pm S_R, y \pm S_R)$. If the sample interval is a constant fraction of the size of the impulse response at each level then the uncertainty of a signal's position will always be the same fraction of its size. More accurate position information may be obtained from the description of the object's boundaries, which is at lower levels in the transform.

Ideally we would like the configuration of peaks that describes a signal to be invariant to the signal's position. However, as a peak moves from one sample to the next, there is a point at which two adjacent samples will have the same peak value as shown here in 5-8.

The frequency of occurrence of such double peaks is dependent on the number of bits used to represent each sample and on the signal amplitude. Double peaks occur most frequently when the signal amplitude is small.

This randomness is also present in the relative position of peaks at adjacent levels as shown in figure 5-9.



Peak Makes Discrete Jumps as Object Moves to Right

Figure 5-8: Location of Peak Sample as Signal Moves to the Right

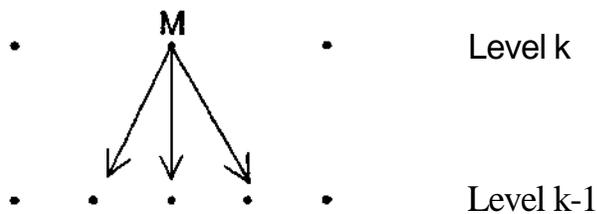


Figure 5-9: Uncertainty of Position of Peaks at Adjacent Levels

A peak could occur with equal likelihood at any of the positions directly under the higher level peak. Thus any matching rule for graphs of peaks from this transform must accept a peak at any of the three positions as a match.

5.5.4 Sampling in Frequency

Each level of the DOLP transform represents an ensemble of samples at a particular spatial frequency range. The center frequencies of the band-pass levels are at discrete, exponentially spaced intervals. The problem of choosing the step size for the center frequencies is discussed in section 4.3.

As with spatial sampling, this frequency sampling defines the resolution in frequency of the DOLP transform. This translates into the changes in the size of signals that the transform can resolve. The interval between center frequencies is given by the scale parameter, S . This parameter also defines the band width of the individual filters. The smaller S is, the better the resolution in size (frequency).

A roughly uniform region with a background of a different intensity results in a local maximum in the three space, (x, y, λ) , defined by the transform. The level at which this peak occurs gives an

estimate of the size of the region. Peak detection between levels produces an uncertainty in a signal's size which is analogous to the uncertainty in the signal's position. That is, as a signal's size increases, the level at which the largest peak occurs will make discrete jumps. In this case, the size uncertainty is bounded by the scale factor, S . That is, a peak at level k places the signal duration somewhere between

$$\frac{R_0 S^{k-1/2}}{2} \leq \text{Signal Duration} \leq \frac{R_0 S^{k+1/2}}{2}$$

The result may be compensated for in a matching rule by permitting a stretching or contraction of one of the signals by a factor limited by $S^{1/2}$ and $S^{3/2}$. The particular stretching may be determined for a given signal by observing the distance between landmarks in the description such as two peaks at some level. Such landmarks for two dimensional patterns are discussed in chapters 7 and 8.

Chapter 6

The Sampled Difference of Gaussian Transform

An Efficient DOLP Transform
Based on Gaussian Filters and ReSampling

This chapter develops an algorithm for computing the two dimensional form of the DOLP transform in $O(N)$ steps (where n is the number of picture points). This algorithm employs a property of Gaussian low-pass filters to obtain a drastic reduction in the number of computations needed to compute the sequence of low-pass images. This property is: when a Gaussian is convolved with itself the result is the same Gaussian scaled larger in standard deviation by a factor of $\sqrt{2}$.

The previous chapter defined a class of reversible transforms referred to as the DOLP transform. It described how the 2-D DOLP transform could be speeded up from $O(N^2)$ multiplies to $O(N \log N)$ multiplies, and its memory requirements reduced from $O(N \log N)$ cells to $3N$ cells by using $\sqrt{2}$ resampling. This subclass of the DOLP transform is referred to as the Sampled DOLP transform.

It is also possible to speed up the DOLP transform by using an algorithm referred to as "Cascade Convolution with Expansion". This algorithm exploits the Gaussian auto-convolution scaling property and an operation referred to as $\sqrt{2}$ expansion. The " $\sqrt{2}$ expansion" operator is a mapping of a function from a Cartesian sample grid to a $\sqrt{2}$ sample grid. Cascaded convolution with expansion reduces the computational cost of a DOLP transform from $O(N^2)$ multiplies to $O(N \log N)$ multiplies. Because this algorithm is based on properties of the Gaussian function the DOLP transform which it produces is referred to as the Difference of Gaussian (DOG) transform. Combining resampling and cascaded convolution with expansion gives a form of DOLP transform which may be computed in $O(N)$ multiplies. This transform is referred to as the Sampled Difference of Gaussian (SDOG) transform.

Chapter 7 shows how to construct a structural description of the contents of a grey-scale image by detecting and linking peaks and ridges in the SDOG transform of the image.

The Sampled Difference of Gaussian (SDOG) Transform is defined in this chapter. The Gaussian function and its use as a finite impulse response low-pass filter are examined. The computational complexity of the SDOG transform is analyzed and shown to be $O(N)$. Two approximations for

scaling the standard deviation of a finite Gaussian filter by $\sqrt{2}$ in standard deviation are introduced: The use of the auto-convolution of a finite Gaussian, and the use of an "expanded" Gaussian.

Section 6.1 describes Gaussian functions and filters and proves the the scaling property. Section 6.2 describes cascaded convolution with expansion. It then examines the effects of the expansion operation on a low-pass filter. Section 6.3 defines the Sampled DOG transform by construction, and shows that this transform requires $3X_0N$ multiplies and produces $3N$ samples for an N sample picture. Section 6.4 describes an experiment that gives the accuracy of the scaling obtained by multiple convolution with a Gaussian kernel. Section 6.5 presents the impulse responses for the level 0 and 1 band-pass filters, and the transfer functions of the level 1 and 2 band-pass filters.

6.1 Gaussian Functions

Even with re-sampling, the DOLP transform of an image is a very costly process in terms of the number of computations that are required. It is possible to reduce the computational complexity by several orders of magnitude by exploiting the properties of Gaussian filters. In this section, the Gaussian function and its properties are reviewed and the construction of 1-D and 2-D low-pass and band-pass filters using Gaussian functions is described.

The Gaussian function is most commonly known in its one dimensional form

$$g(t; \mu, \sigma) \triangleq \frac{1}{\sigma\sqrt{2\pi}} e^{-(t-\mu)^2/2\sigma^2}$$

where: $\mu \triangleq$ The mean and
 $\sigma \triangleq$ The standard deviation

The term $1/\sigma\sqrt{2\pi}$ scales the infinite Gaussian so that it has unit area.

For the discussion that follows, the mean will always occur at the origin ($t=0$), and so will be omitted from the notation. In some of the discussion values such as σ , which determine the specific function, are used as variables. In these cases these values are included within the parenthesis to simplify the notation. They are separated from the independent parameters of the function, such as x and ω , by a semicolon.

The standard deviation, σ , is the square root of the second central moment of the Gaussian function, and thus defines its width. The zero mean Gaussian

$$g(t; \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2}$$

has a Fourier transform

$$G(\omega; \sigma) = e^{-\sigma^2\omega^2/2}$$

6.1.1 Scaling by Auto-Convolution

The scaling property is easily deduced from the formula of a Gaussian function. It has been observed by statisticians, and is used in Communications theory and Linear Systems theory to describe the effect of repeated convolution. In this section it is employed to describe the effects of a finite impulse response Gaussian filter as a kernel for cascaded filtering. This scaling property is only strictly true for the infinite Gaussian function. For a finite Gaussian low-pass filter this scaling property is only an approximation. The accuracy of this approximation is examined in section 6.3.4 and 6.4.

The fast algorithm described in this chapter is based on the following property of Gaussian functions:

Gaussian Scaling Property:

A Gaussian function convolved with itself yields a Gaussian function whose standard deviation (width) is $\sqrt{2}$ larger than the original function.

Proof:

The convolution:

$$\frac{1}{\sigma\sqrt{2\pi}}e^{-t^2/2\sigma^2} * \frac{1}{\sigma\sqrt{2\pi}}e^{-t^2/2\sigma^2}$$

may also be expressed as the product of Fourier transforms

$$e^{-\sigma^2\omega^2/2} \circ e^{-\sigma^2\omega^2/2} = e^{-\sigma^2\omega^2}$$

whose inverse Fourier transform is

$$\frac{1}{\sigma 2\sqrt{\pi}}e^{-t^2/4\sigma^2}$$

To get back to standard form then requires the substitution

$$\sigma_1^2 = 2\sigma^2 \text{ or } \sigma_1 = \sqrt{2}\sigma.$$

Thus the standard deviation, and hence the function width, have been expanded by a factor of $\sqrt{2}$. \square

Note also that the amplitude has been multiplied by a factor of $1/\sqrt{2}$. Auto-convolution preserves the unit area normalization.

6.1.2 Discrete Gaussian Filter

The Gaussian function may be used as a low-pass digital filter. When used as a filter the variance σ^2 is replaced by the ratio of a shape parameter, α , to the support radius squared, R^2 . This gives a family of finite functions with different standard deviations for a particular radius. Adjusting the parameter α permits a trade-off between stop-band ripple, δ , an transition width, ΔF , for the filter. An experiment to determine the effect of α on this trade-off is described in appendix A.

The Gaussian is converted to discrete form by

1. Making the substitution $\sigma^2 = \frac{R^2}{2\alpha}$, and
2. Sampling the continuous function at $2R + 1$ points given by the discrete variable x , $|x| \leq R$.

Implicit in this form is a multiplication by a $2R + 1$ point uniform window (or aperture or support)

$$\text{Rect}_{2R+1}(x) \triangleq \begin{cases} 1 & \text{for } |x| \leq R \\ 0 & \text{otherwise.} \end{cases}$$

This gives a space domain formula.

$$g(x; \alpha, R) = \text{Rect}_{2R+1}(x) e^{-\alpha x^2 / R^2}$$

whose transfer function is

$$G(\omega; \alpha, R) = \frac{\text{Sin}(\omega(2R+1)/2)}{\text{Sin}(\omega/2)} * \sqrt{\alpha} / R \sqrt{\pi} e^{-R^2 \omega^2 / 4\alpha}$$

Where the first term in the convolution is the Fourier transform of the support

$$\mathcal{F}\{\text{Rect}_{2R+1}(x)\} = \frac{\text{Sin}(\omega(2R+1)/2)}{\text{Sin}(\omega/2)}$$

6.1.3 Two Dimensional Digital Gaussian Filter

Generalizing the Gaussian low-pass digital filter to two dimensions can be accomplished by substituting the radial formula, $x^2 + y^2$, for the distance variable x^2 . In addition, the finite support must also be generalized to two dimensions, which presents a choice. The two dimensional support may be the square

$$s(x, y; R) \triangleq \begin{cases} 1 & \text{for } |x| \leq R, |y| \leq R \\ 0 & \text{otherwise} \end{cases}$$

which is separable and has a transfer function [Oppenheim 75]

$$S(u, v; R) = \frac{\text{Sin}(u(2R+1)/2)}{\text{Sin}(u/2)} \frac{\text{Sin}(v(2R+1)/2)}{\text{Sin}(v/2)}$$

Or it may be the disc

$$c(x,y;R) \triangleq \begin{cases} 1 & \text{for } x^2+y^2 \leq R^2 \\ 0 & \text{Otherwise} \end{cases}$$

which is circularly symmetric and has a transfer function [Papoulis 68]

$$C(u,v;R) = \frac{2\pi R J_1(R\sqrt{u^2+v^2})}{\sqrt{u^2+v^2}},$$

where $J_1(\cdot)$ is the first order Bessel function.

The Gaussian is the only two-dimensional function which is both circularly symmetric and separable into one-dimensional components. This property can be used to speed up two-dimensional filtering with a Gaussian by replacing convolution with a $(2R+1) \times (2R+1)$ filter by two convolutions with $2R+1$ point one-dimensional filters (one for each dimension). This requires $4R+2$ multiplications for each picture point instead of $4R^2+4R+1$ multiplications. However, this savings can only be obtained by defining the Gaussian over a separable support, such as $s(x,y;R)$.⁹ Unfortunately, the square support focuses the stop-band ripple of the filter along the u and v axes. This gives a non-circularly symmetric transfer function and a larger worst case stop-band ripple than for the circular support. The stop-band ripple must be minimized if the filter is to be used with re-sampling in order to minimize the maximum aliasing error.

For the experiments described in this dissertation, circular symmetry and the best possible stop-band performance were judged to be more important than the computational savings. However, in a real system, it may be worthwhile to accept some degradation in order to gain a significant savings in processing speed.

The implementation described in this chapter and used for experiments in constructing a representation is based on the Gaussian filter with circular support:

$$g_o(x,y) = c(x,y;R) e^{-\alpha(x^2+y^2)/R^2}$$

Whose Transfer function is

$$G_o(u,v) = \frac{2\pi R J_1(R\sqrt{u^2+v^2})}{\sqrt{u^2+v^2}} * \left(\frac{\sqrt{\alpha}}{R\sqrt{\pi}} \right) e^{-R^2(u^2+v^2)/4\alpha}$$

In the examples given in this dissertation, the parameters $R=4.0$ and $\alpha = 4.0$ were used for the Gaussian filter. These values were obtained by an experimental procedure described below in Appendix A.

To control the filter gain, the filter coefficients are normalized so that they sum to 1.0. This is done by summing the coefficients and then dividing each coefficient by the sum.

⁹Although any uniform rectangle is a separable support, the uniform square has the least effect on the circular symmetry of the filter. Section 4.2 describes the need for circular symmetry in the filters used in a DOLP transform

The following figures show the impulse response, $g_o(x,y)$ for $R=4$, $\alpha=4.0$ and a plot of its transfer function.

				.001488				
	.003150	.006669	.008564	.006669	.003150			
.003150	.010996	.023278	.029890	.023278	.010996	.003150		
.006669	.023478	.049280	.063276	.049280	.023478	.006669		
.001488	.008564	.029890	.063276	.081248	.063276	.029890	.008564	.001488
.006669	.023478	.049280	.063276	.049280	.023478	.006669		
.003150	.010996	.023278	.029890	.023278	.010996	.003150		
	.003150	.006669	.008564	.006669	.003150			
				.001488				

Figure 6-1: Normalized Impulse Response $g_o(x,y)$ for $R=4$, $\alpha=4.0$

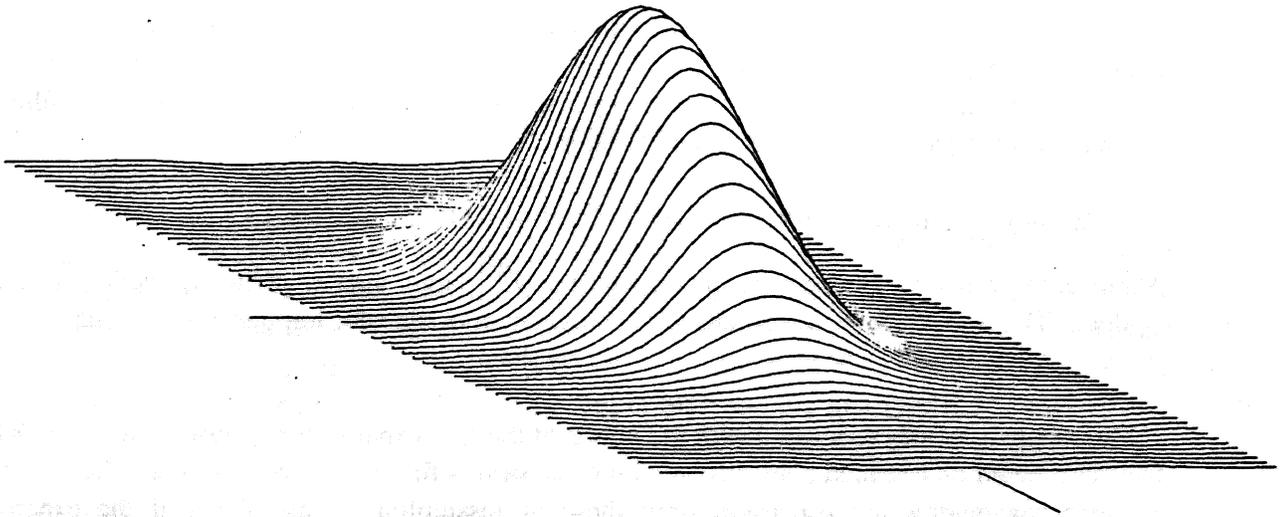


Figure 6-2: Transfer Function $G_o(u,v)$ for $R=4$, $\alpha=4$

In figure 6-2 and all other transfer function plots, the transfer function was evaluated over a 64×64 floating point array representing the Nyquist region $-\pi \leq u, v < \pi$. Because the filters have zero phase, the imaginary part of the function is identically zero. Thus only the real part is plotted. The values were scaled so that the maximum would extend full scale on the plot. Linear interpolation was used to obtain the value between sample points. The range from 0 to maximum response (1.0 for low-pass filters, ≈ 0.25 for band-pass filters) is represented by 4096 increments at 2045 dots/inch.

6.2 Cascaded Convolution with Expansion and Resampling

In this section we introduce a fast algorithm for computing the 2-D Sampled DOLP transform with Gaussian low-pass filters. This algorithm, referred to as "Cascaded Convolution with Sampling", is based on the convolution scaling property of Gaussian filters, the $\sqrt{2}$ expansion operation and resampling. In this algorithm, the image is filtered, re-sampled at $\sqrt{2}$, and then filtered again with a filter that has been expanded out to the sample grid of the re-sampled image.

In chapter 5 it was shown that a DOLP transform could be computed by 2 methods:

1. Convolution of the image signal with a sequence of size-scaled low-pass filters followed by a subtraction of each low-pass signal from the next. i. e.

$$L_k = g_k * p$$

$$B_k = L_{k-1} - L_k$$

2. Convolution with an exponentially size-scaled set of band-pass filters which are formed by subtracting size scaled low-pass filters. i. e.

$$b_k = g_{k-1} - g_k$$

$$B_k = p * b_k$$

This fast algorithm is based on the first of these two approaches. That is the computation cost is reduced by computing each L_k from L_{k-1} . As is shown below this computation may be done by convolving the filter g_0 with L_{k-1} k times, or by a single convolution with a version of the filter g_0 which has been expanded by $\sqrt{2}^{k-1}$ times. That is,

$$L_k = L_{k-1} * E_{\sqrt{2}^k}\{g_0\}$$

Although this expanded filter covers an area which is $\sqrt{2}^k$ larger than g_0 , it has X_0 coefficients just as g_0 does. Thus a set of low-pass signals with an exponential series of impulse response sizes can be formed with cost which is the same for each low-pass signal.

This section is mainly concerned with the effects of the $\sqrt{2}$ expansion operator. A form of DOLP transform based on cascaded convolution with expansion is first introduced to isolate the effects of cascaded convolution and expansion from those of resampling. The effects of the expansion operation are then examined.

The impulse response of the level 0 low-pass signal, L_0 , is $g_0(x,y)$ by definition. At level 1 the desired impulse response is $g_1(x,y)$ as described in section 5.1. The Gaussian scaling property, described in section 6.1, shows that if $g_0(x,y)$ is a Gaussian filter, the level 1 low-pass filter impulse response can be approximated by

$$g_1(x,y) = g_0(x,y) * g_0(x,y).$$

In a Sampled DOLP transform, for each level above level 1, both the impulse response and the

unit sample distance, S_R , are to be scaled in size by an additional factor of \sqrt{l} . This section describes how this sequence of low-pass signals can be formed by repeatedly re-sampling and then convolving with the same filter expanded out to the proper sample grid. The motivation for this algorithm is a great reduction in computational complexity in acquiring the sequence of sampled low-pass signals needed to form a Sampled DOLP transform and its description.

6.2.1 Cascaded Filtering and the VIF Expansion Operation

The cost of computing the DOLP transform without resampling can be reduced from $O(N^2)$ multiplications to $O(N \log N)$ by using the Gaussian scaling property and the \sqrt{l} expansion operation (defined below).

Let us consider the use of the Gaussian scaling property for forming a DOLP transform without the use of $\sqrt{2}$ expansion or resampling, in this version of the DOLP transform the low pass image at level k is formed by 2^{k+1} convolutions of the low pass image at level $k-1$ with the kernel low pass filter g_0 . Thus the level 1 low-pass filter impulse response, g_1 is approximated by

$$g_1 \approx g_0 * S_0$$

and the level 2 low-pass filter, g_2 is approximated by

$$g_2 \approx g_0 * g_0 * g_0 * g_0$$

For each additional level, the number of convolutions with g_0 doubles.

6.2.2 Cascaded Convolution with Expansion

The exponential growth that results from cascaded filtering can be averted by expanding each low-pass filter onto a sample grid which is a $\sqrt{2}$ larger before the convolution to produce the next low-pass level. This expansion operation scales the low-pass filter impulse response larger in standard deviation by $\sqrt{2}$ but it also introduces reflections of the low-pass transfer function in the corners of the Nyquist plane, $-\pi < u, v \leq \pi$. The kernel filter can be formed so that these reflections fall over the stop region of the kernel filter and are thus greatly attenuated, as shown in section 6.2.4 below.

Cascaded convolution with expansion can be used to compute a DOLP transform that is not resampled in $O(N \log N)$ multiplies. This complexity may be arrived at by the following reasoning. The $\sqrt{2}$ expansion operation does not change the number of coefficients in the filter. Thus each low-pass image may be formed from the previous low pass image with the same cost in multiplies. The cost of each convolution is $X_0 N$ multiplies where X_0 is the number of coefficients in the kernel filter and N is the number of samples in the image. Since the impulse response scale grows exponentially, there are $O(\log N)$ low-pass images. Hence the cost of cascaded convolution with expansion is $O(N \log N)$ multiplies. This expansion operation and its effect on the transfer function of a Gaussian low-pass filter is examined in the following Subsections*

6.2.3 $\sqrt{2}$ Expansion and Resampling

In this section we consider the expansion operation in the context of the use of cascaded convolution and resampling. The $\sqrt{2}$ expansion operator is a convenient way of scaling a Gaussian low-pass filter by a factor of $\sqrt{2}$. When images are resampled, expanding the filter onto the same sample grid automatically gives the expansion operation.

The $\sqrt{2}$ expansion operation maps each row from a filter on a cartesian sample grid into every other diagonal. This mapping takes each coefficient from point (x,y) of a filter $g(x,y)$ and places it at point $(x-y, x+y)$ of a filter $g_2(x_2, y_2)$. Points of $g_2(x_2, y_2)$ which receive no coefficient under this mapping are declared to be undefined.

Let us define this mapping as the function $E_{\sqrt{2}}[\cdot]$. Since

$$\begin{aligned}x_2 &= x - y \\y_2 &= x + y\end{aligned}$$

we get

$$x = \frac{-x_2 + y_2}{2}$$

and

$$y = \frac{x_2 + y_2}{2}$$

So that this function may be defined by

$$E_{\sqrt{2}}[g(x,y)] \triangleq g_2(x_2, y_2) = \begin{cases} g((-x_2 + y_2)/2, (x_2 + y_2)/2) & \text{For } x_2 \text{ Mod } 2 = y_2 \text{ Mod } 2 \\ \text{Undefined otherwise} \end{cases}$$

Where $A \text{ Mod } B$ is the remainder of A/B . This mapping is illustrated by figure 6-3. This figure shows the correspondence between points in the mapping. The dashes ("-") illustrate the points which are not defined in the new filter.

The algorithm for cascaded filtering with sampling involves repeatedly re-sampling. Each re-sampling enlarges the actual smallest distance between samples by $\sqrt{2}$ and alternates the direction of that smallest distance between $\pm 45^\circ$ and $0^\circ, 90^\circ$. For each convolution the distance between filter coefficients must be expanded by $\sqrt{2}$ as many times as the image has been re-sampled. For this, a more general expansion operator is needed: $E_{\sqrt{2}}^l[\cdot]$. This more general operator expands the filter to the same grid as an image which has been $\sqrt{2}$ sampled l times.

When l is odd, the filter is mapped onto a grid whose axes are $\pm 45^\circ$, and whose smallest distance between samples is $2^{l/2}$. The points on this grid are those at which

$$x_1 \text{ Mod } 2^{(l+1)/2} = y_1 \text{ Mod } 2^{(l+1)/2} = 0.$$

For even l , the expanded filter will be mapped onto a grid whose axes are at 0° and 90° . The distance between samples along these axes will also be $2^{l/2}$. The mapping $E_{\sqrt{2}}^l$ may be defined as:

.(0,1) .(2,2) .(2,1)
 .(0,0) .(1,0) .(2,0)
 .(0,-1) .(1,-1) .(2,-1)

maps into

.(2,1)
 .(1,1) - .(2,0)
 .(0,1) - .(1,0) - .(2,-1)
 .(0,0) - .(1,-1)
 .(0,-1)

Figure 6-3: Example of mapping given by $E_{\sqrt{2}}[\cdot]$

For even l :

$$g_l(x,y) = \begin{cases} g\left(\frac{x_1}{2^{l/2}}, \frac{y_1}{2^{l/2}}\right) & \text{For } x_1 \text{ Mod } 2 = 0 \text{ and } y_1 \text{ Mod } 2 = 0 \\ \text{Undefined otherwise} \end{cases}$$

For odd l :

$$g_l(x,y) = \begin{cases} g\left(\frac{-x_1+y_1}{2^{(l+1)/2}}, \frac{x_1+y_1}{2^{(l+1)/2}}\right) & \text{For } x_1 \text{ Mod } 2^{(l+1)/2} = y_1 \text{ Mod } 2^{(l+1)/2} \\ \text{Undefined Otherwise} \end{cases}$$

For a circularly symmetric filter this mapping is equivalent to applying the following procedure recursively l times:

$E_{\sqrt{2}}\{ \cdot \}$ Procedure:

For each point (x,y) at which the filter $g_{l-1}(x,y)$ is defined, define a new point in $g_l(x,y)$ at $(x-y, x+y)$ and copy the value from $g_{l-1}(x,y)$ into the point.

This is the procedure which was used for the experimental implementation.

6.2.4 Frequency Domain Effects of $\sqrt{2}$ Expansion

The $\sqrt{2}$ expansion operator has a well defined effect on the transfer function of its argument. As with $\sqrt{2}$ sampling a new Nyquist boundary is created which is a 45° rotation and a $\sqrt{2}$ shrinking of the old boundary. Inside this new Nyquist boundary is a copy of the old transfer function scaled down in size by a factor of $\sqrt{2}$. Outside this new Nyquist boundary is a reflection of the scaled transfer function. This is illustrated by figure 6-4 below, which shows the 3dB contour of a low-pass filter before and after the expansion operation. Figures 6-5 and 6-6 show actual plots of a Gaussian

low-pass filter ($R=4$, $\alpha=4$), before and after the expansion operation. Note the 4 lobes in the corners of figure 6-6. These are the reflections of the pass region. If these were to show up in the composite filter they could cause a large stop-band response, which would add aliasing to the transform because of re-sampling.

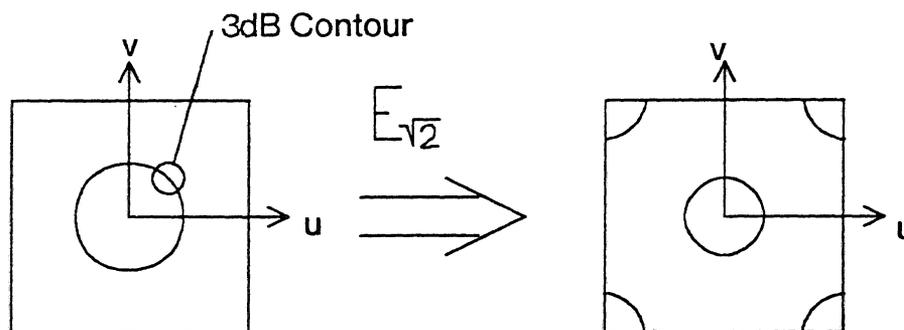


Figure 6-4: Effect on Transfer Function of $E_{\sqrt{2}}$ Expansion Operator

$E_{\sqrt{2}}\{.\}$ scales the size of the transfer function by $\sqrt{2}$ so that it fits into the new smaller Nyquist boundary. That is

$$\mathcal{F}\{E_{\sqrt{2}}[g_o(x,y)]\} = \mathcal{F}\{g_1(x,y)\}$$

$$\text{within } \pi \leq |u+v| \leq \pi \text{ (The new Nyquist boundary)}$$

Because the expansion operation introduces a reflection about the new Nyquist boundary, there is reason to be concerned about the stop-band error introduced by this technique. The stop-band error is not a serious problem for the parameter values $R=4$, $\alpha=4$. The reflected energy from expansion falls into the stop-band of the previous filter. That is, outside of the new Nyquist boundary,

$$\mathcal{F}\{g_o(x,y) * g_o(x,y)\}$$

will be very small (i.e. < -60 dB¹⁰ for $R=4$, $\alpha=4$) and thus the product

$$\mathcal{F}\{E_{\sqrt{2}}[g_o(x,y)]\} \circ \mathcal{F}\{g_o(x,y) * g_o(x,y)\}$$

¹⁰Response is < 95 dB in the area of the corner where the reflected nodes are present

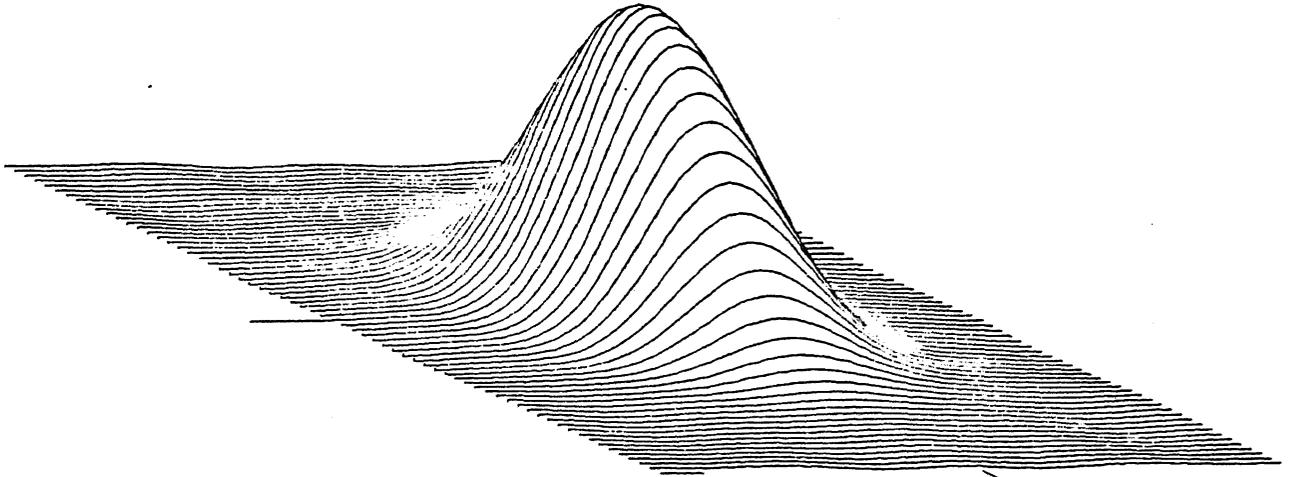


Figure 6-5: Filter $G_o(u,v)$ for $R = 4.0$, $\alpha = 4.0$ Before $\sqrt{2}$ Expansion

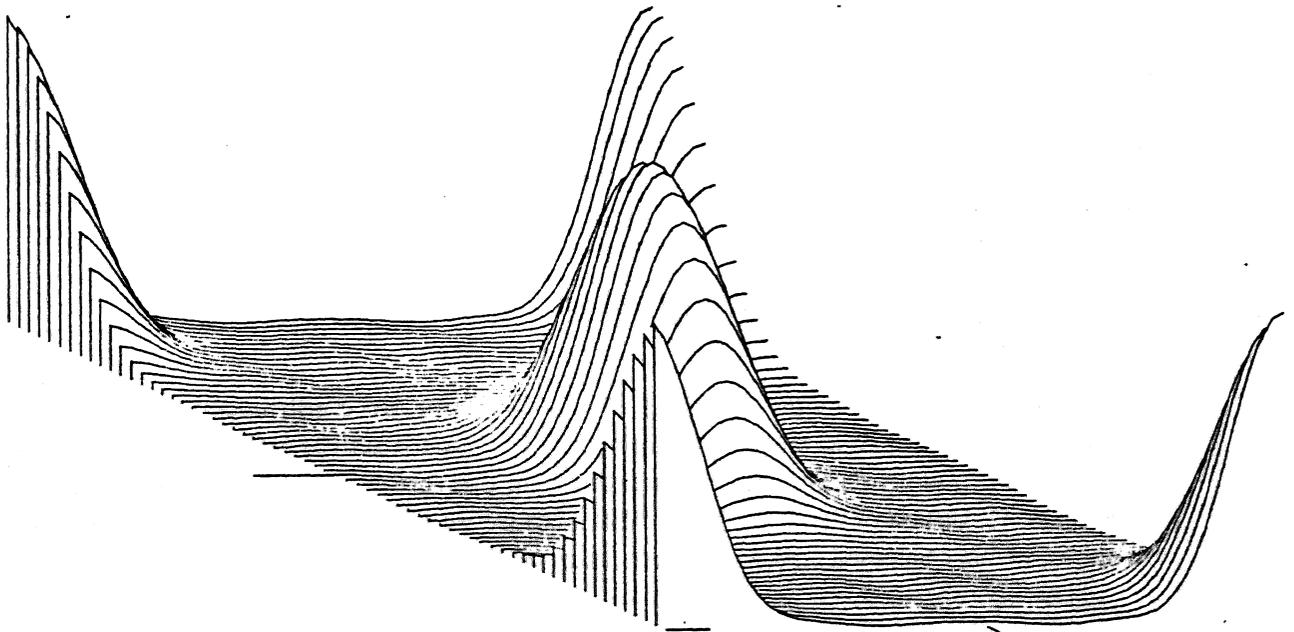


Figure 6-6: Filter $G_o(u,v)$ After $\sqrt{2}$ Expansion

will be very very small outside the new Nyquist boundary. Thus the impulse response at low-pass level 2, \mathcal{L}_2 , which is desired to be $g(x,y; \sigma_2=2\sigma_o)$ that is, $g_o(x,y)$ with its standard deviation scaled larger by a factor of 2, is actually approximated by

$$g(x,y; \sigma_2=2\sigma_0) \approx S_{\sqrt{2}}[g_0(x,y) * g_0(x,y)] * E_{\sqrt{2}}[g_0(x,y)]$$

Where $S_{\sqrt{2}}(\cdot)$ is the $\sqrt{2}$ resampling operation which was defined in section 3.3 as

$$S_{\sqrt{2}}[p(x,y)] = \begin{cases} p(x,y) & \text{for } x \bmod 2 = y \bmod 2 \\ \text{undefined} & \text{otherwise} \end{cases}$$

Figure 6-7 is a plot of the transfer function of the level 2 low-pass filter. As can be seen the response in the corners is so small that it does not register in this plot.

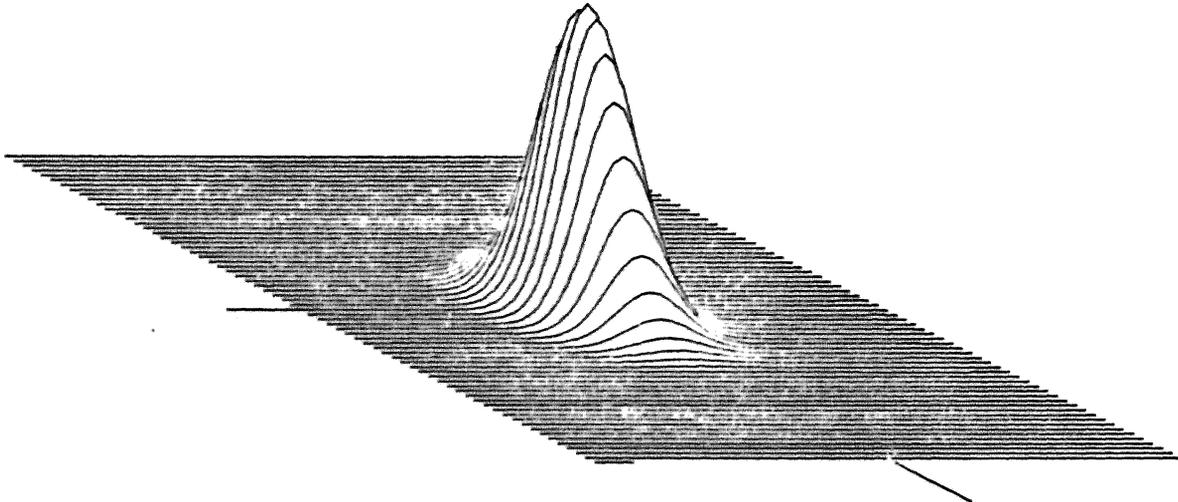


Figure 6-7: Filter $G_2(u,v)$ for $R = 4.0$, $\alpha = 4.0$
 $g_2(x,y) = S_{\sqrt{2}}[g_0(x,y) * g_0(x,y)] * E_{\sqrt{2}}[g_0(x,y)]$

A logarithmic plot of the amplitude of $G_2(u,v)$ is shown in figure 6-8. This plot spans -120 db in amplitude. The scale on the left marks off drops of -10 db. Note that the response in the corner region is well below -100 dB.

6.3 The Sampled DOG Transform

In this section we define the Sampled DOG transform by construction and examine the computational complexity and memory requirements. Unlike the similar sections in chapter 5 on the DOLP transform and the Sampled DOLP transform, in this section we are concerned with only the two-dimensional version of this transform. Also, because we use the Gaussian scaling property and resampling, we are concerned only with a scale factor of, $S_2 = \sqrt{2}$.

As in the similar sections in chapter 5, the number of filter coefficients for the level 0 band-pass filter, X_0 , is related to the radius by:

$$X_0 \approx \pi R_0^2$$

Also, as before, the 2-D image signal is assumed to have N samples. The convolutions are computed for the filter centered over each sample point, with a default boundary value supplied as needed.

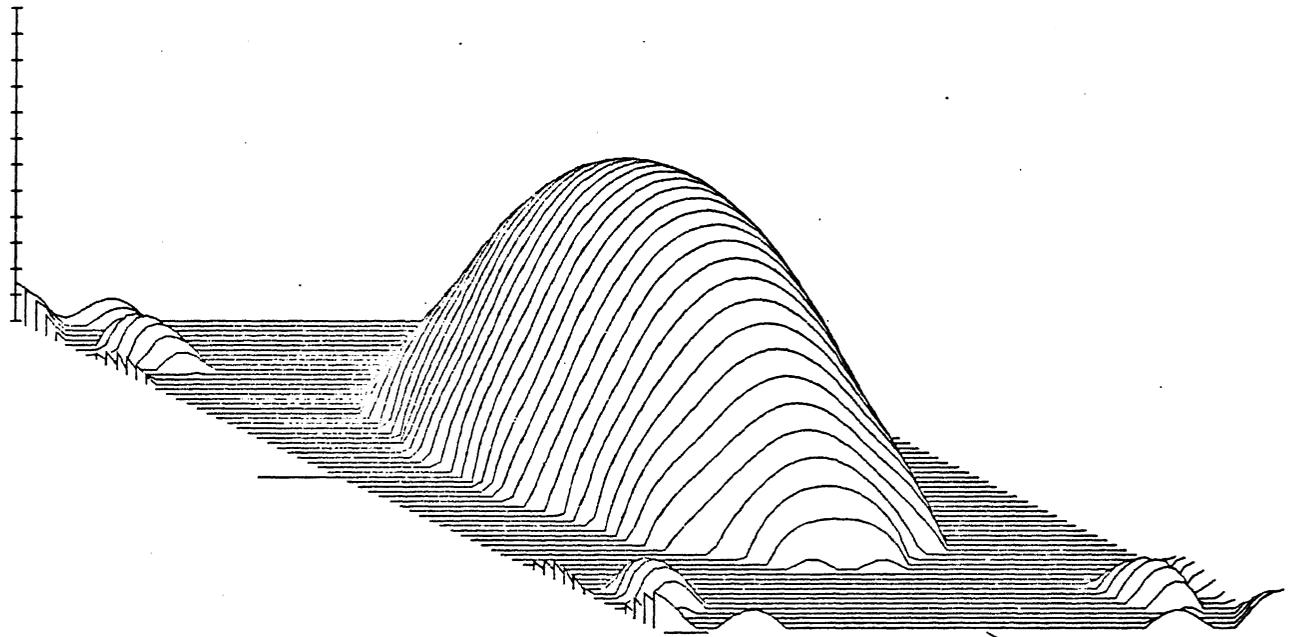


Figure 6-8: Plot of $20 \text{ Log}_{10} [G_2(u,v)]$
Scale (shown at left) spans -120 dB.

6.3.1 Construction of a Sampled DOG Transform

The sampled DOG transform may be expressed by the data flow graph shown below as figure 6-9. The number of points (for an N point image) produced by each step are given in square brackets to the right of each band-pass level.

As with the DOLP and Sampled DOLP transforms, the high-pass residue, \mathfrak{B}_0 , is formed by convolving g_0 with the image, p , to form \mathcal{L}_0 and then subtracting the convolution output at each point from the sample under the center of the filter as it is computed. That is, the low-pass level 0 signal is given by:

$$\mathcal{L}_0 = g_0 * p$$

and the level 0 band-pass signal is given by:

$$\mathfrak{B}_0 = p - \mathcal{L}_0$$

The level 0 impulse response is:

$$b_0 = 1 - g_0$$

Note that when filters of different sizes are subtracted, it is implied that their centers are aligned, and that undefined coefficients are treated as having the value zero. The filter, b_0 , defined above is the same as that given in figure 6.12 below.

Computing \mathfrak{B}_0 requires X_0 N multiplies and produces N sample points.

The low-pass level 1 signal is then formed by convolving g_0 with the low-pass level 0 signal. Thus

$$\mathcal{L}_1 = g_0 * \mathcal{L}_0$$

and

$$g_1 = g_0 * g_0$$

During the convolution, the level 1 band-pass signal \mathfrak{B}_1 is formed by subtracting each sample point of \mathcal{L}_1 from the corresponding point of \mathcal{L}_0 .

$$\mathfrak{B}_1 = \mathcal{L}_0 - \mathcal{L}_1$$

and

$$b_1 = g_0 - (g_0 * g_0)$$

This operation also requires X_0 N multiplies and produces N sample points.

Since the level 1 low-pass filter transfer function has a pass and transition band that has been designed to be inside a $\sqrt{2}$ shrinking of the Nyquist boundary, it can be re-sampled at $\sqrt{2}$. Thus, only the samples along every other diagonal are stored. The result is a low-pass signal, $S_{\sqrt{2}}\{\mathcal{L}_1\}$ which has N/2 sample points.

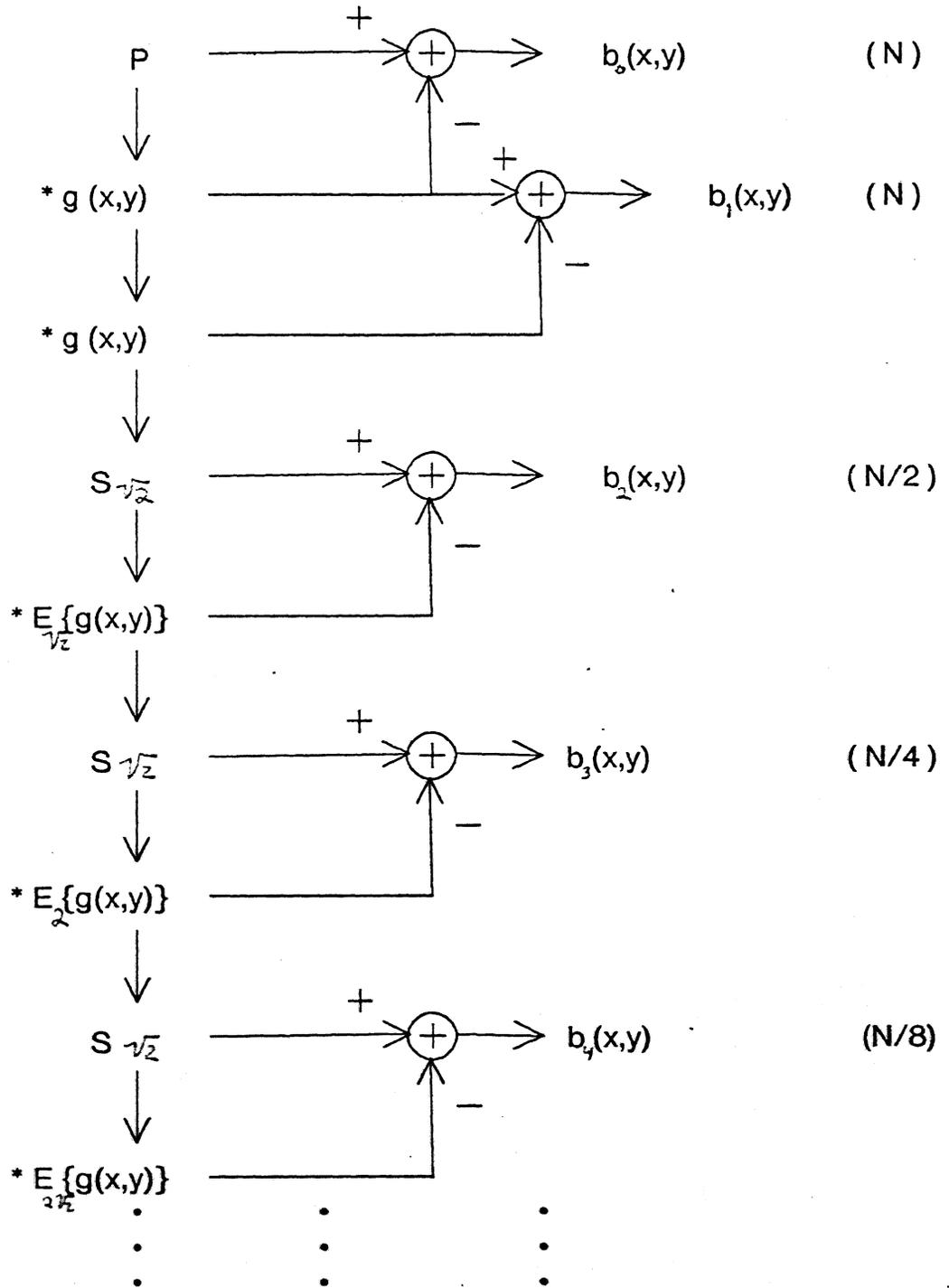


Figure 6-9: Data Flow Graph for Sampled DOG Transform

This sampled low-pass level 1 signal is then convolved with an expanded version of g_0 to produce l_2 . Thus:

$$l_2 = S_{\sqrt{2}}\{l_1\} * E_{\sqrt{2}}\{g_0\}$$

and

$$g_2 = E_{\sqrt{2}}\{g_0\} * S_{\sqrt{2}}\{g_0 * g_0\}$$

During this convolution, the level 2 band-pass filter is formed by subtracting each low-pass sample, l_2 from the sampled version of l_1 .

$$b_2 = S_{\sqrt{2}}\{l_1\} - l_2$$

Thus the level 2 band-pass filter is given by:

$$b_2 = S_{\sqrt{2}}\{g_1\} - g_2$$

Since $S_{\sqrt{2}}\{l_1\}$ has $N/2$ samples, this operation requires $X_0 N/2$ multiplies and produces $N/2$ samples.

