

Image Segmentation via
Multiresolution Extrema Following

Technical Report 87-012

1987

Lawrence Mark Lifshitz

The University of North Carolina at Chapel Hill
Department of Computer Science
New West Hall 035 A
Chapel Hill, N.C. 27514

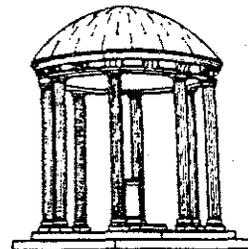


Image Segmentation via Multiresolution Extrema Following

by

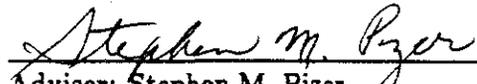
Lawrence Mark Lifshitz

A Dissertation submitted to the faculty of The University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill

April, 1987

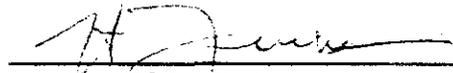
Approved by:



Adviser: Stephen M. Pizer



Reader: John H. Halton



Reader: Henry Fuchs



Reader: Sue E. Goodman

Copyright © 1987

Lawrence Mark Lifshitz

ALL RIGHTS RESERVED

Acknowledgements

My tenure at U.N.C. has been a pleasurable one, made so by the warmth and commitment of many of the people here. I feel a strong bond with many of individuals who have passed through these halls during my stay. Most of these people entered during my early years and so are dispersed to the winds by now, yet the friendship remains.

My committee is, of course, still here. I am indebted to them for the time and energy they have spent on my behalf. To Steve Pizer, my chairman and advisor, I owe a special thanks. Not only has he been the chairman of my committee, he has guided me along the path from student to researcher. My progress was made all the more enjoyable and instructive due to the conducive research environment he was able to provide.

Several friends have been particularly encouraging and supportive through the years. Sandra Bloomberg has been forced to listen to my fears, frustrations, and (occasionally) successes - her office is down the hall. Her outlook and advice have always proved friendly and helpful. Dorothy, whose friendship will always be dear, has seen me through some of the toughest moments; I hope they don't discourage her. My friends and tennis partners, Brad and Duncan, have been there when I needed to vent my frustration by pounding tennis balls - even if most of them went out. Finally, Jodie's love and support have been of immeasurable importance.

I dedicate this dissertation to my family; their love and encouragement have been unfailing. I love them dearly.

LAWRENCE MARK LIFSHITZ. Image Segmentation via Multiresolution Extrema Following
(Under the direction of STEPHEN M. PIZER).

Abstract

The aim of this research has been to create a computer algorithm to segment greyscale images into regions of interest (objects). These regions can provide the basis for scene analysis (including shape parameter calculation) or surface-based shaded graphics display. The algorithm creates a tree structure for image description by defining a linking relationship between pixels in successively blurred versions of the initial image. The image description describes the image in terms of nested light and dark regions. This algorithm can theoretically work in any number of dimensions; the implementation works in one, two, or three dimensions.

Starting from a mathematical model describing the technique, this research has shown that

- by explicitly addressing the problems peculiar to the discreteness of computer representations the segmentation described by the mathematical model can be successfully approximated.
- although the image segmentation performed sometimes contradicts semantic and visual information about the image (e.g., part of the kidney is associated with the liver instead of the rest of the kidney), simple interactive post-processing can often improve the segmentation results to an extent sufficient to segment the region desired.
- the theoretical nesting properties of regions, originally thought to hold in all circumstances, does not necessarily apply to all pixels in a region.

The interactive post-processing developed selects regions from the descriptive tree for display in several ways: pointing to a branch of the image description tree (which is shown on a vector display), specifying by sliders the range of scale and/or intensity of all regions which should be displayed, and pointing (on the original image) to any pixel in the desired region.

The algorithm has been applied to about 15 CT images of the abdomen. While performance is frequently good, work needs to be done to improve performance and to identify and extend the range of images the algorithm will segment successfully.