The Sampled DOG process continues in this manner until the K^{th} level. Thus the level 2 low-pass signal, l_2 is again sampled at a distance of $\sqrt{2}$, corresponding to a sample for every other column of every other row of the original picture, p . This is a total of $N/4$ sample points. This resampled low-pass signal is convolved with a twice expanded low-pass filter:

$$E_2\{g_0\} = E_{\sqrt{2}}\{E_{\sqrt{2}}\{g_0\}\}$$

to form the level 3 low-pass signal,

$$l_3 = E_2\{g_0\} * S_{\sqrt{2}}\{l_2\}$$

and

$$g_3 = E_2\{g_0\} * S_{\sqrt{2}}\{E_{\sqrt{2}}\{g_0\} * S_{\sqrt{2}}\{g_0 * g_0\}\}$$

Thus band-pass level 3 is formed by:

$$b_3 = S_{\sqrt{2}}\{l_2\} - l_3$$

and the level 3 band-pass impulse response is:

$$b_3 = S_{\sqrt{2}}\{g_2\} - (E_2\{g_0\} * S_{\sqrt{2}}\{g_2\})$$

Since $S_{\sqrt{2}}\{l_2\}$ has $N/4$ samples, producing the level 3 band-pass signal requires $X_0 N/4$ multiplies and produces $N/4$ sample points.

In summary, for levels 2 through K we can state the following recursive formulae:

$$l_k = E_{\sqrt{2}}\{(k-1)\{g_0\}\} * S_{\sqrt{2}}\{l_{k-1}\} \quad (6.1)$$

$$g_k = E_{\sqrt{2}}\{(k-1)\{g_0\}\} * S_{\sqrt{2}}\{g_{k-1}\} \quad (6.2)$$

(6.3)

$$I = \{k-i - (EVT^* - fe_o) * Sfc-i\} \quad (6-4)$$

6.3.2 Computational Complexity and Memory Requirements

Producing each band-pass level, k , for the $k-1$ low-pass level requires $X_0 N/2^{k-1}$ multiplies, and produces $N/2^{k-1}$ samples. Thus the cost, C_{SDOG} , of computing a Sampled DOG transform of an image signal with N samples is:

$$C_{SDOG} = X_0(N + N + N/2 + N/4 + N/8 + \dots) \\ \gg 3 X_0 N \text{ multiplies}$$

The total number of band-pass samples produced, M , is:

$$M = N + N + N/2 + N/4 + N/8 + \dots \\ \text{sr } 3N \text{ samples}$$

6.3.3 Comparison of Complexity with Filtering Using FFT

The Sampled DOG Transform is based on a filtering algorithm which we have named "Cascade Convolution with Sampling". Any sampled DOLP transform could alternatively be computed using the Fast Fourier Transform (FFT) algorithm. A Sampled DOLP Transform of an N point signal (1-D or 2-D) could be computed using the FFT algorithm by the following steps:

1. Precompute the coefficients of the level 0 band-pass filter (high-pass residue) and the level 1 band-pass filter. Evaluate the transfer functions of these two filters over N equally spaced points in the nyquist interval. Since the level 2 through K band-pass filters are size scaled copies of the level 1 filter, their transfer functions can be obtained from the level 1 band-pass transfer-function by resampling, as described below. The cost of computing these transfer functions will not be included in this complexity analysis.
2. Compute the Discrete Fourier Transform (DFT) of the signal using the FFT algorithm. This requires $N \log_2 N$ multiplies for an N point 1-D signal or $[M \log_2 M]^2$ multiplies for an $N = M \times M$ 2-D signal. Note that for this step alone is more expensive for:

$$\log_2 N > 3 X_0 \text{ in the 1-D case, and}$$

$$[\log_2 M]^2 > 3 X_0 \text{ in the 2-D case}$$

3. For band-pass levels 0 and 1, multiply the DFT of the signal by the transfer function of each filter. Each product costs N multiplies. For band-pass levels $k=2$ through $k=K$ both the transfer functions and the DFT of the signal must be re-sampled to $N/2^{k-1}$ evenly spaced points. Each re-sampled transfer function is then multiplied by the corresponding re-sampled DFT for a cost of $N/2^{k-1}$ multiplies at each level. The total cost of these multiplies is then:

$$N + N + N[1/2 + 1/4 + 1/8 + \dots] = 3N \text{ multiplies}$$

4. Compute the inverse FFT of each array. This requires

$$\begin{aligned} N \log_2 N + \sum_{k=1}^{\log_2 N} (N / 2^k) \log_2(N / 2^k) & \text{ multiplies} \\ &= 2N \log_2(N) + N/2 \log_2(N/2) + N/4 \log_2(N/4) + \dots \\ &= 2N \log_2(N) + N/2[\log_2(N) - 1] + N/4 [\log_2(N) - 2] \\ & \quad + N/8 [\log_2(N) - 3] + \dots \\ &= 2N \log_2(N) - \sum_{k=1}^{\log_2 N} k N / 2^k \end{aligned}$$

The final series term at the end converges to approximately $2N$. The middle series, as we have seen before converges to N , so that the cost of the inverse FFTs is approximately:

$$3N \log_2(N) - 2N \text{ multiplies}$$

Thus the total cost of using the FFT algorithm is:

$$\begin{aligned} C_{\text{FFT}} &\approx N \log_2(N) + 3N + 3N \log_2(N) - 2N \\ &= 4N \log_2(N) + N \text{ Multiplies} \end{aligned}$$

Recall that the Sampled DOG transform requires approximately:

$$C_{\text{DOG}} \approx 3 X_0 N \text{ multiplies}$$

Thus the Sampled DOG algorithm costs less whenever:

$$3 X_0 < 4 \log_2(N) - 1$$

For the 1-D case, X_0 has a typical value of 9. Thus the Sampled DOG Transform is cheaper whenever:

$$N > 2^{6.5} = 90.5$$

For Circularly Symmetric filters in the 2-D case, X_0 is typically 49. Also the cost of a FFT for an $N = M \times M$ signal is $[M \log_2 M]^2$ multiplies, so that the Sampled DOG Transform is cheaper in terms of multiplies whenever:

$$4[\log_2(M)]^2 > 3(49)$$

$$\text{or } [\log_2(M)]^2 > 36.5$$

$$\text{or } \log_2(M) > 6.04$$

$$\text{or } M > 2^{6.04} = 65.86$$

6.3.4 The Size of Cascaded Filter Impulse Response

As discussed above, the sampled DOG transform employs cascaded convolution with sampling to produce a set of low-pass images whose Gaussian impulse responses are scaled larger in standard deviation by a factor of \sqrt{T} from each level to the next. In chapter 5 this scaling was discussed in terms of the filter radius. Cascaded filtering produces a set of impulse responses whose radii grow faster than a factor of \sqrt{T} .

The level 0 low-pass filter is defined over a disc of radius $R_0=4$. When convolved with itself to produce the level 1 low-pass filter it produces an impulse response which is non-zero over a disc of radius $2R_0$. This is a property of the convolution operation. At the same time, the standard deviation of this impulse response has only grown by \sqrt{T} .

The convolution of two functions which are normalized to sum to one produces a function whose values also sum to one. Thus the autoconvolution of the Gaussian preserves its normalization to unit sum. Since the auto-convolution has its unit sum spread out over a larger area, the coefficient values are slightly smaller than the same coefficients for a unit-sum Gaussian filter which is computed by scaling the R parameter by \sqrt{T} .¹¹ The auto-convolved Gaussian filter has a larger tail and is thus a closer approximation to the infinite 2-D Gaussian function.

The level 1 low-pass image is sampled at \sqrt{T} and so the low-pass filter must be expanded to the same sample grid by the $E^{\{ \}$ operator defined above. From a filter defined over a disc of radius R_0 , the expansion operator $R_{\sqrt{T}j}^{\{ \}$ produces a filter whose furthest coefficient from the origin is at $\sqrt{T}R_0$. That is, for a radius 4 filter, the coefficient from (4,0) is mapped into the point at (4,4). When this filter is convolved with the level 1 low-pass filter, the result is a filter whose radius is $R_0 + R_0\sqrt{T}$.

Each additional expansion of the filter will enlarge it in radius by a factor of \sqrt{T} and will add its size to that of the cumulative impulse response. Thus the radius of the cumulative impulse response, R_k , for the level k low-pass filter is given by the following formula:

$$R_k = R_0 \sum_{n=0}^k [\sqrt{T}]^n$$

This support radius grows much faster than the support radius

$$R_k = R_0 [\sqrt{T}]^k$$

for a simple scaling of the function. This faster growth in support radius is advantageous; it provides a low-pass impulse response at each level which is a closer approximation to the infinite Gaussian function. Thus at each level the error in the auto-convolution scaling that results from the finite duration of the Gaussian filter is reduced.

¹¹ Note that the two functions do have the same standard deviation.

6.4 Verification of Scaling Approximation

Because the discrete two dimensional Gaussian filter defined in section 6.1 is defined over a finite window, the scaling relation described in section 6.1.1 is only approximate for $g_o(x,y)$. Described below are three measures for the accuracy of this scaling for the approximation:

$$g(R=4\sqrt{2}, \alpha=4.0) \approx g(R=4, \alpha=4) * g(R=4, \alpha=4)$$

6.4.1 Diagonal Method in Space Domain:

The easiest measure of the accuracy of scaling by auto-convolution is to compare the coefficients of $g_o(x,y)$ along the axis $x=y$ to the coefficients of $g_o(x,y) * g_o(x,y)$ along the x axis. These sample points have the same ratio of distance from the center to total radius, and thus will have the same value if the filter is exactly expanded by $\sqrt{2}$ and is circularly symmetric. These data are shown in table 6-1 below. The coefficients of $g_o(x,y)$ are generated normalized to a dc response of 1.0. Their auto-convolution also has a dc response of 1.0. The effects of this normalization were removed by dividing each coefficient by the coefficient at 0,0, and this could be a source of small inaccuracy.

x	1	2	3
g	0.7788	0.3678	0.1054
g * g	0.7768	0.3607	0.0952
%error	0.25%	1.9%	9.6%

Table 6-1: Comparison of Filter Coefficients

It should be noted that the auto-convolution, $g_o(x,y) * g_o(x,y)$, has a finite support that is a disc with a radius of $\approx 2R$, as opposed to $g_1(x,y)$ which is defined over a disc of radius $\sqrt{2}R$. Yet both filters are normalized so that their sum is 1.0. For this reason the autoconvolution should be expected to taper slightly faster than the scaled filter. The auto-convolved filter will actually be a closer approximation to a Gaussian function.

6.4.2 Diagonal Method in Frequency Domain:

This method involves comparing values in the real part of the transfer function $G(u, v; R=4, \alpha=4)$ along the diagonal axis $u=v$ to values of $\mathcal{F}\{g(R=4, \alpha=4) * g(R=4, \alpha=4)\}$ along the axis $v=0$. The distance to the origin is $u\sqrt{2}$ for the points from the first transfer function and u for the second. The values are shown for distances of $u=n\pi/32$ where n ranges from 1 to 16.

The maximum error shown by this method is 0.011 and it occurs at $n = 9$ and 10 or frequencies of $u = 9\pi/32$ and $u = 10\pi/32$. As with the diagonal method in the space domain this comparison may be sensitive to any circular non-symmetry in the filter. A larger source of error would be the difference in normalization that occurs because of the larger support for the auto-convolved filter.

n	1	2	3	4	5	6	7	8
$G(u,v)$	0.982	0.931	0.852	0.750	0.636	0.518	0.414	0.302
$G(u,v) * G(u,v)$	0.982	0.932	0.852	0.752	0.639	0.523	0.412	0.312
error	0.000	0.001	0.000	0.002	0.003	0.005	0.008	0.010
% error	0.00	0.10	0.00	0.26	0.46	0.95	1.94	3.20

n	9	10	11	12	13	14	15	16
$G(u,v)$	0.215	0.146	0.095	0.060	0.037	0.024	0.016	0.012
$G(u,v) * G(u,v)$	0.226	0.157	0.104	0.066	0.040	0.023	0.013	0.007
error	0.011	0.011	0.009	0.006	0.003	0.001	0.003	0.005
% error	4.86	7.00	8.65	9.09	7.50	4.34	23.07	71.42

Table 6-2: Diagonal Comparison Of Transfer Function Samples

6.4.3 Expansion Method:

The third technique for measuring the accuracy of the approximation was to form the two filters $g_o(x,y) * g_o(x,y)$ and $E_{\sqrt{2}}\{g_o(x,y)\}$, subtract the expanded filter from the auto-convolved filter, and then compute the transfer function of this difference. A plot of this difference is shown below as figure 6-10. This plot is dominated by the reflection of the center lobe from the expanded filter, which is not present in the auto convolved filter. The idea behind this method is that within the diamond shaped region, $|u+v| \leq \pi$ the expanded filter should be identical to a $\sqrt{2}$ scaling in size of the original filter.¹² The transfer function to the third decimal place shows a number of circular ripples within the region where the two filters should be the same. The largest ripple has a peak of -0.012 which occurs over an arc of constant radius, spanning $u,v = -9\pi/32, -3\pi/32$ to $-3\pi/32, -9\pi/32$.

Table 6-3 below shows the error values along the diagonal $u=v$ for $u = n\pi/32$ for $n \in \{1,2,3,\dots,16\}$.

The errors shown by this method are of the same magnitude, but not identical to those found by the diagonal frequency domain method. In both measures involving transfer functions the error in the approximation was found to be at most 0.012 (out of 1.000) and this maximum error tended to be at or near $u^2 + v^2 \approx 8\pi/32$, which is also the peak frequency, ω_1 , of the band-pass filter at band-pass level 1.

The conclusion formed from these experiments was that the scaling approximation was accurate enough for the finite filters formed using $R = 4$, $\alpha = 4.0$, to permit its use in developing a description technique based on the Sampled DOG transform.

¹²Outside this region the reflection of the center lobe in the auto-convolved filter will dominate the difference as seen in figure 6-10.

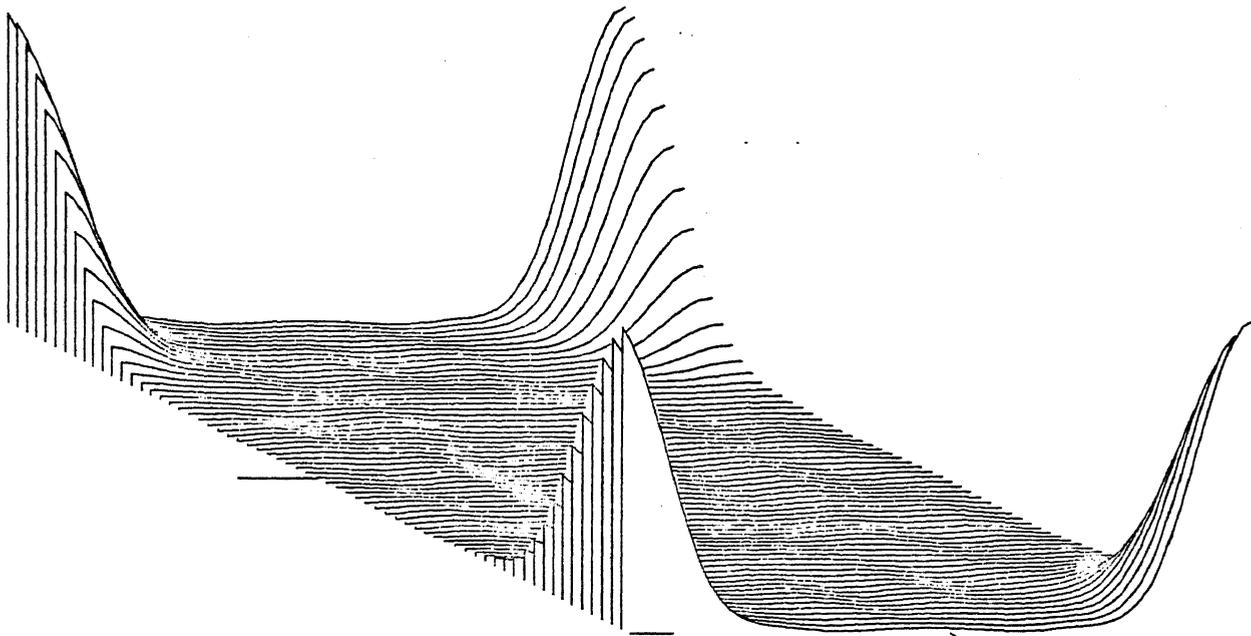


Figure 6-10: Transfer Function of $E_{\sqrt{2}}\{g_o(x,y)\} - g_o(x,y) * g_o(x,y)$

n	1	2	3	4	5	6	7	8
$\mathcal{F}\{E_{\sqrt{2}}\{g\} - (g * g)\}$	0.000	0.001	0.002	0.005	0.008	0.011	0.012	0.010

n	9	10	11	12	13	14	15	16
$\mathcal{F}\{E_{\sqrt{2}}\{g\} - (g * g)\}$	0.005	0.001	-0.005	-0.007	-0.007	-0.004	-0.001	0.000

Table 6-3: Values Along Line $u=v$ in Transfer Function of $E_{\sqrt{2}}\{g\} - (g * g)$

6.5 The Band-Pass Filters

This chapter comes to a close by showing the impulse responses and transfer functions for the smaller filters. Given below are the coefficients for the band-pass filters at levels 0 and 1, and plots of the transfer functions of the level 1 and level 2 band-pass filters.

6.5.1 Size of Positive Center Radius

The scale or size of forms to which each filter in a sampled DOG transform is sensitive depends on the size of the positive center lobe of the impulse response. We have observed by examining the coefficients of the impulse responses that for the Sampled DOG transform based on a Gaussian low pass filter with a radius, $R_o = 4.0$, and a shape parameter of $\alpha = 4.0$, the radius of the zero crossing of this positive center lobe, R_{k+} , at a level, k , may be predicted by the following formula.

$$R_{k+} \gg \sqrt{5} (VI^k)$$

This formula is based on the observations given in table 6-4 below. The radii of the positive center lobes in this table were measured by finding the distance from the center point to the furthest (and smallest) positive coefficient. The filters tend to be most sensitive to objects whose width is $2R_{fc} + 1$. Note that as the radius increases there are more coefficients near the zero crossing, and thus the accuracy to which the zero-crossing radius can be determined increases.

Level	Radius of Center Lobe
1	$\sqrt{5} = 2.23606$
2	$\sqrt{10} = 3.1622$
3	$\sqrt{20} = 4.4721$
4	$\sqrt{40} = 6.4031$

Table 6-4: Radii of Center Lobes
As measured by Distance to Furthest Positive Coefficient

6.5.2 Relative Size of Filters and Their Transfer Functions

Since the filters are circularly symmetric, it is possible to visualize each filter impulse response and transfer function from the values along a line which passes through the center of the filter or its transfer function. Figure 6-11 shows plots of the coefficient values along the X axis of the band-pass filters for levels 1 through 4. Note that the size of each filter increases by a factor of $\sqrt{2}$ from the previous filter and that the maximum response (at the center) decreases by a factor of 2 from the previous filter.

The following figure shows the transfer functions for the band-pass filters from levels 1 through 4. The transfer function values from the u axis ($v = 0$) from $0 \leq u \leq m$ are shown. The spatial frequency values are shown as integers from 0 to 32 because the transfer function was evaluated over a 64 x 64 grid. (Note that $u = 2\pi f = 2\pi r k / 64$).

6.5.3 Filter at Band-Pass Level 0

We start with figure 6-13 which shows the filter which gives the high pass residue, $3B_0$. This filter is the lowpass filter $g_0(x,y)$ with its center coefficient subtracted from 1 and all other coefficients subtracted from zero.

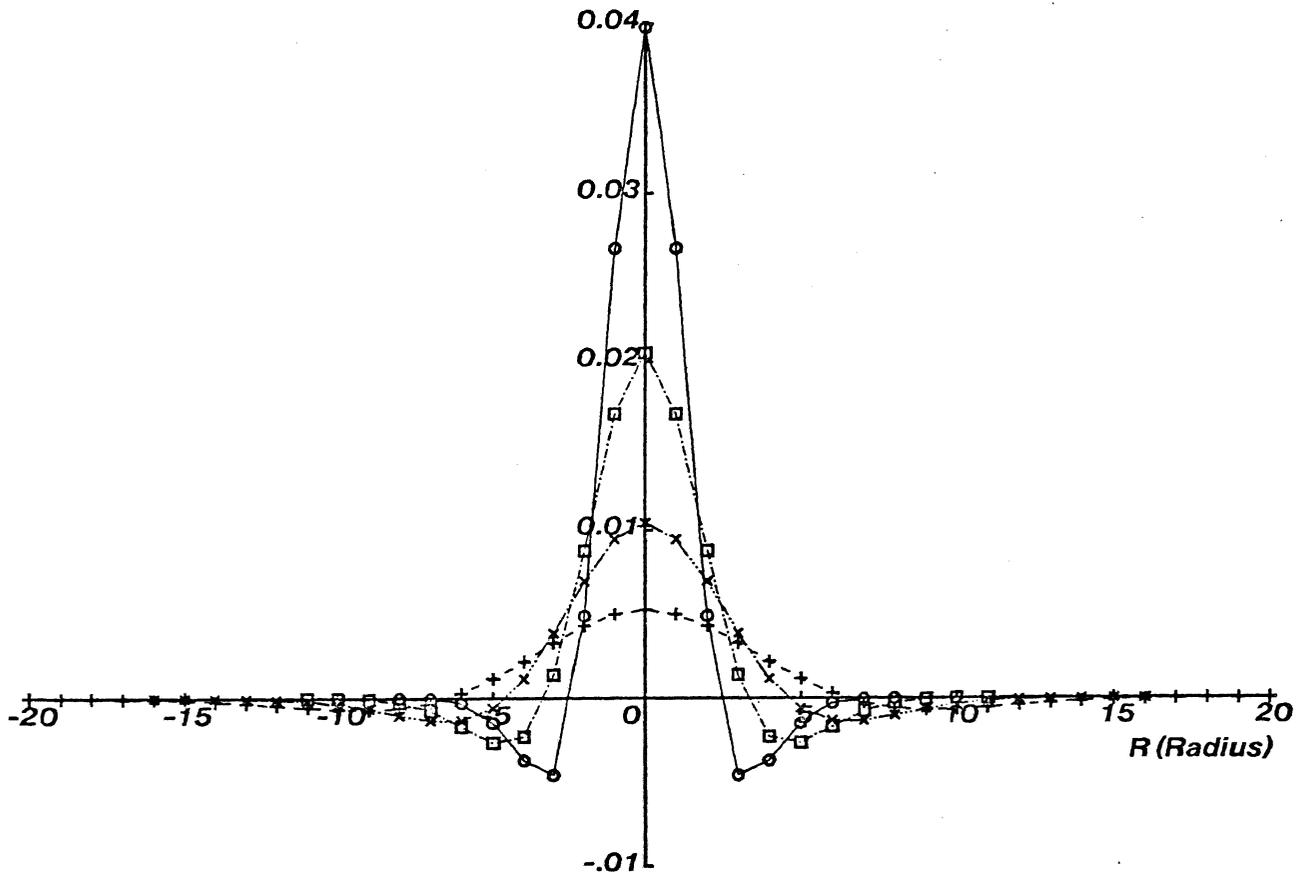


Figure 6-11: Coefficients Along X Axis for Filters from Levels 1 Through 4

6.5.4 Filter at Band-Pass Level 1

Next is figure 6-14 which gives the coefficients for the band-pass filter at level 1. The formula for this filter is:

$$b_1(x,y) \triangleq g_o(x,y) - (g_o(x,y) * g_o(x,y))$$

The values for this filter are shown in two sections so that they fit on a page. The first section is columns -8 to 0, and the second is columns 1 to 8.

Figure 6-15 shows the transfer function, $B_1(u,v)$ for the level 1 band-pass filter. The peak response is 0.250 at $\sqrt{u^2+v^2} = \pi/4$.

Figure 6-16 shows a logarithmic plot of $B_1(u,v)$. This plot spans -40 dB. The scale at the left marks off drops of -10 dB in response. This relatively large ripple is not a concern because the level 1 band-pass image is not resampled.

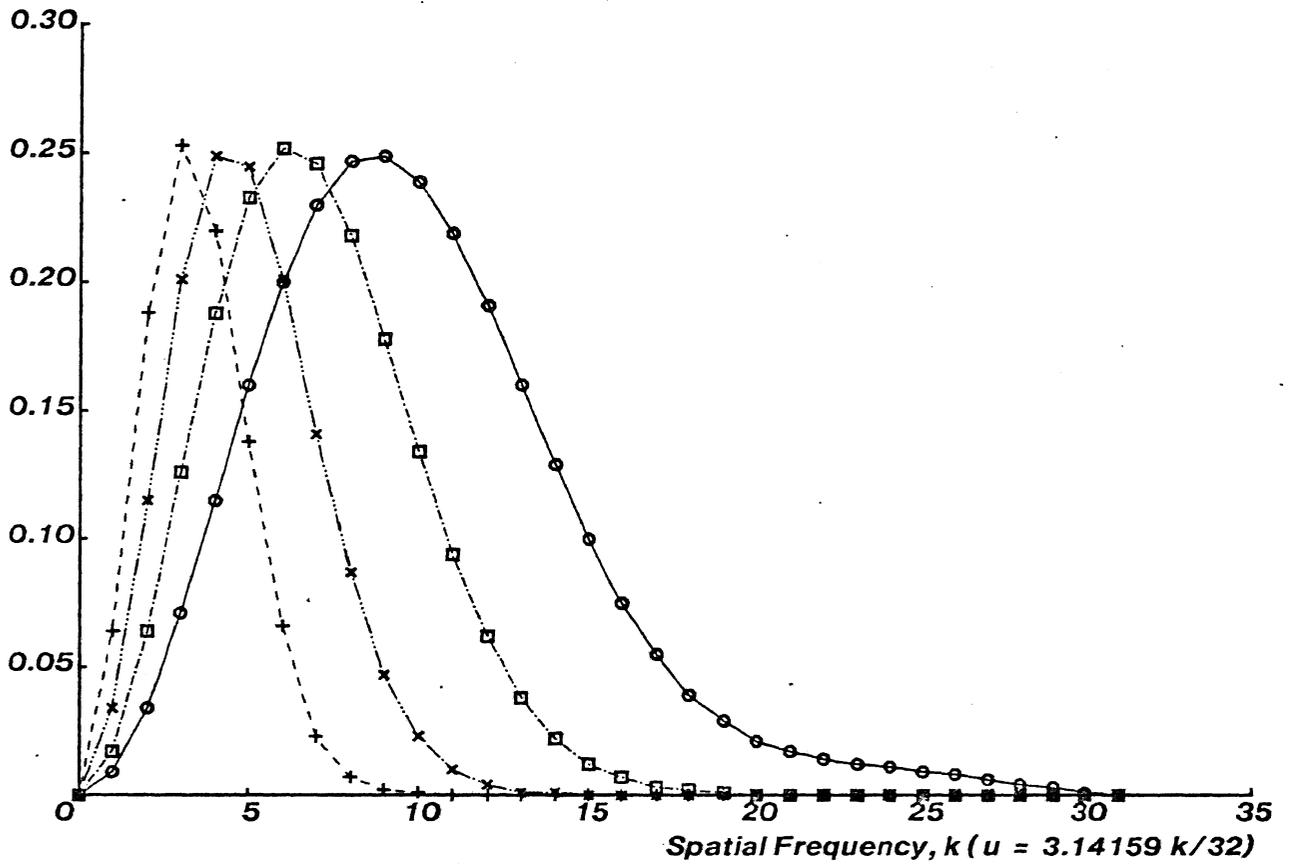


Figure 6-12: U Axis Of Transfer Functions for Band-Pass Filters from Levels 1 Through 4. $u = 2\pi k/64$

-.001488
 -.003150 -.006669 -.008564 -.006669 -.003150
 -.003150 -.010996 -.023278 -.029890 -.023278 -.010996 -.003150
 -.006669 -.023478 -.049280 -.063276 -.049280 -.023478 -.006669
 -.001488 -.008564 -.029890 -.063276 .91752 -.063276 -.029890 -.008564 -.001488
 -.006669 -.023478 -.049280 -.063276 -.049280 -.023478 -.006669
 -.003150 -.010996 -.023278 -.029890 -.023278 -.010996 -.003150
 -.003150 -.006669 -.008564 -.006669 -.003150
 -.001488

Figure 6-13: Filter for High Pass Residue, \mathfrak{B}_0 .

```

-.000002
-.000009 -.000020 -.000025
-.000010 -.000051 -.000131 -.000226 -.000271
-.000020 -.000111 -.000367 -.000798 -.001257 -.001460
-.000010 -.000111 -.000508 -.001461 -.002978 -.004512 -.005172
-.000051 -.000367 -.001461 -.003949 -.004609 -.004849 -.004560
-.000009 -.000131 -.000798 -.002978 -.004609 -.003962 .001282 .004904
-.000020 -.000226 -.001257 -.004512 -.004849 .001282 .017072 .026734
-.000002 -.000025 -.000271 -.001460 -.003684 -.004560 .004904 .026734 .039788
-.000020 -.000226 -.001257 -.004512 -.004849 .001282 .017072 .026734
-.000009 -.000131 -.000798 -.002978 -.004609 -.003962 .001282 .004904
-.000051 -.000367 -.001461 -.003949 -.004609 -.004849 -.004560
-.000010 -.000111 -.000508 -.001461 -.002978 -.004512 -.005172
-.000020 -.000111 -.000367 -.000798 -.001257 -.001460
-.000010 -.000051 -.000131 -.000226 -.000271
-.000009 -.000020 -.000025
-.000002

-.000020 -.000009
-.000226 -.000131 -.000051 -.000010
-.001257 -.000798 -.000367 -.000111 -.000020
-.004512 -.002978 -.001461 -.000508 -.000111 -.000010
-.004849 -.004609 -.003949 -.001461 -.000367 -.000051
.001282 -.003962 -.004609 -.002978 -.000798 -.000131 -.000009
.017072 .001282 -.004849 -.004512 -.001257 -.000226 -.000020
.026734 .004904 -.004560 -.003684 -.001460 -.000271 -.000025 -.000002
.017072 .001282 -.004849 -.004512 -.001257 -.000226 -.000020
.001282 -.003962 -.004609 -.002978 -.000798 -.000131 -.000009
-.004849 -.004609 -.003949 -.001461 -.000367 -.000051
-.004512 -.002978 -.001461 -.000508 -.000111 -.000010
-.001257 -.000798 -.000367 -.000111 -.000020
-.000226 -.000131 -.000051 -.000010
-.000020 -.000009

```

Figure 6-14: Impulse Response of Level 1 Band-Pass Filter

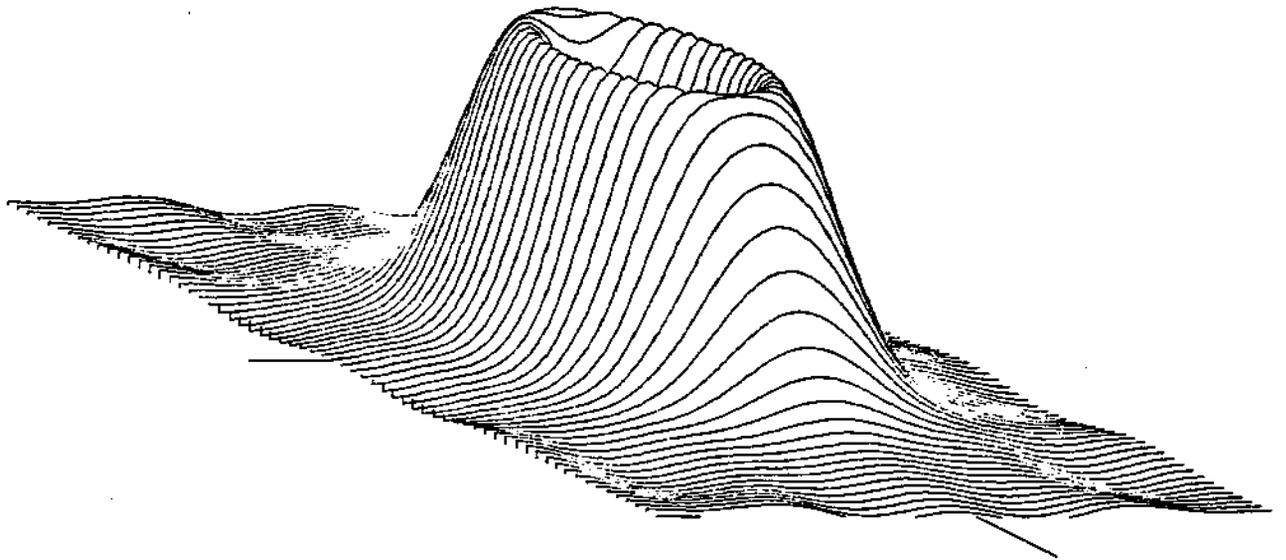


Figure 6-15: $B_L(u,v)$, The Transfer Function of the Level 1 band-pass Filter

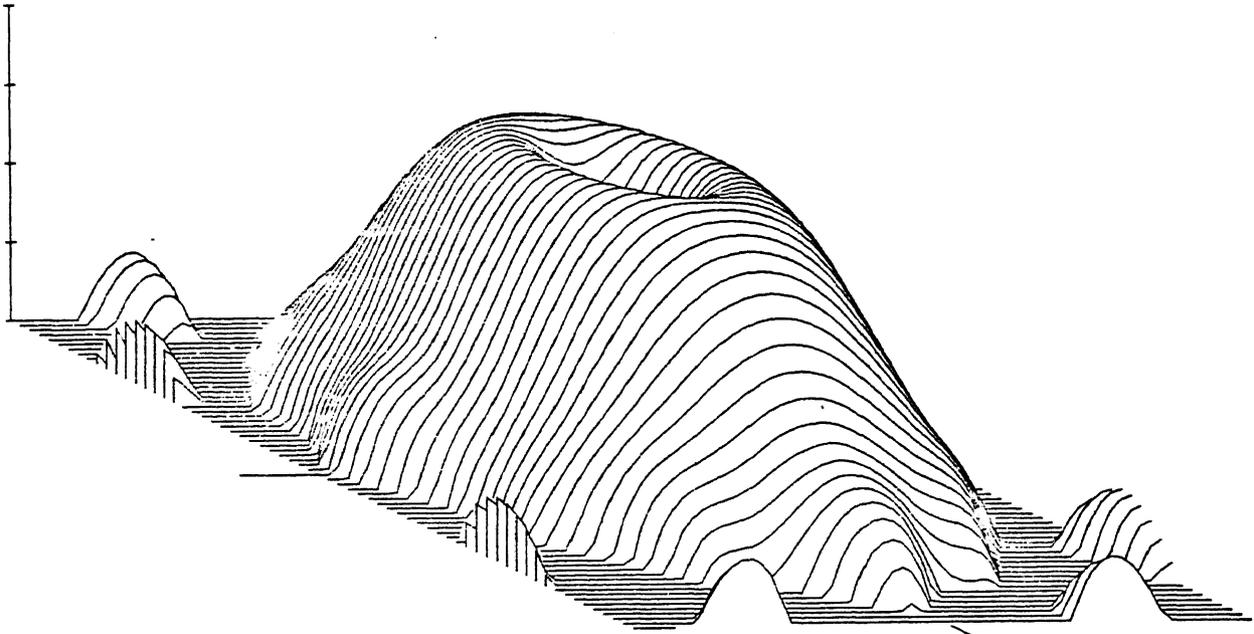


Figure 6-16: $20 \text{ Log}_{10}[B_1(u,v)]$, The Transfer Function of the Level 1 Band-Pass Filter Plotted in dB Scale, shown at left in increments of -10 db, spans -40 dB

6.5.5 Filter at Band-Pass Level 2

The impulse response of the filter at band-pass level 2 requires a 32 column by 32 row table to enumerate. Rather than fill two pages with these coefficients we show its transfer function in figure 6-17 below. The formula for this filter is

$$b_1(x,y) = g_o(x,y) * g_o(x,y) - E_{\sqrt{2}}\{g_o(x,y)\} * g_o(x,y) * g_o(x,y)$$

Figure 6-18 shows a plot of $B_2(u,v)$ in dB, with a scale spanning -80 dB.

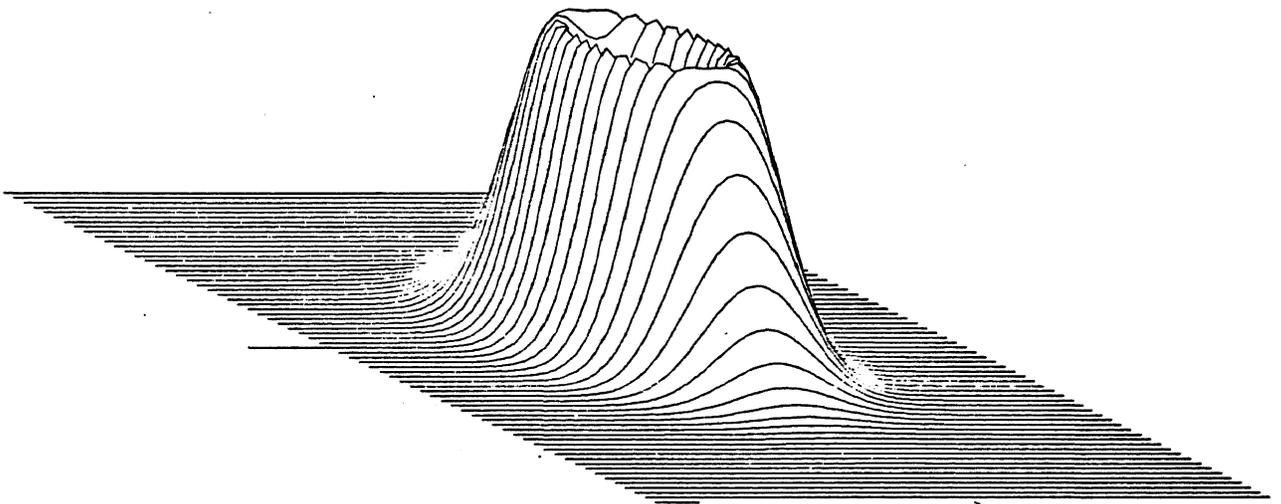


Figure 6-17: $B_2(u,v)$, The Transfer Function of the Level 2 band-pass Filter

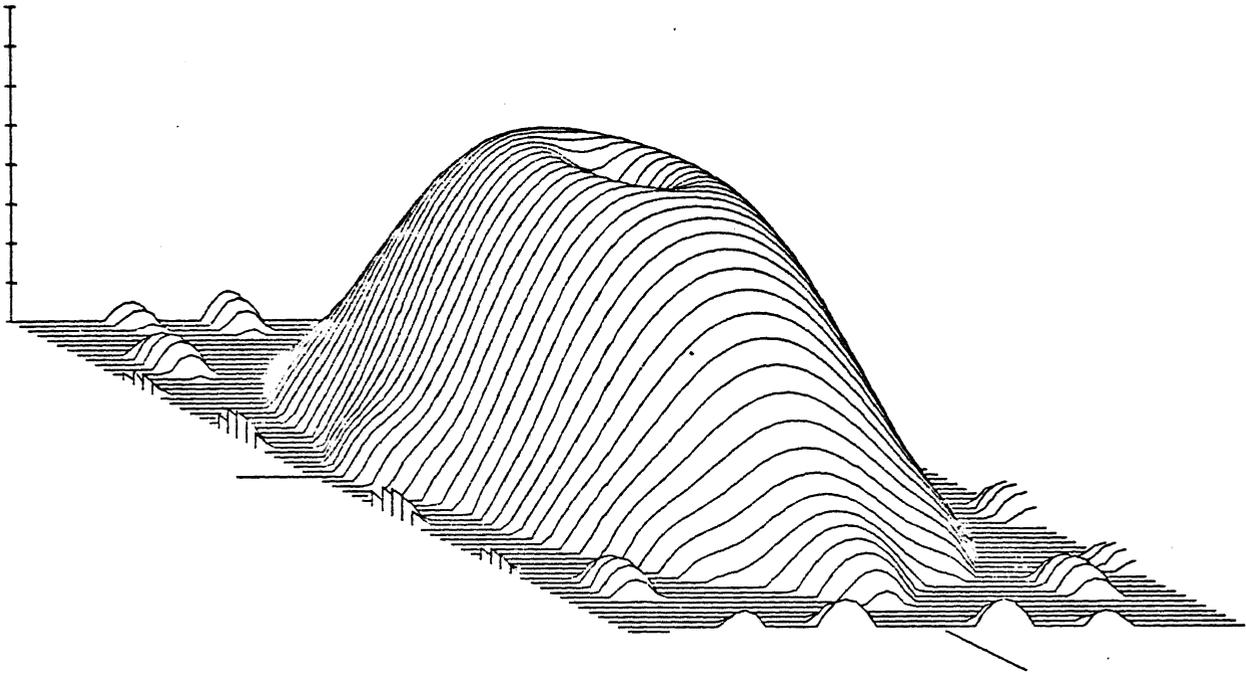


Figure 6-18: $20 \text{ Log}_{10}[B_2(u,v)]$, The Transfer Function of the Level 2 band-pass Filter Plotted in dB Scale, shown at left marks increments of -10 dB to -80 dB

Chapter 7

A Symbolic Representation Based on the Sampled Difference of Gaussian Transform

The previous two chapters described techniques which could be considered within the domain of digital signal processing. In order to demonstrate the usefulness of these techniques, it is necessary to show that the filtered image signals can be used to construct a structural representation of an image. This chapter will describe such a technique. These algorithms were developed to demonstrate the usefulness of the sampled DOG transform, and to explore and develop the principles for using the transform to form a structural representation of gray scale images for object recognition and stereo matching.

The algorithms described below were designed to be local. As with the transform itself, they can be implemented in parallel. Rather than try to develop a single monolithic process that would construct the description, the process was broken down into a series of stages, and a number of competing ideas were evaluated for each stage.

The process was broken into the following stages:

1. Identify and link ridge points (P-nodes) and local peaks (M-nodes) at each band-pass level;
2. Remove small loops and fix short broken connections in the P-paths at each level;
3. Connect together peaks at adjacent levels (M-paths);
4. Use 2-D ridge points (P-nodes) as candidates to find 3-D ridge points (L-nodes) in the three dimensions (x,y,k);

The result of this process is a tree-like graph which contains four classes of symbols:

- P: Points which arc on a ridge at a level.
- M: Points which arc local maxima at a level.
- L: Points which arc on a ridge across levels (i.e. in the three space (x,y,k)).
- M*: Points which arc local maxima in the three space.

Every uniform (or approximately uniform) region will have one or more M*'s as a root in its description. These are connected to paths of L's (L-Paths) which describe the general form of the region, and paths of M's (M-Paths) which branch into the concavities and convexities. The shape of the boundaries are described in multiple resolutions by the paths of P's (P-Paths). If a boundary is blurry, then the highest resolution (lowest level) P-Paths are lost, but the boundary is still described by the lower resolution P-Paths.

Before launching into a discussion of how the values from the Sampled Difference of Gaussian (SDOG) transform may be mapped into symbols, a word about one of the terms used below. The SDOG transform produces values at discrete points in a finite space (x,y,k). Each point in this space has the potential to contain a symbol. When a symbol is assigned to a point, a certain amount of additional state information is encoded at the point. To avoid confusion between the words point and pointer, each point in the space (x,y,k) will be referred to as a sample, when speaking of only the band-pass value, or as a "node" when describing the various labels, flags and pointers assigned at a sample point.

7.0.1 Information Stored at Each Node

In the implementation that is described in this chapter, nodes were subdivided into the fields shown in table

Filter Value	8 bits
Direction	8 bits
E,B,S,*,L,M,P	1 bit flags
P Pointers	8 one bit pointers
Label, U, D	6 bit Symbol ID, Pointer bits Straight up and down
UP (to k+1 level) pointers	For L and M paths (8 Bits, 1 for each neighbor)
Pointers to SAME level	For L and M paths
DOWN (to k-1 level)	For L and M paths

Table 7-1: Fields of a 64 Bit Node

The first 8 bit sub-field holds the value from the Sampled DOG transform. The direction sub-field contains the result of a directionality measure that was employed in early versions of the representation. This number is between 0 and 179 degrees. Next are seven 1-bit flags whose meanings are discussed in the sections 7.2, 7.4, and 7.5. The next subfield contains the 8 pointer bits for connecting P nodes. Each pointer corresponds to one of the adjacent 8 neighbors. The neighbor to the right is pointed to by the pointer at bit 0. Neighbor numbers increase in a counter-clockwise direction. (A number of the algorithms below do modulo 8 arithmetic on the P pointers.) The next

subfield is a 6 bit symbol ID that is assigned based on the configuration of ridges around the node. There are then two 1-bit fields which act as pointers for the L and M paths. The U field can be set to point to the neighbor directly above if that neighbor exists. The D bit can be set to point to the neighbor directly below (at the $k-1^{\text{st}}$ level). The "UP" field contains the pointers for the L and M paths that can point to the 8 neighbors at the $k+1^{\text{st}}$ level. The "SAME" field contains pointers for L paths that can point to any of the adjacent 8 neighbors at the k^{th} level. The "DOWN" subfield points to the 8 neighbors below (at the $k-1^{\text{st}}$ level) for representing L and M paths.

7.0.2 Meaning and Purpose of Peaks and Ridges

Section 3.1 showed that a 2-D sampled correlation is equivalent to a 2-D sequence of inner products between the filter and the neighborhoods centered at the sample points. An inner product has its largest possible value when the two functions are identical. It is also a good measure of how similar two functions are. For example, in communications theory an inner product is used to tell how much of the energy in a received signal is described by a basis function [Wozencraft 65]. Thus a local peak in a band-pass image indicates a local point where the image signal most resembles the impulse response of the band-pass filter.

It is possible for a two dimensional signal to maintain a large amplitude along a line or a curved path such that all of the neighboring values are smaller. When this happens in the band-pass images from a DOLP or SDOG transform it means that the impulse response of the band-pass filters are a best fit to the gray-scale form in the image at a sequence of points. Such a sequence of points is called a ridge. A ridge could be loosely defined as a 1-D sequence of points in a 2-D signal along which the function value is larger than any neighboring points.