Table Of Contents

Acknowledgements	iii
1 Introduction and Overview	1
1.1 The Driving Problem	1
1.2 Thesis	1
1.3 Results	2
2 Background on Segmentation	3
2.1 Segmentation as a Step towards Interpretation	3
2.2 Common Segmentation Techniques	4
2.3 Context	5
2.4 Multiresolution Approaches to Image Segmentation	6
2.4.1 Details of the Pyramid Approach	6
2.4.2 The Difference of Low-Pass Transform	8
2.4.3 Cheng and Lu	12
2.4.4 The Stack	14
2.4.5 Helminck's Work	18
3 Mathematical Properties of the Basic Stack	20
3.1 Morse Theory Basics for the Generic Case	20
3.1.1 The Morse and Splitting Lemmas	21
3.2 Embedding an Image in a Family Based Upon Gaussian Blurring	28
3.3 Containment for Extremal Region Paths in the Generic Case	35
3.4 Noncontainment for Extremal Region Paths in the Gaussian Case	37
4 Theoretical Issues due to Discreteness	44
4.1 Linking Criteria	44
4.2 Blurring	51
4.3 Blurring Rate	52
4.4 Spatial Sampling	54
4.5 Extrema Creation	54
4.6 Artifacts Introduced by Discrete Implementation	55
5 Extensions to the Stack - Embedding Schemes	57
5.1 Anisotropic Stationary Blurring	58
5.2 Nonstationary Blurring	63
5.2.1 Mathematics of Nonstationary Blurring	64
5.2.2 Blurring Strategies	66
6 Use of the Stack Program	69
6.1 Implementation Performance	69
6.2 Interactive Display Based on the Image Description Tree	71
6.3 Future Post-Processing Techniques	77
6.4 Results	79

7	Conclusions and Future Directions	83
7.1	Conclusions	83
7.2	Future Directions	83
7.2.1	Modeling Tree Stability and Deformations	84
7.2.2	Tree Matching	88
7.2.3	Additional Modifications to the Blurring Algorithm	90
7.2.4	Linking Strategies	90

Chapter 1

Introduction and Overview

1 The Driving Problem

Given an array of sampled intensity values that represents an image, it is very difficult to instruct a computer in both a precise and general way to "understand" the contents of the image to a level approaching that of humans. Humans are able to recognize broad classes of objects from many different angles and distinguish them from noisy or cluttered backgrounds. People also understand the relationships between the objects. Typically the class of objects and images on which an algorithm will work is restricted, in order to make the recognition problem easier. Some algorithms also require human guidance at difficult stages of the image analysis. Algorithms also vary in the amount of a priori information about the expected scene that is incorporated and the amount of context which is taken into account during the recognition process.

I have chosen as my driving problem the recognition of organs in abdominal CT scans. Recognition can be broken down into two fairly standard phases. The first is the grouping of pixels into regions such that all the pixels in a region belong to the same organ and there are no pixels in the region that don't belong to the organ. This is called the segmentation step. The second, or labeling, phase is the semantic understanding of what this region represents. I concentrate on the segmentation step in this dissertation, but the tree data structure produced by the algorithm is well suited to guide a semantic understanding phase. These phases would not necessarily have to be performed completely independently of each other.

2 Thesis

My thesis is that the multiresolution extrema following approach to image segmentation can be made to produce useful segmentation results in many instances. This examination of an image at many levels of resolution incorporates global, contextual information into the segmentation process. In order to produce successful results, techniques

have been developed which bridge the gap between the theoretical exposition of the stack method and a practical realization of the method on a digital computer.

The theory upon which the approach is based is all derived for continuous processes. Surfaces are smooth and continuous. Blurring is performed in infinitesimal steps. On the other hand, our algorithms must deal with non-smooth sampled images that are blurred by non-infinitesimal amounts. In many instances where the initial application of the stack algorithm does not yield a completely correct segmentation, the segmentation produced can be easily modified in an interactive fashion to produce an accurate segmentation.