Both ridges and peaks occur in each of the band-pass signals produced by a DOLP transform. This chapter shows that the appearance of an object in an image can be represented by encoding the ridges and peaks from all of the band-pass images from a SDOG transform. To the extent to which the band-pass signal can be reconstructed from knowledge of the position and magnitude of the peaks and ridge paths, this encoding is approximately reversible. This chapter also shows that the concepts of peak points and ridge paths can be extended to the third (or k) dimension, that is between band-pass levels. These peak points and ridge paths in the (x,y,k) space provide sufficient information to uniquely represent descriptions of the 2-D appearances of objects. Chapter 8 shows how this a representation can be used to efficiently match 2-D appearances, despite changes in size, 2-D orientation, or position of the object relative to the camera.

7.1 Phenomena in Each Band-Pass Image

This section describes the manner in which peaks and ridges occur in each band-pass image of a SDOG transform. Section 7.4 describes peaks and ridges in the 3-D space (x,y,k). The phenomena described in these sections are illustrated with filter output from uniform intensity rectangles. These artificial shapes have simple descriptions and yet illustrate the principles on which this representation is based. Examples of the descriptions of the images of real objects are presented in later sections and in the next chapter.

7.1.1 The SDOG Band-Pass Impulse Response

In the following discussions, it is helpful to recall the form of the impulse response of the band-pass filters implemented by the sampled DOG transform. The zero crossings and the center row of this impulse response are illustrated below in figure 7-1. The impulse response is circularly symmetric. The coefficient along any line passing through the origin will resemble the cross-section shown on the right in figure 7-1. The impulse response consists of a positive center lobe, surrounded by a negative side lobe. The sum of the coefficients is zero. The response at any point may be thought of as the sum of the weighted points under the center lobe minus the sum of the weighted points under the outside side lobe.

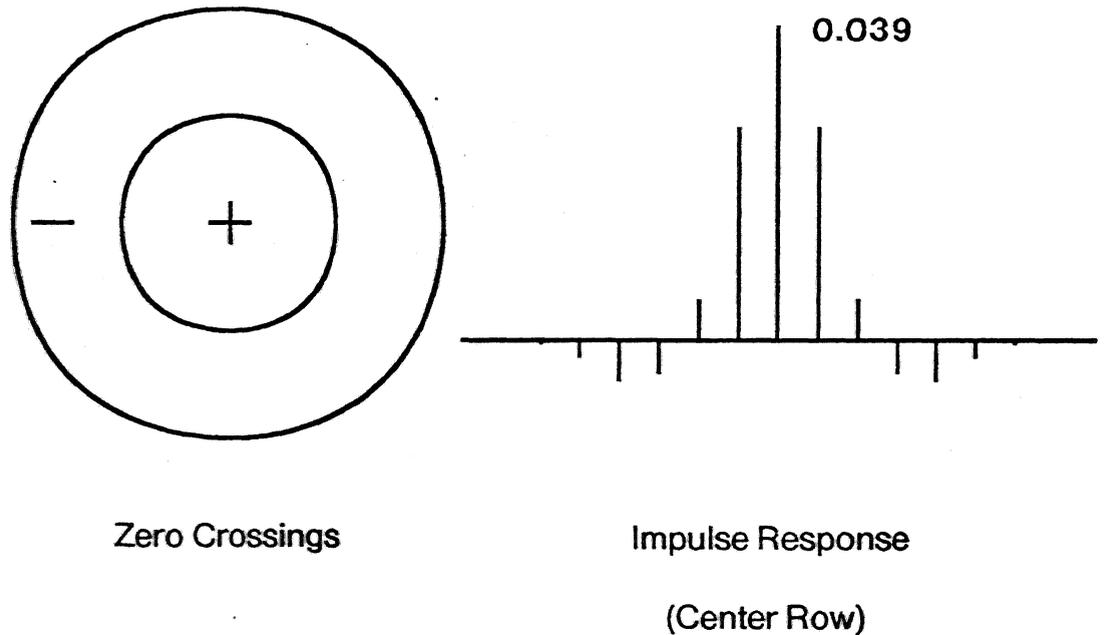


Figure 7-1: Impulse Response of Band-Pass Filter

Ever
descrip
region,
the bot
blurry,
by the

Before
(SDOC
SDOG
has the
additic
and pc
band-p
sampl

7.0.1

In
show

The
cont
repr
mea
for c
the
dire

7.1.2 Edges of Large Regions

Let us start by considering the response of the band-pass filters at the boundary of a much larger uniform region. Consider a square whose side length is much larger than the diameter of the band-pass filter, and whose picture elements are of a larger value than the surrounding background. Let us examine the response of the filter along a line which is perpendicular to the side of the square and passes through the center. This response is illustrated in figure 7-2.

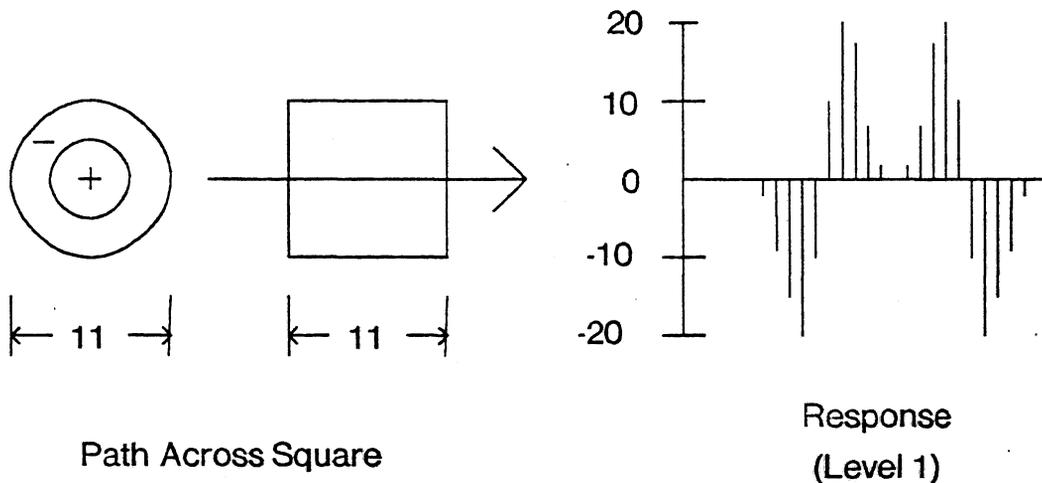


Figure 7-2: Response Across Center of a Square

When the filter support is totally in the uniform background region the response is zero. As the filter's negative side lobe begins to overlap with the square, the inner-product becomes negative. As the edge of the positive center lobe reaches the edge of the square, the inner-product reaches a negative minima. The response climbs through zero as the positive center lobe overlaps with more of the square. Just before the positive center lobe completely overlaps the square, the response will reach a positive maximum and begin to drop. The drop continues until the filter is completely within the square and the response has tapered to zero. Thus the edges of the square result in a pair of peaks of opposite sign, on either side of the edge. The distance of the peaks from the edge can depend on how sharp the edge is, and will occur at approximately $2/3$ the filter radius on either side of the edge. If the edges are blurred at the resolution described by the filter, the amplitude of the peaks will be decreased, the width will be increased, and the peaks will tend to be a little further apart.

The fact that a negative response occurs outside of the square is interesting. Any approximately uniform region will have a negative ridge surrounding it. Artists refer to a similar phenomenon in the human visual system as "negative shape".

7.1.3 Convex Protrusions: The Corner

The filters tend to respond to concave and convex protrusions by producing a peak. When linked between levels, these peaks form an M-path which describes the shape of the protrusion. As an example of a convex protrusion, consider the uniform square described in the previous section. Consider the response along a line which is parallel to and about half the filter radius below the upper edge of the square as shown in figure 7-3.

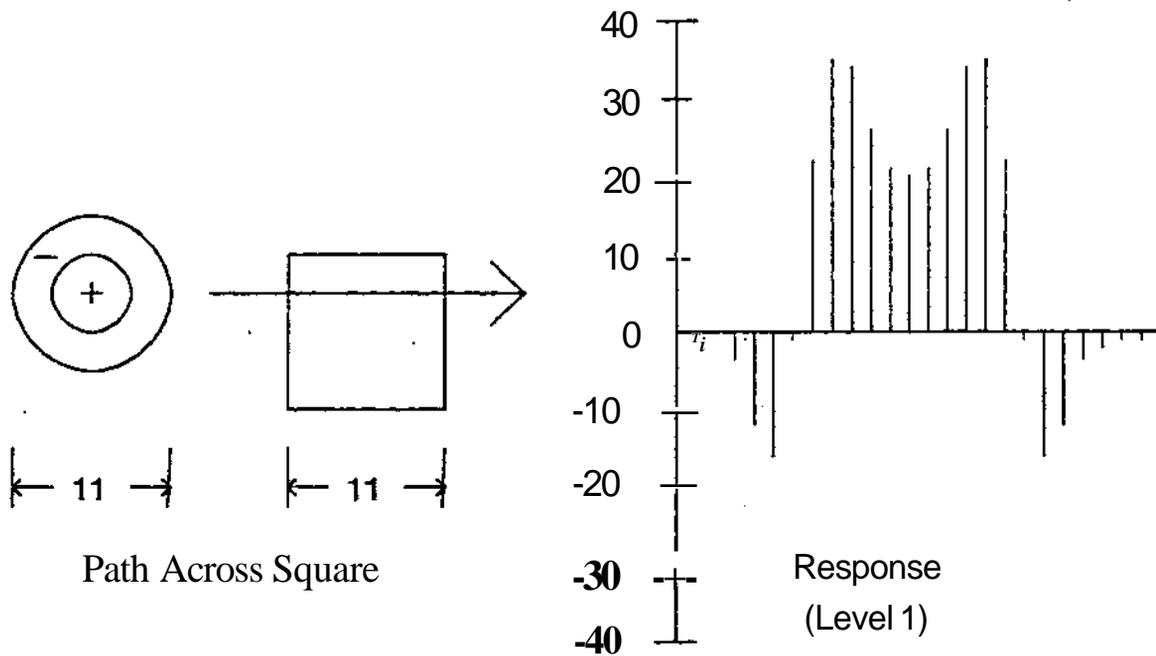


Figure 7-3: Response at Corner of a Square

As before, the filter response is initially zero. As the negative sidclobc moves over the corner of the square, the response will go negative until a minimum is reached. The amplitude of this negative peak will be smaller than for the negative edge at the center of the square. This is because less of the negative side lobe is overlapping with the square. As the positive center lobe comes over the square, the response will rise through zero to a positive maximum. The amplitude of this peak will be approximately twice the amplitude of the positive peak at the center of the square. Again, this is because less of the negative side lobe overlaps with the square. To the right of the positive maximum, the response will decrease to about half of its maximum value. These points arc along the positive ridge that is inside the boundary of the square. The response is symmetric about the middle of the square.

Peaks, such as the one described above, will occur whenever there is a protrusion. Protrusions which have sharp straight edges appear the same over a range of scales. For such protrusions the height of the peaks at several adjacent band-pass levels will be approximately the same. If the

protrusion does not have sharp straight edges, then there will exist levels at which the peak is larger than the peak at adjacent levels. An example of such a shape would be a square in which the corners are rounded.

7.1.4 Across a Long Thin Rectangle

Let us consider the response of a filter along a line crossing a rectangle (or bar) whose width is approximately the same as the radius of the filter's positive center lobe. This situation is illustrated in figure 7-4.

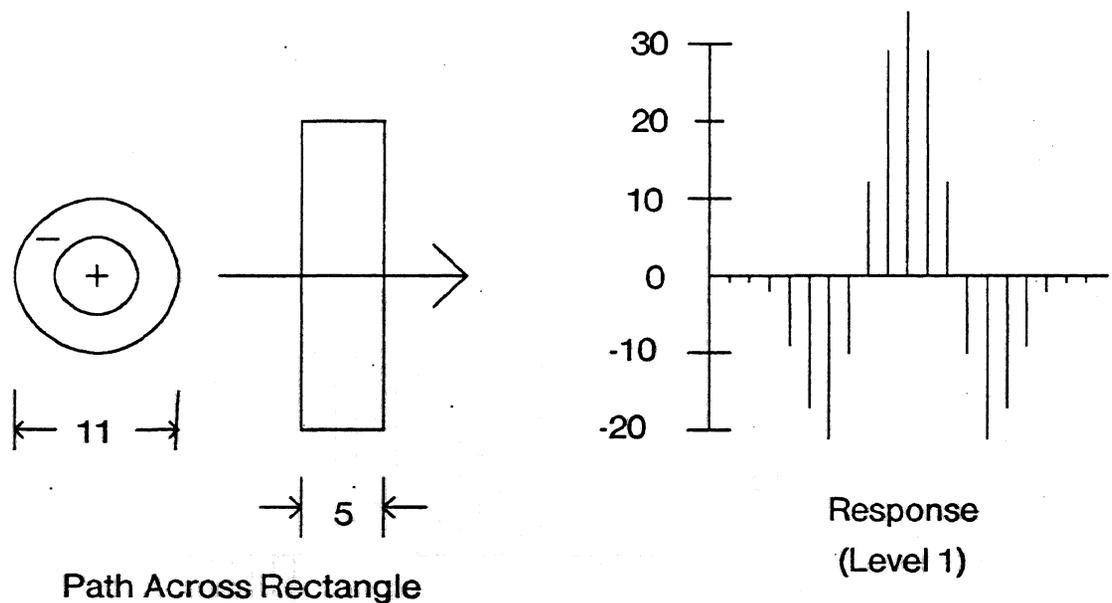


Figure 7-4: Response of Filter Across a Rectangle

As with the first square example, the response starts out as zero, and falls to a negative peak as the side lobe overlaps with the rectangle. However, since the side lobe passes beyond the rectangle as the center lobe comes over the bar, the positive response will rise faster and reach a peak which is approximately twice that of the positive edge of the square. The response is symmetric about the center of the rectangle. What is important about this example is that the response of the filter whose positive inner lobe is the same width as the rectangle will be larger than the response for filters which are larger or smaller. Such a ridge results in a path of L-nodes; that is, a ridge between band-pass levels. The index of the level at which the L path occurs gives an estimate of the width of the rectangle.

7.1.5 At the Ends of the Rectangle

Let us now consider the response of the same filter along the long axis of the same rectangle. This is illustrated by figure 7-5.

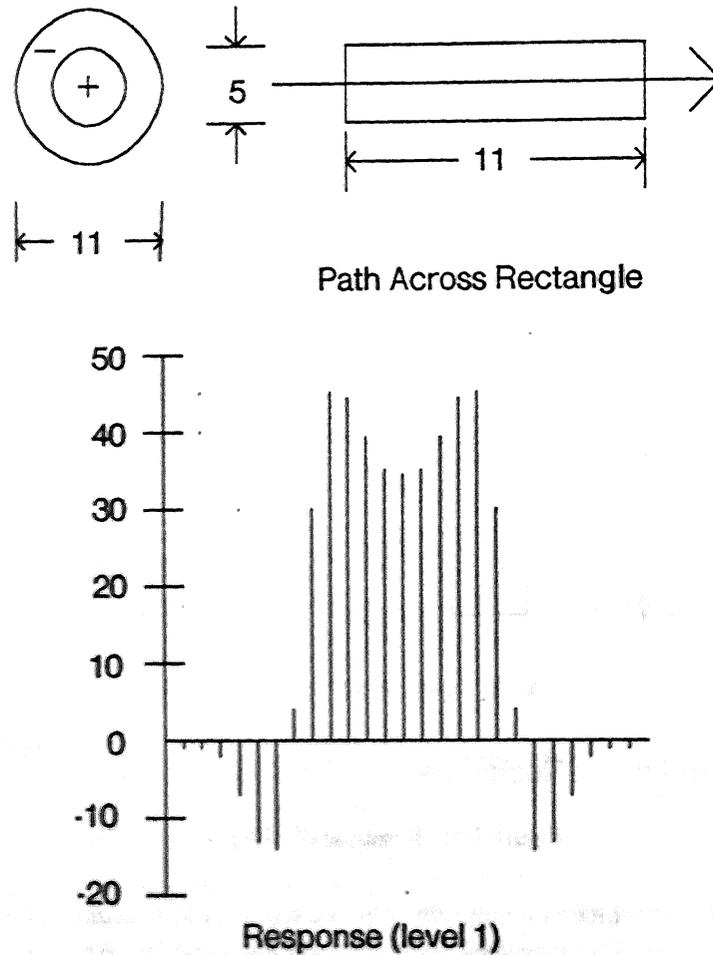


Figure 7-5: Response of Filter Along a Rectangle

The negative minimum that occurs as the filter comes over the end of the rectangle will be smaller than the negative minimum beside the rectangle, because less of the negative side lobe will be overlapping with the rectangle. As the positive center lobe comes over the end of the rectangle, the response will rise to a positive maximum which is even larger than for the center of the rectangle. This is because at the end of the rectangle, only about a quarter of the negative side lobe overlaps with the rectangle, whereas in the center almost half of the negative side lobe overlaps. Thus at the ends of a rectangle, a local peak occurs. For the filter whose center lobe most closely fits the

rectangle, the amplitude of this peak will be larger than for filters that are smaller or larger. Such a peak will be detected as a peak between levels, and labeled as an M^* . The levels below it will contain an M path which splits into two parts, one for each corner. Above it another M -path will lead to the center of the rectangle. This M -Path may or may not join with one from the other end of the rectangle, depending on both the length to width ratio, and the difference in gray level between the rectangle and the background.

7.1.6 A Square Which is Smaller Than the Filter

As a final illustration, let us consider the response of a filter to a square whose size is approximately the same as the positive center lobe of the filter. This is illustrated by figure 7-6.

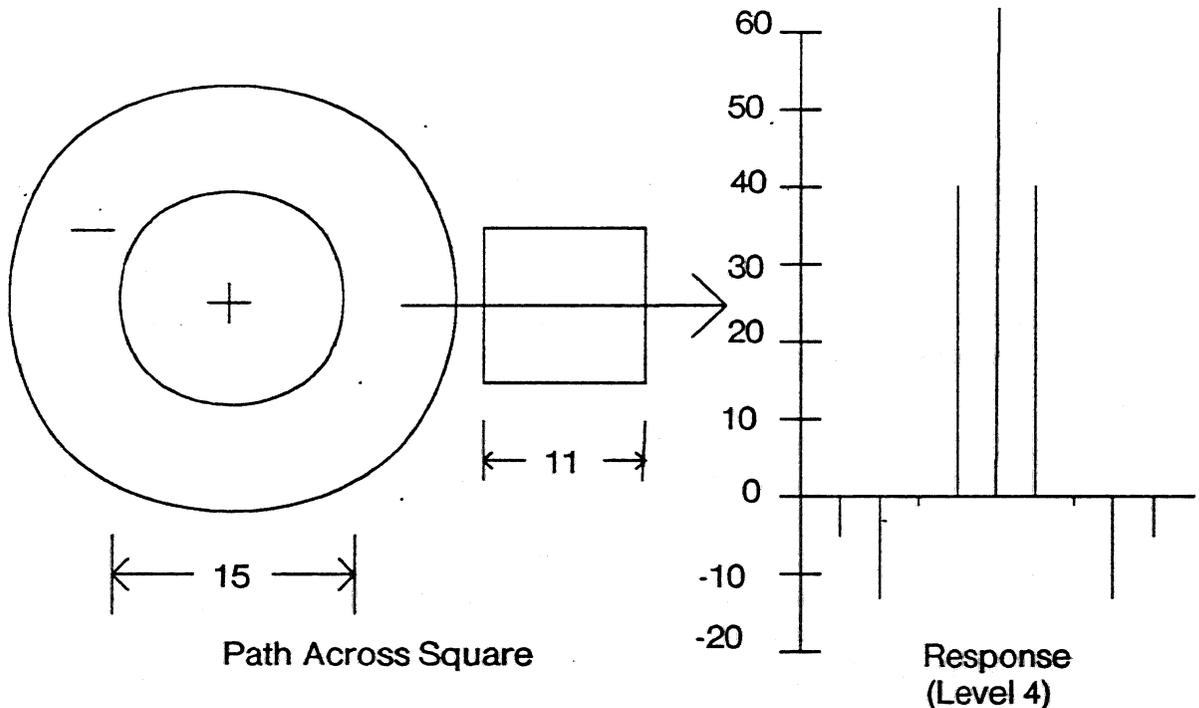


Figure 7-6: Response of Filter To a Square

As with the earlier examples, there is a negative ridge surrounding the square. As the center of the filter moves over the square the response rises to a strong peak. The height of the peak will be approximately four times the amplitude of the negative ridge outside the square. The peak that occurs for the filter whose center lobe just covers the square is the largest response to the square which any of the filters will have. This peak is detected as an M^* point, and serves as a root for the graph which represents the square. An M Path will extend above this peak for several levels. Below the peak an M Path will split into four parts, one for each corner.

7.2 Peak and Ridge Path Detection at Each Band-Pass Level

Detecting a local peak in a band-pass level from the SDOG transform is simple because of the smoothness given by the band-pass impulse response. Unambiguous detection of the path of a ridge with an algorithm that may be implemented in parallel has proved to be a more difficult problem.

It was originally believed that the detection of points on a ridge would require measuring the direction of least change (local directionality) and then finding the local ridge by scanning perpendicular to that direction. Several techniques for measuring local directionality were investigated. A particularly reliable and efficient measure based on a 4 point DFT of the inner-product from 1-D filters at four directions will be described in a separate report.

The simplest measure of local directionality at a point is to compare the filter output at each of the 8 neighbors. At any point, the directions at which the largest neighbors exist is the most likely direction of the nearest ridge. By definition, the largest neighbors of points on a ridge are also points on a ridge. This simple principle serves as a basis for the ridge detection algorithm described below. Because it is not based on a costly directionality measurement function, this algorithm is simpler to program and executes faster than any of the other algorithms for ridge detection that were investigated.

None of the algorithms that were developed for detecting and linking ridge path points always produced unbroken paths. The problems with these algorithms is that the data consists of fixed point numbers which exist at discrete locations. While the algorithm described below was sufficient for the purpose of demonstrating this thesis, there is room for further research.

7.2.1 Detecting Local Peaks

Local peaks (positive maxima and negative minima) at a band-pass level are easy to detect. A local peak (M) is defined as any sample in a band-pass level for which none of the adjacent 8 neighbor samples has a value of the same sign and larger magnitude. Note that this definition allows adjacent samples with the same value to both be detected as peaks. This situation occurs because of the fixed point quantization and is handled by interpreting adjacent peak points as part of a single peak. If two samples have the same value, and only one of them has an adjacent neighbor with a larger value, then neither sample is labeled as a peak.

By this definition, an area of uniform filter output is composed of all peaks. Only a constant signal will produce a uniform response over an area in a band pass image, and the values in this response are zero. Such areas are easily detected and excluded. It is possible to have small regions of width < 4 which have a constant value if the amplitude is very small (e.g. < 3). This is because of quantization with fixed point numbers. This problem is avoided by not allowing a point where the magnitude is less than 10 to be labeled as a peak.

It is mentioned above that a situation can occur where two adjacent samples have the same value,

and only one of the samples has a *larger* neighbor. An example of this occurs in figures 7-8 and 7-9 below at row 54 column 142. Such false peaks are eliminated by setting the E flag for any M-node which has an equal valued neighbor. A second pass is made through the image during which the M and E flags are cleared for any M-node which has its E flag set and is not adjacent to another M-node.

Thus peaks are detected by comparing a value to its neighbors, and to the quantization threshold. If implemented by itself, this algorithm requires 8 references to the image array for each sample. This simple detection procedure is easily implemented as part of the more complex ridge path detection procedure described below.

7.2.2 Detecting Ridge Paths at a Band-Pass Level

This section describes an algorithm for detecting samples which are on a ridge in a 2-D band-pass image. This algorithm is based on the principle that the largest neighbors of a point on a ridge are also on the same ridge. Thus any pair of samples which point to each other as largest neighbors are on a ridge (detected as P-nodes).

The algorithm for detecting ridge path nodes consists of two stages and requires 8 "pointer" bits. The following is an informal explanation of this algorithm: The eight neighbors of a point are assembled into a circular list, with the nodes of the opposite sign marked as zero. This list is then scanned looking for local maxima. For each local maxima, the corresponding pointer bit is set. After this process has been executed for every node in the level the second stage commences. At this stage, at each node, any neighbor for which the pointer has been set is tested. If the neighbor has its corresponding pointer (pointing back) set, then both points are labeled as ridge nodes, and marked by setting a P flag. By deleting all unanswered pointers, the ridge nodes are left with a two way linked list giving the path of the ridge.

This algorithm consists of the following steps:

- Stage 1: At each node:

1. Make a circular list of the absolute value of the 8 neighbors.
2. For any neighbor where the sign of the value is different then the center node, enter a zero.
3. Scan the list (A finite state process works nicely here). For any list element for which there is no larger adjacent value, set a pointer for that neighbor.
4. Store the pointers for the next stage.

- Stage 2: For each point:

1. Scan the pointers. For each pointer that is set, get the pointer of that neighbor that points back.

2. If this pointer is also set, mark the node as a P. Otherwise delete the pointer.

The two way linked list of pointers is used in later processes.

This process is illustrated by the examples shown in figures 7-7 through 7-9 below. Figure 7-7 shows the raw values filter values from level 2 of the piston rod test image, columns 141 through 152, rows 47 through 57. Note that this data is on a $\sqrt{2}$ sample grid.

	Values for nodes - Level 2										rod.dat		raw data		
	141	142	143	144	145	146	147	148	149	150	151	152			
47	13		7		-3		-6		-11		-12				
48		-2		-9		-15		-18		-20		-19			
49	-5		-18		-19		-17		-18		-18				
50		-18		-14		-7		-3		-1		-3			
51	-16		-11		1		11		14		14				
52		-3		8		13		15		17		15			
53	0		14		15		8		1		1				
54		14		7		-9		-18		-19		-16			
55	12		1		-20		-29		-36		-38				
56		0		-26		-38		-38		-39		-43			
57	0		-27		-37		-29		-24		-23				

Figure 7-7: Values at Level 2 of rod.swf

Figure 7-8 shows the pointers that are created by the first stage of the ridge path detection process. The pointers are marked by the symbols { / ! \ - }. Also shown is the symbol M wherever a peak has been detected.

The result of the second stage is shown in figure 7-9 below. At this stage the ridge path points have been marked with a P and only answered pointers are not deleted.

7.2.3 Eliminating Small Loops

In most cases the algorithm described above produces a unique path of largest values. Occasionally two points occur with the same value such that the direction between them is perpendicular to the ridge path. This occurs because a continuous ridge is represented by fixed point numbers at discrete sample points. This phenomenon becomes more likely as the signal intensity becomes weaker.

Such small loops complicate the programming for later stages of the process. Fortunately, they are easily detected and eliminated by deleting one of the sub-paths.

The set of all such loops involving 3 or 4 points may be divided into three classes by grouping together those that are rotational equivalents. These classes are listed in figure 7-10 with the equal samples shown as "E" and the other samples as "P". Note that in classes 1 and 2 the loop on the right is on a $\sqrt{2}$ sample grid.

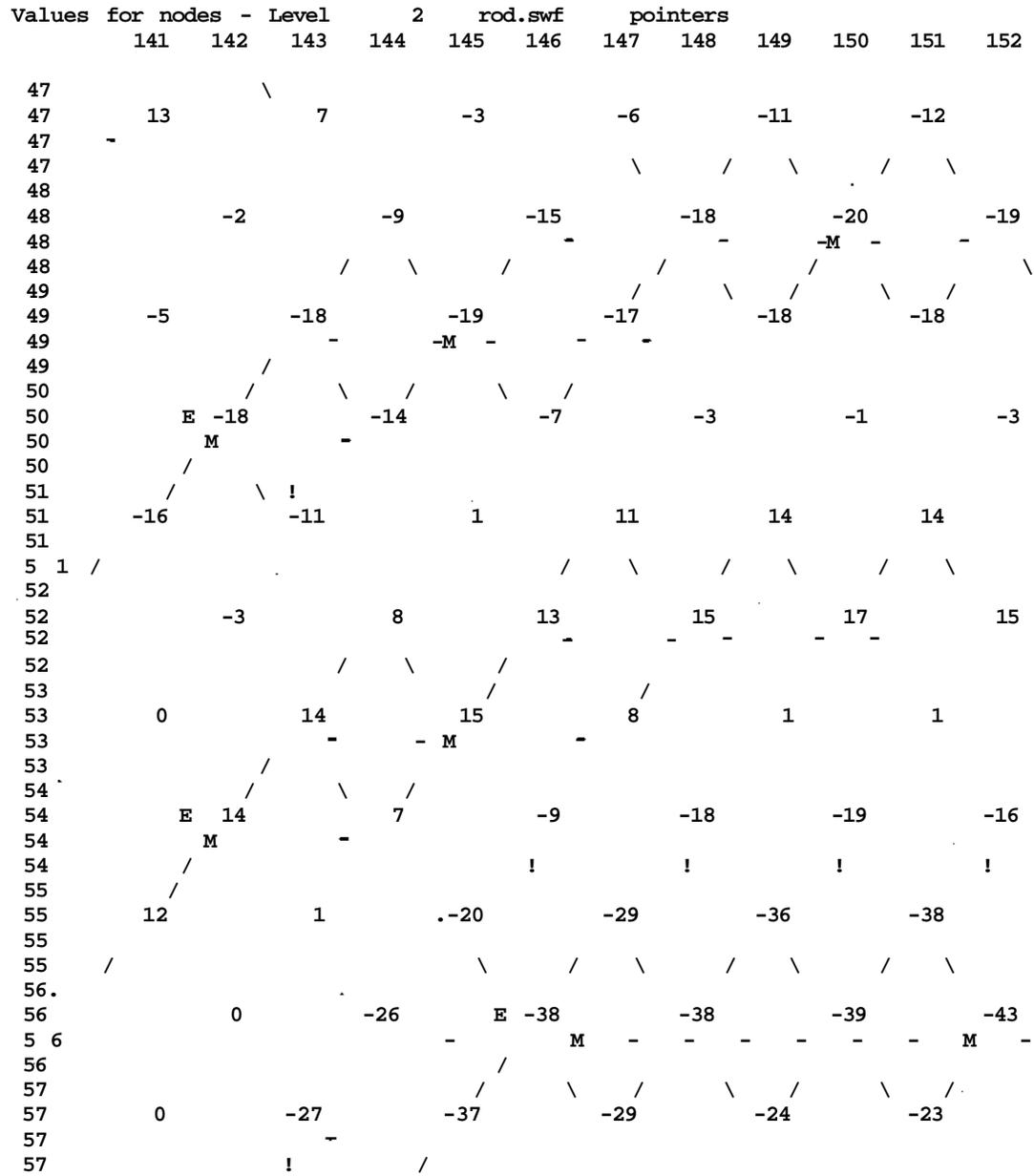


Figure 7-8: Pointers From First Stage of Ridge Path Detection Procedure

The possible presence of such a loop is signaled by a sample having a pair of pointers in adjacent directions. When such an adjacent pair of pointers is detected the node is marked by setting its S flag. A second stage process then makes a test of the directions of the pointers in the next sample in the path. Loops are broken by deleting the P flag and the pointers of one of the equal valued samples. The sample that is deleted is chosen such that path length is kept as short as possible and as straight as possible. When these two criteria are not sufficient to choose an equal valued point to be removed, the more clock-wise sample is chosen arbitrarily.

Figure 7-11 shows a path that includes a small loop. The nodes with adjacent pointers are marked

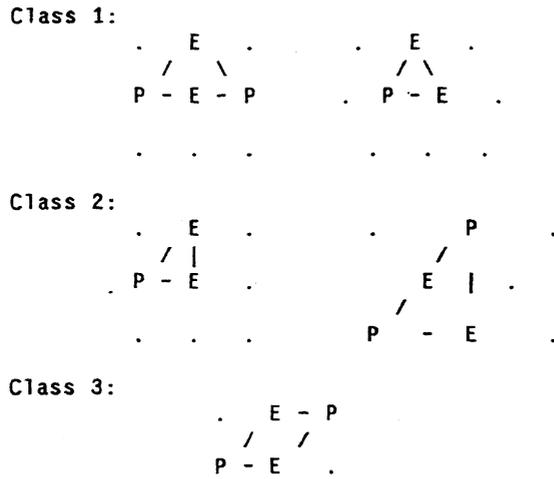


Figure 7-10: Classes of Small Loops

Level	49	51	53	55	57	59	61
61							!
61	-26	-22	-16	-5	8	24	37
61							P
61							!
63							!
63	-29	-28	-24	-18	-7	11	23
63	P	--	P				P
63							!
65							!
65	-18	-25	-29	-24	-15	1	15
65			S	P			PB
65			!	!			
67			!	!			
67	2	-17	-29	-29	-20	-4	10
67			P	P			
67			!	!			
69			!	!			
69	20	-4	-25	-33	-22	-6	9
69			S	MP			
69			!	!			
71			!	!			
71	26	3	-23	-32	-25	-8	8
71				P			
71				!			

Figure 7-11: Ridge Path Containing Small Loop

7.2.4 Unterminated Ridge Paths

In most cases a ridge path will terminate at both ends at an M node. There are, however, several situations where this does not occur. In the following sections we describe these situations and how they are treated.

Whenever a node has only one P pointer, a flag, called the B flag (for Broken) is set. A B node can occur for the following reasons:

Values for nodes - Level				3	rod.swf		Small Loops Removed	
49	51	53	55	57	59	61		
61							!	
61	-26	-22	-16	-5	8	24	37	
61							P	
61							!	
63							!	
63	-29	-28	-24	-18	-7	11	23	
63	P	--	P				P	
63							!	
65							!	
65	-18	-25	-29	-24	-15	1	15	
65			P				PB	
65								
67								
67	2	-17	-29	-29	-20	-4	10	
67				P				
67				!				
69				!				
69	20	-4	-25	-33	-22	-6	9	
69				MP				
69				!				
71				!				
71	26	3	-23	-32	-25	-8	8	
71				P				
71				!				

Figure 7-12: Path After Removal of Small Loop

1. When a ridge path is broken, usually because of an abrupt change in the ridge amplitude. Such cases are an error and are handled by attempting to extend the path as described in section 7.2.5 below.
2. A "Spur": This is an extra point which occurs to the side of a ridge path, usually connected to an M node. Spurs are deleted only when they are a single node and not connected to an M node, as described by section 7.2.7.
3. A Fading Ridge: This can legitimately occur for some patterns. For example, when a bar ends by fading into the background, or when a large area has square wave "teeth" that are longer than they are wide.
4. An Isolated Pair. This is the case when two P nodes are connected to each other and only each other. This can be the result of a small region which is described at lower levels and should be ignored at this level, or it can occur at a saddle point along a ridge.

The action which is taken at a B node is first determined by the number of pointers which the connected neighbor of the B node has. The following situations occur:

1. One pointer: This signals an Isolated Pair.
2. Two pointers: This usually indicates a break along a ridge path, although a fading path or a long spur might be the cause. Which of these is the case is determined by attempting to extend the path as described in section 7.2.5 below.
3. Three (or more) pointers: The B node is a spur.

7.2.5 Repairing Broken Paths

Under some conditions the amplitude of a ridge can make a sharp increase or decrease. Such a rapid transition can result in a ridge path point not being detected or in a pair of pointers not being formed along a ridge. An example in which this occurs in 4 places is shown in figure 7-13. The pointers are used in the process for detecting the L-nodes. Thus it is necessary to correct such broken paths.

A one pass process is executed for each node with its B flag set which is connected to a node with 2 pointers. This process attempts to extend the ridge path for up to 2 samples. If it is possible to close the path with samples of the same sign, and without creating an adjacent pointer condition (as defined above), then the path is closed. The algorithm runs as follows:

1. Determine the direction of the single pointer.
2. For the opposite direction, and the two directions adjacent to the opposite direction, get the neighbor node.
3. If any of these neighbors are also a P-node and have the same sign, and linking to that node will not create an "adjacent pointers" condition (see exception below), link to the P-node with the largest magnitude and quit.
4. If none of these three nodes are P nodes, choose the largest of them (with the same sign) and repeat steps 2 and 3. Use the direction between the starting point and the chosen neighbor for choosing the next set of three neighbors.
5. Steps 2 and 3 are repeated twice if the largest neighboring node is always found in the same direction. Otherwise steps 2 and 3 are only repeated once to avoid creating small loops.

Exception: At step 3, an adjacent pointer condition does not inhibit linking to a node if the adjacent pointer points to a B-node. In such a case the the link is made and the B-node is deleted.

Figure 7-13 shows the inner oval region from a piston rod at band-pass level 3 before it is processed by the algorithm to connect broken ridge paths. Figure 7-14 show the result after the extension algorithm. This figure also illustrates that the extension algorithm has a preference for connecting to the adjacent node that has the largest value. The procedure also deleted the B-nodes that remained as spurs after the linking.

7.2.6 Isolated Pairs

The configuration of two P nodes with only 1 pointer (i.e. connected only to each other) is a rare but troublesome one. It usually occurs in areas where the signal is weak, and if extended can often cause a spur of length 2 or 3. It has been observed that when the amplitude of a ridge makes a dip this configuration will occur. In this case, the broken path on either side of the pair of isolated

Values for nodes	Level			3	rod.swf		Ridge Paths
	39	41	43	45	47	49	
67							
67	2	20	29	28	22	2	
67			PB				
67			!				
69			!				
69	16	35	39	38	34	20	
69		PB	MP	P	PB		
69							
71							
71	20	32	30	31	32	26	
71		PB			PB		
71		!			!		
73		!			!		
73	16	27	21	22	26	24	
73		P			P		
73		r			!		
75		I			!		
75	15	21	19	16	25	23	
75		P			P		
75					t		
77		t			i		
77	15	22	20	21	22	20	
77		P			P		
77		1			i	*	
79		J			!		
79	21	29	27	25	31	24	
79		PB			PB		
79							
SI							
81	19	34	38	36	33	21	
81		Pi	HP	P	PB		
81			I				
83			!				
S3	5	26	37	36	27	10	
S3			P				
83			t				
35			\				
85	-14	3	15	15	7	-11	
85			PB				
85							

Figure 7-13: Example of Broken Ridge Paths Before Extension

P-nodes will extend to the P-nodes, thus connecting the broken path. Thus these points are not extended. If they both remain as B nodes after the extension process they are deleted.

7.2.7 Deleting Spure

Occasionally the algorithm for detecting ridge nodes will leave a node which is adjacent to, but not on the path of, the ridge marked as a P^h node. Such P-nodes, which are referred to as "spurs" are easily detected. Spur nodes have only one pointer, and they are connected to a node with 3 pointers. When a spur P-node is detected, if the node to which it points is not an M node, its P flag and pointer are deleted. A spur which points to an M point is retained as a potential point on an L-path.

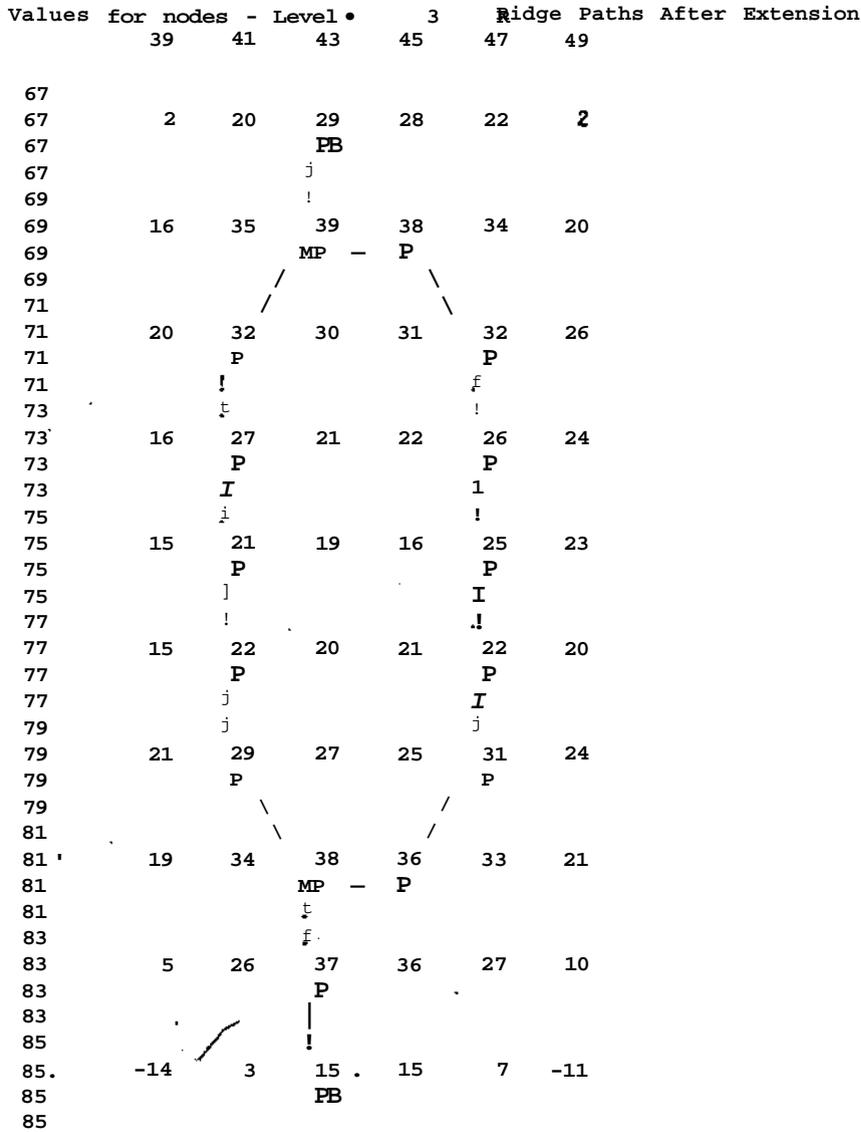


Figure 7-14: Example of Repaired Ridge Paths After Extension

7.3 Phenomena Between Levels in the Transform Space

In this section we review some of the structures that occur in the sampled 1X)G transform of some common forms. We first describe the chain of M-nodes (the M-path) that result from non-elongated forms, ends of elongated forms and corners. We then describe the chains of L-nodes (the L-path) that result from elongated forms and edges. This section describes the purpose and principles behind the algorithms for forming M-paths and L-paths that are described in the next section.

7.3.1 Connectivity of Peaks: M-Paths

In our first experiments with the band-pass detection functions [Crowley 78b] we observed a phenomenon which has proved fundamental to constructing a size invariant representation of gray scale forms from a SDOG Transform. This phenomenon is: Any non-elongated gray scale form will cause a peak at approximately the same location in several adjacent band-pass levels. Furthermore, except for certain degenerate cases, the magnitude of the peaks will rise monotonically across levels to a maximum and then decrease.

These peaks may be detected individually at each level as described above in section 7.1. The peaks may then be linked by starting at each and examining its neighbors in the next upper level for a peak of the same sign. The largest peak may be found during this linking process by comparing the values of the peaks as they are linked. This process, which is called "flag stealing", is described in section 7.4.

To see why this connectivity occurs, let us consider the Sampled DOG Transform of a uniform intensity 11×11 square. Each band-pass filter will respond most strongly to a uniform region which just fills its positive center lobe. However the response of a filter falls off gradually as the size of a uniform region grows larger or smaller. We have observed that the response will decrease by about a factor of 2 for a factor of 2 increase or decrease in the width of a square. Since the filters are scaled by a factor of $\sqrt{2}$ a local peak occurs within several adjacent band-pass levels. The band-pass signals for an 11×11 square are shown below in figure 7-15. In this figure we have plotted the values along a line which passes through two corners of the square for the band-pass levels 6 through 1. The largest peak occurs for the filter at level 4, which has a positive center region of diameter $2\sqrt{2} + 1$ (See equation (6.5)) or diameter of approximately 9.9 samples.

In fact there are distinct types of M-paths that occur in a DOLP transform. The following three sub-sections examine the three most common classes of M paths. Each of these classes has been given a name. These names, "spots", "bar-ends", and "corners", are not intended to imply that these peaks only occur in patterns which an English speaking human would call a spot, bar, or corner. These are merely labels with which we can refer to these classes. These labels could just as easily be labeled with numbers (as indeed they are in our programs).

In this subsection we are concerned with regions of pixels in which the values are approximately uniform. These regions must have a background which is predominantly darker or lighter than the region for these results to hold

7.3.1.1 "Spots" or Non-Elongated Forms

Let us consider such a region which is not more than twice as long as it is wide. We refer to this class of gray scale forms as "spots". The square in figure 7-15 is an example of a form that includes a **spot M-path**.

A spot will result in M-nodes at a set of adjacent levels of a DOLP transform. These M-nodes will

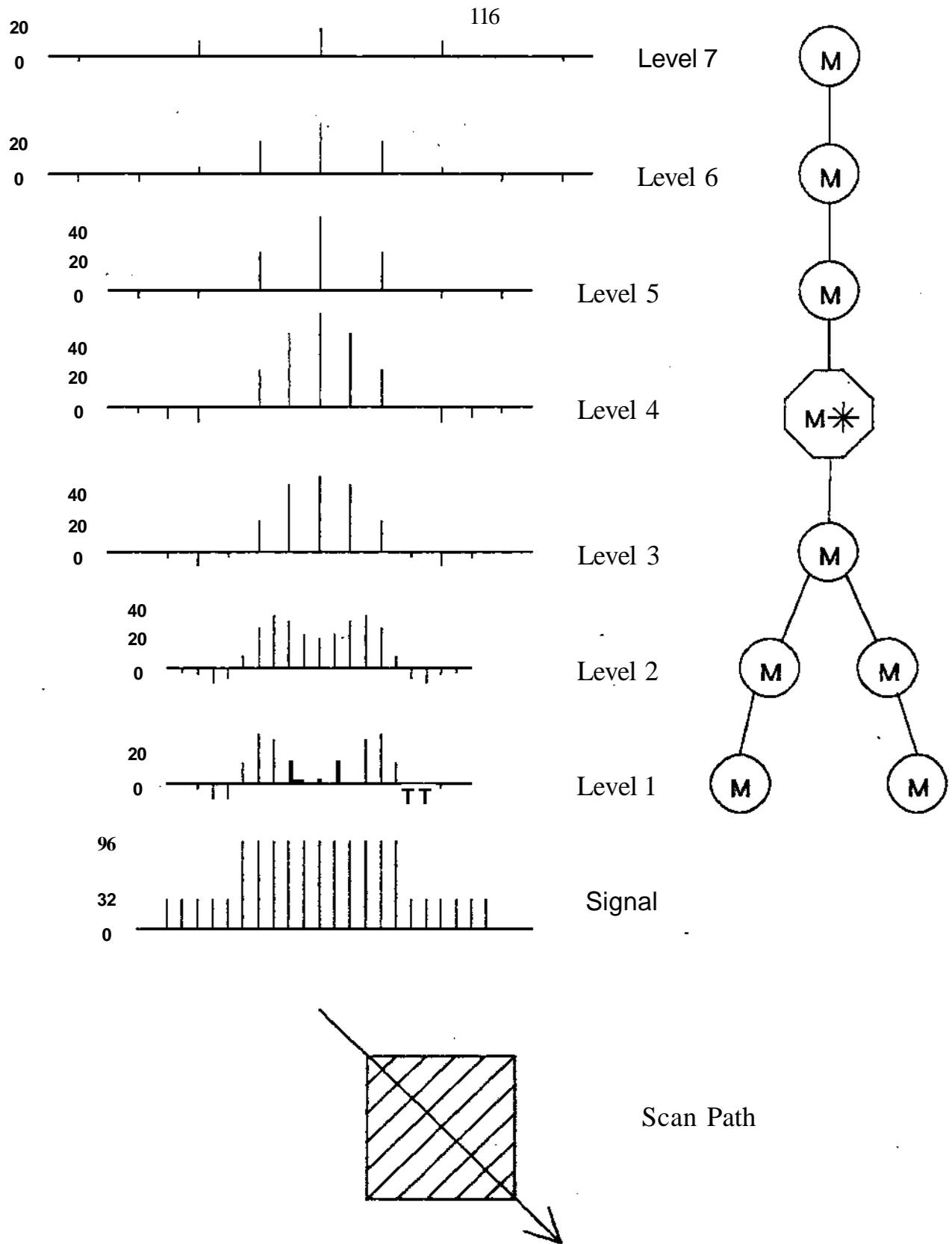


Figure 7-15: Response to an 11 x 11 Square Across Diagonal for Levels 1 Through 7

be located at the sample at each level closest to the center of the form. As a result, these M's will tend to be almost directly under one another. An example of such a sequence of peaks is shown in levels 7 through 3 in figure 7-15.