I also claim that modifications to the basic stack algorithm based upon the expected image content can perform better, under some circumstances, than the unmodified stack which does not include any a priori information. Specifically, allowing the blurring kernel to vary its shape based upon information about suspected object positions should improve the segmentation phase of the algorithm.

3 Results

I have shown that

1. by explicitly addressing the problems peculiar to the discreteness of computer representations we can successfully mimic the segmentation abilities of the theoretical stack.
2. although stationary blurring creates links and nesting relationships which sometimes contradict semantic and visual information about an image (e.g., part of a kidney links to the liver instead), interactive post-processing can still produce useful image segmentations. In addition, varying the blurring scheme can improve the segmentation results. The specific variation of blurring to perform depends upon the certainty of the a priori information provided.
3. the theoretical nesting relationship which was thought to hold for the stack algorithm does not have to hold. From a practical point of view nesting relationships are still maintained.

Chapter 2

Background on Segmentation

1 Segmentation as a Step towards Interpretation

An image as it is stored by a computer is just a multi-dimensional array of pixel values. Although we as humans may look at the displayed image and recognize it as meaningful, the computer must algorithmically analyze the array of pixel values before it can reach any conclusions about the content of the image.

A computer must deal with objects in an image, as objects and not just as unrelated pixels, for many reasons:

1. computer vision (e.g., robotics)
2. computer analysis of quantitative properties of objects
3. computer manipulation of objects for image display via man-machine interactions
4. object-based nonstationary image restoration.

Before any of the above mentioned object-related actions can be taken, one or two conditions must be met:

1. The object must be recognized as an entity distinct from other objects in the image (i.e., pixels belonging to the object must be understood to be related in some way).
2. The entity must be labeled. It must be understood that it is, in fact, the specific object that is being searched for.

Most image processing techniques perform step # 1 first and independently of step # 2. Step # 1 is commonly called the *image segmentation* step. Sometimes image segmentation is performed in tandem with tentative labeling, thereby gaining the use of additional semantic information in order to perform the segmentation more accurately. This approach is envisioned for the iterative version of the stack.

2 Common Segmentation Techniques

There exists a large number of image segmentation techniques. Most techniques, however, fall into one of two broad categories:

1. Region growing techniques
2. Boundary detection techniques

Both sets of techniques proceed on the assumption that there is some property (or group of properties) which is similar or slowly varying among pixels belonging to one object. Region growing techniques start with a set of pixels which are similar and keep adding neighboring pixels to the set, until no "similar" neighboring pixels remain. Boundary finding techniques search the image for locations which display marked *dis*-similarity between neighboring pixels, and thereby locate the *boundary* of an object. Typically some post-processing is necessary to join together the pieces of boundary found throughout the image into *closed* boundaries. Region growing and boundary finding techniques tend to work in some limited circumstances.

The above approaches fail to take into account *global* information during the segmentation process. Region growing tends to take a slightly more global view than boundary finding, but still the decision to create a boundary or add pixels to an object is made by looking at the pixels in question and their local neighborhood. Since it is questionable whether even humans could correctly segment an image if forced to always look at only very local small regions of an image, we would expect better performance by somehow creating an algorithm with a more global view.

Humans also approach the recognition process with a priori knowledge about relations which are valid and to be expected between various objects in the image as well as knowledge about objects as entities with a purpose distinct from their "visual" attributes. Thus a house painted green and yellow is still recognized as one object, and not two - one green and one yellow. Similarly, we understand that in some instances we will want to distinguish between brown and green grass in a field, and sometimes we want to treat them entirely as one entity, depending upon our a priori knowledge and goals.