These M-nodes may be detected individually at each level. They may then be linked together by a quite simple process to form a two-way linked list. We call such a linked list of M nodes an M-path. The magnitude of the values of the M nodes along such an M-path will rise to a maximum and then drop off. The level at which the maximum occurs provides an estimate of the size of the spot. This estimate may be obtained from the formula for the radius of the positive center lobe of the level k band-pass filter. This formula is given as equation (6.5) in chapter 6.

In most cases each peak in the spot M-path will be surrounded by a ridge path of the opposite sign at a distance of 3 to 5 samples. One way to classify a peak as part of a spot M-path is to detect such an opposite signed ridge at all directions within a distance of 6 samples. We have employed a process which scans at multiples of 45° searching for such opposite signed ridges to classify individual peaks with satisfying results. The classification accuracy can be improved by combining the result of such a scan from the peaks within several levels of the largest, or M* peak. This provides a label for the M* peak.

7.3.1.2 "Bar-end": The Ends of an Elongated Form

If a gray scale form is more than twice as long as it is wide, a sequence of peaks will occur at several adjacent levels at the ends of the form. This is illustrated by figure 7-16. This figure shows one end of a uniform intensity rectangle. Circles are drawn over this rectangle to represent the locations where difference of gaussian filters from an SDOG transform best fit the rectangle. Each circle has a radius which is that of the zero crossing of the inner positive center lobe of the corresponding filter. The circles are centered at legal sample points from the level of the SDOG transform of the filter which they represent.

To the right of the partial rectangle is a tree of M-nodes. Each symbol corresponds to one of the circles on the left and represents the location of a peak in the SDOG transform of the partial rectangle. The largest circle corresponds to the top symbol, the second largest circle corresponds to the second symbol, etc. The labels "Bar-End" and "Corner" are those which were assigned on the basis of the out side negative ridge. The labeling process employed a search scan in 8 directions that returned one of three states: no ridge, same-signed ridge, or opposite-signed ridge. The base three number was then used to index into a table of labels. The table was constructed by a training process. This labeling procedure will be described in a report.

The position of these peaks will move from the center toward the ends of the form as the level index, k, decreases. As with a spot M-path, the magnitude of the peaks will rise to a largest value and then fall off. This largest value, which is labeled an M*, corresponds to the filter whose positive center lobe best fits the ends of the form.

At each level, the peaks at the end will be connected by a ridge path of the same sign. The entire configuration will be surrounded by a ridge of the opposite sign. For bar-end M-Paths a scan of its

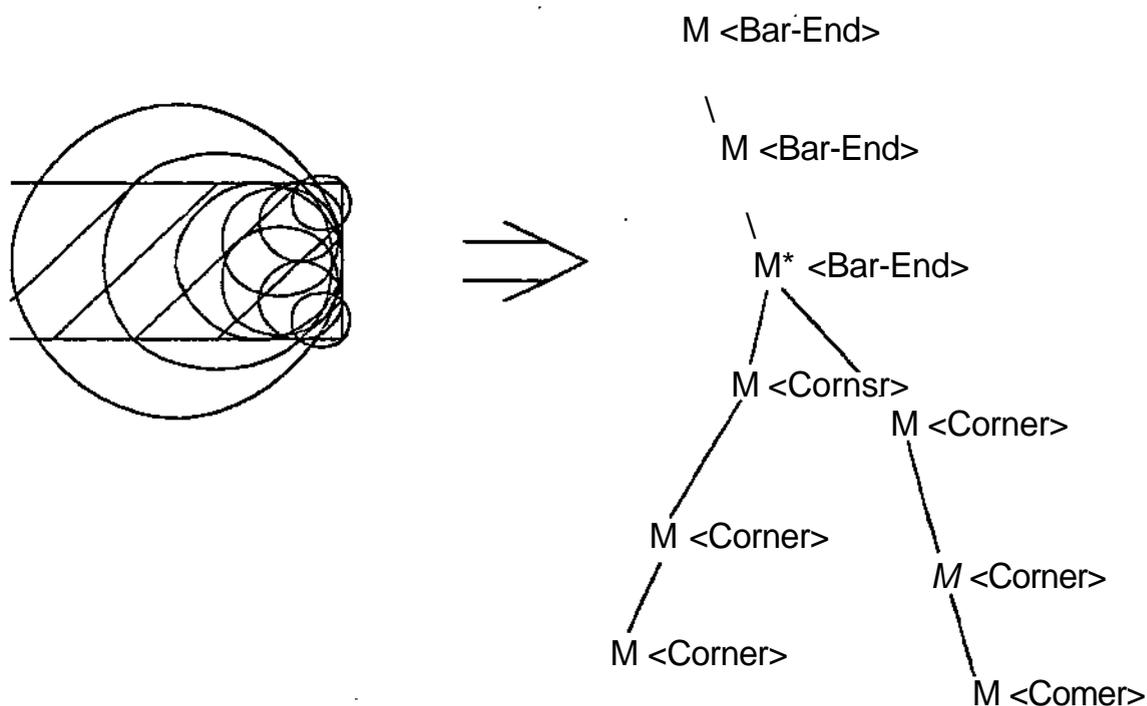


Figure 7-16: Examples of Bar-End M-Paths

neighbors to a distance of 6 samples will show this opposite signed ridge spanning an angle of approximately 270° . This fact, and the presence of the single ridge of the same sign can be used to label the peaks as "bar-ends". As before, a label may be assigned to the M^* peak on the basis of the labels of the other M 's in the M -Path.

7.3.L3 "Corners" and Other Protrusions

A corner or a sharp protrusion will also result in a sequence of peaks at several adjacent levels. However, if the edges of this corner or protrusion are straight, then we have a shape which is the same at several resolutions. In this case the magnitude of the peaks will tend to be constant (In fact, small fluctuations can cause spurious M^* 's to be detected.) If the protrusion is rounded, the value of the peaks will rise to a maximum and then diminish as k decreases. The M -Path may even end before the lowest ($k = 1$) level. In this case there will likely be a largest M node. For a peninsula that is more than twice as long as it is wide, this M -path will be a bar-end. Both of these situations are illustrated in figure 7-17.

In most cases, comers will have two ridges (P -paths) of the same sign connected to them, usually at right angles. Also, within a distance of 6 samples there will be an ridge of opposite sign spanning an arc of about 180° .

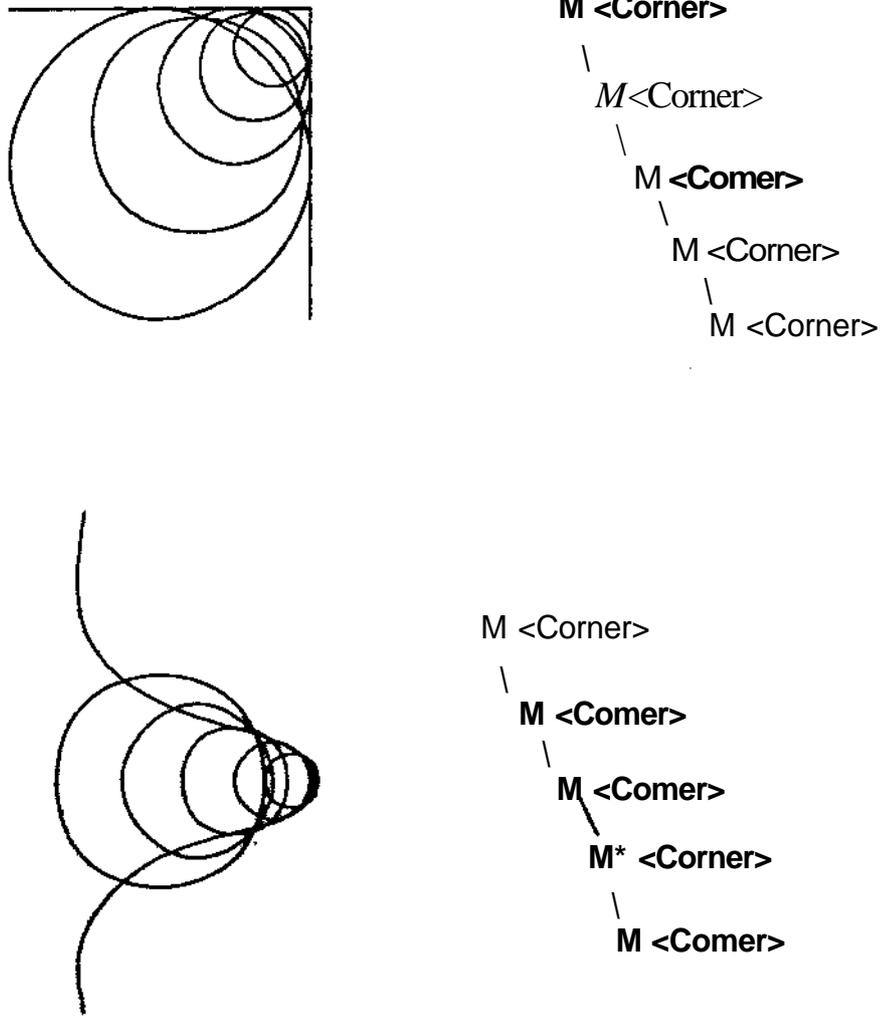


Figure 7-17: Two forms that Cause "Corner" M-Paths

7.3.2 3-D ftdtM: I Paths

Whenever an elongated gray scale form occurs, the DOI-P transfer of the form will contain a ridge at several adjacent levels. The sample points along these ridges correspond in points in (x,y,k) where the primitive center of a bandpass filter is a close fit to the width of the gray scale form. These points are defined by the ridge detection process described above and labeled as F nodes. As multiple M nodes, P nodes occur at approximately the same location in several adjacent levels. At the level where the filter center fits the gray scale form, the magnitude of the filter output falling the ridge will have a larger value than at adjacent levels.

These largest ridge nodes (called L-nodes) can be detected from the ridge nodes (P-nodes) at each level by a process which is similar to the "flag stealing" process used for detecting the largest M-node on an M path. Unfortunately this detection process is somewhat more complex because of the directional nature of ridges and the difference of sample rates at different levels. Once the L-nodes have been detected they can be linked into a two-way linked list call an L-path.

In the following paragraphs we will examine the patterns of ridges that occur for uniform width bars, bars of changing width , and edges of regions.

7.3.2.1 Ridge Paths for a Uniform Bar

Consider the uniform rectangle which was used as an example in figure 7-5 above. The response at levels 6 through 1 of the Sampled DOG transform along a line through the center of the rectangle is shown in figure 7-18 below. At level 2, an M* occurs at both ends of this rectangle. Between these M*-nodes there is a ridge node that is larger than the ridge nodes above and below it. This ridge node is detected as an L node by the process described in the next section. This rectangle produces a graph as shown in figure 7-18. We can abstract all of the M* nodes and L-paths in this graph to obtain a description of a class of forms that resemble this bar. This class of forms is defined by the presence of the symbols:

$$M^* - L - M^*$$

If we held the width of the rectangle constant and increased its length the number of L-nodes between the M* nodes would increase. We can define the class of bars as those forms which have a pair of M* nodes connected by some number of L-nodes between them, and then encode the cartesian distance between the M* nodes (measured in samples at some reference level) as an attribute of the form.

7.3.2.2 Bars of Changing Width

Suppose, instead of a rectangle, we have a four-sided form which changes in width by a factor of 2 along its length. Such a form is shown in figure 7-19. As the width of the form decreases, the level of the filter which best fits the form decreases. As a result the M* nodes occur at different levels, and the L-Path changes levels. We can define a class of bars that includes bars that change width, by collapsing the L-path into a single symbol. The L-path should retain the attributes of its length (Measured in number of samples at some reference level) and the change in levels between the M* nodes that it connects (Δk).

7.3.2.3 Edges of Regions

A straight line edge of a uniform region will result in a set of ridge paths at several levels in which the values are approximately the same. If the edge is blurry, then the value along these ridge path will decrease with decreasing k. If, on the other hand, the figure is washed out, the values along the ridge path will be largest at some level, and will be detected as L-nodes.

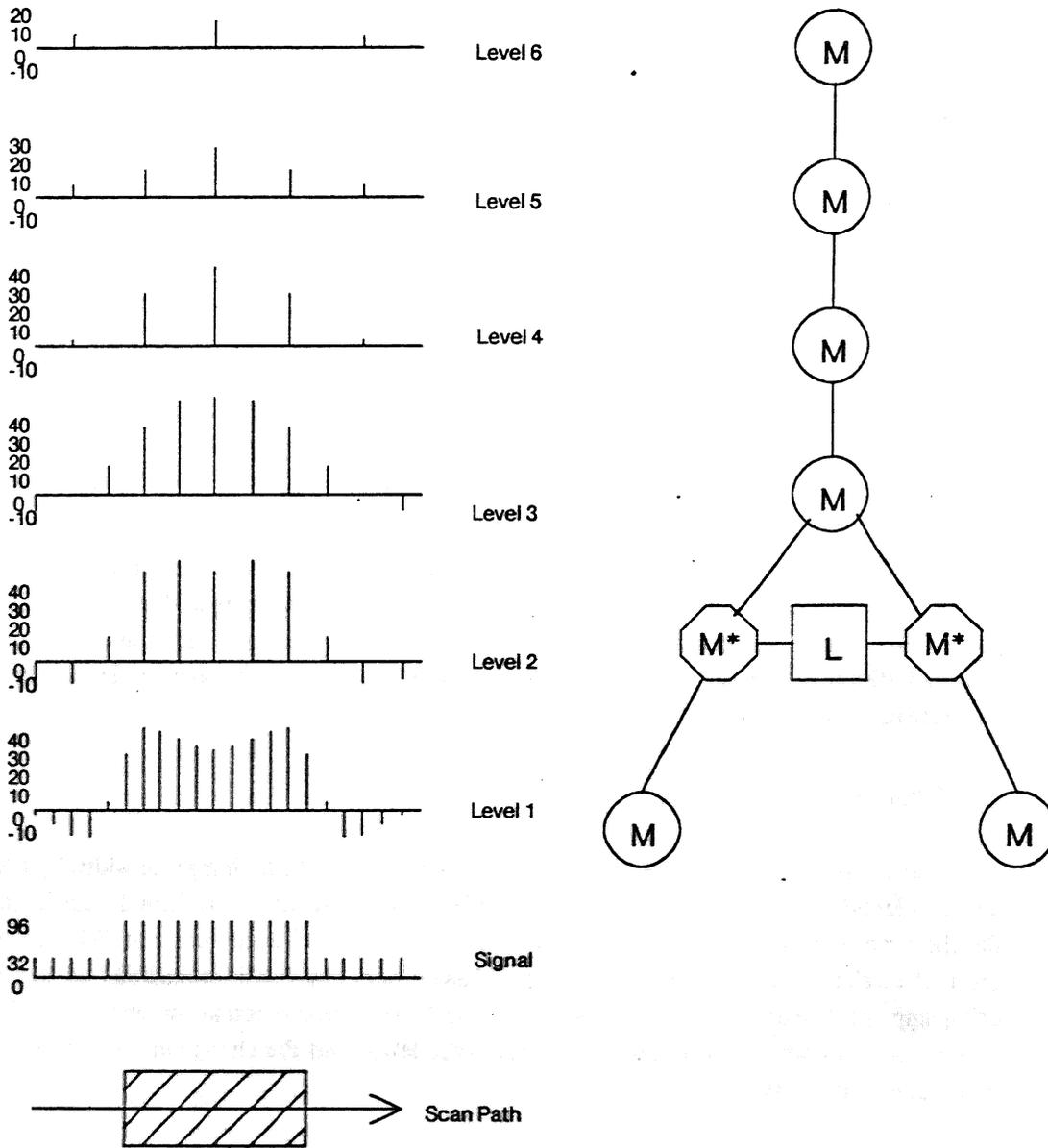


Figure 7-18: Response to a 5 by 11 Rectangle

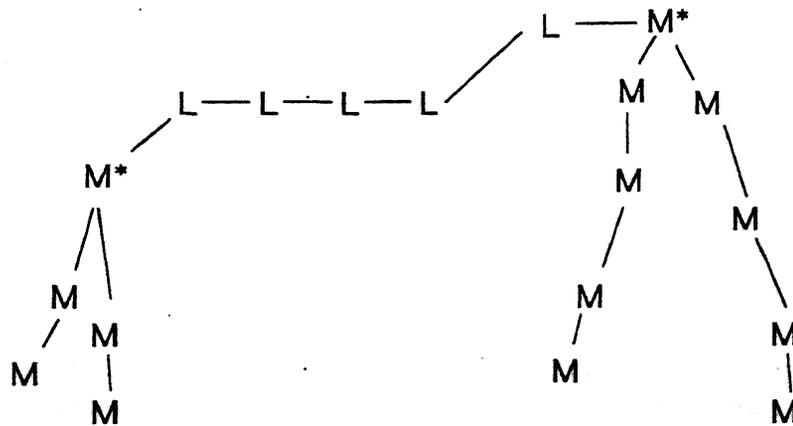
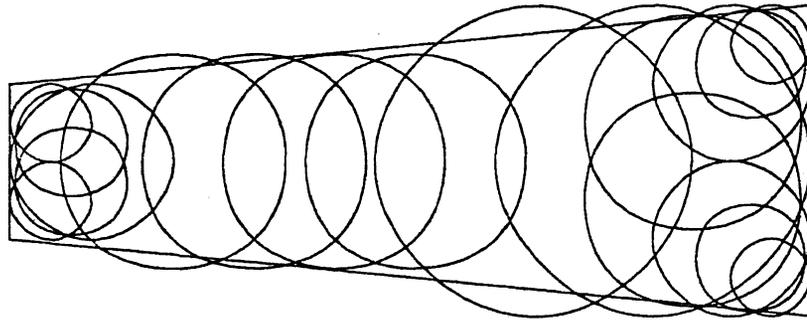


Figure 7-19: An Elongated Form That Changes Width

The fact that an L node is part of an edge can be detected by the same scan procedure described above for labeling M-nodes. An L node or P-node which is part of an edge will have a single ridge of opposite sign running parallel to it within a distance of 6 samples. It may or may not have a same signed ridge parallel to it in the opposite direction within 6 samples, depending on how wide the form is. An L-path which is part of a "bar" or other elongated form will have opposite signed ridges running parallel to it on two sides. Figures 7-2 through 7-6 show examples of the ridge points and opposite signed ridge points that occur for an edge. These figures show the response along a line at one level. Figure 7-4 shows an example of a ridge point which is an L node and detected as a bar with ridge points of the opposite sign on both sides. Both of these cases are illustrated with a piston rod image shown in figures 7-26(a) through 7-26(h) and 7-27(a) through 7-27(h) at the end of this chapter. Figure 7-27(h) is a good 2-D example of the ridges that occur on both side of an edge.

7.3.3 Connectivity of L-Paths and M-Paths

One of the properties that permits us to construct a representation of an image using only local operations is the property that L-paths will almost always terminate at an M-path.

An L-path follows the length of an elongated form. As the form widens, the L-path moves upwards in the k dimension. As the form narrows, the L-path moves downward in the k dimension. At the ends of an elongated form the response of a DOLP (or SDOG) transform increases due to the presence of more background area in the negative side-lobe of the band-pass filter. This increase results in an M-node. Unless the form fades into the background very gradually there will be an M-node at its end, and thus the L-path will terminate at an M-path. Because the same band-pass filter will best respond to the width of a form both along the form and at its ends, an L-path will usually terminate within one level of an M^* node.

7.4 Connecting Peaks Between Levels

- This section describes a process which links peaks (M nodes) which are at adjacent levels in the DOG transform to form M-paths. This process also detects the largest M nodes in a path and labels these as M^* nodes. An M^* node is an M node which is part of an M-path and which has a larger value than the adjacent M nodes in the M-path.

7.4.1 Linking NTs

The principle behind the process for linking M nodes is simple. Starting at the highest level, K , at each level k each M node looks at the nodes within a local neighborhood above it, at level $k+1$. A 2-way pointer is made to all M nodes that are found within this neighborhood.

This process proceeds as follows: For each level k , from K through 1, each M node at level k examines the nodes which are adjacent to it at level $k+1$. There may be either 4 or 9 such adjacent nodes due to the VT sampling. The nodes which are adjacent to these nodes at level $k+1$ are also examined. Thus either 25 or 16 total nodes are examined. If any of the adjacent 4 or 9 nodes at level $k+1$ are M nodes and have a value of the same sign, then a 2-way pointer is formed. This pointer is formed by setting the appropriate down pointer of the node at level $k+1$ and setting the up pointer corresponding to that upper neighbor in the node at level k . See table 7-1 and section 7.1 for an explanation of the up and down pointer bytes.

If any of the neighbors of the neighbors at level, $k+1$ are an M node an indirect 2-way pointer is made. An indirect pointer goes through the adjacent neighbor's pointer. The set of possible indirect paths are illustrated in figure 7-20. The fact that a pointer is indirect may be determined by examining the L and M flags of a node. If both these are zero then any pointers for L and M paths are indirect pointers.



Figure 7-20: Possible Set of Indirect 2-Way Pointers for M-Paths

7.4.2 Detecting M*'s

M* nodes are detected by a process which we refer to as "flag stealing". When an M node detects another M node at level $k+1$, it compares values. If the M node at level k has a value of smaller magnitude it clears its own * bit. If the M node at level k has a value of larger magnitude it clears the * flag of the node at level $k+1$ and sets its own * flag. If more than one M node is detected at level $k+1$ they must all be smaller for the node at level k to set its * flag. If no M nodes are found at level $k+1$ then the * flag is cleared; This prevents any isolated M nodes from becoming M* nodes. If more than one node at level k link to an M node at $k+1$ any of them will clear the * flag of the node at level $k+1$ if they have a larger value. Thus * flags propagate down an M-path until they reach a node with the latest magnitude.

7.4.3 Example

Figure 7-21 shows the M-paths and the M* node that occur at level 7 through 1 for a uniform intensity square of width 11 pixels, and grey level 96 on a background of 32.

7.5 Detecting Ridge Nodes in $\{x,y,k\}$ Space

This section describes the processes for detecting ridge nodes (L-nodes) in the 3-D SDOG transform space. The section starts with a discussion of the approach which is used and a description of some of the problems that complicate such detection. A description of the search procedure for P-nodes within two neighborhood sizes above each P-node is then given. A discussion of the "flag stealing" process that is used and modifications to this process is then presented.

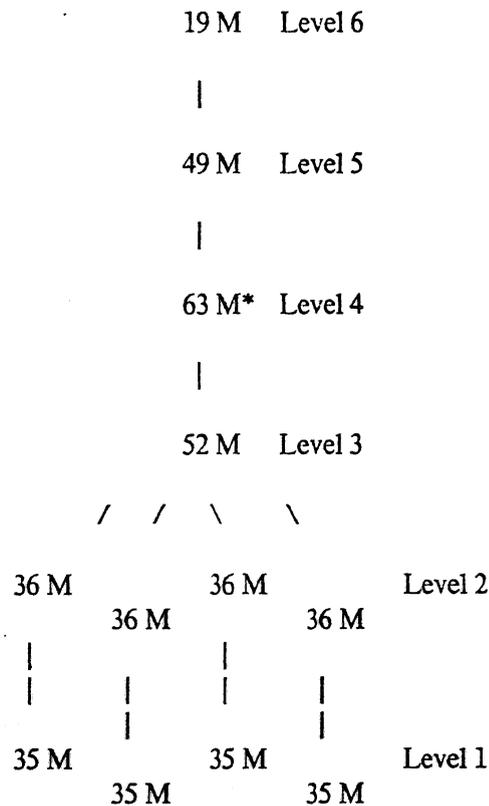


Figure 7-21: M Paths For Square of Size 11 Pixels

7.5.1 Problems and Approach

Ridge nodes in the (x, y, k) space produced by the SDOG transform are detected with a form of flag stealing process. As with detection of M^* -nodes from M -nodes, the P -nodes which have been detected as ridge points at each level are used as candidates for L -nodes.

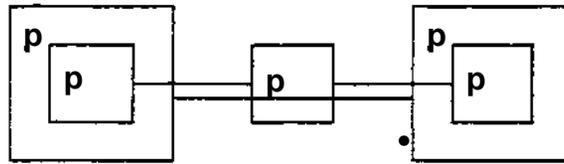
These P -nodes examine the P -nodes within a neighborhood at the level above them. This examination occurs during a two stage search procedure. Initially a small neighborhood at level $k + 1$ is examined above each P -node at level k . If no P -nodes are found in this small neighborhood, then the nodes within a larger neighborhood are searched for P -nodes. This second search is inhibited for directions within 45° of any P -path pointers in the P -nodes at level k to prevent a P -node at level k from stealing the L -flag from a P -node at level $k + 1$ over a different part of the ridge.

The situation is more complicated than with detection of M^* -nodes, because:

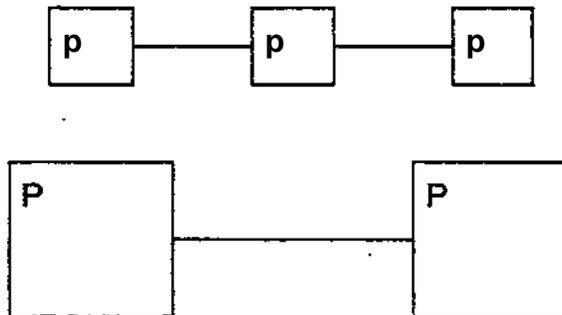
- Ridge paths (L -paths) are directional and may travel through as well as along the levels.
- Ridge paths that describe an edge tend to move sideways toward the edge as the level decreases. This creates situations where each P -node at level $k + 1$ is examined by several P -nodes at level k .

- Two connected P-nodes at level k may, because of VT resampling, have a P-node at level $k-1$ between them, as illustrated by upper part of figure 7-22. In this figure, the larger squares represent the P-nodes at level $k+1$, and the smaller squares represent the P-nodes at level k . Which of the nodes at level $k+1$ should the node in the center at level k compare its value to?

The problem illustrated by figure 7-22 is even more severe when the P paths at adjacent levels are displaced side-ways as shown in the lower part of figure 7-22. This situation is handled by a modification to the flag stealing process described in section 7.5.3. This modification is based on the principle that an L-flag is stolen only if all its lower P-node neighbors have a larger value.



Overlapping Ridges at Adjacent Levels



Displaced Ridges at Adjacent Levels

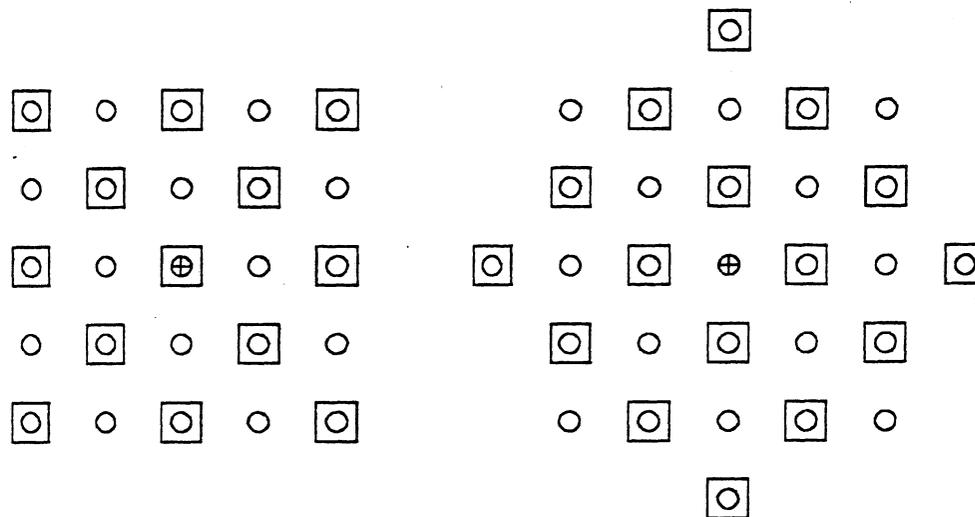
Figure 7-22: Two Configurations of Ridge Paths at Adjacent Levels

7.5.2 Search Paths

At each P-node at a level k , the upper neighborhood at level $k+1$ is searched for P-nodes. The P-node at level k from which the search originates is referred to as the "source" node.

A source node at (x, y, k) can have two possible neighborhoods at level $k+1$ depending on whether a sample exists at $(x_0, y, k+1)$. These two neighborhoods are illustrated in figure 7-23. In

this figure, circles represent sample points at level k while boxes represent sample points at level $k+1$. The source node has a cross through it. If k is even (i.e. on a $\sqrt{2}$ sample grid), these two neighborhoods are rotated by 45° .



Upper Neighbor Exists

No Upper Neighbor Exists

Figure 7-23: Two Possible Upper Neighborhoods

There are two search procedures that are used to detect P-nodes at an upper level, depending on whether the source node at (x, y, k) has a sample directly above it, i. e. at $(x, y, k+1)$. The test which tells whether a sample exists at $(x, y, k+1)$ is used to determine which search procedure is used. That is, if:

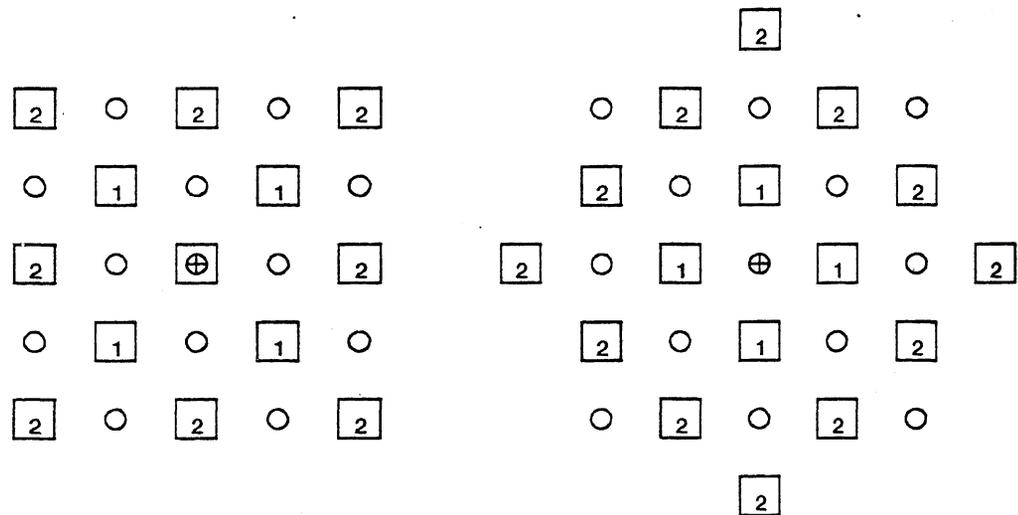
$$x \bmod 2^k = y \bmod 2^k = 1$$

is true then the source node at (x, y, k) has a sample directly above it.

If a sample exists above the source node, then it is tested to see if it is a P-node. If it is a P-node, then only this node is examined.

If no sample exists above the source node, or the sample above the source node is not a P-node, then a two stage search procedure is employed. The first stage examines the nearest 4 upper neighbors. If no P-node is found in this first stage, a second stage searches for P-nodes in an enlarged neighborhood. The neighborhoods examined by these search algorithms are illustrated in figure 7-24. In this figure the sample points at level k which have no neighbor are illustrated with a circle. Points where samples exist at both levels are indicated by a 1, or a 2. Those points with a 1 are examined in the first stage, those with a 2 are examined in the second stage if no P-nodes are found in the first stage.

The second stage search does not occur for any direction within 45° of a P-path pointer in the source node. This helps prevent nodes from interfering with the flag stealing process at other points on the P-path.



Upper Neighbor Exists

No Upper Neighbor Exists

Figure 7-24: Upper Search Neighborhoods for Stage 1 and Stage 2

7.5.3 The Modified Flag Stealing Process

The principles of "flag stealing" were described during the discussion of detection of M^* -nodes given in section 7.4.2. This process must be modified to use with detecting L-nodes, because each L-node at level $k+1$ is likely to be examined by several P-nodes at level k , some of which may be displaced along the P-path ridge. Since the value can change along a 3-D ridge, nodes further along the ridge might improperly clear the L-flag of nodes above them, breaking the L-path. The modification is based on the principal that all of the lower neighbors must have a larger value, before the upper P-nodes L flag will be reset.

Modified flag stealing employs two temporary bits at each node which denote whether any lower neighbors have a smaller value (flag T1) or a larger (or equal) value (flag T2). After flag stealing is executed at level k , the L-nodes at level $k+1$ are examined, and any with node which has its T2 flag set and its T1 flag clear has its L flag cleared.

A search neighborhood which is of restricted duration along a ridge is also used. A larger neighborhood is needed for directions perpendicular to the ridge because of the lateral drift that can occur with P-paths as the level decreases.

7.5.3.1 Modified Flag Stealing

If a source P-node at (x, y, k) has an upper neighbor at $(x, y, k+1)$ which is also a P-node, then only this neighbor is examined by this source node.

If the source P-node at (x, y, k) has no upper neighbor, or the upper neighbor is not a P-node, then

this process is applied to the nearest upper 4 neighbors. If no P-nodes are found in the nearest upper neighbors, the search is applied to an enlarged upper neighborhood. As mentioned above, the second stage search is inhibited for all samples within 45° of a P-path pointer in the source node.

When a P-node is found at level $k+1$, its value is compared to that of the source node. If the value of the upper neighbor is larger and the upper neighbor has its L flag set, then the T2 flag of the upper neighbor is set to indicate that the upper neighbor has a lower neighbor with a smaller value. If the value of the source node is larger, then the L flag of the source node is set. Also, if the L flag of the upper neighbor is set, then the T1 flag of the upper neighbor is set to indicate that the upper neighbor has a lower neighbor which attempted to steal its flag.

7.5.3.2 Resolving the T1 and T2 Flags

After the L node detection process has been run at level k , the L-nodes at level $k+1$ are processed to resolve the T1 and T2 flags. At each L-node at level $k+1$, if its T1 flag is set and its T2 flag is not set, then all of its neighbors at level k are larger. In this case, its L flag is cleared.

This modified flag stealing process will permit two or more P-nodes at the same location in adjacent levels to be L-nodes. This can occur when an elongated form has a sudden decrease in width. For such a form, the L-path can travel straight down through the levels. An example of this occurs with in the Piston Rod images and can be seen at column 41, rows 97 to 109 in levels 7 and 6 of the Piston Rod description shown in figures 7-27(d) and 7-27(e). The L-nodes at the upper level are inhibited from losing their L-flags, because other P-nodes at in the lower level P-path have smaller values, and thus set their T1 flag.

7.5.3.3 Linking L-nodes

After the T1 and T2 flags have been resolved, a process is executed to form two way pointers between all adjacent L-nodes. This process runs as follows. Each L-node at level $k+1$ examines all of its neighbors at level $k+2$ within its 2nd stage neighborhood and all neighbors at level $k+1$ for which it has a P-path pointer but no L-path pointer. If any of these neighbors are an L-node, an M-node, or an M*-node a two way pointer is made by setting the appropriate pointers in the UP, SAME and DOWN pointer bytes of the neighbor and the source L-node.

7.6 Examples

This section shows some examples of M*'s, M Paths, L Paths and P Paths. These examples are from levels 10 through 3 of the right most piston rod in the image shown in figure 7-25 below. This image is from the GM "Bin of Parts" data base [Baird 77].

Figures 7-26(a) through 7-26(f) show the upper third of the left most piston rod. These figures are shown with nodes spaced at 4 pixels, which is the sample rate at level 5. Figures 7-26(g) and 7-26(h) show a smaller window which is from the upper left corner of the window shown in parts a through f. In parts g and h the sample rate is $2\sqrt{2}$ and 2 respectively.

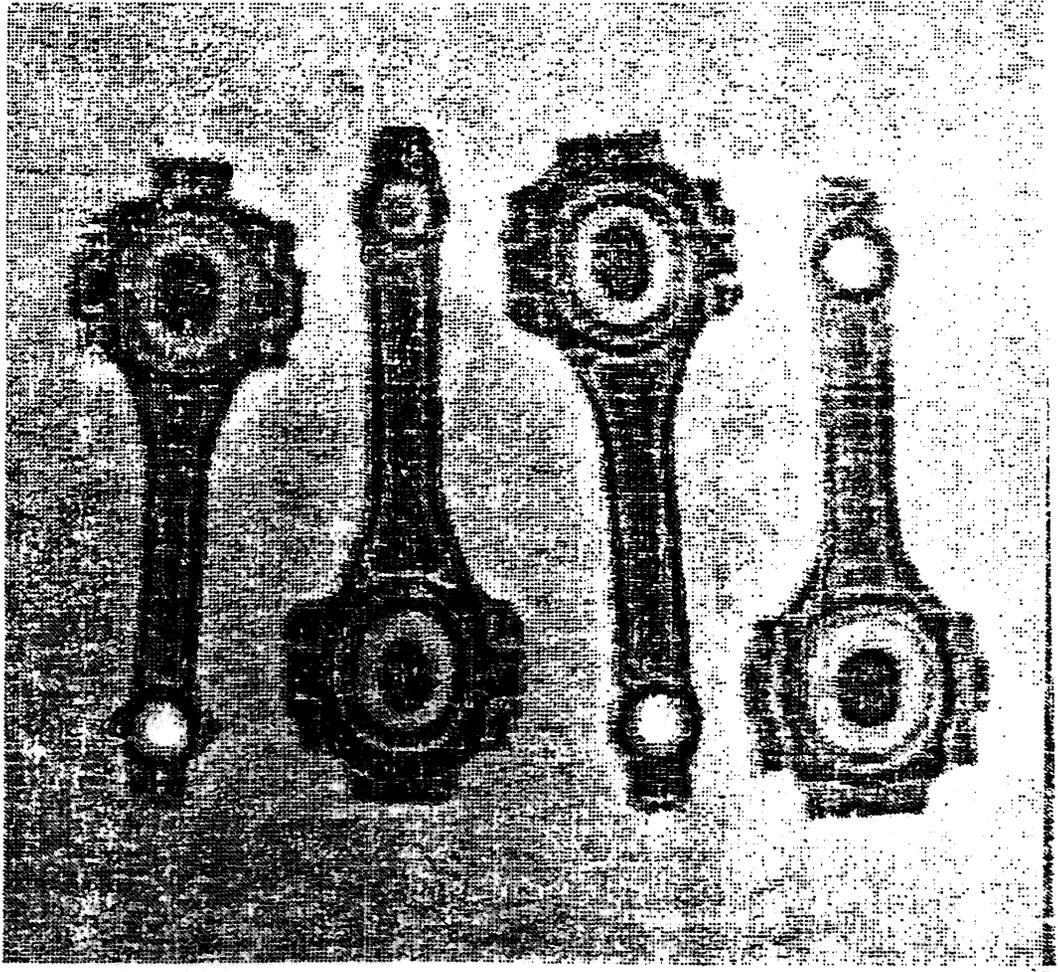


Figure 7-25: Piston Rod Image. Sampled at 256 by 256.

Figure 7-26(a) is from level 10 of the DOG transform. At this level the data has been sampled at $16\sqrt{2}$ and so this figure is very sparse. Note the M node at row 81, col 49. This is the start of an M path that leads into the piston rod.

Figure 7-26(b) shows the same window at level 9. As is often the case there are short spurs hanging off of the M node at row 81, col 33.

Figure 7-26(c) shows the same window at level 8. At row 73, col 41 is the M* node which serves as a landmark for the upper part of any piston rod. The two L nodes at row 65 are spurs; they do not connect to anything else. The L node at row 89 is part of an L path that travels down through the levels and down through the rows to become the long part of the piston rod.

Figure 7-26(d) shows a phenomenon which is very rare; This is the only instance that we have observed. On rows 73 and 81, The values in columns 41, 49, and 57 are the same. The result is a pair of parallel adjacent ridges of the same sign. This is not a serious problem as these points are not strong enough to be L nodes. Note also that the M path has split into two parts. Both parts have two way pointers to the M* node at level 8.

In figure 7-26(e) the shape of the upper part of the piston rod begins to become apparent. Note that an M node has appeared in the middle, at row 77, col 45. This M node is attached by P paths to nearby M nodes in 4 directions. These paths resulted when the spurs attached to this central M node were extended. This central M node evolves at lower levels into the oval shaped region which occurs in the center of the top of the piston rod.

Figure 7-26(f) shows level 5 of the description. Note the M* node on row 49, column 45. This marks the large region at the top of the piston rod. Notice also that two L paths extend from this M* node. These L paths drop down to lower levels as that part of the piston rod narrows. Also note that at this level the negative ridge surrounding the inner oval has appeared. The oval is not connected to the rest of the piston rod in this or any of the lower levels.

Figure 7-26(g) shows the upper right corner of the window from the previous subfigures, as scene in level 4. At this level the data is sampled at $2\sqrt{2}$. Note that the L path begun in level 5 continues into this level. Note also that at this level the negative ridge which surrounds the oval also forms a part of an L path.

Figure 7-26(h) shows the transform at level 3. The L path that describes the ring of the upper part of the piston rod dips into this level in its narrow parts. The P path for this form is broken at this level. This is an artifact of the ridge detection process. The negative ridge outside of the piston rod has an M* at this level. This indicates that a rounded corner occurs in the background (A negative corner!) The M* occurs because this corner is not sharp. The negative ridge between the outer positive ring, and the inner oval also contains two M*'s at this level. These correspond to negative corners in the inside of the ring. The L path attached to these negative M*'s extends up to level 4.

Values for nodes	Level 10											
	17	21	25	29	33	37	41	45	49	53	57	61
45												
45												
45												
45												
49												
49	10								13			
49	PB											
49												
53												
53												
53												
53												
57												
57												
57												
57												
61												
61												
61												
61												
65												
65												
65												
65												
69												
69												
69												
69												
73												
73												
73												
73												
77												
77												
77												
77												
81												
81	14											
81												
81												
85												
85												
85												
85												
89												
89												
89												
89												
93												
93												
93												
93												

Figure 7-26a: Top OfPiston Rod at Level 10

	Values for nodes - Level 9				rod.swf		L Paths and M Paths					
	17	21	25	29	33	37	41	45	49	53	57	61
45												
45												
45												
49												
49	5				18				16			
49					PB							
49					!							
53												
53												
53												
57												
57												
57												
57												
61												
61												
61												
61												
65					!							
65	14				35				30			
65					P							
65					!							
69												
69												
69												
69												
73												
73												
73												
73												
77												
77												
77												
77												
81					!							
81	18				36				29			
81	PB-				- MP							
81					!							
85												
85												
85												
89												
89												
89												
89												
93												
93												
93												
93												

Figure 7-26b: Top of Piston Rod at Level 9

Values for nodes - Level	8 rod.swf L Paths and M Paths											
	17	21	25	29	33	37	41	45	49	53	57	61
45												
45												
45												
49												
49	-2				20				20			
49												
49												
53												
53												
53												
57												
57			18				34				14	
57												
57												
61												
61												
61												
61												
65												
65	10				37				35			
65					L PB				L PB			
65					\				/			
69												
69												
69												
69												
73							\	/				
73			29				44				24	
73							*LMP					
73							!					
77												
77												
77												
77												
81												
81	13				37				34			
81												
81												
85												
85												
85												
85												
89							!					
89			21				36				11	
89							L P					
89							!					
93												
93												
93												

Figure 7-26c: Top of Piston Rod at Level 8

Values for nodes - Level	7						L Paths and M Paths					
	17	21	25	29	33	37	41	45	49	53	57	61
45												
45												
45												
45												
49	!						!					
49	-10		2		17		28		25		7	
49	P						P					
49	/						!					
53												
53												
53												
57							!					
57	-2		14		29		34		29		14	
57							MP -		- PB			
57							!					
61												
61												
61												
61												
65							!					
65	6		21		30		30		26		20	
65							P					
65							/					
69												
69												
69												
69												
73							/					
73	12		26		34		30		26		21	
73	PB-		- P -		-S MP -		- P -		- P -		- P -	
73												
73												
77												
77												
77												
77												
81							/					
81	8		26		30		30		26		21	
81							P -		- P -		- PB	
81							!					
85												
85												
85												
85							!					
89							28		24		15	
89	0		15		23		L P					
89							!					
89												
93												
93												
93												
93												

Figure 7-26d: Top of Piston Rod at Level 7

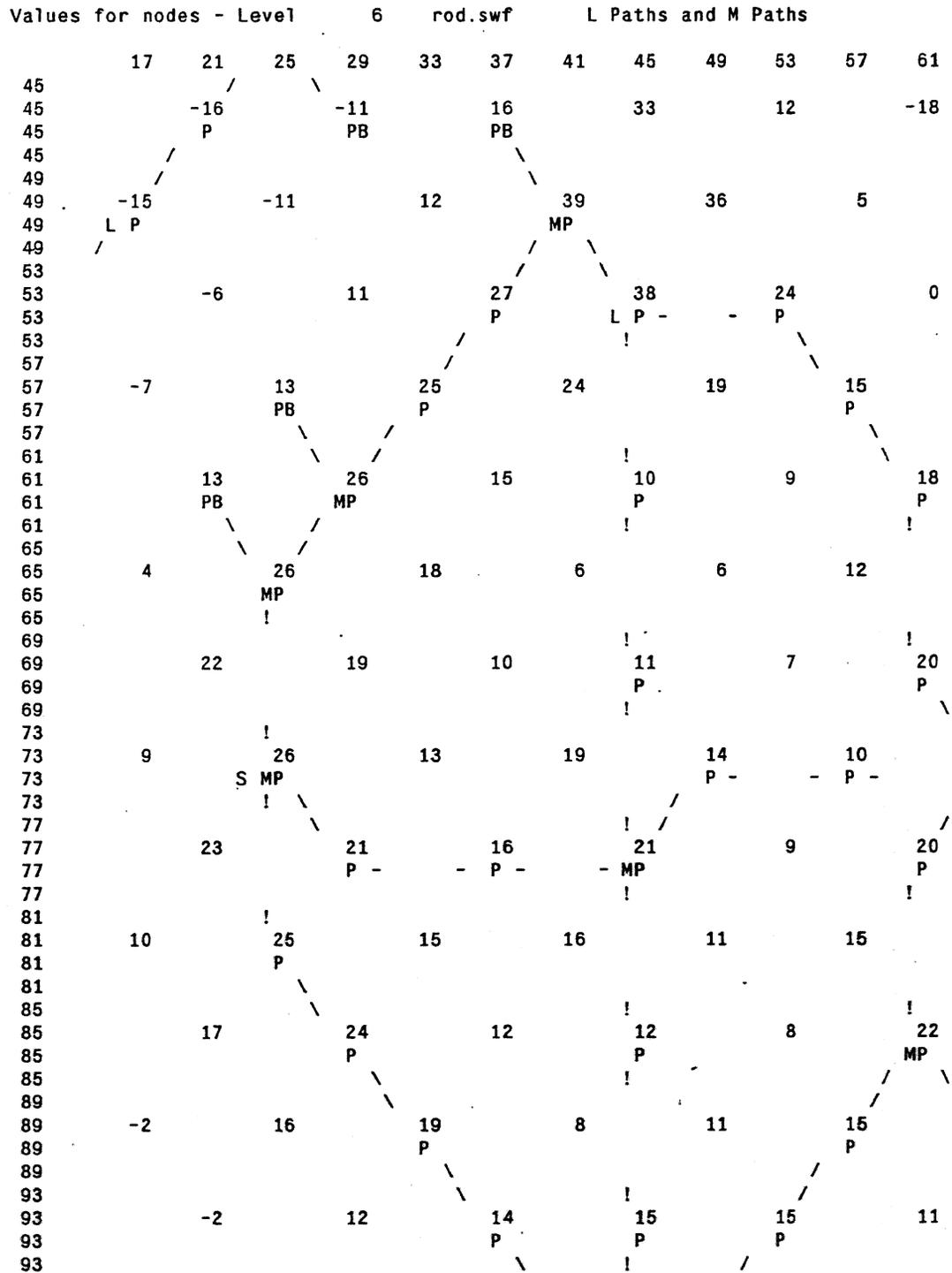


Figure 7-26e: Top of Piston Rod at Level 6

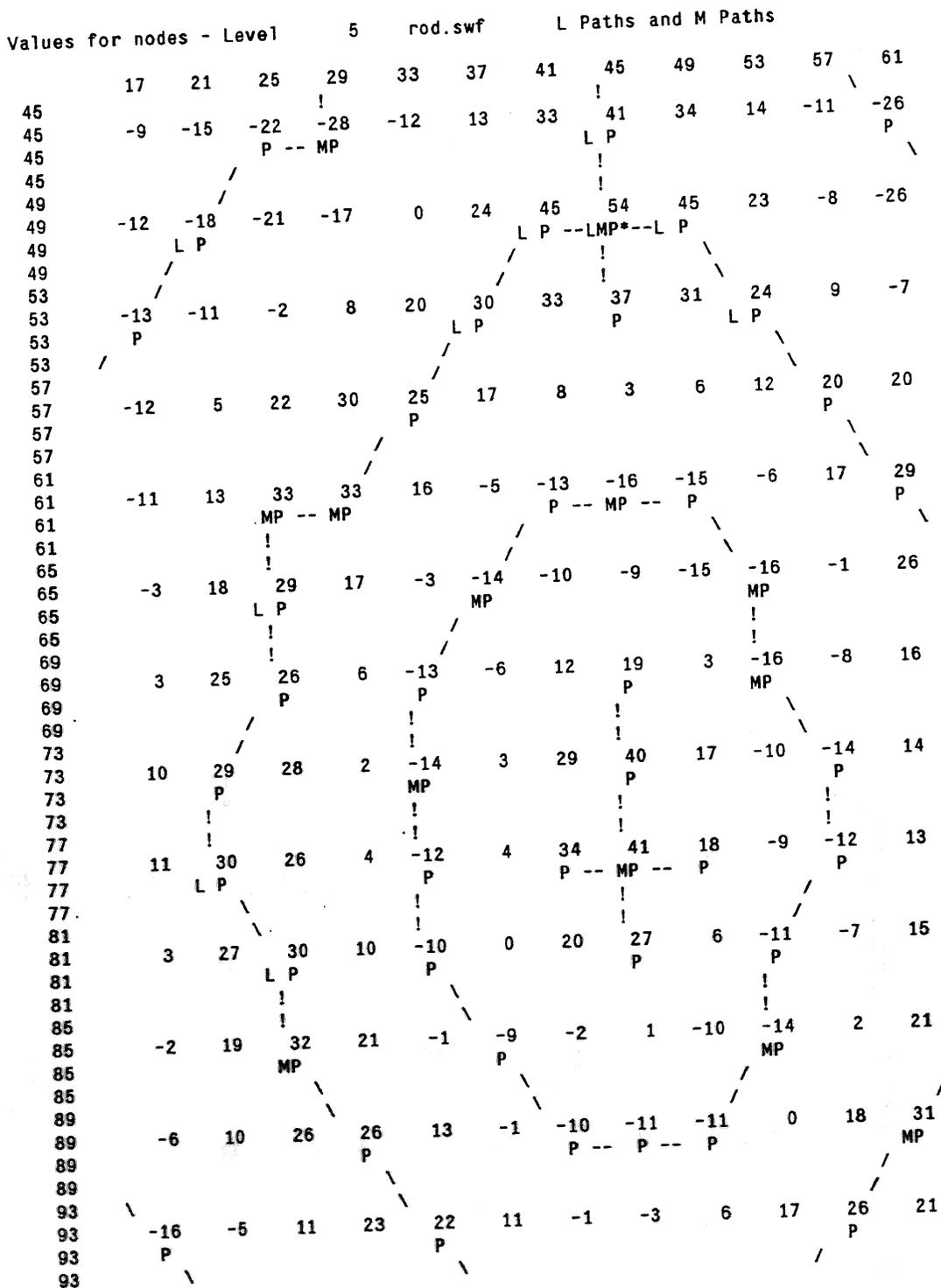


Figure 7-26f: Top of Piston Rod at Level 5

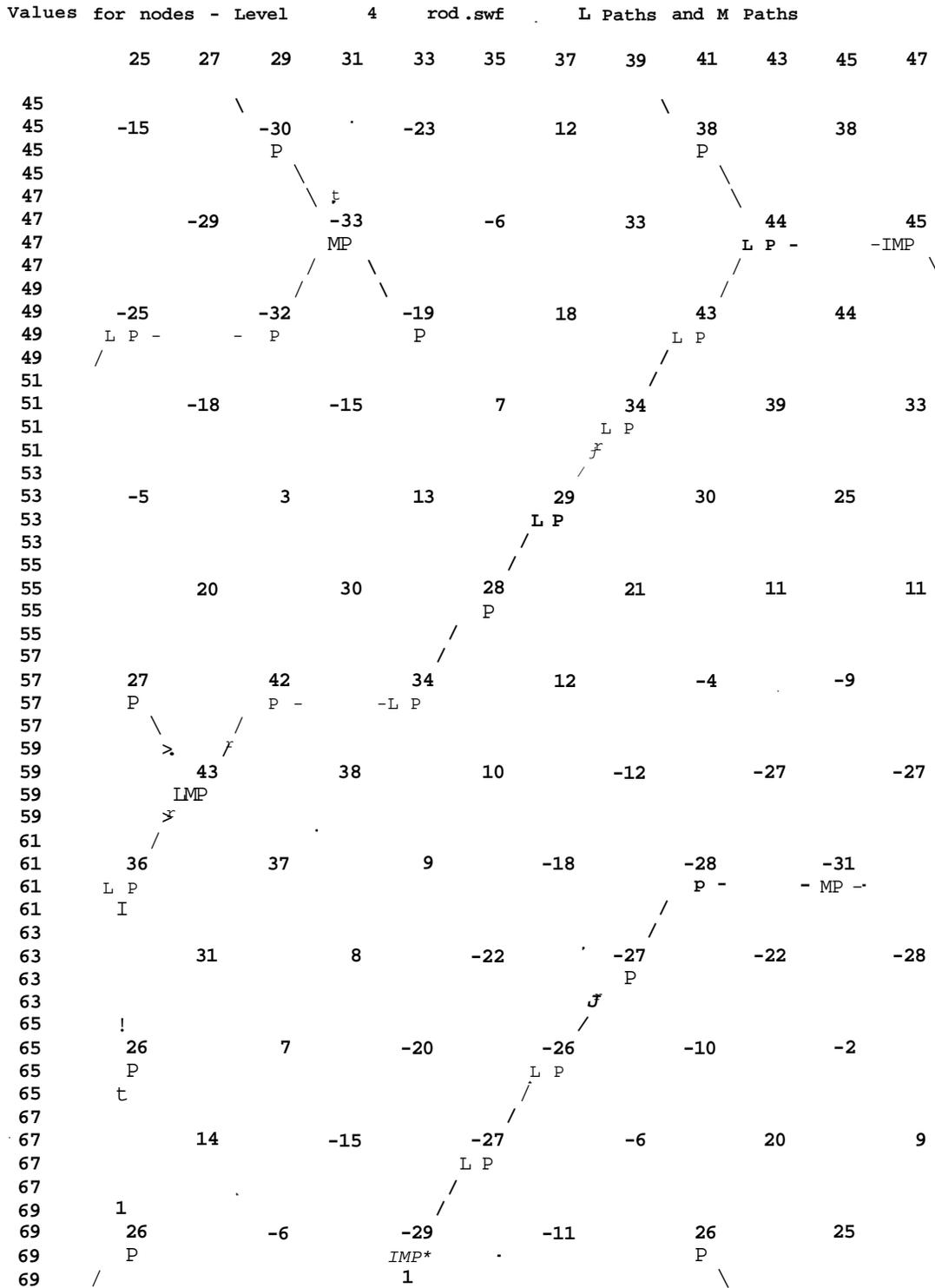


Figure 7-26g: Top I.left Corner of Piston Rod at Level 4
 (Note that Sample Rate is $2\sqrt{2}$)

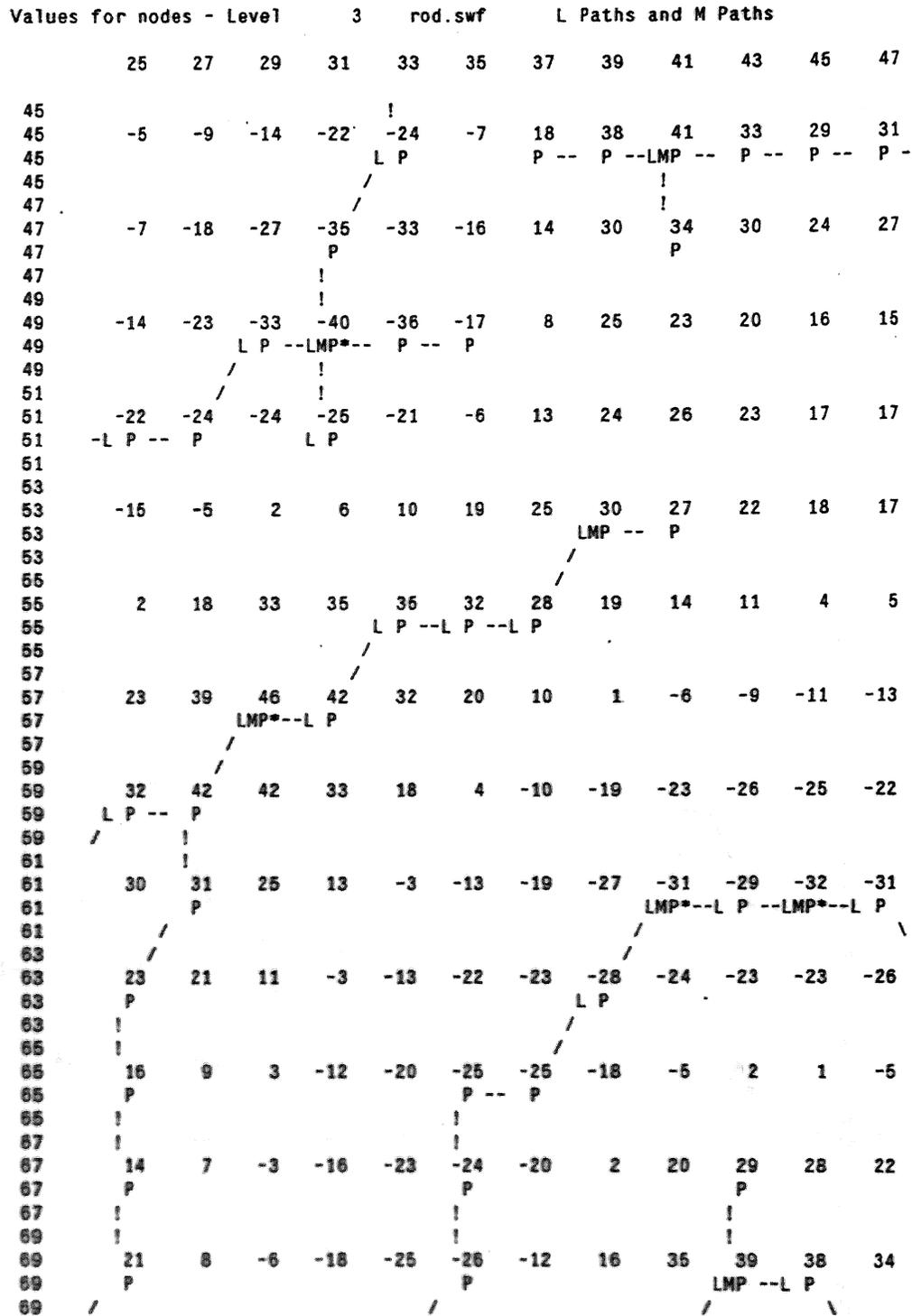


Figure 7-26h: Top Left Corner of Piston Rod at Level 3
(Note that Sample Rate is 2)

Figures 7-27(a) through 7-27(h) show the description for the middle of the same piston rod. The window within which these points are shown is immediately below that for figures 7-26(i) through 7-26(h).

Figure 7-27(a) shows this window at level 10. Because of the sparse sampling, there are only 2 P nodes, which are an extension of the ridge path for the middle of the piston rod. The same is true for levels 9 and 8, although one can see the values increasing as the level decreases.

At level 7, figure 7-27(d) shows this P path with two L nodes at rows 97 and 105. These L nodes are part of the L path that started with the M* node at row 73, col 41 of level 8 shown in figure 7-27(c). This L path continues into level 6, as shown in figure 7-27(e) as the upper part of the piston rod narrows. Note, also, how the negative ridges move closer to the positive ridge as the filter radius becomes smaller. This is a classic example of the configuration of ridges that occurs for a uniform width longish object.

The L path finally settles into level 5, as shown in figure 7-27(f). This L path connects to the M* node at row 133 col 41, and then continues down the piston rod.

Figures 7-27(g) and 7-27(h) show blown up versions from the middle of the window shown in the previous figures. In these two figures, the nodes are printed with a spacing of two columns; the sample rates are $2\sqrt{2}$ and 2, respectively. Figure 7-27(g) shows this smaller window at level 4. The positive ridge at this level has a lower value than at level 5. Figure 7-27(h) shows this smaller window at level 3. At this level the positive ridge has split into two ridges, representing the edges of the piston rod. The spurs attached to the M nodes at this level extended to reach each other, giving an

occasional path between the two positive ridges.

	Values for nodes - Level 10				rod.swf		L Paths and M Paths					
	17	21	25	29	33	37	41	45	49	53	57	61
97												
97					20							
97												
97												
101												
101												
101												
101												
105												
105												
105												
109												
109												
109												
109												
113									!			
113	6								13			
113									P			
113									!			
117												
117												
117												
117												
121												
121												
121												
121												
125												
125												
125												
125												
129												
129					9							
129												
129												
133												
133												
133												
133												
137												
137												
137												
137												
141												
141												
141												
141												
145									!			
145	4								11			
145									P			
145												

Figure 7-27a: Middle of Piston Rod at Level 10

Values for nodes	Level 9				rod. swf		L Paths and M Paths					
	17	21	25	29	33	37	41	45	49	53	57	61
97					I							
97	9				21				17			
97					L P							
97					!							
101												
101												
101												
101												
105												
105												
105												
105												
109												
109												
109												
109												
113					!							
113	3				11				4			
113					P							
113					!							
117												
117												
117												
117												
121												
121												
121												
121												
121												
125												
125												
125												
125												
129					!							
129	0				8				-2			
129					P							
129					!							
133												
133												
133												
133												
137												
137												
137												
137												
141												
141												
141												
141												
145					!							
145	4				W				2			
145												
145					!							

Figure 7-27b: Middle of Piston Rod at Level 9

Values for nodes - Level	8 rod.swf L Paths and M Paths											
	17	21	25	29	33	37	41	45	49	53	57	61
97												
97	0				22				19			
97												
101												
101												
101												
101												
105							!					
105			5				18				-8	
105							P					
105							!					
109												
109												
109												
109												
113												
113	-7				12				5			
113												
113												
117												
117												
117												
117												
121							!					
121			1				13				-18	
121							P					
121							!					
125												
125												
125												
125												
129												
129	-4				13				4			
129												
129												
133												
133												
133												
133												
137							!					
137			9				16				-17	
137							P					
137							/					
141												
141												
141												
141												
145												
145	1				17				4			
145					MP							
145					!							

Figure 7-27c: Middle of Piston Rod at Level 8

Values for nodes - Level	7 rod.swf L Paths and M Paths											
	17	21	25	29	33	37	41	45	49	53	57	61
97							!					
97	-9		2		15		23		19		2	
97							L P					
97							!					
101												
101												
101												
101												
105							!					
105	-15		-9		9		22		15		-10	
105	P						L P					
105	!						!					
109												
109												
109												
109												
113	\ !						!					
113	-16		-12		7		25		15		-14	
113	MP -		- P				P					
113	!						!					
117												
117												
117												
117												
121	!						!					
121	-11		-9		7		26		13		-17	
121	P						P					
121	!						!					
125												
125												
125												
125												
129	!						!					
129	-9		-7		15		31		13		-14	
129	P						P					
129	!						!					
133												
133												
133												
133												
137	!						!					
137	-9		-3		15		32		14		-20	
137	P						MP -		- P			
137							!					
141												
141												
141												
141												
145							!					
145	-6		-3		19		32		9		-22	
145							MP					
145							!					

Figure 7-27d: Middle of Piston Rod at Level 7

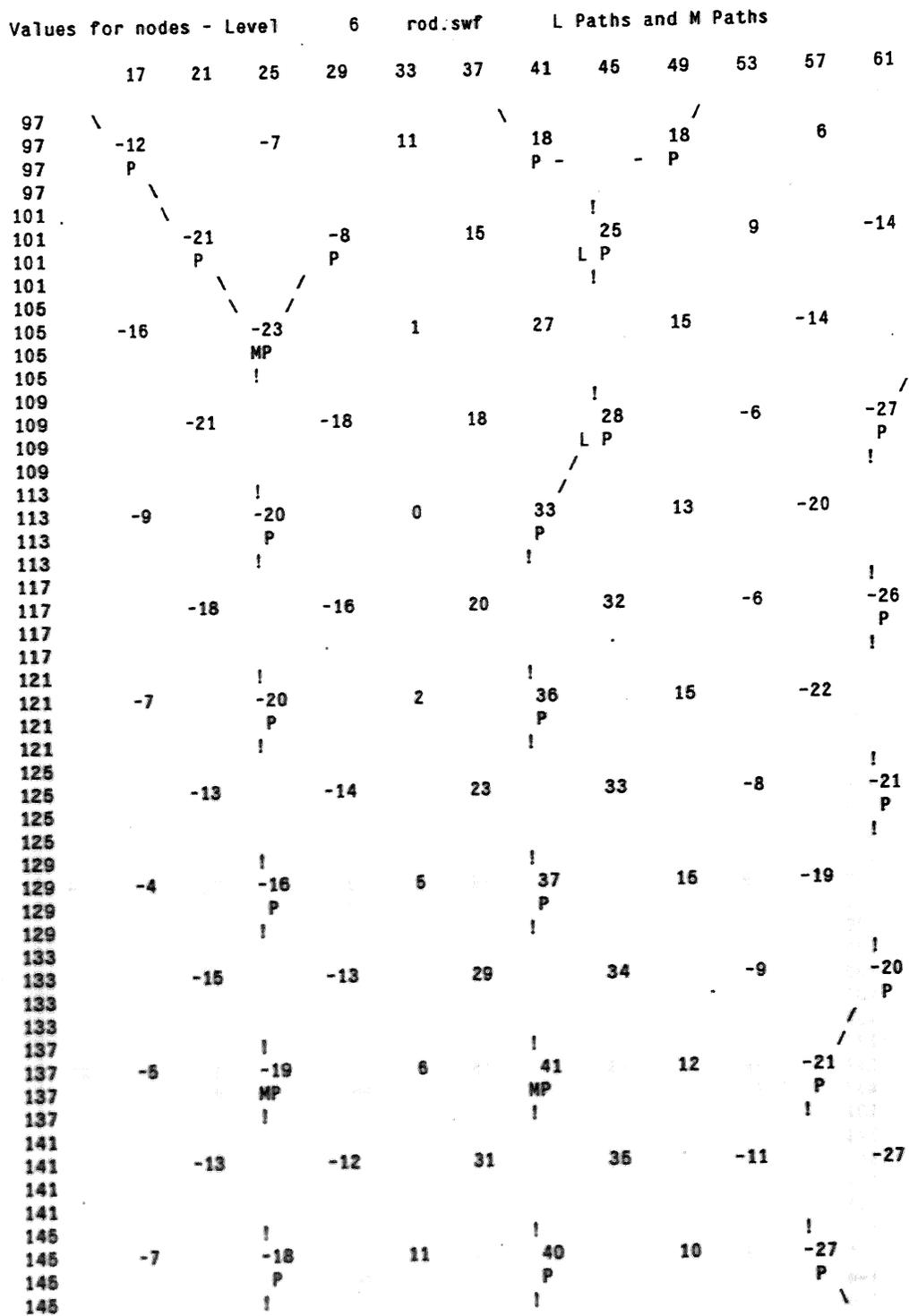


Figure 7-27c: Middle of Piston Rod at Level 6

Values for nodes - Level		5		rod.swf		L Paths and M Paths						
	17	21	25	29	33	37	41	45	49	53	57	61
97												
97	-19	-21	-14	3	16	16	14	16	21	19	12	-3
97		P				L P			P --	P		
97		!										
101		!										
101	-15	-26	-26	-13	2	10	15	21	19	5	-10	-23
101		LMP*	LMP*				P --	P				
101		!						!				
105		!										
105	-8	-18	-26	-24	-8	11	20	23	14	-4	-24	-29
105			LMP*	P --	P			P				LMP
105		!										
109		!										
109	-3	-13	-21	-22	-8	12	28	27	12	-10	-28	-24
109				P			P				L P	
109				!			!				!	
113				!			!				!	
113	0	-9	-19	-22	-8	18	37	34	14	-13	-27	-19
113				P			L P				L P	
113				!			!				!	
117				!			!				!	
117	-2	-6	-19	-22	-8	18	37	36	10	-16	-27	-17
117				P			L P				LMP	
117				!			!				!	
121				!			!				!	
121	0	-9	-17	-23	-9	20	40	33	9	-21	-25	-17
121				MP --	P	P --	LMP				P	
121				!			!				!	
125				!			!				!	
125	0	-8	-16	-22	-8	18	38	32	5	-20	-26	-15
125				P			L P				MP	
125				!			!				!	
129				!			!				!	
129	1	-7	-15	-22	-8	22	43	34	4	-17	-23	-13
129				P			L P				P	
129				!			!				!	
133				!			!				!	
133	2	-6	-19	-22	-9	25	43	36	7	-18	-20	-11
133				L P			L P				P	
133				!			!				!	
137				!			!				!	
137	0	-8	-18	-22	-6	24	43	34	6	-18	-21	-12
137				P			L P				P	
137				!			!				!	
141				!			!				!	
141	-2	-11	-21	-22	-2	28	44	33	2	-22	-24	-12
141				P			L P				P	
141				!			!				!	
145				!			!				!	
145	-1	-10	-20	-22	-3	28	46	32	-1	-21	-24	-14
145				P			L P				P	
145				!			!				!	

Figure 7-27f: Middle of Piston Rod at Level 5

Values for nodes	Level 4			rod.swf		L Paths		and M Paths		49	51	53
	29	31	33	35	37	39	41	43	45			
117	!	/					/					
117	-24		-13		18		30		29		10	-20
117	IMP						MP					
117	!						!					
119												
119		-22		2		24		29		20		-12
119												
121	!						!					
121	-24		-17		17		29		27		1	"26
121	MP						P					
121							!					
123												
123		-24		0		22		28		15		-16
123		MP										
123		!										
125							!					
125	-23		-15		14		30		27		-1	-25
125	P						P					P
125	!											*
127		I										
127		-23		0		26		33		18		-15
127		P						P				
127		!						I				
129	I											»
129	-23		-16		16		33		31		2	-23
129	P											P
129												!
131							!					
131		-24		2		27		36		19		-17
131		MP						IMP				
131								!				
133												!
133	-23		-14		17		35		32		0	-24
133	P											MP
133												!
135								I				
135		-22		3		29		36		18		-17
135								IMP				
135												
137	!											I
137	-23		-8		18		34		30		-3	-23
137	L P						P					t P
137	!						!					*
139												
139		-23		8		31		33		15		-19
139	P											
139												

Figure 7-27g: Middle of Piston Rod at Level 4

	Values for nodes - Level				3	rod.swf		L Paths and M Paths					
	29	31	33	35		37	39	41	43	45	47	49	51
117		j			!					/	j		
117	-20	-28	-17	9	20	15	13	15	19	21	15	-7	-24
117		I			P		P	---	P	--	P		P
117		/ j			j						j		
119		/ I			j						I		/
119	-20	-27	-19	3	21	15	13	15	18	19	7	-14	-26
119	-	P	P		MP	--	P	--	P	--	P		P
119		t			j						j		j
121		I			t						t		J
121	-16	-27	-18	4	17	15	9	12	17	16	2	-19	-26
121		P			P				P*				L P
121		!			!				J				t
123		«			I				f				»
123	-14	-24	-19	4	20	14	8	12	18	15	2	-20	-28
123		P			MP				P			P	-LMP
123		i			!				!			j	
125		j			»				\			t	
125	-16	-25	-16	6	18	15	13	13	20	18	2	-20	-25
125		L P			P				P			L P	
125		I			j				I			f	
127		j			t				j			J	
127	-18	-27	-19	3	19	16	13	17	21	20	1	-20	-22
127		LMP	--	P	MP	--	P		P			P	
127		l			!				I			J	
129		!			i				•	»		I	
129	-20	-26	-18	4	16	16	17	21	25	22	0	-17	-24
129		P			P		P	--	P	--	MP	--	P
129		!			t				!			l	
131		l			l				l				
131	-18	-25	-14	4	15	15	16	21	22	20	-1	-22	-25
131		L P			P				P			P	--LMP*
131		I			l				l			!	
133		I			f				j			!	
133	-16	-25	-14	7	15	15	17	22	25	20	-1	-22	-25
133		L P			P		P	--	P	--	MP	--	P
133		!			t				S			!	
135		I			!				f			/	!
135	-1E	-24	-11	10	18	15	14	20	24	20	-4	-21	-21
135		L P			P	--	MP		P			P	f
135		j			!				I			l	l
137		I			!				I			I	I
137	-20	-23	-11	9	18	16	15	19	24	17	-2	-20	-20
137		P			MP				P			P	P
137		I			t				I			J	l
139		!			t				l			l	!
139	-18	-22	-11	9	13	17	17	22	24	15	-4	-22	-22
139		P			m	--	P		P			P	P
139		l			t				/	I		t	I

Figure 7-27h: Middle of Piston Rod at Level 3

Chapter 8

Matching the Representation

This chapter concerns matching the representations of pairs of gray scale forms, particularly in situations where:

- the two forms are in digitized images of the same object (or very similar objects), and
- one of the objects was at a different distance and/or 2-D image plane orientation from the camera than the other at the time of digitization.

This chapter provides examples of the rotational quasi-invariance and the size quasi-invariance of the representation developed in the previous chapters. However the techniques involved in such matching can also be used for stereo image interpretation and object recognition. Thus, it is worthwhile to develop principles and approaches to such matching while demonstrating the properties of the representation.

The remainder of this *section* discusses the role which correspondence plays in stereo interpretation and structural pattern recognition. Section 8.2 summarizes the matching techniques which are illustrated in this chapter. These techniques are preliminary; matching was not within the domain of this research. These techniques were explored to assist in demonstrating the usefulness of the representation and as a preliminary tool at an important problem which we will address when this **disserctatton** is complete. This is followed by a section which presents the test data (section 8.2) **which** was used to verify the **skel** and **rotational** invariance of the representation.

Sections 8.3 and 8.4 concern the use of M-nodes (local peaks at a level), M*-nodes (local peaks among the levels), and P-paths (ridges at a level) for determining the relative position, orientation and size of two representations of the same (or similar) gray scale forms. In section 8.3 the concept of connected M*-nodes is defined and an example is **presented**. Section 8.4 illustrates the **zmtmpQn^mtt** of M-nodes **aid** M*^Hnode& in rotated and scaled images of an object using the teapot images. This section ends by showing the correspondence of the M-nodes in a stereo pair of paper wad images. Section 8.5 discusses the use of the M-node **aHYcqxtudnce** to align L-paths (ridges among the levels) from rotated and scaled images of an object and describes a simple similarity measure for **digbed** (-paths. Its section ends with examples of **miichiiiig** the **L-paths** from the **right** **HLtlc** **inkHuw** of the teapot **inaSe***

8.0.1 Applications of Correspondence Matching

This Subsection briefly introduces the matching problem in the domains of stereo matching and structural pattern recognition. It also describes the properties of the representation that make it useful in these domains.

In image understanding there are several problem domains where it is desirable to determine the correspondence between parts of two representations. One such problem domain is interpretation of pairs of stereo images to obtain depth information. Depth information is obtained from a stereo pair of images by triangulation. Triangulation depends on knowledge of the relative positions and orientations of two cameras, the so-called "camera parameters" [Duda 73]. The "stereo correspondence" of surface points in the images is also required. This is the positions of pixels in the two images that correspond to the same point on the surface of an object. It is then possible to set up the projective geometry that relates the two cameras to points on the surface of objects. Given this geometry, the distance may be computed from one of the cameras to each surface point for which correspondence is known. These distances provide a map of the 3-D form of a scene.

Before the depth to a surface point can be computed, it is necessary to determine the location of the pixels which correspond to that surface point in each of the images. This stereo correspondence problem is the most difficult problem in stereo image interpretation. The usual approach to this problem is to correlate patches in the two images. But this is an expensive process, and there are problems with determining how large a neighborhood to correlate.

The representation developed in the previous chapters has properties which greatly simplify the process of determining the correspondence of patterns of pixels in two images.

1. Only peaks correspond to peaks. The existence of peaks or M-nodes provides a set of landmarks which can be used as tokens in the matching process.
2. The multi-resolution hierarchical structure of the representation permits the correspondence process to commence with the most global M^* nodes for each form. Since very few such symbols exist at the coarsest resolution, the complexity of this process is kept small.
3. The connectivity of M-paths permits the match information from a coarse resolution to constrain the possible set of matches at the next higher-resolution level. Thus what could be a very large graph matching problem is repeatedly partitioned into several small problems.

Another important problem domain in image understanding is classifying two dimensional gray scale forms. The representation developed in this dissertation can be used for a structural pattern recognition approach to this problem. That is, a gray scale form may be classified by measuring the similarity of its representation to a number of prototype representations for object classes. This approach was described briefly in chapter 1 for both 2-D gray scale forms and for 3-D shapes.

The properties of the representation cited above facilitate its use for constructing object-class

prototypes and for matching prototypes to object representations. An object class prototype may be formed by constructing the representations of a training set of images. The configurations of M-paths and L-paths that occur for a given class of objects can be determined by matching the representations from this training set. The prototype description can be composed of the M-paths and L-paths that occur in the majority of the descriptions.¹³ This provides a simplified representation which can serve as an object class prototype. The multi-resolution hierarchical structure of the representation permits the set of possible matching prototypes to be reduced on the basis of the few coarsest resolution symbols.

The study of creating and matching such prototypes could be a dissertation in itself. Only a few of the more obvious principles and techniques are described below.

8.1 A Matching Procedure for Descriptions of Similar Grey Scale Forms

This section describes a matching procedure for descriptions of the same or similar objects from two images. The investigation of such matching is a research topic which we expect to pursue in the near future. The procedures described below are very preliminary; matching techniques were not within the scope of the research proposed for this dissertation. These techniques were investigated to assist the demonstration of the usefulness of the representation for matching, and to show the invariance of the representation to changes of the size and orientation of a gray-scale form.

Matching is treated as a problem of comparing a reference description to a measured description. In this process the reference description is transformed in size, orientation, and position so as to bring its components into correspondence with the measured data. The goal of this process is to determine:

- the overall relative position, orientation, and size of the of the forms represented in the two descriptions,
- which M*-nodes, M-nodes, and L-nodes in the reference description correspond to which M*-nodes, M-nodes, and L-nodes in the measured description (the correspondence mapping),
- local relative changes in position, orientation, and size between parts of the reference description and the corresponding parts of the measured description,
- parts in either of the descriptions that do not occur in the other description.

Such matching consists of several steps:

1. Initial alignment: In this stage the most global M*-node(s) is(are) used to determine the relative positions and sizes of the two descriptions.

¹³Although this technique has been tried for a few hand examples, we have not, as of this writing, tried to implement it in code.

2. Orientation: Given the relative positions and sizes, the correspondence of M-nodes and L-nodes in the few levels below the most global M*-node(s) can be used to estimate the relative orientations of the two descriptions. This correspondence can be found by the same procedure used for the following task.
3. Correspondence of M-nodes: Each level in which there is more than one M-node in the description of a form, gives a graph composed of M-nodes connected by ridges (P-paths). Each P-path has the attributes of distance and orientation between the M-nodes at either end. Techniques exist for determining the correspondence between nodes in such a pair of graphs. Indeed, when the number of nodes is small it is not unreasonable to exhaustively examine every possible correspondence. A similarity measure, such as the average difference in the lengths and orientations of the P-paths may be used to determine the correspondence which is most likely. A fundamental principle in matching descriptions from an SDOG transform is to use the correspondence at the previous (lower frequency) level to constrain the set of possible correspondences at the next (higher frequency and higher resolution) level. This prevents the computational complexity of matching M-nodes from growing exponentially as the number of M-nodes grows exponentially with increasing resolution.
4. Correspondence of L-nodes: Forms which are elongated can result in a description which contains few M-nodes. The shape of such forms can be compared by comparing the L-paths in their descriptions. Comparing L-paths consists of two stages:
 - alignment of the L-paths by aligning the M*-nodes which terminate the L-paths at each end, and
 - computing the distance of each L-node in the reference L-path to the nearest L-node in the measured L-path.

Determining the correspondence of individual L-nodes in two descriptions is not a reasonable approach because the distance between L-nodes in an L-path varies by as much as a factor of $\sqrt{2}$ with orientation. Measuring the distance from each L-node in one description to the nearest L-node on the second description allows the measures of maximum distance and average distance to be used to compare the entire L-path

8.2 Test Data

The matching techniques described in this chapter are illustrated with representations from five teapot images.¹⁴ These images were formed by photographing a scene composed of a teapot flanked on either side by a cup; all of these objects are on a white table cloth. The photographs were taken with a 35 mm camera using a 55 mm lens and Pan-X black and white film. The negatives were digitized by SRI-International to 512 by 512 by 8 bits. Test images of the teapots were formed by cropping 256 by 256 pixel sections from each image. The pixel values in these cropped sections were then normalized to have a mean of 128 and a standard deviation of 32.

¹⁴A sixth teapot image was also formed and processed but the tape on which the image was stored became unreadable during preparation of this dissertation

Images were formed at three scales by moving the teapot away from the camera. This movement changed the position of the teapot and cups with respect to the lights, causing some changes in shading and shadows among the images of different sizes. The distances are such that if the size of the smallest teapot image is defined as 1.0, the middle scale images are larger by a factor of 1.14 and the largest images are larger by a factor of 1.36.

At each distance, a second photograph was taken with the camera tilted by approximately -15° . Thus there were originally six teapot images. The scales and 2-D orientations of the five images shown in this chapter are summarized in table 8-1.

Teapot	Size	Orientation
1	1.0	0°
2	1.14	0°
3	1.36	0°
4	1.0	-15°
5	1.14	-15°

Table 8-1: Size and Orientation of five Teapot Images

Reproductions of these five test images are displayed below in figures 8-1 through 8-5. To produce these figures, the original digitized images were displayed with the Grinnell image display on the C-MU Computer Science Dept. IUS VAX. Each display was zoomed by a factor of 2 to simulate the cropping that produced the teapot image. The zoomed images were then photographed with the Dunn film recorder attached to the Grinnell monitor. The resulting 8" by 10" glossy prints were then half-toned to produce the images shown in figures 8-1 through 8-5.

Section 8.4 below describes the results of matching for teapot images #1 through #5.

8.2.1 Example of Band-Pass Images of Teapot

Following the pictures of the test data is a picture showing the band-pass images for teapot #1. The format for this band-pass image is shown in figure 8-6. The actual band-pass images for teapot #1 are shown in figure 8-7. The level 0 band-pass image (also known as the high-pass residue) is shown in the lower right corner. The upper left corner shows the level 1 band-pass image. The level 2 band-pass image is shown in the upper right corner. The level 3 and 4 band-pass images are shown underneath the level 1 image and so on, down to level 13.

The even level images (levels 2, 4, 6, ..., 12) are sampled at $\sqrt{2}$. In order to display these images on a raster display, each pixel on an odd row is used to fill the undefined location to its right, and each pixel on an even row is used to fill the undefined location on its left. This creates an interlocking brick-like texture in the display. This filling was done only for display purposes.

The band-pass levels 12 through 5 are important to the examples given in section 8.4. Since these levels are so hard to see in figure 8-7, they are shown enlarged in figure 8-9. This figure was formed by zooming the display of levels 12 through 5 by a factor of 4. The format for this image is shown in



Figure 8-1: Teapot #1. Size = 1.0, Orientation = 0.0° .



Figure 8-2: Teapot #2. Size = 1.14, Orientation = 0.0°



Figure 8-3: Teapot #3. Size = 1.36, Orientation = 0.0°



Figure 8-4: Teapot #4. Size = 1.0, Orientation = -15.0°

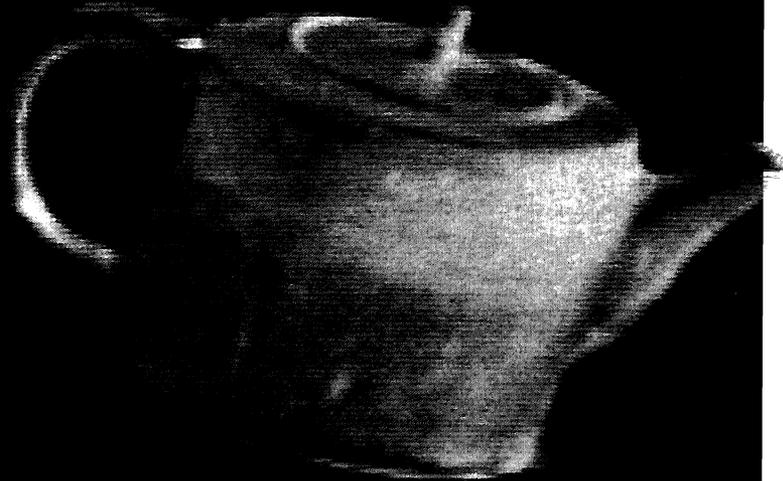


Figure 8-5: Teapot #5. Size = 1.14, Orientation = -15.0°

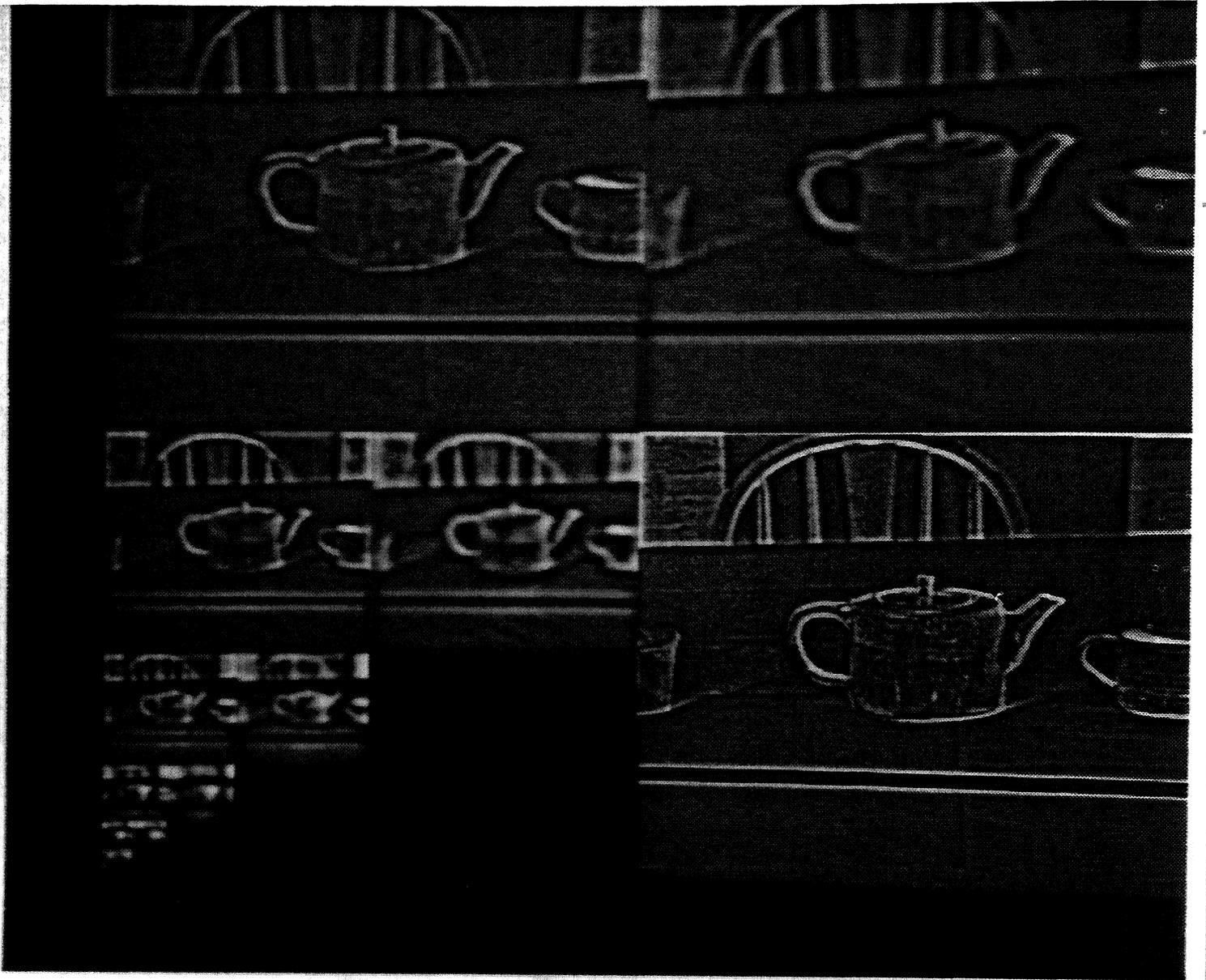


Figure 8-7: Band-Pass Images for Levels 13 Through 0 of Teapot #1

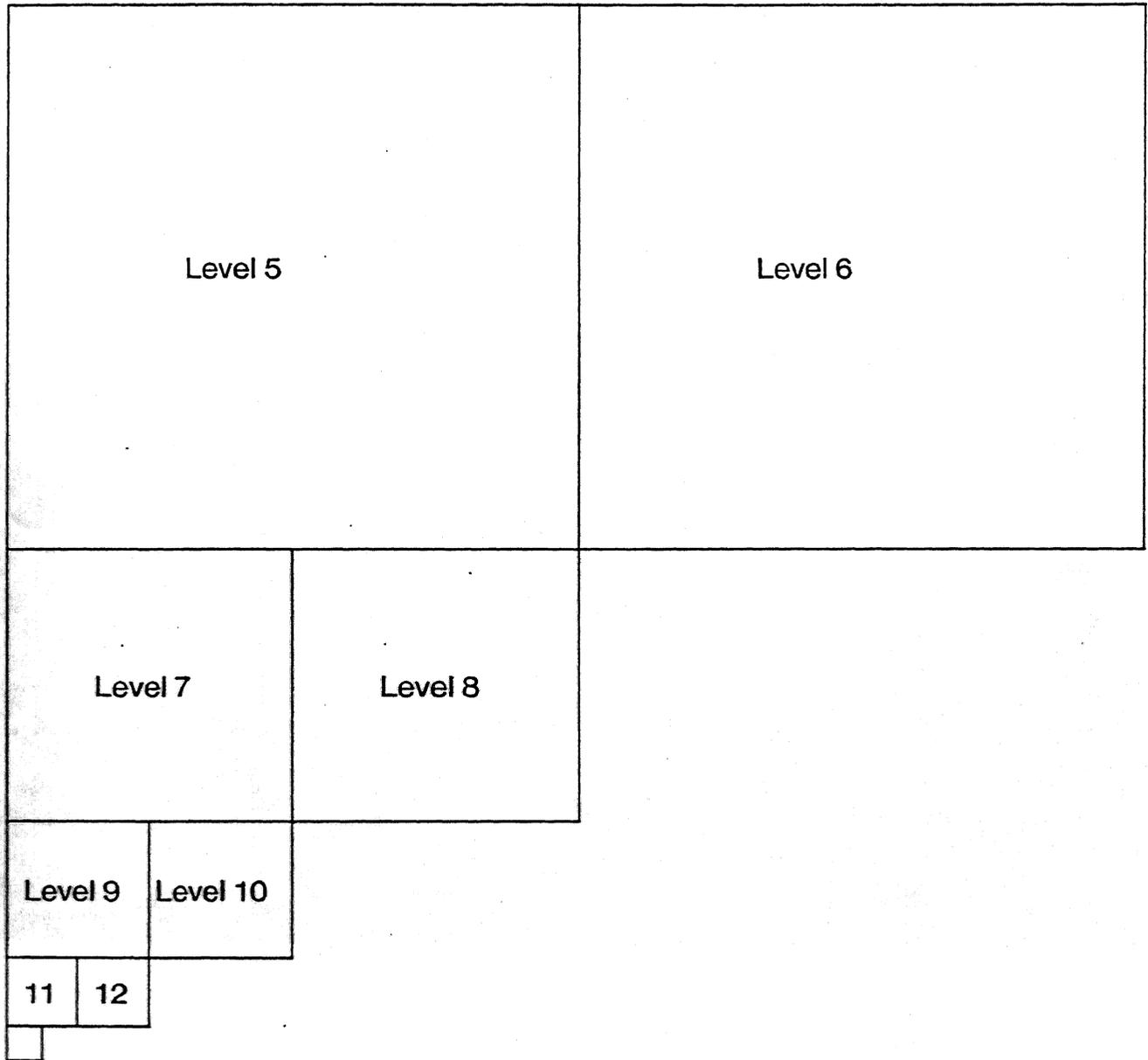


Figure 8-8: Format for Display of Zoomed Band-Pass Levels 13 through 5



Figure 8-9: Zoomed Band-Pass Images for Levels 13 Through 5 of Teapot #1

8.3 Matching M-Paths

This section describes how the M-paths from two representations may be matched to determine the correspondence of M-nodes. The techniques described in this section employ only information that is intrinsic to M-paths and P-paths. For clarity the section starts by describing how this information is obtained from the representation. This additional information may be thought of as either an abstraction from the representation, or as something that is computed from the representation "on the fly". After this M-node representation is described, the process of obtaining the initial alignment based on the highest level (lowest resolution) M* node is described. The correspondence of lower level nodes in the test images is then shown.