3 Context

Use of global information is an attempt to use contextual information for image segmentation and labeling. To beneficially examine and understand most images, contextual information must be used. Many researchers have used contextual information to help in the analysis of images. The way the information has been used varies considerably. Oftentimes a primarily statistical approach to specifying relationships is employed. Other times something more akin to simply specifying the allowable consistent relationships works well enough. Many applications employ contextual information only for the labeling phase of a recognition scheme. Often regions or tokens have already been identified by use of more local means. Toussaint [Toussaint, 1978] gives a survey of several uses of context in pattern recognition and scene analysis. He compares the approaches under the unifying theme of compound decision theory. One of the conclusions he reaches is that "in order to use statistical contextual information in an efficient manner, it is necessary to use suitable approximations to otherwise unmanageable probability distributions".

Haralick [Haralick, 1983] examines several different statistical approaches to incorporating context. He reframes each in terms of a Bayesian decision theoretic framework. The specific problems he was examining had to do with the general labeling problem. Haralick reaches the significant conclusion that "more is to be gained by discovering sub-optimal ways of handling context than by discovering optimal ways of handling local structure". He also analyzes probabilistic relaxation labeling in terms of Bayesian theory. He discovers that

each iteration computes the conditional probability of each unit's category interpretation given a new context which is the context of the previous iteration enlarged by one neighborhood width... relaxation iterations must only continue until either the conditional independence assumptions no longer hold or until the entire context is taken into account.

So relaxation labeling is also a way of incorporating context into the labeling process.

Contextual information can also be used in the segmentation phase of image analysis. Feldman and Yakimovsky [Feldman and Yakimovsky, 1974] use Bayesian decision theory to decide which regions in an image to merge and when to stop merging. They incorporate two levels of learning. The most restrictive assumption in their program is that the interpretation of a region only depends upon adjacent regions. Chen [Chen and

Pavlidis, 1980] also performs image segmentation as a sequence of decision problems within the framework of a split-and-merge algorithm. Regions are not labeled. A statistically ambiguous boundary region is also defined.

Much has been written on the topic of statistical pattern classification using contextual information. See, for example, the work by Fu and Yu [Fu and Yu, 1980]. A less statistical, but perhaps more manageable, approach to using the global contextual information present in an image for segmentation purposes can be found in the multiresolution methods.

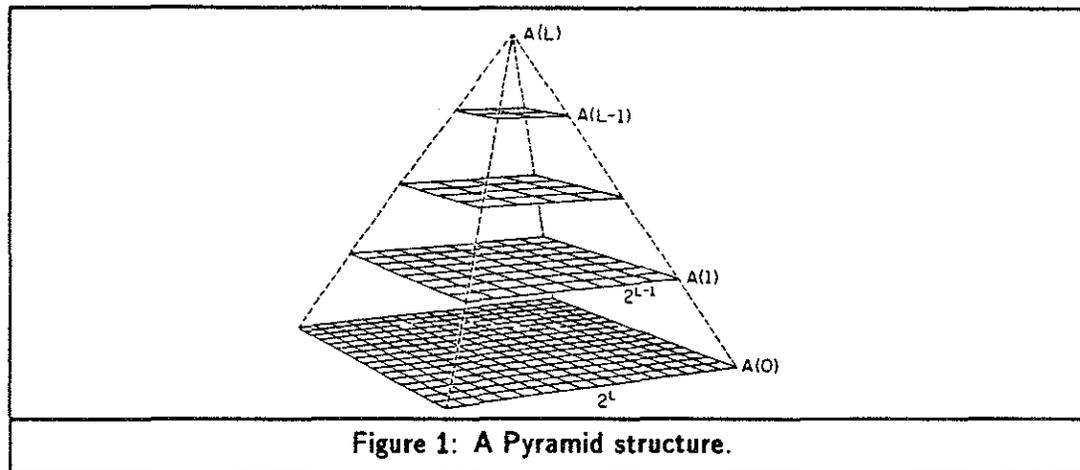
4 Multiresolution Approaches to Image Segmentation

Multiresolution techniques attempt to gain a global view of an image by examining it at many different resolution levels. The lower resolution provides a global view of the image, and the higher resolution provides the detail.

There have been several different attempts to segment images based upon a multiresolution approach. Two of the most well known examples of this are the pyramid approach of Burt and Hong [Burt *et al.*, 1981] and the stack approach of Koenderink. These are summarized below, along with a few other relevant approaches.