The processes described in this section will not work for gray-scale forms which are very long and thin (e.g. roads, rivers, bars, stripes etc.) and do not have ends within the image. These forms are described primarily by L-Paths. Matching L-paths is discussed in section 8.5.

8.3.1 Abstracting M-Paths from the Representation

Unless a gray scale form is a thin form with its end off of the image, it will have one or more M-Paths in its representation. The M-nodes in these M-paths provide tokens for aligning pairs of representations and determining whether structures that exist in one image also exist in another, as well as determining how the structures differ in two images. Determining the correspondence of M-Paths in two representations depends on information which is intrinsic to the M-nodes and the P-paths that connect M-nodes. In order to illustrate M-path correspondence more clearly this section describes this information and how it may be obtained from the representation. The first concept that must be elucidated is that of connected M-nodes.

8.3.1.1 Strongly Connected M-Nodes

Definition: Two M-Nodes are said to be "strongly connected" if and only if:

1. They exist at the same level of the same representation,
2. They are not adjacent to each other (i.e. are not part of the same M-path),
3. They are linked by a P-Path or sequence of P-Paths.

In most cases, M-nodes which are at the same level and of the same form will be strongly connected. When two M-nodes are connected by a P-Path with no intervening M-Nodes along the P-Path between them, they are said to be "directly" strongly connected. If a third M-Node occurs along the P-Path between the two M-Nodes, then the two (outer) M-Nodes are said to be "indirectly" strongly connected. This distinction will come in handy when discussing M-Path matching in the presence of spurious or missing M-Nodes.

8.3.1.2 Weakly Connected M-Nodes

Definition: Two M-Nodes are said to be "weakly connected" if and only if:

1. They exist at the same level of the same representation,
2. They are not adjacent,
3. They are not linked by a P-Path at their level,
4. Other M-Nodes within one level in their M-Paths are strongly connected.

The concept of weakly connected M-Nodes provides for the case where a P-Path has been broken either for reasons intrinsic to the form or because of an error in the P-Path detection algorithm.

Weakly connected M-Nodes can be detected by examining the connectivity above or below them in their M-Paths.

M-Nodes have certain attributes based on their position in the transform space (x,y,k). They also have an attribute that is the value of the filter at that level and location. Also, if desired, they can be assigned a label on the basis of the configuration of oppositely signed ridges around them. Such labeling can simplify the correspondence process.

Connected M-Paths are "linked" by two way pointers. Each half of a pointer may also be assigned the attributes of distance (D) and orientation (θ), which are defined as:

Distance: The distance between two M-nodes is the cartesian distance measured in terms of the number of samples at that level. In levels with a $\sqrt{2}$ sample grid, the distance along the x and y axes are in units of $\sqrt{2}$.

Orientation: The orientation between two M-nodes is the angle between the line that connects them and the x axis in the positive direction (right). For convention, this angle ranges from 0° to 359° in the counter-clockwise direction. Up is 90° , left is 180° and down is 270° .

8.3.1.3 Example of Abstracted M-nodes and P-Paths

Several figures are shown in the next sections to illustrate connected M-Nodes and M-Paths from the upper levels of the teapot images. The following example illustrates how these figures are derived from the representation.

Figure 8-10 shows the M-nodes and P-nodes from level 7 of teapot image #1. Level 7 is the highest level with more than one M-node. Because of space limitations this figure does not include all of the negative ridges surrounding the teapot. This figure shows three positive M-nodes, connected by P-paths. Also present is the negative ridge above the teapot, the negative peak inside the handle of the teapot, and a part of the negative ridge below and to the left of the teapot. The

most important feature of this figure is the presence of the three connected positive M-nodes (peaks) and the P-paths that connect them.

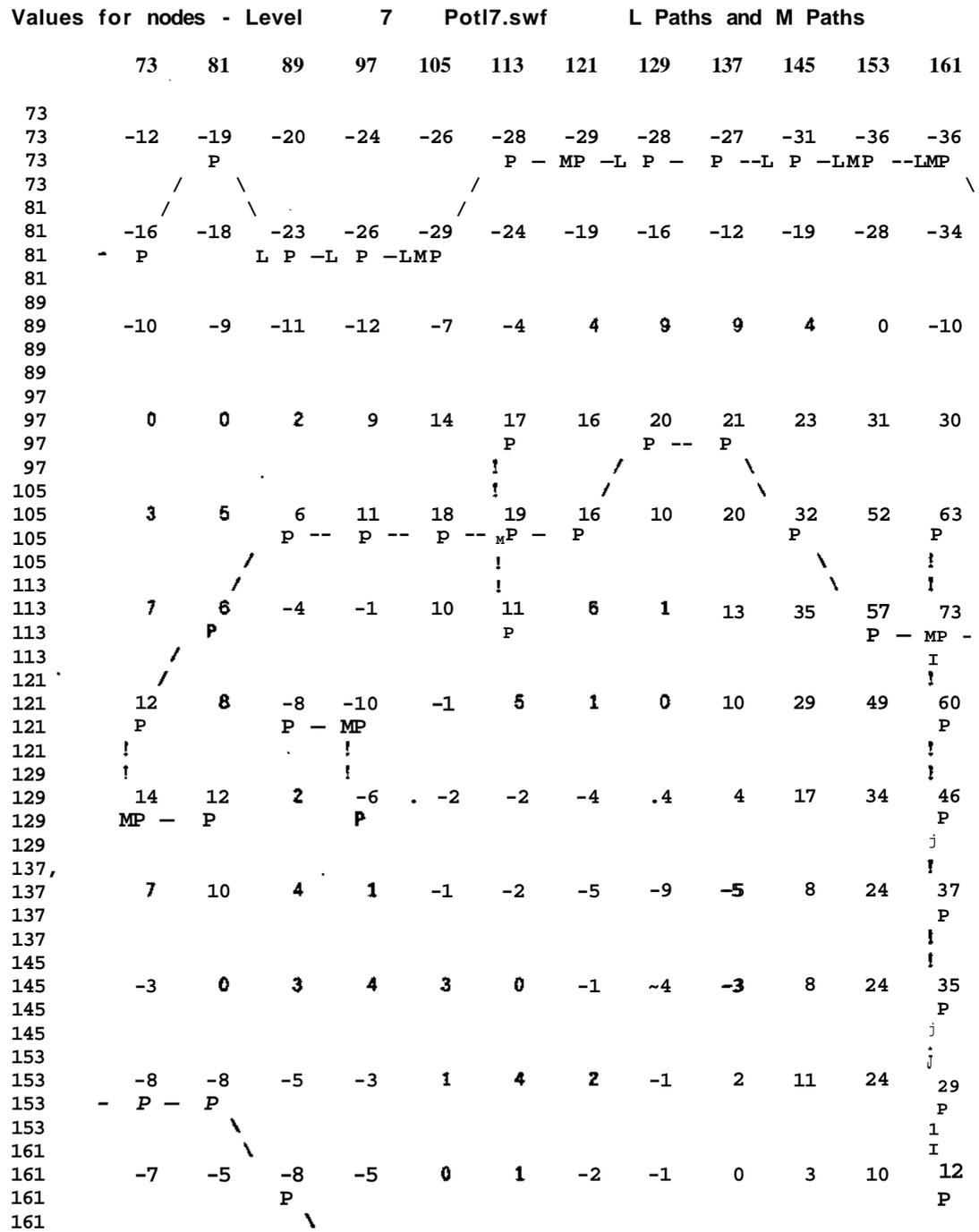


Figure 8^10: Level 7 from Teapot Image # 1

The three positive peaks from level 7 of teapot # 1 are shown abstracted from the band-pass data in figure 8-11. The direct P-Path links between these M-nodes are illustrated with solid arrows and labeled with circled numbers. The indirect P-Path link between the right-most and left-most M-nodes is shown as a dotted arrow labeled with the circled number 3. The numbers are an index into a table

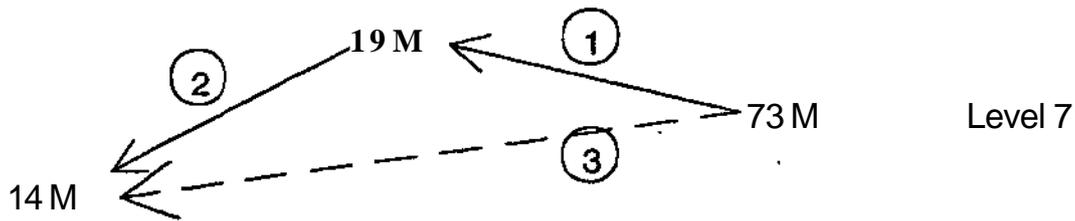


Figure 8-11: M-nodes and P-Paths for Level 7 of Teapot #1

of attributes. The attributes for these particular links are given in table 8-2 in the next section. This same set of links is included in figure 8-12. These numbers are also used to show the correspondence which was assigned by hand matching between these links and the same links in the other teapot images,

8.4 Examples of M-node Correspondence

This section presents examples of M-node correspondence using the most global levels of the teapot images. In each of the examples, the M-nodes from the most global level (level 12) to the second highest level with more than one M-node are used.

This section begins with the M-node graph for levels 12 through 6 of teapot image #1. This is followed by the results of hand matching this graph to teapot image #3 (scale = 1.36, orientation = 0°) and to teapot image #4 (scale = 1.0, orientation = -15°). Other examples of M-node matching for the teapot images are then presented and discussed. The section ends with M-node matching for the upper levels of the stereo pair of paper wad images.

Figure 8-12 shows the upper M-nodes, M-Paths and P-path links for teapot image 1. In figures 8-12 and the other M-nodc graphs, the M-path links are shown as a dark line. Lighter solid arrows are shown between directly linked M-nodes at each level. Dashed arrows are shown connecting some indirectly linked M-nodes.

Each P-path link in the M-nodc graphs (such as figure 8-12) is labeled with a circled number. These labels were assigned by hand on the basis of the length and relative orientations of the P-paths. In the assignment of the labels in the second level with more than one M-nodc, the correspondence of the M-nodes in the level above this level was used to constrain the possible set of correspondences*. As mentioned above, these numbers also serve as an index into a table of attributes for the links.

These attribute tables give the values for dx , dy , D , and θ for each P-path link. The positive directions for dx and dy are the same as used in the image: $+x$ points right, $+y$ points down. However, note that θ increases in the counter-clockwise direction. In these tables; in the levels which

are at a $\sqrt{2}$ sample grid, the distances dx and dy are recorded in units of $\sqrt{2}$. In cases where an M-node spans two adjacent samples, the M-nodes position is assigned at the mid-point between them. This results in values of dx or dy that have fractional parts of .5 in the cartesian sampled levels, and .25, .5 or .75 in the $\sqrt{2}$ sampled levels.

In these tables, orientation (θ) is measured in degrees. On a cartesian grid, at distances that are typically 5 to 10 pixels, angular resolution is typically 5 to 10 degrees. Of course, the longer the distance, the more accurate the estimate of orientation.

8.4.1 M-nodes for Teapot Image # 1

The M-nodes for levels 12 through 6 of teapot image #1 are shown in figure 8-12. As shown in table 8-1 this is the smallest "non-rotated" teapot image. In levels 12 through 9 of figure 8-12 only a single M-node occurs in the teapot. These M-nodes all occur within a distance of two samples of the M-node above them, and are thus linked into a single M-Path.¹⁵ This M-path is referred to as the principal M-Path. The M-node at level 8 has the largest value along this M-path and is thus marked as an M*-node. This M*-node corresponds to a filter with a positive center lobe of radius $R_+ \approx 18$ pixels¹⁶ (see equation (6.5)) or a diameter of 37 pixels. This corresponds to the form in the image that results from the overlap of the shadow on the right side of the teapot and the darkly glazed upper half of the teapot which appears as a light region in figure 8-1.¹⁷ At level 7, additional detail begins to emerge. M-nodes occur over the upper right corner of the teapot and over the handle region. These M-nodes are joined to the M-node on the principal M-path by a P-Path. These P-Paths are illustrated by a solid arrow.

The indirect links between the M-node on the principal M-path and other M-nodes are shown as dashed arrows. There are two reasons for showing the attributes of the indirect links between these M-nodes:

1. In some of the teapot images, the M-node corresponding to the M-node of value 19 at level 7 does not occur. In such a case the indirect link labeled as 3 occurs as a direct link.
2. Quantization introduces an error into the attributes D and θ . The magnitude of the error in the D term is independent of D . Thus the proportion of D dominated by the error decreases as D increases. The error in θ decreases as D increases. Thus longer links provide a more accurate measure of the scale and orientation of the object.

Five M-nodes occur in level 6. Three of these M-nodes occur underneath (within 2 samples) of M-nodes from level 7. These three M-nodes are thus part of three M-paths. The remaining two

¹⁵The M-path links appear as straight dark lines in figure 8-12 although in fact there can be a lateral shift of up to two samples between their positions. M-path linking was described in section 7.4.

¹⁶A pixel is the sample rate in the original image

¹⁷The teapot images were digitized from negatives. Thus dark forms appear light in figures 8-1 through F86.

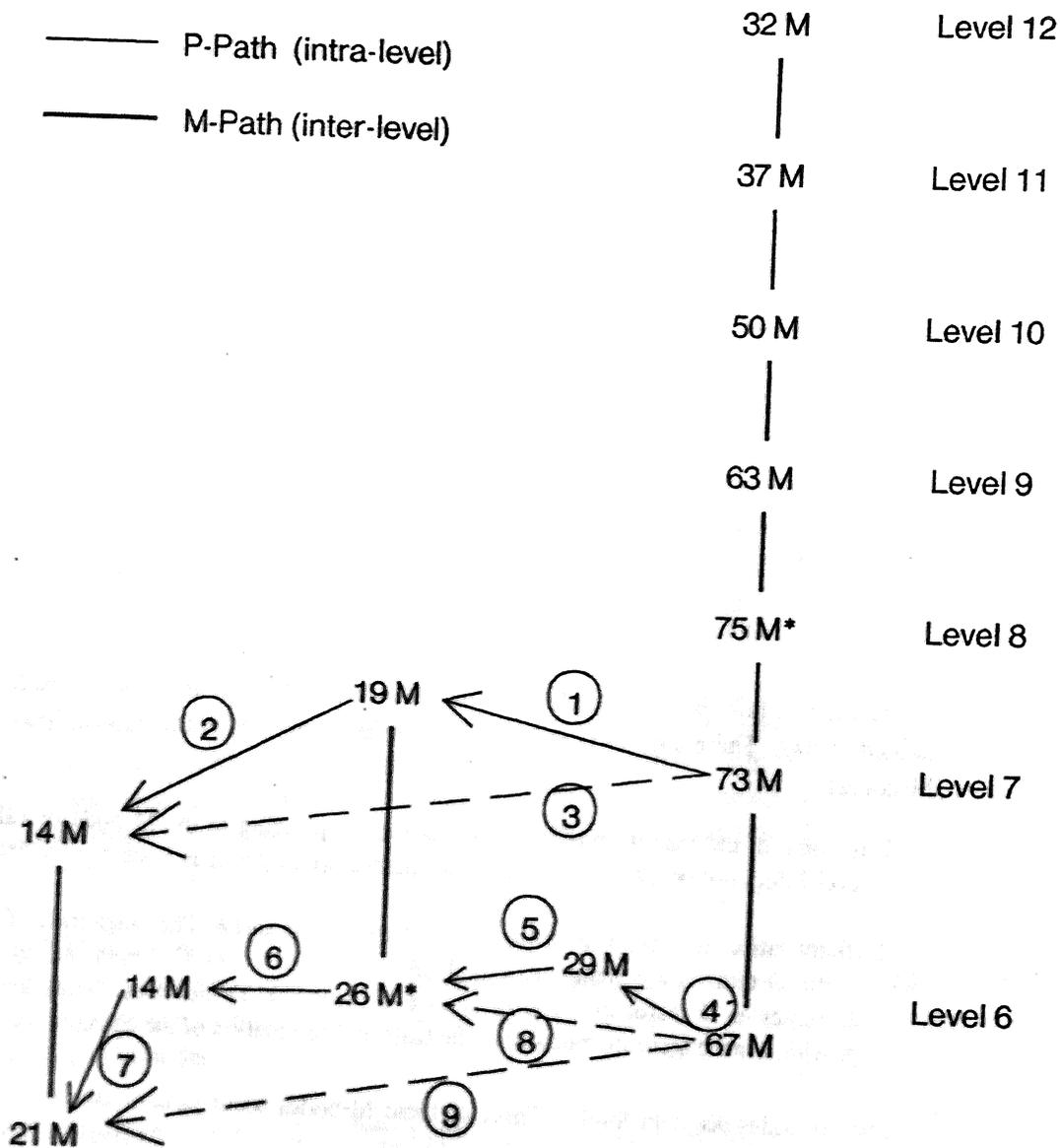


Figure 8-12: M-nodes and P-Paths for Levels 12 to 6 of Teapot #1

P-Path	Level	dx	dy	D	θ
1	7	-6	-2	6.32	161.5°
2	7	-5	3	5.83	210.9°
3 (1&2)	7	-11	1	11.04	185.2°
4	6	$-4.0\sqrt{2}$	$-2.0\sqrt{2}$	6.32	153.4°
5	6	$-3.25\sqrt{2}$	$1.5\sqrt{2}$	5.06	205.8°
6	6	$-3.0\sqrt{2}$	0.0	4.24	180°
7	6	$0.25\sqrt{2}$	$3.25\sqrt{2}$	4.6	265.6°
8 (4&5)	6	$-7.25\sqrt{2}$	$-0.5\sqrt{2}$	10.2	176.1°
9 (4&5&6&7)	6	$-10\sqrt{2}$	$2.75\sqrt{2}$	14.6	195.3°

Table 8-2: P-Path Links for Levels 7 and 6 of Teapot #1

M-nodes are in fact the highest levels of two more M-paths. For simplicity, this illustration shows only the indirect links for the M-nodes that are part of established M-paths at level 6.

Note that one of the M-nodes at level 6 is an M* node. This M-node corresponds to the upper left corner of the teapot. This M*-node marks the left end of the dark region of glaze on the upper half of the teapot. The width of the positive center lobe of the filter which corresponds to this M*-node gives an approximation of the width of the darkly glazed region.

8.4.2 Initial Alignment to Obtain Size and Position

An initial estimate of the alignment and relative sizes of two gray scale forms may be constructed by making a correspondence between their highest level M*-nodes. This is illustrated by comparing the M-nodes and links in figure 8-12 to those in figure 8-13 shown below. Figure 8-13 shows the M-nodes and P-Path links for teapot number #3. Recall from table 8-1 that teapot #3 has the same orientation as teapot #1 and is scaled larger in size by a factor of 1.36 which is just less than $\sqrt{2}$. The distance and orientation for each P-Path link in teapot #3 levels 12 through 7 is shown in table 8-3 below.

The highest level M*-node in teapot #3 occurs at level 9. The fact that this M*-node is one level higher than the highest level M*-node for teapot #1 confirms that teapot #3 is approximately $\sqrt{2}$ larger than teapot #1.

The correspondence of the highest level M*-nodes from these two teapots gives an estimate of the alignment of the two teapots as well as the scaling. The correspondence tells us the position at which teapot #1, scaled by $\sqrt{2}$ in size will match teapot #3. The tolerance of the initial alignment is dependent on which of the teapots is designated as a reference pattern. The reference pattern is the one which is scaled, rotated and translated so that its components are brought into correspondence with the second, observed pattern. In this matching (as well as with stereo interpretation) which image is used as the reference image and which image is used as the data image is arbitrary. The tolerance of the initial position alignment is \pm the sample rate at the level of the M*-node in the data image. If teapot #3 is designated as the data image, then the sample rate at level 9 determines the tolerance. The positioning tolerance at level 9 is $\pm 8\sqrt{2}$ pixels.

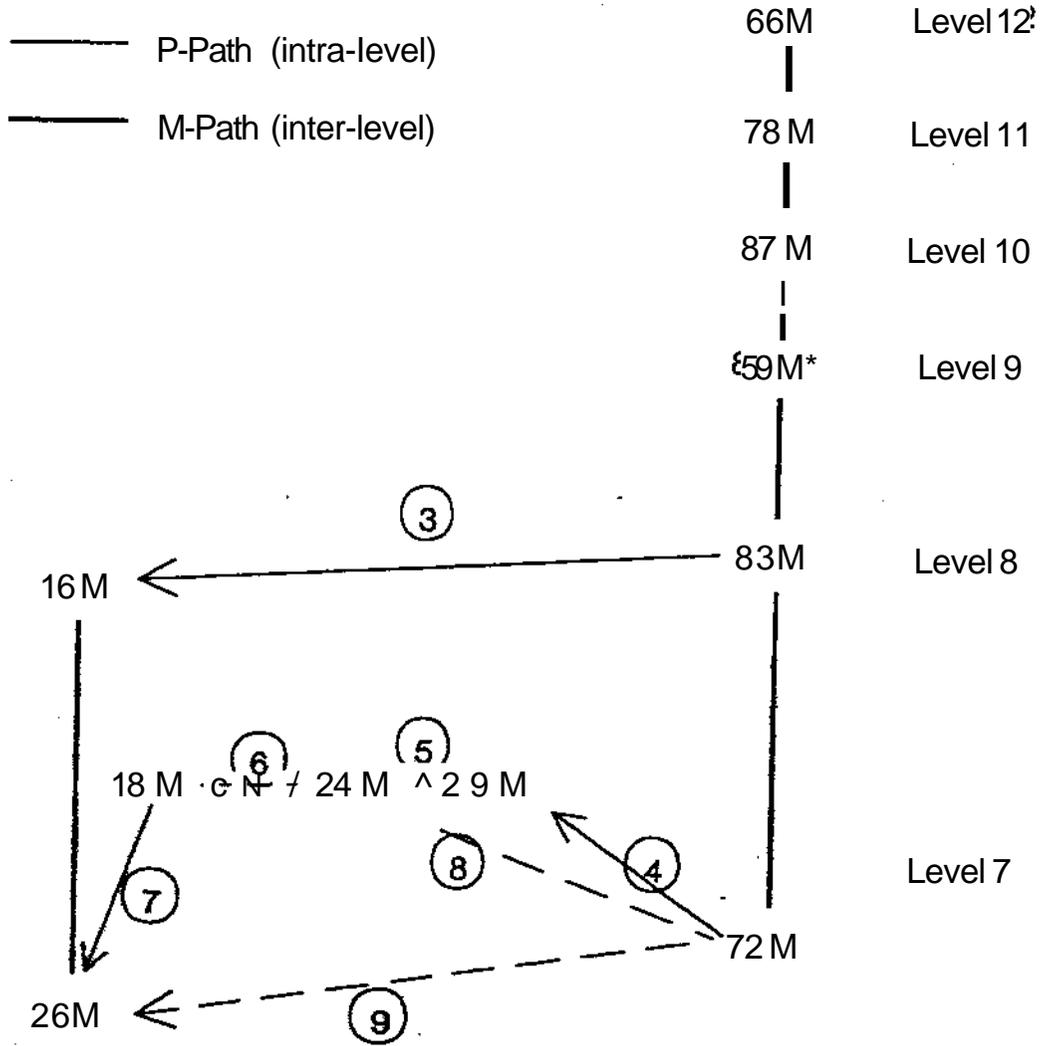


Figure &-13: M-nodes and P-Paths for Levels 12 to 7 of Teapot #3

P-Path	Level	dx	dy	D	θ
3	8	-7.5VT	1.5V $\sqrt{2}$	10.81	191.3°
4	7	-3.5	-6.0	6.94	149.7°
5	7	-4.0	1.0	4.12	194.0°
6	7	-4.5	1.0	4.61	192.0°
7	7	-0.5	5.0	5.02	264.3°
8(4&5)	7	-10.0	-1.5	10.11	171.5°
9 (4&5&6&7)	7	-15	3.5	15.4	193.1°
11	7	-1.5	6.5	6.67	257.0°

Table 8-3: P-Path Links for Levels 8 and 7 of Teapot # 3

The tolerance of the size scaling is less than $\pm\sqrt{2}$. The correspondence of the highest level M*-nodes provides an estimate of the size scaling factor which is a power of $\sqrt{2}$. Such an estimate is sufficient to constrain the correspondence process. A more accurate estimate can be obtained from the average of the ratio of D's for links whose correspondence has been found. An example of this will be given in the next section.

8.4.3 Determining Further Correspondence and Orientation

The matching process starts by finding the correspondence for the highest level M*-nodes. This provides the process with an initial estimates of the size and position of the two forms. The next step is to find the correspondence of lower level M-nodes to refine the estimates of relative size and position, discover the relative orientations, and discover where one of the forms has been distorted by parallax or other effects.

Let us continue with our example. An M-nodc for the upper left corner of teapot #3 does not occur. The change in scale from teapot # 1 to teapot #3 was not enough to bring this M-node up to level 8. This may also be a result of the slight difference in shading that resulted from moving the teapot with respect to the lights and camera in order to size scale the object, the fact that the M-node of value 16 in level 8 of teapot #3 corresponds to the M-nodc of value 13 in level 7 of teapot #1 must be discovered from the position relative to their principal M*-nodcs and the distance and orientation from the M-nodc on the principal M-path at the same level.

The values for D and θ for the link attributes in levels 7 and 6 of teapot I are compared to the attributes in the corresponding links from levels 8 and 7 of teapot 3 in table 8-4. The reader should remember that all of these links are constrained to begin and end at samples in their respective levels*. Because we are dealing with distances of between 4 and 15 samples at arbitrary angles, there is quantization noise in these attributes. The differences in orientation are shown in the column labeled $\theta - \theta_r$. Except for link 3, these values show a consistent small rotation in the counter-clockwise direction for the links from teapot 3. In light of this, the image data was re-examined after compiling this table. landmarks were chosen at the base of the handle and the base of the spout in both images. In teapot # 1, this baseline had an angle of 3.8° relative to the raster line. In teapot #3, this baseline

P-Path	Teapot 1		Teapot 3		$\theta_1 - \theta_3$	D_3/D_1	Difference	
	D_1	θ_1	D_3	θ_3			$D_3 - D_1$	$100 \times (D_3 - D_1)/D_3$
3	11.09	185°	10.8	191°	-6°	0.974	-0.2	-1.8%
4	6.3	153°	6.9	148°	5°	1.095	0.6	8.7%
5	5.1	206°	4.1	194°	12°	0.804	1.0	24.4%
6	4.2	180°	4.6	192°	12°	1.09	0.4	8.7%
7	4.6	266°	5.2	264°	2°	1.13	-0.6	-11.5%
8	10.2	176°	10.1	171°	5°	0.99	-0.1	-1.0%
9	14.6	195°	15.4	193°	2°	1.05	0.8	5.2%
Average Error					4.57°	1.020	0.257	4.3%

Table 8-4: Comparison of D and θ attributes for Teapots 1 and 3

had an angle of 7.1°. Thus it appears that the two teapots actually have a relative change in orientation of approximately 3.3°. The actual values of θ fluctuate more than this due to quantization error from sampling and changes in shading.

The ratio D_3/D_1 would show a factor by which the lengths consistently shift when the teapot is scaled by 1.36. Since this shift in scale was enough to drive the corresponding P-paths in teapot #3 up to a new level, but less than the $\sqrt{2} = 1.41$ scale change between levels, an average ratio of $D_3/D_1 = 1.36/1.41 = 0.96$ was anticipated. In table 8-4 we see that this average ratio worked out to 1.02. Our conclusion is that quantization noise and changes in shading accounted for most of this difference. The actual differences in length, $D_3 - D_1$, show that the lengths were always within one sample. Except for link 5, the percentage differences, $(D_3 - D_1)/D_3$ were generally small ($\leq 10\%$). The conclusion from this experiment is that the correspondence between M-nodes from similar gray-scale forms of different sizes can be found, provided that the matching tolerates variations of the lengths of P-paths of up to 25% and variations in the relative angles of up to 12°.

8.4.4 Correspondence of M-nodes Under Rotation

Figure 8-14 shows the M-nodes, M-paths, and P-path links for levels 12 through 6 of teapot image #4. This teapot image is the same size as teapot image #1, but rotated by approximately -15°. Figure 8-14 contains all of the M-nodes found in figure 8-12 (teapot #1) plus one additional M-node at level 6. The values for dx, dy, D, and θ for the links in teapot 4 are shown in table 8-5. These values are compared to those from teapot #1 in table 8-6.

This comparison shows an average rotation for the P-Paths in teapot #4 of -13.7° with respect to the P-Paths in teapot #1. This is very close to the -15° which the rotation was estimated to be from the photographs. As with the size scaling example in the previous section, all of the lengths match within one sample. The percentage difference in the length of links ranges from -9% to 14%.

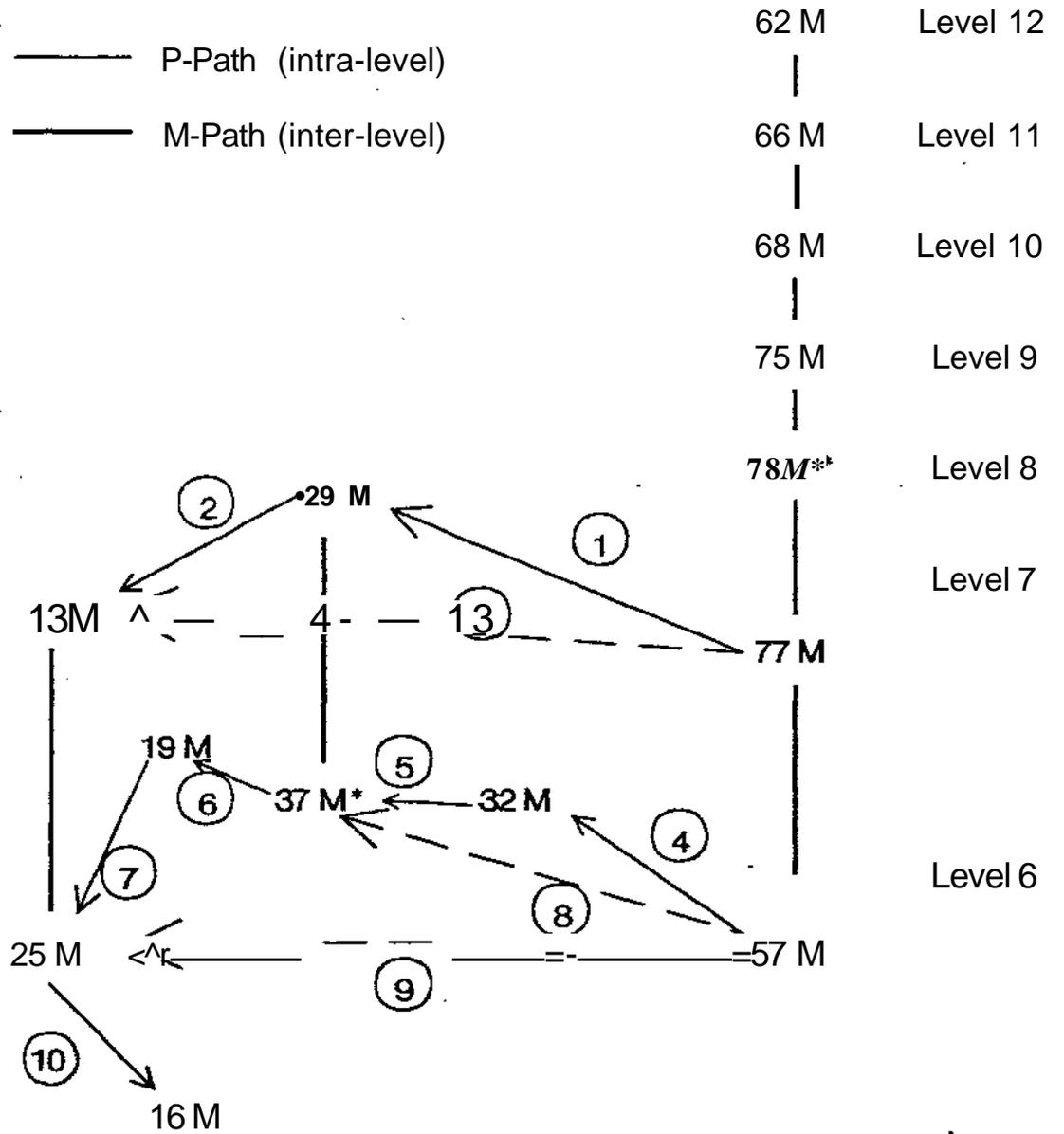


Figure 8-14: M-nodes and P-Paths for Levels 12 to 6 of Teapot #4

P-Path	Level	dx	dy	D_4	θ_4
1	7	-6	-3	6.71	153°
2	7	-5	2	5.38	202°
3 (1&2)	7	-11	-1	11.04	185°
4	6	$-2.5\sqrt{2}$	$-3.0\sqrt{2}$	5.52	130°
5	6	$-3.75\sqrt{2}$	$0.25\sqrt{2}$	5.31	184°
6	6	$-3.25\sqrt{2}$	$-0.75\sqrt{2}$	4.72	167°
7	6	$-0.75\sqrt{2}$	$3.75\sqrt{2}$	5.4	256°
8 (4&5)	6	$-6.25\sqrt{2}$	$-2.75\sqrt{2}$	9.65	153°
9 (4&5&6&7)	6	$-10.25\sqrt{2}$	$0.25\sqrt{2}$	14.50	179°
10	6	$2.5\sqrt{2}$	$2.5\sqrt{2}$	5.0	315°

Table 8-5: P-Path Links for Levels 7 and 6 of Teapot #4

P-Path	Teapot 1		Teapot 4		Difference			
	D_1	θ_1	D_4	θ_4	$\theta_1 - \theta_4$	D_4/D_1	$D_4 - D_1$	$100 \times (D_4 - D_1)/D_4$
1	6.3	161°	6.7	153°	8°	1.06	0.388	5.7%
2	5.8	211°	5.3	202°	9°	0.914	-0.5	-9.4%
3	11.0	185°	11	185°	0°	1.0	0.0	0.0%
4	6.3	153°	5.52	130°	23°	0.876	-0.7	-12.7%
5	5.1	206°	5.3	184°	22°	1.039	0.2	3.7%
6	4.2	180°	4.7	167°	13°	1.119	0.5	10.6%
7	4.6	265°	5.4	256°	9°	1.174	0.8	14.8%
8	10.2	176°	9.6	153°	23°	0.931	-0.7	-7.3%
9	14.6	195°	14.5	179°	16°	0.992	-0.1	-0.72%
Average Error					13.7°	1.012	-0.121	0.52%

Table 8-6: Comparison of D and θ attributes for Teapots #1 and #4

8.4.5 Examples of Size Change Less than $\sqrt{2}$

This subsection shows the result of hand matching the upper levels of teapots #2 and #5. Teapot #2 is the same orientation as teapot #1, but digitized approximately 1.14 larger. Teapot #5 is approximately the same size as teapot #2, but oriented at -15° . Because of the change in scale and lighting, both of these teapot images contain additional M-nodes in their upper levels.

Figure 8-15 shows the M-nodes, M-paths, and P-paths links for levels 12 through 6 of teapot image #2. Level 7 of teapot #2 contains 3 additional M-nodes that did not occur in level 7 of Teapots #1 and #4, or level 8 of teapot #3. These M-nodes are all at the top of M-paths that start at level 6 of teapots #1 and #4 and level 7 of teapot #3. The small scale change between teapot #1 and teapot #2 was enough to bring these M-nodes up to the next level. These P-paths are not labeled in figure 8-15 and their attributes are not included in table 8-7.

Table 8-7 shows the attributes of the P-paths in figure 8-15 which were matched by hand to the P-paths from teapot #1. These values are compared to those of teapot #1 in table 8-8.

This comparison shows that each of the P-Paths links in teapot #2 are slightly larger than the corresponding links in teapot #1, with the average ratio of lengths being 1.19. This is slightly larger than the 1.14 estimated from the photographs, but well within the expected range. The average mismatch of P-path links was 1.57 samples. The percentage change in the lengths of the P-paths ranged from 8% to 27% with an average of 14%.

The M-nodes, M-paths, and P-path links for teapot #5 are shown in figure 8-16 below. Teapot #5 is scaled larger than teapot #1 by approximately 1.14 and rotated in the image plane by approximately -15° . This teapot was supposed to have been a rotation of teapot #2. However, the lighting was changed between the photographing of teapot image #2 and teapot image #5. As a result the shadow on the right side of teapot #5 appears to be slightly larger than that of teapot #2. This slight increase in size is sufficient to cause the M-node in the upper left corner to appear at level 8, and to shift the M* node from level 8 to level 9. It also causes an additional M-node (value 32) to appear along P-path number 5. Despite these changes, the P-paths which were identified in the earlier examples are still detectable in teapot #5. The attributes for the P-paths of teapot #5 are shown in table 8-9. These attributes are compared to those of teapot #1 in table 8-10 and to those of teapot #2 in table 8-11.

The average values for the comparison of the lengths and orientations of the P-paths from teapot #5 to those of teapot #1 are very close to the expected values. As shown in table 8-10, the difference in orientation ranges from 4° to 26° , with an average value of 14.22° , which is very close to the 15° difference of orientation that was measured from the photographs. The ratio of the lengths of P-paths range from 0.93 to 1.45, with an average value of 1.13. This is also very close to the change in size of a factor of 1.14 which was estimated from the photographs.

The results of comparing the lengths and orientations of P-path links from teapot #5 to those of teapot #2, shown in table 8-11, are also reasonably close to the expected values. Teapot #5 is approximately the same size as teapot #2, but rotated by approximately -15° . The ratio of the lengths of the P-paths ranged from 0.77 to 1.34 with an average value of 0.96. The difference in orientation of the P-paths ranged from -13° to 32° with an average value of 10.34° . The match of P-path 6 stands out in this table as having the largest difference in orientation (32°) as well as the smallest ratio of lengths (0.77). P-path 7 seems to correct for this aberration by having a ratio of lengths of 1.34 and an difference of orientation of 9° . The cause of this aberration seems to be that the M-node to which P-path 6 points in teapot image #2 is "out of place" by 1 or 2 samples. Checking back to the comparison of teapot #1 to teapot #2, shown in table 8-8, shows that this same P-path was the largest source of error in both orientation and length in that table also. Our conclusion is that because of a change in shading, this M-node seems to have been shifted in position in the image of teapot #2. This aberration illustrates that when an M-node is slightly shifted in position, the error is averaged out by the lengths and orientations of the P-paths going to the M-node and those coming from it. The conclusion is that the average ratio of lengths and the average orientation of P-paths is a reasonable feature to use in determining the best correspondence of a set of M-nodes from a level of the descriptions of two images.

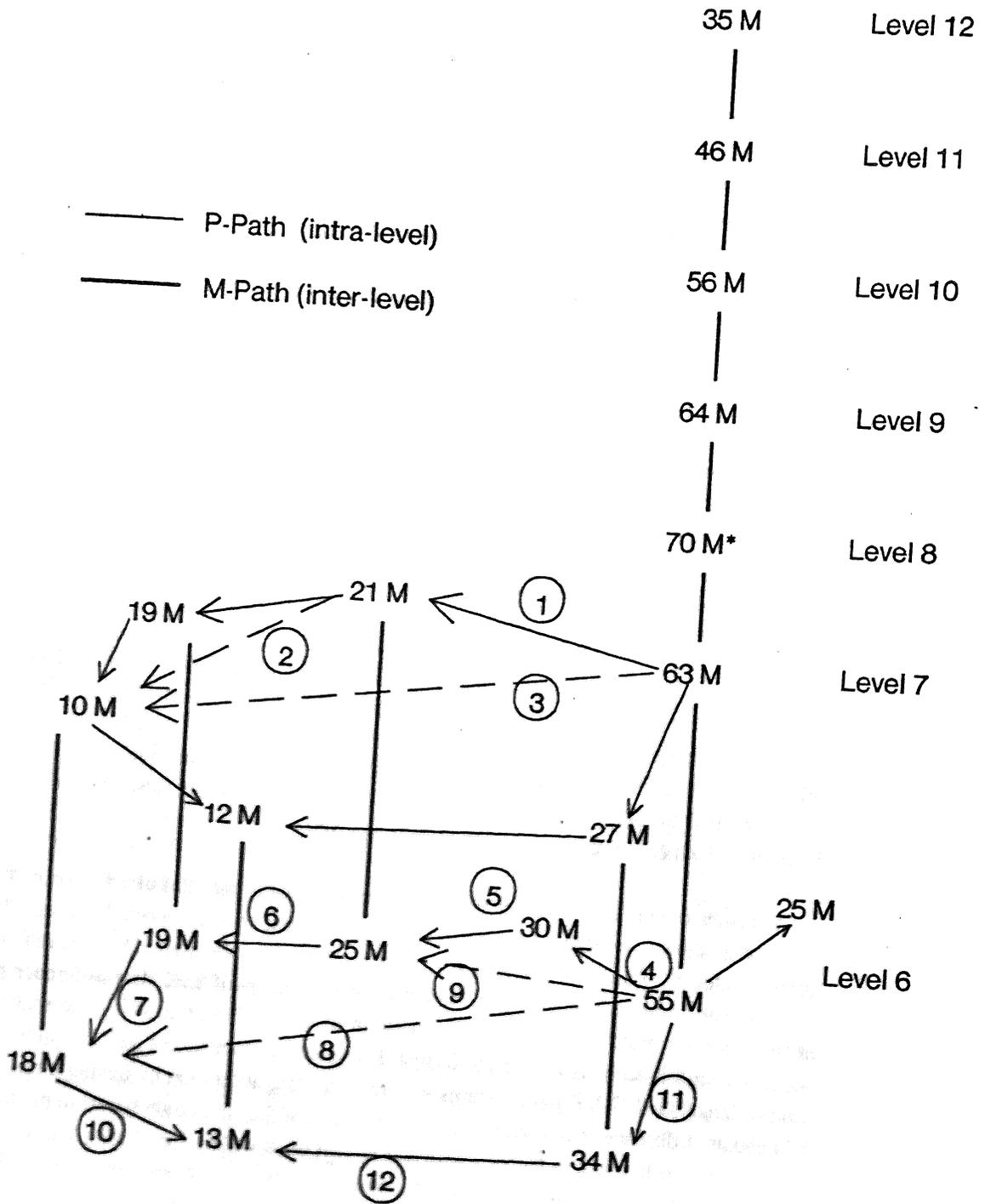


Figure 8-15: M-nodes and P-Paths for Levels 12 to 6 of Teapot #2

P-Path	Level	dx	dy	D	θ
1	7	-7	-2	7.28	164°
2	7	-6	1	6.08	189°
3 (1&2)	7	-14	2	14.14	188°
4	6	$-4.5\sqrt{2}$	$-2.5\sqrt{2}$	7.28	151°
5	6	$-4.0\sqrt{2}$	$1.0\sqrt{2}$	5.83	194°
6	6	$-4.0\sqrt{2}$	$1.0\sqrt{2}$	5.83	194°
7	6	$0.5\sqrt{2}$	$3.5\sqrt{2}$	5.0	262°
8 (4&5)	6	$-8.5\sqrt{2}$	$-1.5\sqrt{2}$	12.2	170°
9 (4&5&6&7)	6	$-13.0\sqrt{2}$	$3.0\sqrt{2}$	18.6	193°

Table 8-7: P-Path Links for Levels 7 and 6 of Teapot #2

P-Path	Teapot 1		Teapot 2		Difference			
	D_1	θ_1	D_2	θ_2	$\theta_1 - \theta_2$	D_2/D_1	$D_2 - D_1$	$100 \times (D_2 - D_1)/D_2$
1	6.3	161°	7.28	164°	-3°	1.16	0.98	13.4%
2	5.8	211°	6.0	189°	22°	1.048	0.2	3.2%
3	11.0	185°	14.14	188°	-3°	1.285	3.0	21.2%
4	6.3	153°	7.28	151°	2°	1.16	0.98	13.4%
5	5.1	206°	5.83	194°	12°	1.143	0.73	1.2%
6	4.2	180°	5.83	194°	-14°	1.388	1.63	27.9%
7	4.6	265°	5.0	261°	4°	1.087	0.4	8%
8	10.2	176°	12.2	170°	6°	1.196	2.0	16.4%
9	14.6	195°	18.8	193°	2°	1.287	4.2	22.2%
Average Error					3.11°	1.19	1.57	14.1%

Table 8-8: Comparison of D and θ attributes for Teapots #1 and #2

8.4.6 Summary of Teapot Matching Examples

The examples shown above illustrate that the graphs of M-nodes connected by P-path links from two images of similar objects can be matched despite changes in the size and orientation of the object between the two images. Before advancing to a simple example of how the representation can be used to find stereo correspondence, let us summarize the examples that have been presented.

This section began with an example of how the graph of M-nodes, connected by P-paths, is formed from a level of the description. This example showed how the M-nodes and P-path links are abstracted from level 7 of teapot image #1.

Next, it was shown how M-nodes from several adjacent levels form M-paths that give an increasingly detailed description of structures in an image. The M-nodes from levels 12 through 6 of teapot image #1 were presented, with the P-path links that connect M-nodes at each level. The table of attributes for each P-path link was also presented.

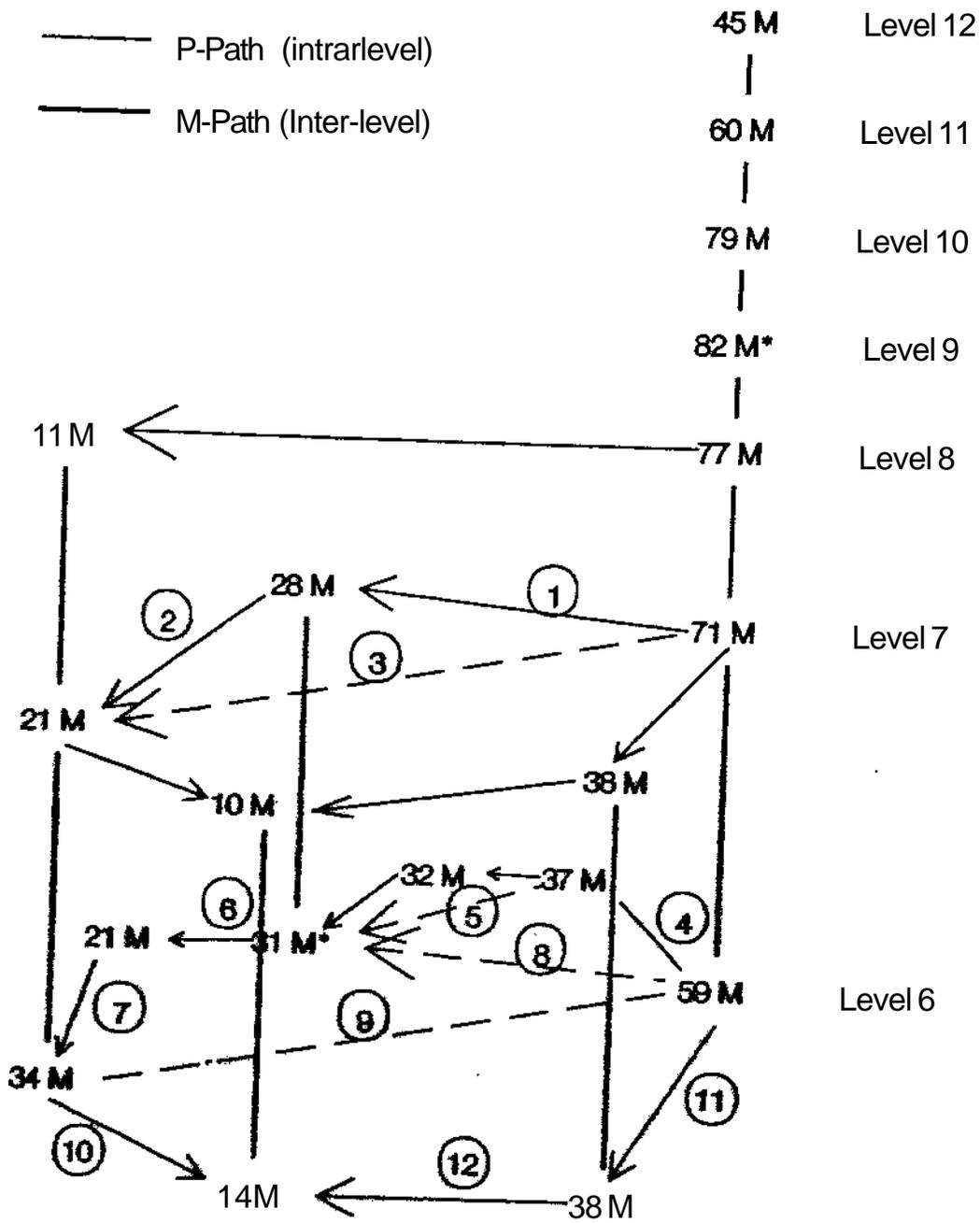


Figure frifc M-nodcs and P-Paihs for Levels 12 to 6 of Teapot #5

P-Path	Level	dx	dy	D	θ
1	7	-7	-3	7.61	157°
2	7	-5	2	5.39	202°
3 (1&2)	7	-12	-1	12.0	175°
4	6	$-3.5\sqrt{2}$	$-3.5\sqrt{2}$	7.0	135°
5	6	$-4.0\sqrt{2}$	0	5.65	180°
6	6	$-3.0\sqrt{2}$	$-1.0\sqrt{2}$	4.47	162°
7	6	$-1.5\sqrt{2}$	$4.5\sqrt{2}$	6.70	252°
8 (4&5)	6	$-7.5\sqrt{2}$	$-2.5\sqrt{2}$	11.18	162°
9 (4&5&6&7)	6	$-12.0\sqrt{2}$	0	16.97	180°
10	6	$3.0\sqrt{2}$	$3.0\sqrt{2}$	6.0	315°
11	6	$2.0\sqrt{2}$	$5.0\sqrt{2}$	7.6	248°
12	6	$-7.0\sqrt{2}$	-1.5	10.12	168°

Table 8-9: P-Path Links for Levels 7 and 6 of Teapot #5

P-Path	Teapot 1		Teapot 5		Difference			
	D_1	θ_1	D_5	θ_5	$\theta_1 - \theta_5$	D_5/D_1	$D_5 - D_1$	$100 \times (D_5 - D_1)/D_5$
1	6.3	161°	7.62	157°	4°	1.21	1.32	17.3%
2	5.8	211°	5.39	202°	9°	0.93	-0.41	-7.6%
3	11.0	185°	12.04	175°	10°	1.09	1.04	8.6
4	6.3	153°	7.0	135°	18°	1.11	0.70	10.0%
5	5.1	206°	5.65	180°	26°	1.10	0.55	9.7%
6	4.2	180°	4.47	162°	18°	1.06	0.27	6.0%
7	4.6	265°	6.70	252°	13°	1.45	2.1	31.3%
8	10.2	176°	11.2	162°	14°	1.09	1.0	8.9%
9	14.6	195°	16.97	180°	15°	1.16	2.37	13.9%
Average Error					14.22°	1.13	0.99	10.9%

Table 8-10: Comparison of D and θ attributes for Teapots #1 and #5

P-Path	Teapot 2		Teapot 5		Difference			
	D_2	θ_2	D_5	θ_5	$\theta_2 - \theta_5$	D_5/D_2	$D_5 - D_2$	$100 \times (D_5 - D_2)/D_5$
1	7.28	164°	7.62	157°	7°	1.05	0.34	4.5%
2	6.0	189°	5.39	202°	-13°	0.90	-0.61	-11.3%
3	14.14	188°	12.04	175°	13°	0.85	-2.10	-17.4%
4	7.28	151°	7.0	135°	16°	0.96	-0.28	-4.0%
5	5.83	194°	5.65	180°	14°	0.97	-0.18	-3.0%
6	5.83	194°	4.47	162°	32°	0.77	-1.36	-30.4%
7	5.0	261°	6.70	252°	9°	1.34	1.7	25.4%
8	12.2	170°	11.2	162°	8°	0.92	-1.0	-8.9%
9	18.8	193°	16.97	180°	13°	0.90	-1.83	-10.8%
Average Error					10.34°	0.96	-0.591	-6.2%

Table 8-11: Comparison of D and θ attributes for Teapots 2 and 5

The use of the principal M-path and highest level M*-node was then shown for aligning two

descriptions to get an initial estimate of the difference in size and position. * In this subsection a comparison was made of the M-node graphs from teapot #1 to the M-node graphs of teapot #3. It was shown that the correspondence could be found despite a change in size of approximately 136 by shifting the M-node graph from the larger image down by one level. It was also shown that this shift was dictated by the difference in the level at which the highest M*-node occurred in the two descriptions.

An example was then given of the correspondence that occurs when the object has been rotated. The P-path links from teapot #1 were compared to those of teapot #4, which is of the same size, but rotated by $\sim -14^\circ$. Further examples were then presented which showed how the matching is affected by changes of size which are less than a factor of $\sqrt{2}$.

The next section illustrates how this representation can be used to determine the correspondence from a stereo pair of images.

8.4.7 Stereo Matching/Example

A stereo pair of images was formed of a paper-wad to test the use of the representation for determining the correspondence between structural components in a stereo pair of images. The original images are shown with the output from the low pass filters in figures 8-19 and 8-21. The format of the low-pass images is shown in figure 8-18. Unlike the band-pass images, it is the odd numbered low-pass images which are defined on a $\sqrt{2}$ sample grid. In forming these low-pass images, the undefined pixels were left with a value of zero. Thus the odd numbered low-pass levels appear with much less intensity than the even numbered low-pass images. In each of the low-pass figures the original image appears in the lower right corner.

The resulting band-pass images are shown in figures 8-20 and 8-22. The format for these band-pass images is the same as shown in figure 8-6 in section 8Z.

The scene was formed by placing the paper wad on a dark lab bench under a desk lamp. A vidicon camera, mounted on a tripod, was placed approximately 14 inches from the paper wad, and the left image was digitized using the Grinnell digitizer. The camera was then moved to the right approximately 8 inches and tilted so that the paper wad was located in roughly the same part of the image. The right image was then digitized.

The purpose of this experiment was to test the use of the representation for determining the correspondence of parts of the two images. No attempt was planned or made to use this correspondence to determine the actual distances to surface points on the paper wad.

The M-nodes for Levels 13 through 9 of the two paper wads are shown in figure 8-17 below. The correspondence between M-nodes was assigned by hand. This correspondence is illustrated by the dashed arrows in figure 8-17. Each correspondence is labeled with the displacement, dx, dy, between the actual positions of the M-nodes in the two images. Assigning these correspondences was a trivial task because of the small number of M-nodes at each level. Even when the number of M-nodes

increased at the levels below level 9, the correspondences at the previous level constrain the possible correspondences so that there is often no choice as to which M-nodes correspond.

Note that at level 10, two M-nodes occur in the right image, while only a single M-node occurs in the left image. This difference in structure is the result of the parallax created by the difference in perspective. This illustrates one of the problems in determining stereo correspondence: shape changes when seen from different perspectives. Thus a stereo correspondence algorithm must be capable of assigning a sample from one image to more than one sample in the second.

The conclusion from this experiment is that the representation can provide an efficient technique for determining the correspondence of structural components in a stereo pair of images.

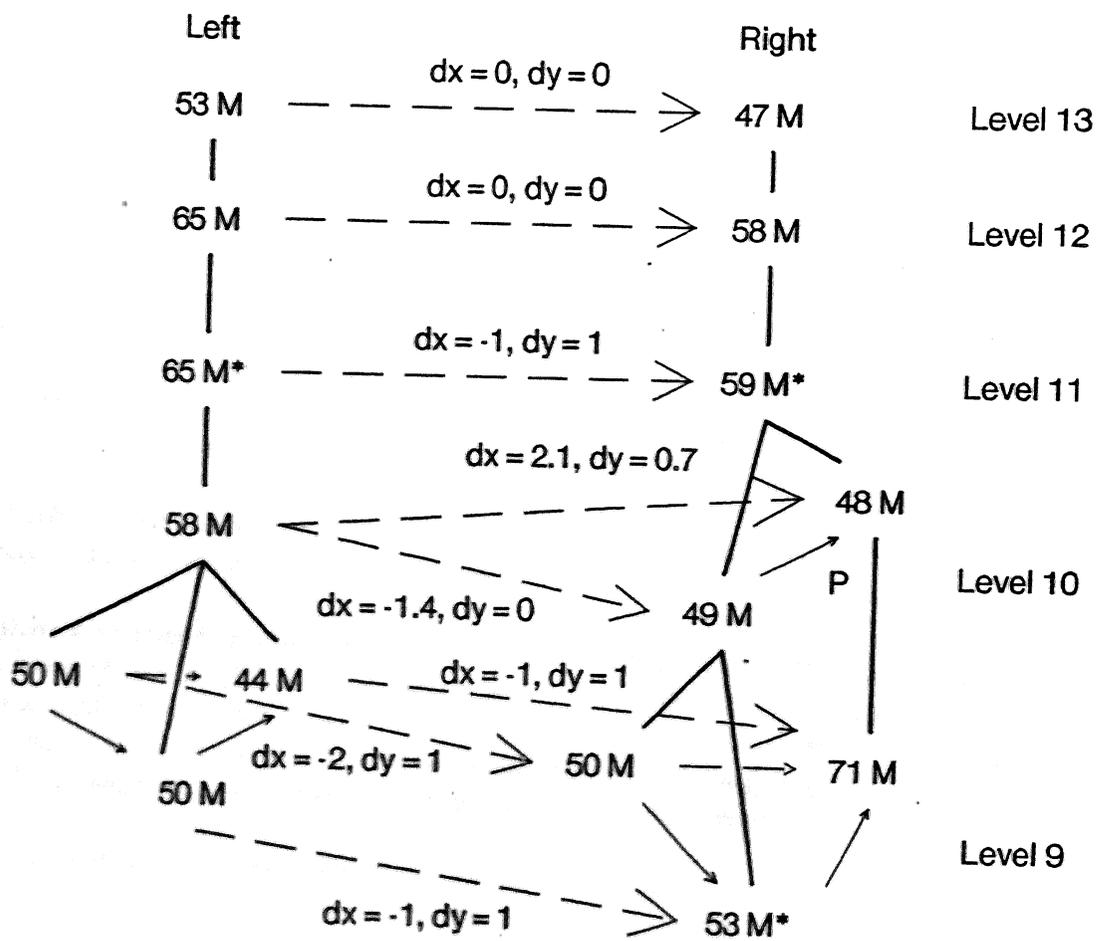


Figure 8-17: Stereo Correspondence of M-nodes for Paper Wads, Levels 13 through 9

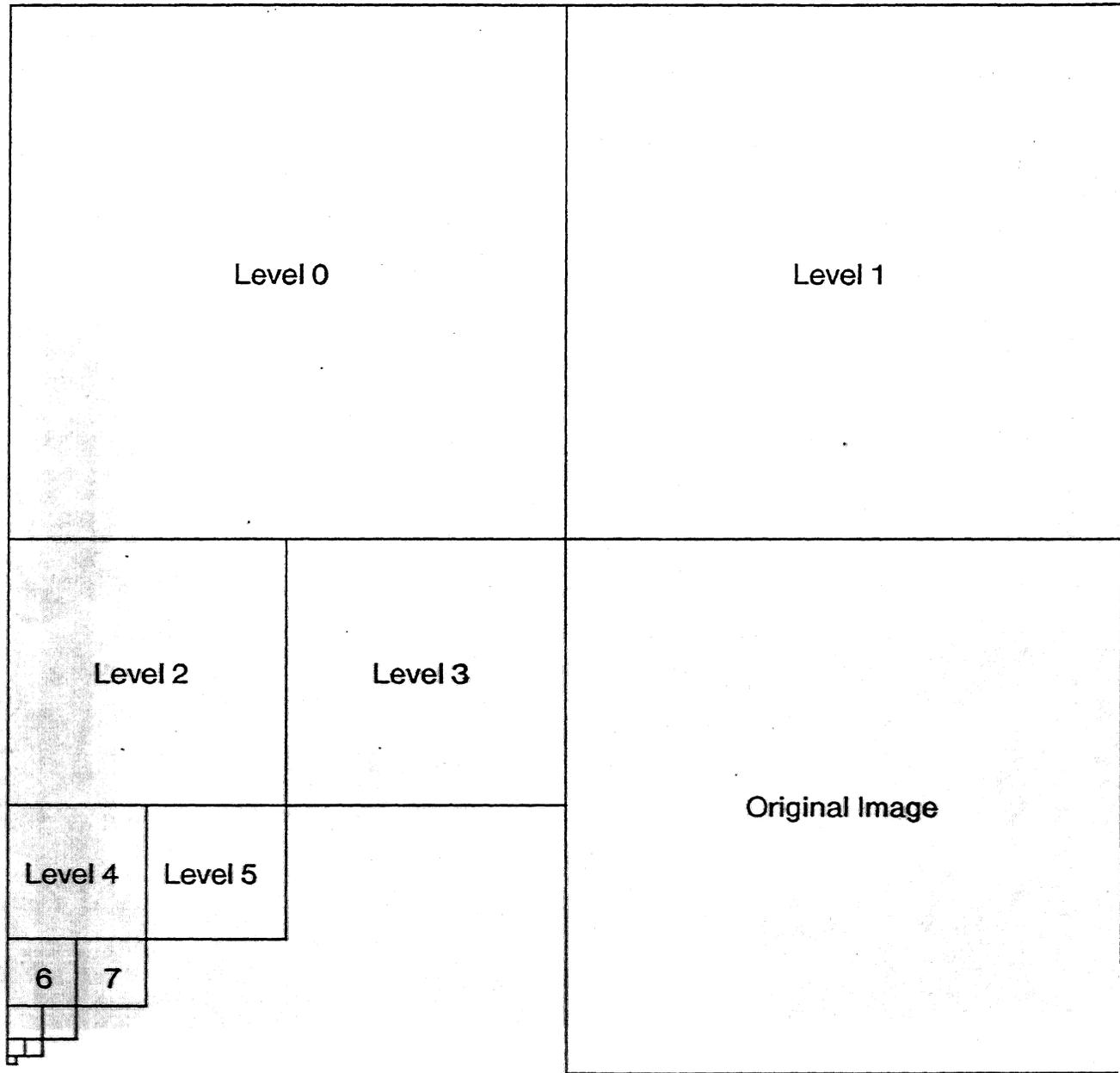


Figure 8-18: Format for Paper Wad Low-Pass Images

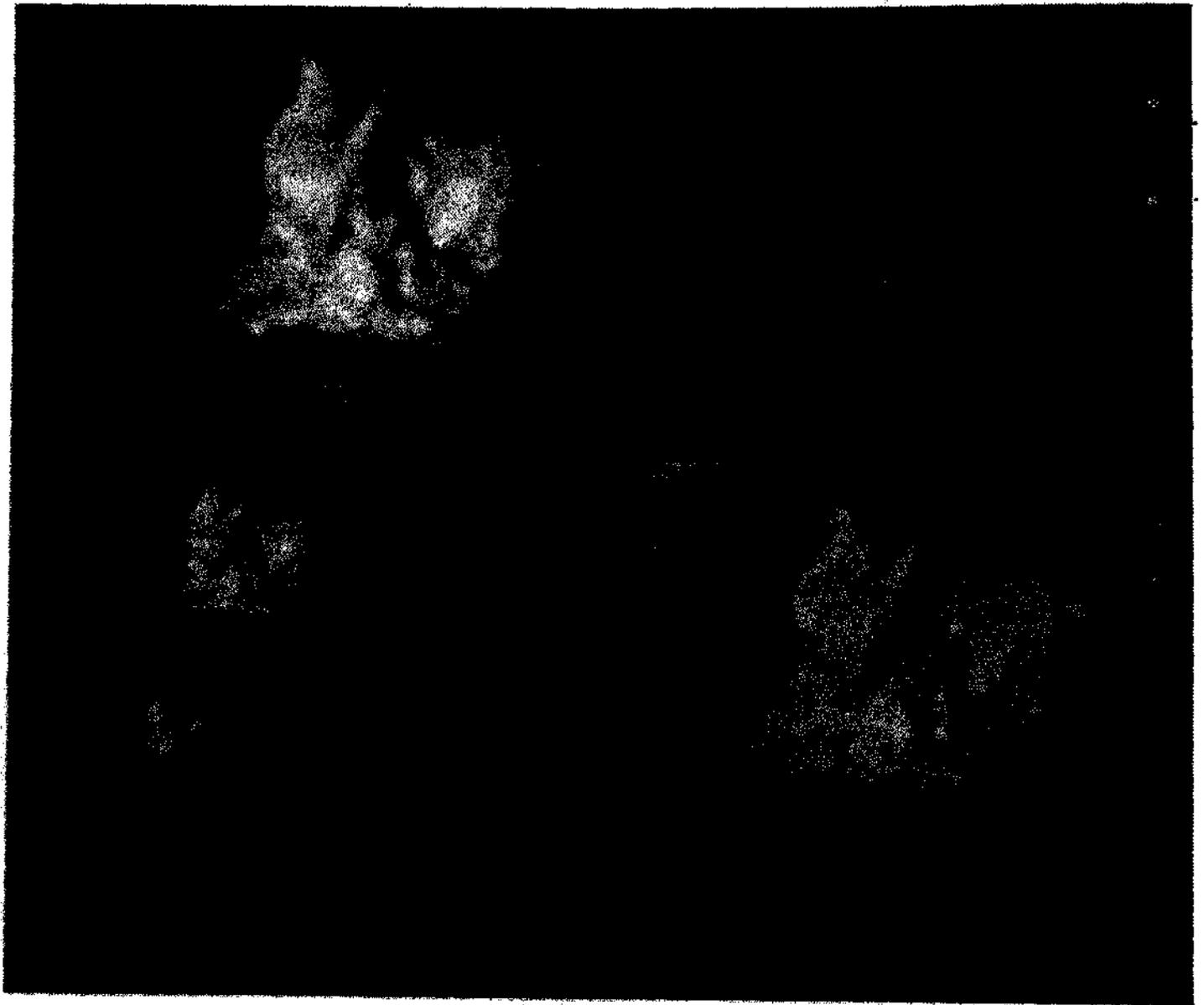


Figure 8-19! Left Paper Wad and Lev:-Pass Images



Figure 8-21: Right Paper Wad and Low-Pass linages

8.5 Matching L-Paths

When a gray scale form has components which are long and thin, ridges, or P-paths occur along this component in several adjacent levels in the Sampled DOLP (or SDOG) transform. This information is encoded by finding the level where the response of the DOLP filter is strongest along the path followed by the ridges. These strongest P-nodes are labeled as L-nodes by a process described in the previous chapter and connected together to form an L-path. In some situations, particularly in structural pattern recognition, identifying or discriminating objects requires being able to measure the similarity of L-paths from two representations. This section is concerned with this problem.

8.5.1 Two stages of Matching

As with any curve matching problem, there are two stages to matching L-paths:

1. An alignment stage: In this stage the L-path from the reference representation is positioned, oriented, and scaled so that will be in its closest correspondence with the measured L-path.
2. A Similarity Measure: In this stage, some measure of the "goodness of fit" is calculated between the two L-paths.

8.5.2 L-Path Alignment

The previous section concerned the problem of determining the correspondence between the representations of two gray-scale forms, which are at different positions, scales, and/or orientations. These techniques employed M-nodes and M*-nodes as landmarks which are brought into correspondence. In most cases, L-paths are terminated at each end by an M*-node. Two L-paths are aligned by aligning their terminating M*-nodes. This section shows how the correspondence of the terminating M*-nodes is used to scale, shift and rotate the reference L-path so that it is in correspondence with the measured L-path.

8.5.2.1 L-Path Notation and Attributes

Let us define the values along an L-path as a sequence: L_i . Each L-node has attributes of filter value and location as well as a set of pointers to adjacent L-nodes or M-nodes on the L-path. The location of the i^{th} L-node in the L-path before applying these linear transformations is (x_i, y_i, k_i) . This location is in terms of pixels from the original image.

One of the two M*-nodes must be selected as a "distinguished" for the orientation attribute, for indexing and for computing the linear transforms. If one M*-node is at a higher level than the other, this is chosen as the distinguished M*-node. Otherwise, the choice is arbitrary.

The entire L-path also has a set of attributes which are similar to those described for P-paths in the previous section. The attributes of an L-path are determined by the relative positions in the SDOG space¹⁸ of the terminating M*-nodes. The L-path attributes are:

- AL: The difference in levels between two terminating M*-nodes. This is computed as the level of the distinguished M*-node minus the level of the other M*-node.
- D_L: The cartesian distance between the M*-nodes measured in pixels from the original image.
- #_L: The orientation of the vector from the distinguished M*-node to the other M*-node.

8-5.2.2 Alignment Parameters:

Matching occurs by aligning a reference representation to a measured representation. Finding the correspondence between the terminating M*-nodes of the reference L-path and the M*-nodes of the measured L-path gives the parameters for position, scale, and orientation for aligning the reference L-path to the measured data. These parameters are used by a set of linear transforms that are applied to the reference L-path to bring it to correspondence with the measured L-path. These transforms and their parameters are as follows:

- Ak: the change in level that must be applied to one L-path so that it may match a second L-path. Each increment of 1 in Ak scales the L-path by a factor of \sqrt{V} in size.
- Ad: A small scale change determined by the correspondence of the terminating M*-nodes after they have been shifted to the same levels. $Ad = D_m/D_r$ where D_m is the length attribute of the measured L-path and D_r is the length attribute of the reference L-path after it has been scaled to account for shifting by Ak levels. This small scaling accounts for minor deviations in the total length of the L-path. This scale change is applied to the distance between each L-node and the M*-node which is used as a starting point for the matching.
- A θ : The rotation of the L-path. The L-paths are originally encoded on cartesian and \sqrt{V} sample grids. A θ rotates one of the L-paths so that its L-nodes occur at real valued (or high resolution integer valued) points. The result is a requirement for a rule which relates the value at such a real-valued point to the values at nearby discrete sample points. A nearest-neighbor rule is described below for this.
- (x_r, y_r): This is the location of the distinguished M*-node.

¹⁸The SDOG space is the set of points defined by the set of band-pass images (x*, y, I).

8.5.4 Examples of L-path Alignment and Matching

This subsection gives examples of the use of the alignment function and the similarity measure. The L-path that describes the shadow on the right side of each teapot is used in these examples. This shadow does not have a well defined shape.¹⁹ At the upper right corner of the teapot, the shadow merges with the darkly glazed upper half of the teapot. In the lower half of the teapot, the left edge of the shadow is very hard to discern. As is often the case in a cylindrical shaped object, the intensity falls gradually as the surface orientation moves away from the light source. Visually determining the edge of the shadow is further complicated by the surface texture of the teapot. Thus this shadow is a good example of the description by an L-path of a form without distinct boundaries.

Figure 8-23 shows this L-path for teapot #1.²⁰ In this figure, each node is represented by two lines of letters and numbers. The top line consists of the SDOG transform value, the node type (M*, M, or L), and the level (in angle brackets). For example, 75 M* <8>, refers to an M*-node of value 75 at level 8. The second line gives the relative position of the node with respect to the distinguished M*-node in pixels from the original image. These numbers are (Δx , Δy). In the distinguished node, the second line gives the actual position of the node. Also shown are the attributes of the entire L-path:

- ΔL : (written as dL) the change in levels between the M*-nodes;
- D: the length of the L-path in pixels; and,
- θ : (written as Angle) the orientation of the vector from the distinguished M*-node to the other M*-node.

Each L-node has a circled number beside it. These numbers serves as an identifier in the tables that illustrated L-node correspondence and distance.

Figure 8-24 shows the L-path which describes the same shadow in teapot #3. The correspondence between L-nodes after the L-path from teapot #3 has been rotated and scaled to match the L-path from teapot #1, is shown in figure 8-25 and table 8-12. The correspondence in figure 8-25 is shown with dashed arrows. Table 8-12 lists the locations to which the L-nodes from teapot #3 were transformed and the closest L-node from teapot #1. The column labeled distance is the cartesian distance between the transformed reference node and the nearest measured node expressed in pixels (samples in the original image). The column labeled "error" shows the result of dividing this distance by the sample rate at the level of to which the reference node was transformed. At the bottom of the table is the average error and the largest error.

¹⁹See figures 8-1 through 8-5

²⁰Note: the sign of the "y" term is reversed in all of the figures and tables in this section. This has the effect of making angles increase positively in the counter-clockwise direction. Thus γ and θ are consistent with the right-handed coordinate system usually used by humans instead of the left-handed coordinate system usually used in image processing. This also keeps the angles used in this section consistent with those given in the examples in section 8.4.

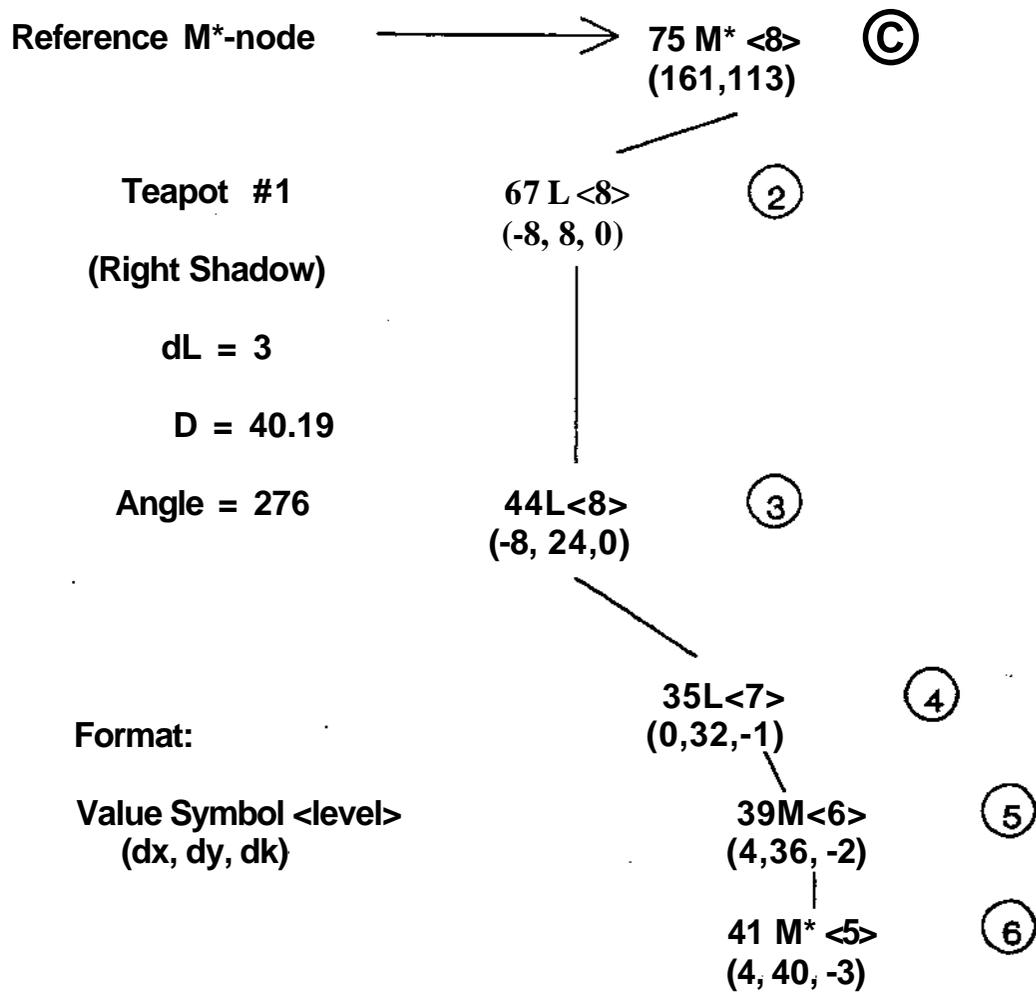


Figure 8-23: L-path from Teapot #1

The top line of table 8-12 shows the change in attributes between the two L-paths. AL is the difference in levels between the distinguished M*-nodes. l/r is the ratio of the lengths of the measured (m) to the reference (r) L-paths. This ratio is computed with length measured in pixels before the reference L-path is shifted by Ak levels. Thus this ratio is the product of the match parameters Ad and $2^{Ak/2}$ that were described above. A0 is the difference in angles. The program that matched these two L-paths transformed the reference L-path by dividing each distance by the ratio of the lengths and rotating by the difference in angles. Table 8-13 shows the results of transforming the L-path from Teapot #1 to match that of Teapot #3. In both table 8-12 and table 8-13 a one-to-one correspondence was found between L-nodes and the error is always less than one sample.

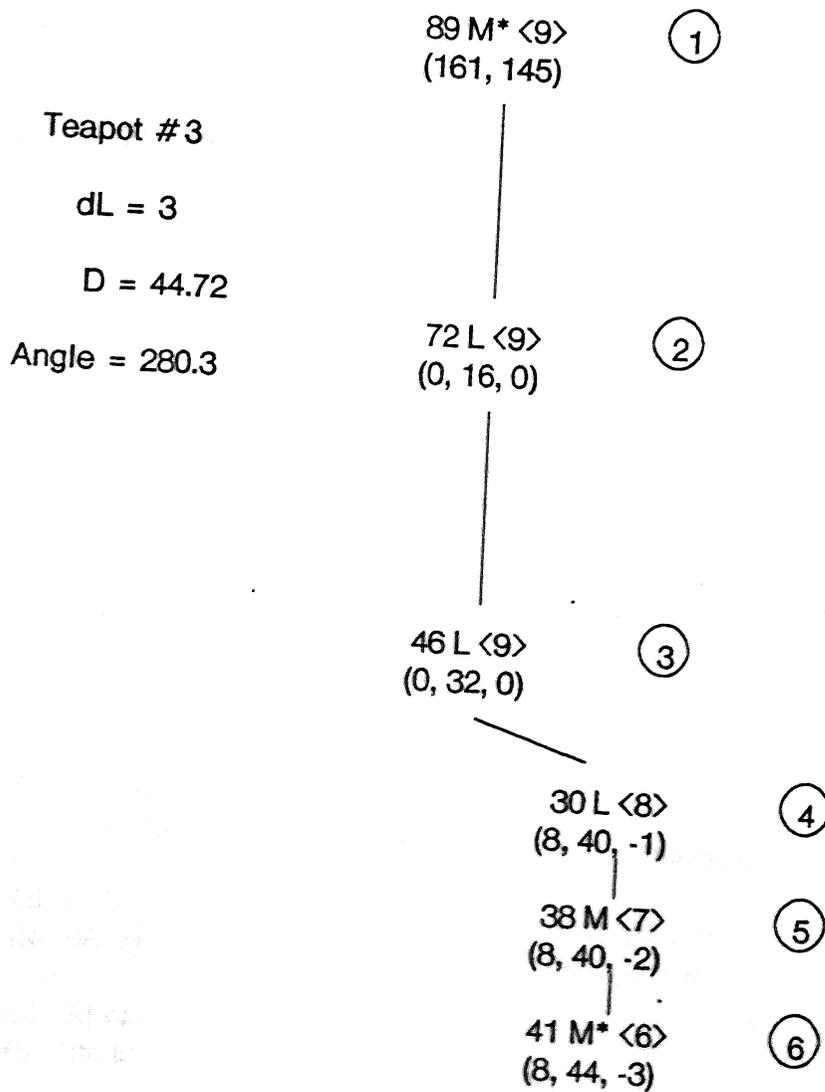


Figure 8-24: L-path from Teapot #3

Transforming the L-path from teapot #3 to be in correspondence with the smaller L-path from teapot #1 gave a worst case error is 0.824 samples and the average error is 0.32. Matching the L-path from the larger teapot #1 to the larger teapot #3 gave a worst case error of 0.648 samples and an average error of 0.30 samples. Thus, despite a scale change of ≈ 1.36 between the two images, aligning the terminating M*-nodes brought the L-path from the each image into a reasonably close correspondence with the L-path from the other image.

Figure 8-26 shows the L-path from the shadow in teapot image #4. The correspondence of transformed L-nodes from teapot #4 to the L-nodes of teapot #1 is shown in figure 8-27. The

Transform of Teapot # 3 to Match Teapot # 1

$$\Delta L = -1, D_m/D_r = 0.89, \Delta\theta = -9.18^\circ$$

Nodes from teapot # 3				Nodes from teapot # 1					
Transform of Reference Node				Closest Measured Node					
Node	$\Delta\tilde{x}$	$\Delta\tilde{y}$	Δk	Node	Δx	Δy	Δk	distance	error
1	0.00	0.00	0	1	0.00	0.00	0	0.000	0.000
2	-1.15	14.33	0	2	-8.00	8.00	0	9.329	0.824
3	-2.30	28.67	0	3	-8.00	24.00	0	7.366	0.651
4	4.28	36.41	-1	4	0.00	31.99	-1	6.155	0.769
5	4.28	36.41	-2	5	4.00	35.99	-2	0.505	0.089
6	4.00	40.00	-3	6	4.00	40.00	-3	0.000	0.000

Average Error = 0.38
Worst Error = 0.82

Table 8-12: Correspondence and Distance for Transform of L-path from Teapot # 3 to Match L-path from Teapot # 1

Transform of Teapot # 1 to Match Teapot # 3

$$\Delta L = 1, D_m/D_r = 1.11, \Delta\theta = 9.18^\circ$$

Nodes from teapot # 1				Nodes from teapot # 3					
Transform of Reference Node				Closest Measured Node					
Node	$\Delta\tilde{x}$	$\Delta\tilde{y}$	Δk	Node	Δx	Δy	Δk	distance	error
1	0.00	0.00	0	1	0.00	0.00	0	0.000	0.000
2	-8.15	9.58	0	2	0.00	15.99	0	10.378	0.648
3	-6.73	27.32	0	3	0.00	31.99	0	8.195	0.512
4	2.85	35.48	-1	4	8.00	40.00	-1	6.847	0.605
5	7.64	39.56	-2	5	8.00	40.00	-2	0.562	0.070
6	7.99	44.00	-3	6	8.00	43.99	-3	0.000	0.000

Average Error = 0.30
Worst Error = 0.64

Table 8-13: Correspondence and Distances for Transform of L-path from Teapot # 1 to Match Teapot # 3

Table 8-15 shows the result of transforming and matching the L-nodes from the L-path in teapot # 1 to the L-path from teapot # 4. The correspondence between L-nodes in this table is different than those for the match from teapot # 4 to teapot # 1. In this case the worst case error was 0.901, which is less than a sample. The average error, 0.48 is also smaller in this case. Node 2 from teapot # 1, which gave the largest worst case distance in table 8-14 was not found to be a closest neighbor to any of the L-nodes from teapot 4. Node 3 from teapot # 4, which appeared to be spurious, actually fell within 0.552 samples of a L-node 3 from teapot # 1.

The L-path for the right shadow in teapot # 2 is shown in figure 8-28. The result of matching this L-path to that of teapot # 1 is shown in table 8-16. Despite the change in scale of 1.14 between these two images these two L-paths have exactly the same lengths and orientations. Differences in position

Teapot # 4
 dL = 3
 D = 36.2
 Angle = 264

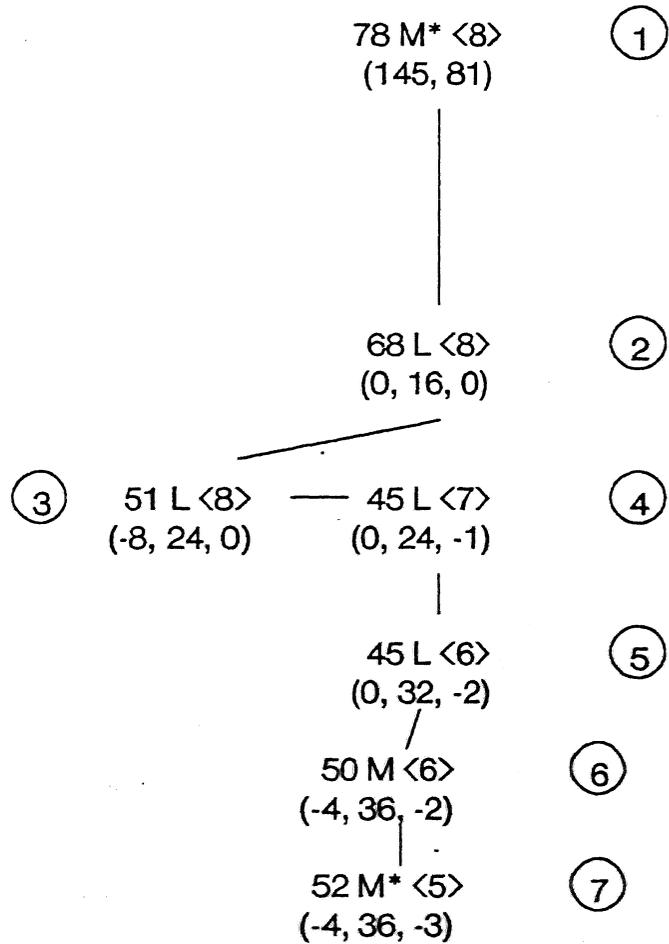


Figure 8-26: L-path from Teapot #4

relative to the sample, however, cause L-nodes 4 and 5 in these L-paths to each be off by 1 sample at their levels.

Figure 8-29 shows the L-path from teapot #5. This image is scaled by a factor of 1.14 and rotated by $\sim 15^\circ$ from teapot #1. The M*-nodes in the L-paths occur such that there is an angle of 37.4° between them. The reader may recall that teapot #5 had an M*-node that occurred at level 9, when it was expected to occur at level 8. As a result, this L-path spans 4 levels. This L-path also has two L-nodes that are -2 levels below the root M*-node. The results which this had on finding the correspondence is shown in table 8-17.

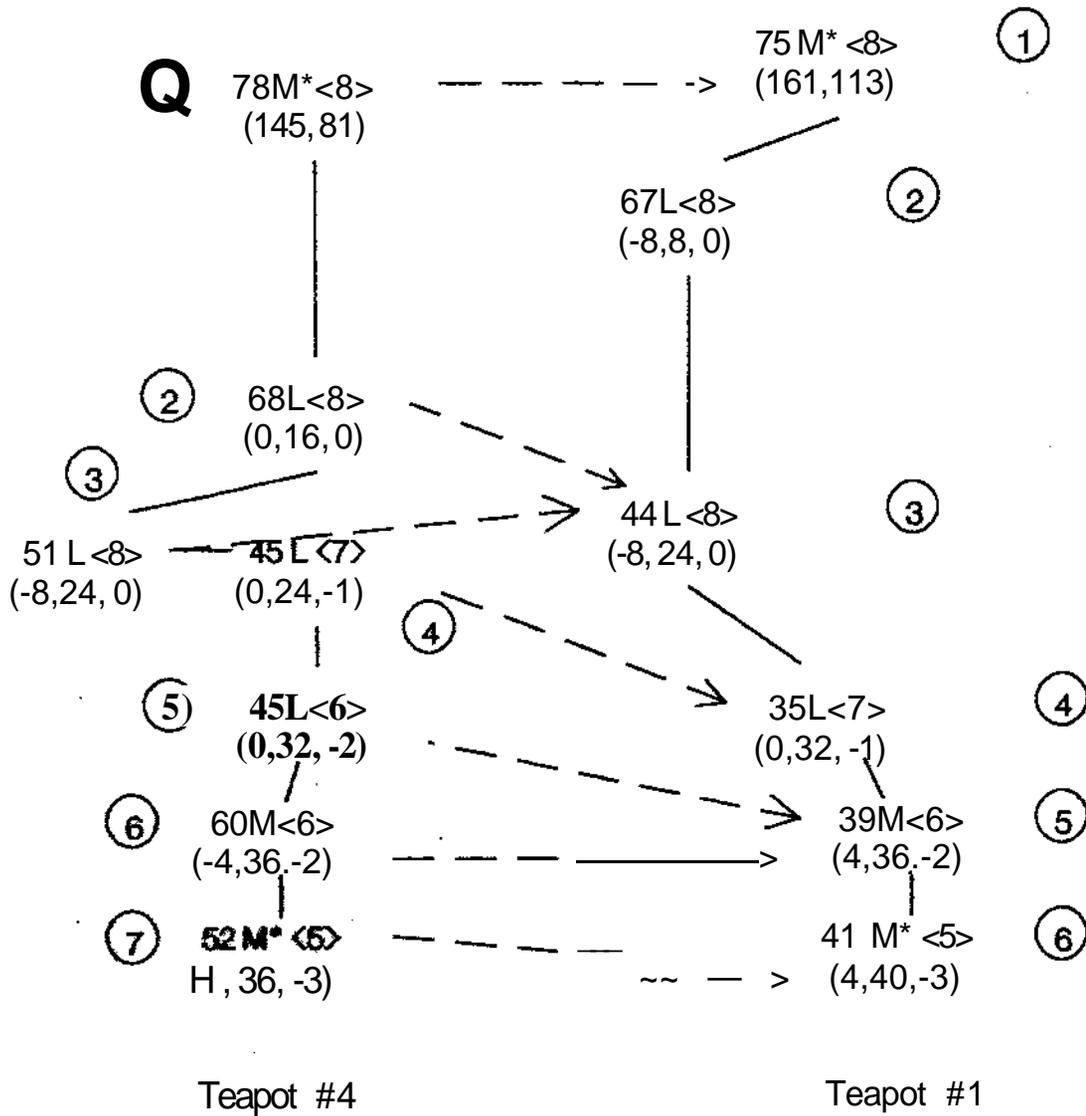


Figure 8*27: I/path Correspondence:
 L-Path from Teapot #4 Transformed to Match I-path from Teapot #1

As can be seen from table 8-17, the alignment of the highest level M*-node from teapot #1 with that of teapot #4 caused a search of the L-nodes from teapot #4 to find their nearest neighbor at a lower level. Such a search matches mid a weight of 1 sample to the error distance. Both K*nodes 2 and 3 from teapot #4 found node 3 of teapot #1 to be the nearest neighbor after alignment. Node 3 from teapot #4 had to find this match, with an error of 1.090 samples. Node 4 from teapot #1 found node 5 of teapot #4 in the upper level, giving an error of 1.269 samples. Partly as a result of all the above level matches the average error was 0.85 samples and the worst case error was 1.7 samples.

Transform of Teapot # 4 to Match Teapot # 1

$$AL = 0, D_m/D_f = 1.10, A0 = 24.10^\circ$$

Nodes from teapot #4				Nodes from teapot #1					
Transform of Reference Node				Closest Measured Node					
Node	$\bar{A}\bar{X}$	$\bar{A}\bar{Y}$	$\bar{A}k$	Node	Ax	AY	Ak	distance	error
1	0.00	0.00	0	1	0.00	0.00	0	0.000	0.000
2	3.70	17.36	0	3	-8.00	24.00	0	13.456	1.189
3	-3.12	27.90	0	3	-8.00	24.00	0	6.246	0.552
4	5.56	26.04	-1	4	0.00	31.99	-1	8.145	1.018
5	7.41	34.73	-2	5	4.00	35.99	-2	3.642	0.643
6	4.00	39.99	-2	5	4.00	35.99	-2	3.999	0.707
7	4.00	39.99	-3	6	4.00	40.00	-3	0.000	0.000

Average Error = 0.58
Worst Error = 1.18

Table 8-14: Correspondence of Transformed L-nodes from Teapot #4 to L-nodes from Teapot #1

Transform of Teapot # 1 to Match Teapot # 4

$$AL = 0, D_m/D_f = 0.90, A0 = -24.10^\circ$$

Nodes from teapot #1				Nodes from teapot #4					
Transform of Reference Node				Closest Measured Node					
Node	$\bar{A}\bar{X}$	$\bar{A}\bar{Y}$	$\bar{A}k$	Node	Ax	AY	Ak	distance	error
1	0.00	0.00	0	1	0.00	0.00	0	0.000	0.000
2	-8.55	5.54	0	1	0.00	0.00	0	10.194	0.901
3	-11.56	19.64	0	3	-8.00	24.00	0	5.628	0.497
4	-6.01	28.19	-1	4	0.00	24.00	-1	7.339	0.917
5	-3.24	32.47	-2	5	0.00	31.99	-2	3.282	0.580
6	-4.00	36.00	-3	7	-4.00	35.99	-3	0.000	0.000

Average Error = 0.48
Worst Error = 0.91

Table 8-15: Correspondence of Transformed L-nodes from Teapot #1 to L-nodes from Teapot #4

8.5.5 Summary of L-path Matching Examples

The first example presented above was the match of the L-paths between teapot # 1 to teapot #3. This illustrated matching between images when the object has been scaled by close to VT in size. In this example, there was a one-to-one correspondence between the L-nodes from the two images^ for both the case when the L-path from teapot #1 was scaled and rotated and the nearest neighbor was sought from teapot #3 and when the L-path from teapot #3 was scaled and the nearest neighbor from teapot #1 was sought. In both cases all of the correspondences were found within one sample.

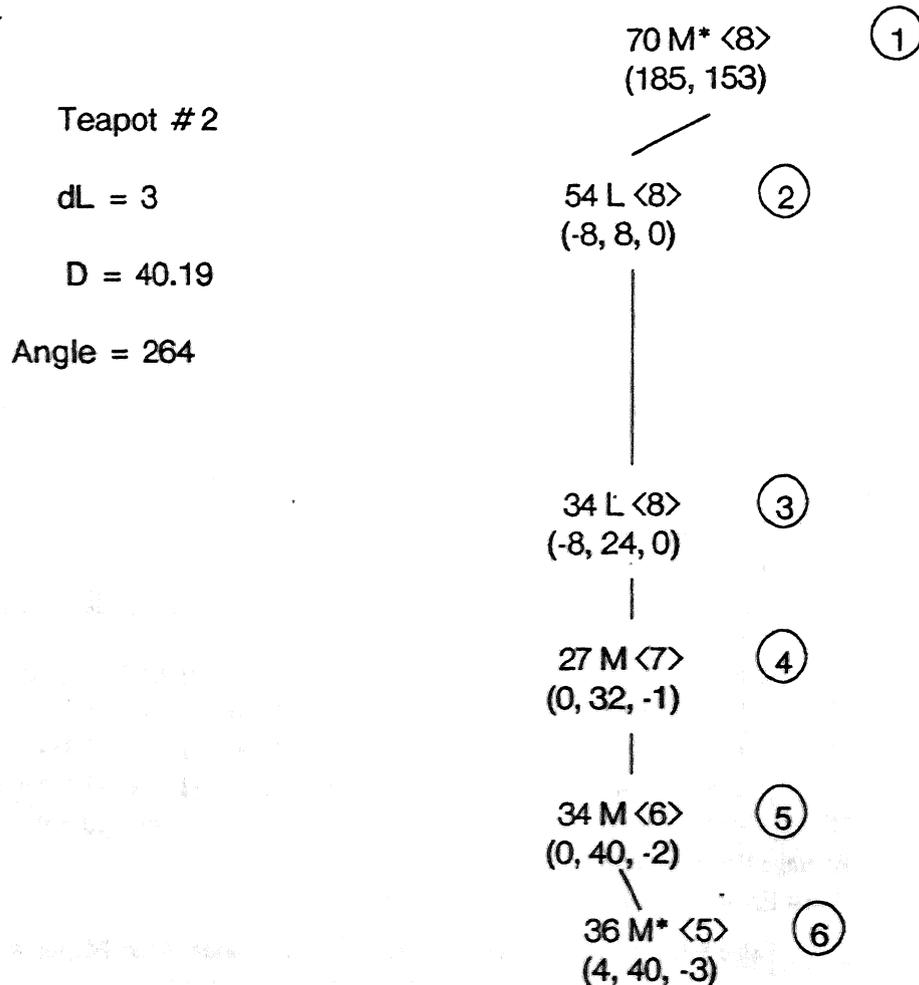


Figure 8-28: L-path from Teapot #2

In the third and fourth examples, the L-path from teapot #1 was matched to that of teapot #4. Teapot #4 is of the same scale as teapot #1, but rotated by approximately -15° . The difference in position of the terminating M*-nodes led to a difference of angle between the two L-paths of approximately 24° . Also, the L-path from teapot #4 was 0.90 the length of the one from teapot #1. This difference in length and orientation led to a difference in the number of L-nodes in the two L-paths. There was not a one-to-one correspondence in the matches of the two L-paths. When the L-path from teapot #4 was scaled and rotated to match the one from teapot #1, two of the L-nodes found their nearest match more than one sample away, with the worst being 1.189 samples away. The average distance was 0.58 samples. When the L-nodes from teapot #1 were compared to those of teapot #4, the worst case matches was 0.91 samples and the average error was 0.48 samples.

Transform of Teapot # 2 to Match Teapot # 1

$$AL = 0, D_m/D_r = 1.00, A\theta = 0.00^\circ$$

Nodes from teapot #2				Nodes from teapot #1					
Transform of Reference Node				Closest Measured Node					
Node	$\bar{A}\bar{X}$	$\bar{A}\bar{Y}$	A_k	Node	A_x	A_y	A_k	distance	error
1	0.00	0.00	0	1	0.00	0.00	0	0.000	0.000
2	-8.00	8.00	0	2	-8.00	8.00	0	0.000	0.000
3	-8.00	24.00	0	3	-8.00	24.00	0	0.000	0.000
4	-8.00	31.99	-1	4	0.00	31.99	-1	8.000	1.000
5	0.00	40.00	-2	5	4.00	35.99	-2	5.656	1.000
6	4.00	40.00	-3	6	4.00	40.00	-3	0.000	0.000
Average Error =				0.33					
Worst Error =				1.00					

Table 8-16: Correspondence of L-nodes and Distances for Transform of L-path from Teapot #2 to Match Teapot #1

In the next matching example the L-path from teapot #2 was matched to that of teapot #1. Teapot #2 is 1.15 larger than teapot #1. The two L-paths had exactly the same length and orientation. All of the L-nodes except two found their nearest neighbor at a distance of 0.0 samples. These two L-nodes found their nearest neighbor 1.0 samples away.

In the final example, the L-path from teapot #5 was compared to that of teapot #1. Teapot #5 is rotated by -15° and scaled by 1.15 from teapot #1. The principal M*-node in teapot #5 was one level higher than expected, and this had a big effect on the matching of these two L-paths. Many of the nearest neighbors in this example were found across level.

Our conclusion from these experiments is that the L-path matching procedure and similarity measure described above gives a reasonable estimate of the similarity of L-paths from two images. The worst mismatch between individual L-nodes in all of these examples was 1.37 samples while the worst average error distance was 0.85. This matching procedure gives the ability to compare L-paths from any orientation and length, and spanning any number of levels. The simple similarity measures of worst distance and average distance provide a useful measure of the similarity of L-paths from two images.

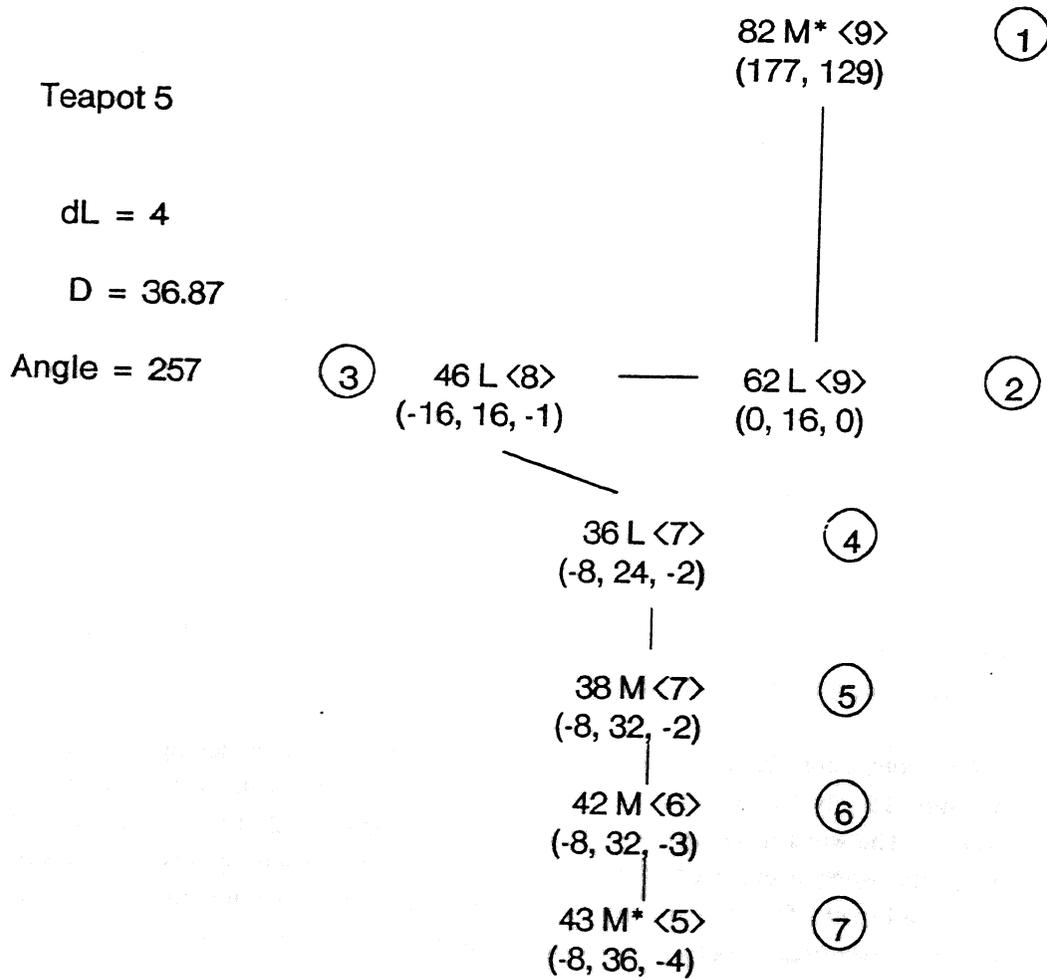


Figure 8-29: L-path from Teapot #5

Transform of Teapot # 5 to Match Teapot # 1

$$\Delta L = -1, D_m/D_r = 1.09, \Delta\theta = 37.42^\circ$$

Nodes from teapot #5			Nodes from teapot #1						
Transform of Reference Node			Closest Measured Node						
Node	$\Delta\tilde{x}$	$\Delta\tilde{v}$	Δk	Node	Δx	Δv	Δk	distance	error
1	0.00	0.00	0	1	0.00	0.00	0	0.000	0.000
2	5.59	16.52	0	3	-8.00	24.00	0	15.516	1.371
3	-10.92	22.11	-1	3	-8.00	24.00	0	8.723	1.090
4	0.13	27.58	-2	4	0.00	31.99	-1	7.178	1.269
5	2.93	35.84	-2	5	4.00	35.99	-2	1.080	0.191
6	2.93	35.84	-3	5	4.00	35.99	-2	4.143	1.035
7	4.32	39.97	-4	6	4.00	40.00	-3	2.847	1.006
Average Error =									0.85
Worst Error =									1.37

Table 8-17: Transform of L-path from Teapot #1 to Match Teapot #5

Chapter 9 Discussion

This chapter presents a discussion of applications of the DOLP transform and a discussion of how the properties of the representation for gray scale shape could be proven with experiments.

9.1 Applications of the DOLP Transform

The DOLP transform, in both its 1-D and 2-D form, can be useful as a representation for a variety of applications requiring signal detection or signal description. Characteristics of the DOLP transform that make it useful in signal detection situations are:

- It provides a function for detecting pulses that is not dependent on the sharpness of the boundary or the uniqueness of the amplitude of the pulse;
- It separates pulses of different durations so that they may be detected independently;
- It provides a way of detecting a pulse whose width is **not** known a priori;
- It provides a way to find the resolution at which some desired signal occurs;

The following paragraphs elaborate on these characteristics.

9.1.1 Detecting IUndefined Puts**

The DOLP transform provides a technique for detecting pulses in 1-D signals and regions in 2-D signals which is not dependent on the sharpness of the boundary of the pulse or region. Indeed, within the DOLP transform there is a separate signal at a higher resolution. In a 1-D signal this ability can be used to find blurred pulses of a particular frequency, even in the presence of noise. For a 2-D signal the DOLP transform provides a simple technique for detecting and describing small 2-D regions. A 2-D region will appear as a local maxima in the DOLP transform. This maxima may be tracked in consecutive frames without a search process.

The DOLP transform is also useful for detecting the orientation of a surface from an image. An image is usually composed of elements at a particular set of scales. In many natural textures, the shapes of the individual elements may be random. If the *orientation* of the physical objects which

correspond to the regions is known, the distance to the surface may be inferred from the size of the texture elements. Furthermore the orientation of the surface may be inferred from the gradient of the size. For either process, the size of the texture elements may be measured by detecting local maxima in the 3-space of the DOLP transform. The level at which the maxima occurs gives an estimate of the size of the element. This simple detection scheme will even work when the shapes of the individual elements vary randomly.

9.1.2 Detecting Pulses of Different Durations

The DOLP transform separates a signal into band-pass components. Each band-pass channel responds to signals of a particular range of durations (in 1-D) or widths (in 2-D). This property can be used to detect overlapping signals of different durations which are superimposed in the same image. For example, consider printing on a textured or nonuniform surface, such that the patterns or blotches on the surface are much larger than the printed letters. A DOLP transform of the image will separate the characters of the writing from the pattern on the paper, allow either the pattern or the writing to be detected by thresholding.

9.1.3 When Width is not known A-Priori

The DOLP transform channels are sensitive to frequency ranges which are exponentially spaced and cover the range from the smallest to the largest signal representable in the image. This property can be useful for detecting a signal whose width (or duration) is not known a-priori. Such a signal will result in a local maximum in at least one of the DOLP channels.

9.1.4 Automatic Focus

When a camera is out of focus the effect is the same as convolving a low-pass blurring function with the image. It is possible to measure whether a lens is moving toward or away from correct focus by detecting the change amplitude with which a high frequency pattern (e.g. a thin bar) is detected by a DOLP transform channel. In the case where the scene does not contain an artificial focusing pattern of known spatial frequency it is possible to servo the focus from the highest frequency level at which significant signal energy is observed in a DOLP transform.

9.2 Evaluating Claims

This research was undertaken to show that it was possible to represent an image with a set of band-pass filters and to determine the properties of such a representation. This research was undertaken with very limited resources. This resource limitation has restricted the **investigation** to forming the representation of only a few images.

The research has gone well beyond its original goals; we have shown that it is not computationally

prohibitive to compute the convolution of an image with an exponentially spaced set of band-pass filters; we have shown that such a set of convolutions can be organized into a reversible transform; we have shown that the image shapes can then be represented by detecting peaks and ridges in the band-pass images; We have shown that these peaks and ridges can be detected by local processes.

9.2.1 Claims Concerning the Representation for Shape

The primary claim of this dissertation is that the representation of a shape based on the 2-D Sampled DOLP transform which is described in chapters 6 and 7 can be matched efficiently. A secondary claim is that this representation can be matched regardless of changes in the size, position, or 2-space orientation of the shape.

The ability to match hierarchically from global to local is intrinsic to the structure of the representation. In chapter 8 we have demonstrated how this matching is done. Having such a representation does not completely solve the problem of how to best do such matching. Issues of how to organize the search for a match and what criteria to use to measure the overall goodness of the match must also be settled. This representation presents the data in a structure that allows a matching procedure to proceed hierarchically, and to use the results of each match to constrain the search for matching features at a more local level.

The hierarchical nature of the representation is intrinsic to the DOLP transform; it can not be disputed. To prove the usefulness of such a representation for matching, it is necessary to develop a matching algorithm based on the representation. The ability of the algorithm to produce correct results must be demonstrated in a large number of different images. This will provide proof that the technique works.

The computational complexity of the matching algorithm must then be analyzed. The resulting measure of computational complexity should then be compared to the complexity of other matching algorithms.

9.2.1.1 Invariance to Size and Rotation

Experiments have shown that the representation composed of M-nodes, M*-nodes, L-nodes and P-nodes is subject to cyclic distortions when a pattern shifts in position, size or orientation. As a shape increases in size, the M-nodes, L-nodes, and M*-nodes must make the transition to a higher level in discrete steps. Since these transitions are not constrained to occur simultaneously, the specific configuration of nodes does change. This is a cyclic distortion; after the change in scale has advanced by a factor of $\sqrt{2}$, the pattern will have returned to its starting configuration. The effects of change in position are similar; as a pattern moves over a distance which is one sample rate at the level of its highest M*-node, the M-nodes, L-nodes, M*-nodes and P-nodes in the representation move to the next sample at in discrete steps that are not constrained to occur simultaneously. However, after the pattern has shifted by the distance of one sample at any level, all of the nodes at the level and lower will have returned to the same configuration. This behaviour is suggested by reasoning and confirmed with experiments with squares and rectangles. The exception to the cyclic degradation from a position shift occurs when a pattern shifts closer (less than its diameter) to a second pattern.

It is possible to construct a second, more abstract, description which compensates for the cyclic distortions. This description, described in chapter 8, is composed of M-paths, M*-nodes, and L-paths. While this representation is not subject to the cyclic distortions, there remain certain illusions which can alter the representation of a shape as it undergoes a transformation in size, position, or orientation. So far all of the illusions which have such an effect also cause distortions in the perception of the form by the human visual system.

Chapter 10

Summary and Conclusions

This chapter presents a summary of the contents of the preceding chapters, a discussion of the results presented in each chapter, and the salient conclusions that can be drawn from these results.

10.1 Major Results of this Dissertation

This dissertation presents results in three areas.

1. A reversible transform (The Difference of Low Pass or DOLP transform) for detecting and mathematically representing signals of any number of dimensions. Signals are filtered into exponentially-spaced spatial frequency bins by convolution with circularly symmetric band-pass filters. The filters are size-scaled copies of a low-pass filter minus the same filter scaled larger by a scaling factor, S (typically $\sqrt{2}$). This transform resolves a signal into components of different spatial frequencies.
2. Techniques for greatly speeding up the calculation of a DOLP transform using resampling and cascaded filtering with expansion.
3. A representation for 2-D gray-scale pictures, based on the sampled DOG transform, which greatly simplifies matching of picture information for structural pattern recognition and stereo interpretation.

This dissertation may be divided into the following sections:

- Background Material (Chapters 1, 2 and 3);
- Measurement, detection and mathematical representation of nonperiodic signals (Chapters 4 and 5);
- Fast computation techniques for the detection technique (Chapter 6);
- Converting the mathematical representation to a symbolic representation which describes gray-scale shape heirarchically by spatial frequency (Chapter 7);
- Examples of the representation and its use for matching, including demonstrations of the invariance of the structure of a description to the size and orientation of the pattern (Chapter 8).

10.2 Summary of Background Chapters

Chapter 1 introduced the problem context for this research: model based recognition of 2-D patterns and 3-D objects by matching structural descriptions to prototypes. This chapter also contains a discussion of the methodologies used in this research and a summary of the results.

Chapter 2 reviewed related work on the problems of measuring and representing 2-D signals. This chapter began with a discussion of the two popular approaches to image description: edge detection and region segmentation. Both approaches are based on the assumption that an image is composed of approximately uniform regions. Careful examination of most images of "real world objects" in unconstrained lighting shows this assumption to be inaccurate. This chapter also described inadequacies in the representations produced by both of these approaches:

- the description of shape in terms of small events,
- the inability to describe gradual transitions in intensity, and
- the inability to describe textured regions.

A number of detection functions for edges are then described. This was followed by a review of several multi-resolution algorithms that have been used to solve various problems involving two dimensional signals. The chapter ended with a review of two representation techniques which give object-centered descriptions of shape.

Chapter 3 provided a brief review of mathematics and terminology from the field of digital signal processing which are employed in later chapters. Definitions were presented for convolution and correlation, the two operations were shown to be the same for a symmetric filter, and correlation was shown to be equivalent to a sequence of inner products. The transfer function of a linear operator was derived based on the properties of the eigenfunctions of linear systems. Resampling, aliasing, and the 2-D Nyquist boundary were then described. The $\sqrt{2}$ resampling operation was defined and its effects on the frequency content of an image were described. Chapter 3 ended with a review of the parameters that are commonly used to specify a digital filter.

10.3 Measurement, Detection and Mathematical Representation of Non-Periodic Signals

Chapter 4 described the foundation on which the techniques described in the later chapters are based. Chapter 4 began by describing the concept of a parameterized family of detection functions. This idea was conceived early in this research and led to the development of the DOI.P transform.

Chapter 4 then reviewed principles for the design of detection functions which are to be used to detect and describe non-periodic signals using ridge and peak detection. These principles were conceived early in this research and played a key role in the development of the DOI.P transform; they served as a guide which directed the research. These principles also show the assumptions on which the research proceeded.

One of the major innovations resulting from this research is the Difference of Low-Pass (DOLP) transform, described in chapter 5. The DOLP transform consists of a set of exponentially size-scaled band-pass filters which are formed by subtracting a sequence of size-scaled low-pass filters. The DOLP transform expands an N point signal into $\text{Log}_S(N)$ band-pass signals, where N is the number of samples in the signal, and S is the scale factor for size scaling the filters (typically $\sqrt{2}$). The band-pass signals, and a convolution of the largest low-pass filter with the signal may be added together to recover the original signal. Thus the DOLP transform is reversible; it preserves all of the information in a signal. The DOLP transform separates a signal into overlapping frequency channels. This has the effect of decomposing a signal into components of different sizes, even if the boundaries of the components are poorly defined. The configuration of peaks in the DOLP transform of a signal describes its components in a tree whose structure is invariant to the scale of the signal.

The DOLP transform may be defined for signals of any dimensionality, and may be computed by analog filters as well as digital filters. Based on this dissertation, a 1-D form of DOLP transform has been recently used to detect and discriminate defects in the coatings of florescent light bulbs [Handelsman81]. An investigation is being launched into the use of a form of DOLP transform for tracking formants in speech spectrograms. Another effort is being started to investigate the use of a form of DOLP transform to describe range data from a depth sensor. Also, we have recently proposed the use of a 3-D form of DOLP transform to represent 3-D shape in terms of primitives which are fuzzy spheres.

As the band-pass impulse responses are scaled larger in size it becomes possible to resample the band-pass signals at a rate proportional to the scaling of the band-pass filter. This resampling can greatly reduce the complexity of computing the DOLP transform as well as the amount of storage required. Resampling at a rate proportional to the scaling of the band-pass impulse response can be designed so that the no information is lost to the description from aliasing, while the computational cost is reduced from $O(N^2)$ to $O(N \text{ Log } N)$ and the storage requirements are reduced from $O(N \text{ Log } N)$ to $3N$. (N is the number of sample points in the image.) The resampled DOLP transform was also defined and described in chapter 5.

10.4 Techniques for Fast Computation of a DOLP Transform: The DOG and Sampled DOG Transforms

Chapter 6 concerned techniques for which were developed in this research to greatly reduce the cost and speed of computing a 2-D DOLP transform. Two properties of the Gaussian function can be used to obtain substantial decreases in the cost of computing a DOLP and a sampled DOLP transform:

1. the Gaussian auto-convolution scaling property, and
2. The separability of the circularly symmetric 2-D Gaussian function.

The Gaussian auto-convolution scaling property provides that when a Gaussian function is convolved with itself, the result is the Gaussian function scaled larger in standard deviation by a factor of $\sqrt{2}$. This suggests that the DOLP transform may be speeded up by producing each low-pass image from

the previous low-pass image by convolving by the appropriate Gaussian function. In fact, the DOLP transform may be reduced in cost from $O(N^2)$ multiplies to $O(N \log N)$ multiplies by using an additional technique for scaling a Gaussian function by \sqrt{V} : The VT expansion function. The VT expansion operation maps each row of a function on a cartesian sample grid onto each diagonal of a \sqrt{V} sample grid. The expanded function is zero or undefined for points between those on the VT grid. This expanded Gaussian filter has a transfer function with a Gaussian center-lobe which is scaled smaller (in frequency) by a factor of \sqrt{V} . There are also reflections of this center lobe in the four corners of the (u, v) Nyquist plane. By proper choice of filter parameters, these reflections can be formed such that they fall over a region of the auto-convolved Gaussian's transfer function where the response is very small (i. e. < -70 dB). Thus, when the two functions are convolved, the center lobes are attenuated to a very small response (< -100 dB in our examples).

By repeated VT expansion the original filter may be scaled to the same size as the cumulative low-pass impulse response at each level. Thus each low-pass image for level k -HI can be formed by convolving the low-pass image at level k with a copy of the low-pass filter that has been expanded k times.

An algorithm for computing a DOLP transform using Gaussian filters, auto-convolution, and expansion was described in section 6.2. This algorithm, called "Cascaded Convolution with Expansion", produces a form of DOLP transform (the DOG transform) in $O(N \log N)$ multiplies.

Further speed-up, and a reduction in storage requirements are possible by including VT resampling in the algorithm. This algorithm, called "Cascaded Convolution with Resampling", gives a form of sampled DOLP transform, the SDOG transform, in $3 X_0 N$ multiplies, where X_0 is the number of coefficients in the kernel Gaussian filter. As with the Sampled IXL P transform, $3N$ storage cells are required.

Chapter 6 defined:

- The Gaussian function
- The 2-D Circularly Symmetric Gaussian filter
- The Gaussian auto-convolution scaling property
- the \sqrt{V} expansion operation
- Cascaded convolution with expansion and the DOG transform
- Cascaded convolution with resampling and the SDOG transform

In this chapter the complexity of the cascaded convolution with resampling was derived. This complexity was compared to that of computing a SDOG transform using FFF convolution. Cascaded convolution with resampling was shown to be more efficient whenever the image signal is larger than 65×65 samples.

Chapter 6 also examined the attenuation of the reflections that result from the expansion operator, and the accuracy of the auto-convolution scaling property when used with a finite Gaussian filter with a circular support. At the end of chapter 6, the impulse responses of the level 0 and level 1 band-pass filters were shown, and linear and log plots were shown of the transfer functions of the level 1 and level 2 band-pass filters.

10.4.0.1 Conclusions Concerning Signal Detection

The principal conclusions to draw from chapter 6 are that:

- A DOLP transform is not prohibitively expensive to compute.
- A DOLP transform can be implemented using Gaussian filters and cascaded convolution with expansion such that the computational cost is less than that of a Fast Fourier Transform.
- Cascaded convolution with expansion can be used to produce a sequence of low-pass images such that the impulse response with which the images are convolved have standard deviations which form an exponential sequence, $\sigma_k = \sigma_0 \sqrt{2}^k$.
- Cascaded convolution with expansion can be implemented such that the impulse responses have stop bands which are kept very small, (i. e. < -80 dB).

The work described in chapter 6 could be extended in several ways.

- A substantial speedup (a factor of 49/18) can be achieved by using the separability property of the circularly symmetric Gaussian function. However this technique will result in a slightly higher worst-case stop-band ripple because a square support is needed for separable filtering. An investigation into the extent of the degrading of the stop band rejection from this method would be useful. Such an investigation is to be carried out in the near future.
- The cascaded-filtering-with-expansion algorithm approximates the Gaussian low-pass filters with an auto-convolved Gaussian convolved repeatedly with expanded Gaussians. This is illustrated in figure 6-9. The measures which were used to determine the accuracy of this approximation are somewhat crude. It would be interesting to compute the standard deviations of the sequence of filters produced in this manner. It would also be interesting to find a measure for how closely these composite filters approximate *true* Gaussian functions.
- The effects of the Gaussian filter parameters R and α have only been examined over a limited region of the R, α space. This examination showed that for $R=4.0$ and $\alpha = 4.0$ the transfer function tapers monotonically along the u and v axes of the spatial frequency plane to a response of approximately zero at the Nyquist boundary points $u = \pm\pi, v = 0$.²¹ An exhaustive exploration of the effects of R and α would be interesting. However

²¹The function is symmetric and so u and v are interchangeable.

M-nodes serve as markers for distinct features in visual forms. M-nodes occur at several levels for forms such as corners, ends of bars, and other convex and concave parts in a visual form. They also denote the presence of forms which are not elongated. Examples of the forms that cause M-nodes are given in section 7.1 and 7.3. Because M-nodes denote distinct visual features they provide excellent tokens for matching images. Correspondence matching in an SDOG transform is a process of determining the correspondence between M-nodes, M*-nodes, and L-nodes in the descriptions from two images.

The fact that each band-pass impulse response is a copy of the impulse response from the next lower level scaled larger by $\sqrt{2}$ provides that the M-nodes from adjacent levels occur within two sample distances of each other. Thus it is possible to connect M-nodes between the band-pass levels by having each M-nodes search for M-nodes in a small neighborhood in the band-pass image above it. Such adjacent M-nodes form a two-way pointer between themselves. Sequences of M-nodes at several levels such that each M-node is connected to one M-node above it and/or one M-node below it are called M-paths. M-paths that describe a visual form give a tree. At the top levels of the tree there are M*-nodes that provide an estimate of the size of the visual form. Aligning the M*-nodes from two images gives an initial estimate of the relative position and size of the two visual forms. The relative orientation is provided by determining the correspondence of the M-nodes, M*-nodes and L-nodes in lower levels of the tree. Such matching is described in Chapter 8.

Forms that are long and thin result in ridges at several adjacent band-pass levels. Comparing the values of ridge points at adjacent levels gives ridge points in the three dimensional SDOG transform. The 3-space ridge points are labeled as L-nodes. L-nodes are linked to adjacent L-nodes with two-way pointers to form an L-path. Except for certain degenerate forms, L-paths begin and end at M*-nodes. An L-path describes the points along the center of an elongated form. The level of each L-node gives an estimate of the width of the form at that point along the center of the form. The alignment of the M*-nodes at each end of an L-path provides an initial estimate of the best alignment of the L-paths from two images. A nearest neighbor matching rule was described for comparing two L-paths in section 8.5.

A conclusion that can be drawn from the algorithms described in chapter 7 is that a structural description of an image can be constructed without the use of explicit measures of directionality. The issue of whether a measure for directionality was needed to detect (or even define what is meant by) ridges in each band-pass image was raised at the outset of our investigation into techniques for constructing a description of an image from a DOLP transform. The outcome was that such a measure is not necessary: a two pass process can be used to detect ridges. In the first pass of this process samples are linked to their largest neighbors. In the second pass, samples which link to each other are marked as ridge nodes. This process was found to be sufficient for detecting ridges.

A fundamental reason why the processes described in chapter 7 work is the smoothness of each band-pass image. This smoothness is a result of the band-pass characteristics of the filters used in the DOLP transform. The DOLP band-pass filters sufficiently constrain the spatial frequency content of each band-pass image so that relatively simple processes may be used to detect peaks and ridges in each image. The $\sqrt{2}$ scaling between filters constrains the changes between adjacent band-pass images so that nearest neighbor comparisons may be used to detect the local peaks and ridges among the band-pass images in the transform space.

The matching examples that were shown in chapter 8 were intended to both illustrate the size and rotation invariance of a structural description based on a DOLP transform, and to show kinds of matching which can be done with such descriptions. In some sense these were the results of a preliminary investigation. These preliminary results were promising. M^* -nodes and M -paths were found to be particularly useful in finding the correspondence of components in two descriptions. We are preparing to launch a thorough development of matching techniques for descriptions based on the DOLP transform within the problem domains of structural pattern recognition and stereo image correspondence. This promises to be an exciting and fruitful investigation.

Appendix A Selection of Filter Parameters

This appendix describes the choice of filter parameters, $R = 4.0$ and $\alpha = 4.0$, for the experimental implementation of the SDOG transform which was used to develop the structural representation.

The choice of R and α must balance two opposing constraints. On one hand, the low pass filters must sufficiently attenuate response at frequencies outside of the Nyquist boundary at each low-pass level to avoid aliasing from resampling. Such aliasing would result in random errors in the position of peaks and ridges as well as the detection of spurious peaks and ridges. The filter response can be made arbitrarily small outside the Nyquist boundary by increasing the number of coefficients of the filter, (i.e. by increasing R). It is also possible to adjust the position of the stop band towards the origin, at the expense of increasing the stop-band ripple, by decreasing the parameter, α .

On the other hand it is desirable to keep the number of coefficients and thus the computational cost of the SDOG transform as small as possible.

The R parameter determines the cost of a DOLP transform (Given the size of the image, and the scaling value $S = \sqrt{2}$). R should be chosen to be the smallest value which gives *acceptable* low levels of aliasing when the low pass images are sampled. The meaning of *acceptable* remains a topic of debate. We have suggested that the stop band ripple is acceptable if the magnitude of the worst case stop-band error is less than the quantization resolution used to represent the samples. In our actual choice of R and α we were much more conservative than this guideline.

The α parameter specifies the standard deviation of the filter for a given R . Since α controls the tapering of the coefficients at the boundary of the filter support, it gives a trade-off between the transition width (ΔF) and the magnitude of the ripples (δ) in the stop band. Increasing α decreases the size of the ripples in the stop band region while making the transition region wider and moving the edge of the stop band away from the origin. For any value of R , α should be chosen as large as possible, so that the stop band ripple is as small as possible. The upper limit for α is the value at which the largest filter response at the Nyquist boundary is of the same magnitude as the stop-band ripple.

The first re-sampling occurs at the level 1 low pass image, where the impulse response of the low-pass filter is the kernel filter, $g(x,y;R,\alpha)$ convolved with itself. Thus the transfer function of the composite filter at level 1 is the square of the transfer function of the the kernel filter.

It was decided to design the kernel filter so that the outer edge of its transition region would just

touch the new nyquist boundary for $\sqrt{2}$ sampling. This meant that the sampling distance at each level would be approximately $\sqrt{2}$ smaller than needed to minimize aliasing. This provides a factor of $\sqrt{2}$ better positional accuracy in the description, although it tends to make peaks and ridges less sharp. This also meant that the worst case stop-band ripple would be the square of the ripple in the kernel filter.

Parameters for the kernel filter were tested to determine:

1. The worst case ripple outside the Nyquist boundary for $\sqrt{2}$ sampling.
2. The values at $u, v = \pm w/2$, the four points on the new nyquist boundary that are closest to the origin.

As a first pass, filters and their transfer functions were computed at each of the 9 points given by all combinations of:

$$R \in \{3, 4, 5\}$$

$$C \in \{3, 4, 5\}$$

These starting values were chosen from earlier experience with circularly symmetric Gaussian filters. The values obtained for maximum amplitude of stop band ripple (S) and for $G(u=w/2, v=w/2)$ (G is for the real part of the transfer function) are shown below in table A-1. The symbol N/A is given for S when the ripple did not come to a peak inside the u, v plane.

	$a = 3.0$	$a = 4.0$	$a = 5.0$
	$\delta, G(w/2, w/2)$	$S, G(w/2, w/2)$	$\delta, G(w/2, w/2)$
R = 3	0.031, 0.025	N/A, 0.063	N/A, 0.109
R = 4	-0.018, 0.013	-0.008, 0.011	0.003, 0.021
R = 5	-0.003, 0.0111	-0.006, -0.006	-0.002, 0.002

Table A-1: Results of Initial Parameter Trial

From this experiment it was learned that $R=3$ was not quite adequate to keep the transition region within the Nyquist boundary for $\sqrt{2}$ sampling. $R=5$ was rejected because $R=4$ was judged to be adequate. The value of $a = 4.0$ was judged to be the best of these three trial points due to the closeness of the stop band ripple magnitude and the maximum stop band error. The transfer functions were then computed for $R = 4$ and $a = 3.80$ to $a = 4.20$ in steps of 0.05. The value $a = 4.0$ was finally chosen to put the first zero crossing at the points $(u, v) = (\pm w/2, 0)$ and $(0, \pm w/2)$, and thus was selected for use in developing the symbolic description technique described in the chapters 7 through 9.

From the above it can be seen that the worst case aliasing when the level 1 low pass image is sampled, occurs at $(u, v) = (\pm w/2, \pm w/2)$. These points are on the Nyquist

boundary, and for them the filter response is $0.011^2 = .000121$ or -78.34 dB down from the maximum response (1.0 at DC). All other aliased frequencies are less than or equal to $-.008^2 = 0.000064$ or -83.8 dB or smaller. This was judged to be adequate and attention was turned to other matters.

References

- [Agin 72] Agin, G. X.
Representation and Description of Curved Objects.
Technical Report A I Memo 173, Stanford, 1971
- [Agin 76] Agin, G. J. and T. O. Binford.
Computer Description of Curved Objects.
IEEE Transactions on Computers O25(4):439-449, April, 1976.
- [Agin and Binford 73] Agin, G. J. and T. O. Binford
Computer Description of Curved Objects.
In *Proceedings of the Third InL Conf on Artificial Intelligence*, pages 629-640.
IJCAI March, 1973.
- [Baird 77] Baird, M. L.
A Computer Vision Data Base for the Industrial Bin of Parts Problem.
Technical Report Technical Report GMR-2502, GM Research Labs, Aug., 1977.
- [Barrow ed] Barrow, H. G. et al.
Parameteric Correspondence and Chamfer Matching: Two New Techniques for Image Matching.
Technical Report Technical Note 153, SRI International, undated
- (Blum 67) Blum, H.
A Transformation for Extracting New Descriptors of Shape.
In W* Wathen-Dunn (editor), *Models for the Perception of Speech and Visual Form*,
MXT. Press, Cambridge Mass., 1967,
- [Campbell 68] Campbell R W. and Robson, I G.
Application of Fourier Analysis to the Visibility of Gratings.
Journal of Physiology 197:551-566. 1968.
- fCrowley 76J Crowley, J. L.
Experiments in the Analysis of Image Texture*
Master's thesis, C-MU Dept of Elect Eng[^] Supt., 1976.
- [Crowley 78aJ Crowley, J. L. and A. C Parley
The Analysis, Synthesis* and Evaluation of Local Measures for Discrimination and
Segmentation of Textured Regions.
In *Conference on Pattern Recognition and Image Processing** pages 172-378. IEEE
Computer Society, June, 1978,
Not listed in table of contents, but it's there!
- fCrowley 78bJ Crowley, J. L.
The Application of Signal Processing to the Description of Digital Pictures.
Technical Report. CMU, Dept of Elect Eng[^] June, 1971
Thesis Project

- [Crowley 80] Crowley, J. L. and A. C. Parker.
Transfer Function Analysis of Picture Processing Operators.
In Robert M. Haralick and J. C. Simon (editor), *Issues In Digital Image Processing*,
chapter 1 pages 3-30. Sijthoff & Noordhoff, 1980.
- [Duda 73] Duda, R. O. and Hart P. E.
Picture Processing and Scene Analysis.
Wiley, New York, 1973.
- [Eastman 70] Eastman, C. M.
Representations for Space Planning.
Comm. of the ACM 13(4):242-250, April, 1970.
- [Freeman 61] Freeman, H.
On Encoding of Arbitrary Geometric Configurations.
IRE Transactions on Electronic Computers EC-10:260-268, June, 1961.
- [Hall et. al. 76] Hall, E. L., Rouge, Lt. J. and Wong, R. Y.
Hierarchical Search for Image Matching.
In *Proc. 1976 IEEE Conf. on Decision and Control*, pages 791-796. IEEE,
December, 1976.
- [Hamming 77] Hamming, R. W.
Digital Filters.
Prentice Hall, Englewood Cliffs, N. J., 1977.
- [Hanson and Riseman 74] Hanson, A. and Riseman, E.
Pre-processing Cones: a Computational Structure for Scene Analysis.
Technical Report COINS Tech Report 74C-7, Univ. of Mass., Sept., 1974.
- [Hanson and Riseman 78] Hanson, a. and Riseman, E.
Visions: A Computer System for Interpreting Scenes.
In Hanson and Riseman (editor), *Computer Vision Systems*, . Academic Press,
1978.
- [Harmon and Julesz 73] Harmon, L. D. and Julesz, B.
Masking in Visual Recognition: Effects of Two Dimensional Filtered Noise.
Science 180:1194-1197, 1973.
- [Horowitz 76] Horowitz, S. I. and Pavlidis, T.
Picture Segmentation by a Tree Traversal Algorithm.
Journal of the ACM 23:368-388, 1976.
- [Hu and Rabiner 72] Hu, J. V. and I. R. Rabiner.
Design Techniques for Two Dimensional Digital Filters.
IEEE Transactions on Audio and Electroacoustics AU-20(4):249-257, October,
1972.

- * [Huang 72] Huang, T. S.
Two Dimensional Windows.
IEEE Transactions on Audio and Electroacoustics AU-20(2):88-89, March, 1972.
- [Hueckel 71] Hueckel, M. H.
An Operator which locates edges in Digitized Pictures.
JACM 18:113-125, Jan, 1971.
- [Hueckel 73] Hueckel, M. H.
A Local Visual Operator Which Recognizes Edges and Lines.
JACM 20(4):634-647, Oct, 1973.
- [Kelly 71] Kelly, M. D.
Edge Detection in Computers by Computer Using Planning.
In B. Meltzer and D. Mitchie (editor), *Machine Intelligence*, . Unknown, 1971.
- [Kender 80] Kender, John R.
Shape from Texture.
PhD thesis, Dept. of Computer Science, Carnegie-Mellon University, November, 1980.
- [Klinger and Dyer 76] Klinger, A. and Dyer, C. R.
Experiments in Picture Representations Using Regular Decomposition.
Computer Graphics and Image Processing 5(1):68-105, March, 1976.
- [Landau 61] Landau, H. J. and H. O. Pollack.
Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - II.
Bell Sys. Tech. Journal 40:65-84, Jan, 1961.
- [Levine 78] Levine, M. D.
A Knowledge-Based Computer Vision System.
In Hanson and Riseman (editor), *Computer Vision Systems*, . Academic Press, 1978.
- [Levine and Leemet 76] Levine, M. D. and Leemet, J.
A Method for Non-Purposive Picture Segmentation.
In *Proc. Third Joint Conf. on Pattern Recog., Coronado, Ca.*, pages 494-488. 1976.
- [Marr 78] Marr, D.
Representing Visual Information.
Technical Report A. I. Memo 415, MIT, 1978.
- [Marr 79a] Marr, D. I., and Hildreth, E.
Theory of Edge Detection.
Technical Report A.I. Memo 518, M.I.T., April, 1979.
- [Marr 79b] Marr, D. and Poggio, T.
A Computational Theory of Human Vision.
Proc. R. Soc. Lond. B, 1979.

- [McKee 77] Mckee, J. W. and J. K. Aggarwal.
Computer Recognition of Partial Views of Curved Objects.
IEEE Trans. on Comp. C-26(8):790-799, Aug, 1977.
- [Minsky 63] Minsky, M.
Steps Toward Artificial Intelligence.
In *Computers and Thought*, pages 406-450. McGraw Hill, New York, 1963.
- [Nevatia and Binford 74] Nevatia, R. and T. O. Binford.
Structured Description of Curved Objects.
Proceedings of the 4th International Conf. on Artificial Intelligence 2, 1974.
- [Newell 59] Newell, A., Shaw, J. C., and Simon, H. A.
Report on a General Problem Solving Program.
In *Proc. Int. Conf. on Inf. Proc.*. Paris Unesco House, 1959.
- [Nilsson 69] Nilsson, N. J.
A Mobile Automaton: An Application of Artificial Intelligence Techniques.
In *Proc. Int. Joint Conf. on Artificial Intelligence*. April, 1969.
- [Ohlander 75] Ohlander, R.
Analysis of Natural Scenes.
PhD thesis. Carnegie-Mellon University Computer Science Dept., June, 1975.
- [Öppenheim 75] Oppenheim, A. V. and Schafer, R. W.
Digital Signal Processing.
Prentice-Hall inc., Englewood Cliffs, N. J., 1975.
- [Papoulis 68] Papoulis, A.
Systems Sciences: Systems and Transforms with Applications in Optics.
McGraw-Hill, New York, 1968.
- [Parks 72] Parks, T. W. and McClellan, J. H.
A Program for the Design of Linear Phase Finite Impulse Response Filters.
IEEE Trans on Audio and Electroacoustics AU-20(3):195-199, Aug., 1972.
- [Perkins 78] Perkins, W. A.
A Model Based Vision System for Industrial Parts.
IEEE Trans. on Comp. C-27(2):126-143, February, 1978.
- [Prewitt 66] Prewitt, J. M. S. and M. L. Mendelsohn.
The Analysis of Cell Images.
Ann. N. Y. Acad. of Sci. 128:1035-1053, 1966.
- [Prewitt 70] Prewitt, J. M. S.
Object Enhancement and Extraction.
In B. S. Lipkin and A. Rosenfeld (editor), *Picture Processing an Psycho-Pictorics*,
pages 75-149. Academic Press, 1970.

- [Price 76] Price, K.
Change Detection and Analysis in Multi-Spectral Images.
 PhD thesis, Dept of Computer Science, Carnegie-Mellon University Science.,
 December, 1976.
- [Rabiner 75] Rabiner, L and Gold, B.
Theory and Application of Digital Signal Processing.
 Prentice Hall, inc., Englewood Cliffs N. J., 1975.
- * [Roberts 65] Roberts, L. G.
 Machine Perception of Three Dimensional Solids.
 In J. T. Tippett et. al. (editor), *Optical and Electrophysical Information Processing*,
 MIT Press, 1965.
- [Rosen and Nilsson 69] Rosen, C. A. and Nilsson, N. J.
Application of Intelligent Automata to Reconnaissance.
 Technical Report SRI Project 5953, Rome Air Development Center, 1969.
- [Rosenfeld 69] Rosenfeld, A.
Picture Processing by Computer.
 Academic Press, 1969.
- [Rosenfeld and Vanderbrug 77] Rosenfeld, A. and Vanderbrug, G. J.
 Coarse-Fine Template Matching.
IEEE Trans. on Man, Systems, and Cybernetics SMC-7(2):104-107, Feb., 1977.
- [Slepian 61] Slepian, D. and H. O. Pollack.
 Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty - I.
Bell Sys. Tech. Journal 40:43-63, Jan, 1961.
- [Tanimoto 76] Tanimoto, S. L.
 Pictorial Feature Distortion in a Pyramid.
Computer Graphics and Image Processing (3):333-352, Sept., 1976.
- * [Tanimoto 78] Tanimoto, S. L.
 Regular Hierarchical Image and Processing Structures in Machine Vision.
 In Hanson and Riseman (editor), *Computer Vision Systems*, Academic Press,
 1978.
- [Tanimoto and Pavlidis 75] Tanimoto, S. I., and Pavlidis, T.
 A Hierarchical Data Structure for Picture Processing.
Computer Graphics and Image Processing (2):104-119, June, 1975.
- [Uhr 72] Uhr, L.,
 'Layered Recognition Cones': Networks that Preprocess, Classify, and Describe.
IEEE Trans on Computers C-21:758-768, 1972.

- [Uhr78] Uhr, L
Recognition Cones and Some Test Results; the Imminent Arrival of Well-Structured Parallel-Serial Computers; Postions, and Positions on Positions, In Hanson and Rtecman (editor), *Compuler Vision Systems**. Academic Press, 1978.
This paper reads like Its title!
- fWamock 67] Waraock, 1E
AHiddenSurfaceAlgorithmforCompulerGeneratedHalf-TonePictures. Technical Report TR-445, Univ. OfUtah Computer Science DcpL, 1967.
- [Wozncraft 65J Wozencraft J. M. and Jarobs, L M.
PrinciplesofCommunicationEngineering. John Wiley & Sons, 1961