4.1 Details of the Pyramid Approach

The standard pyramid has the original, or "ground plane", image averaged so as to produce an image one-half the length and width of the original. This new image has one-quarter the number of pixels as the original, each pixel being the average of four pixels. The same process is applied to the new image to create a third image one-quarter its size, and so on. The result is a pyramid of images (see Figure 1, from [Rosenfeld, 1984]). The higher up one goes in the pyramid the smaller the image and the lower the resolution. The lower resolution image provides a more global view of the image, and the higher resolution image provides the detail. Pixels at each level of resolution are linked to father pixels in the image above them. Each pixel is given the option of linking to one of four possible father pixels. Thus a father can have anywhere from zero to 16 sons (with an average of four sons since there are four times as many son pixels). Exact linking schemes vary [Hong *et al.*, 1982]. Typically a pixel links to the prospective parent most similar to it in grey level. The result is a tree structure of linked pixels. Each pixel is at the root of a



subtree. By following the links in the tree all the way to the leaves, one determines which pixels in the ground plane belong to this subtree.

Better blurring and thus linking is achieved by recomputing the grey levels for all non-leaf pixels by averaging only over their ground plane descendants and then relinking. This averaging subtree may be thought of as a segment or part of a segment. The whole blurring and linking procedure is iterated. At each iteration the decision as to which pixel to link to is made based upon the *new* grey levels associated with each pixel. The iteration process is halted once the links have stabilized.

Segmentation is performed by noting which leaf nodes (ground plane pixels) are associated with each of the highest level (lowest resolution) root nodes (the number of which is usually specified in advance). The literature [Rosenfeld, 1984, for example] gives many details and variations of the pyramid approach.

The advantages of the pyramid approach are summarized below.

1. By linking up pixels between images at different resolutions based upon closeness in intensity, it is able to involve global context in segmentation.
2. By iterating the blurring and linking process, the new global information can be incorporated into the next iteration of the process via variable shape blurring within segments.
3. Adjusting the linking scheme (probabilities, weights, etc) allows one to tailor the approach somewhat.
4. Disjoint regions can be considered as one logical entity (this may also be considered a disadvantage).

There are some deficiencies in the Burt and Hong approach:

1. The basis of segmentation, which is merely following down a tree structure from a particular level once convergence has been reached, is ad hoc. This makes comprehension of why it works difficult, and extension of the technique also becomes uncertain.
2. The segmentation is often based on only a single level of resolution (all root nodes are at the same level) despite the fact that different objects are at different scales. Some versions of the pyramid algorithm address this problem by allowing root nodes to exist at other than the lowest resolution level. This is done by designating any node whose intensity is too different from all of its potential father nodes as a root node.
3. The amount of blurring between levels of the pyramid is greater than considerations of error generation would suggest. Thus information can be lost that can result in incorrect linking and thus incorrect segmentation.
4. The type of blurring, non-Gaussian, theoretically can produce artifacts [Yuille and Poggio, 1986].
5. There is no attempt to describe the *structure* of the set of blurred images.
6. There is no obvious way to incorporate a priori knowledge about the image structure into the process (with the exception of the ability to specify the number of segments and slightly change connectivity by changing weighting functions for the links).

These deficiencies are addressed by Koenderink's stack approach.

4.2 The Difference of Low-Pass Transform

Crowley and Parker [Crowley and Parker, 1984] create a graph (with some tree-like characteristics which aid in matching and picking correct resolutions) from a sequence of different resolution images. Each image in the sequence is a band-passed version of the original image, with those later in the sequence having a lower high pass cutoff than those earlier in the sequence. The actual implementation is done using a difference of low pass images with slightly different cutoff frequencies. This permits an efficient implementation of the filtering process. The sequence of pass-band filters used creates some overlap between sequential filters to preserve some continuity between image features at each resolution. This is equivalent to creating a pyramid of Laplacian images.

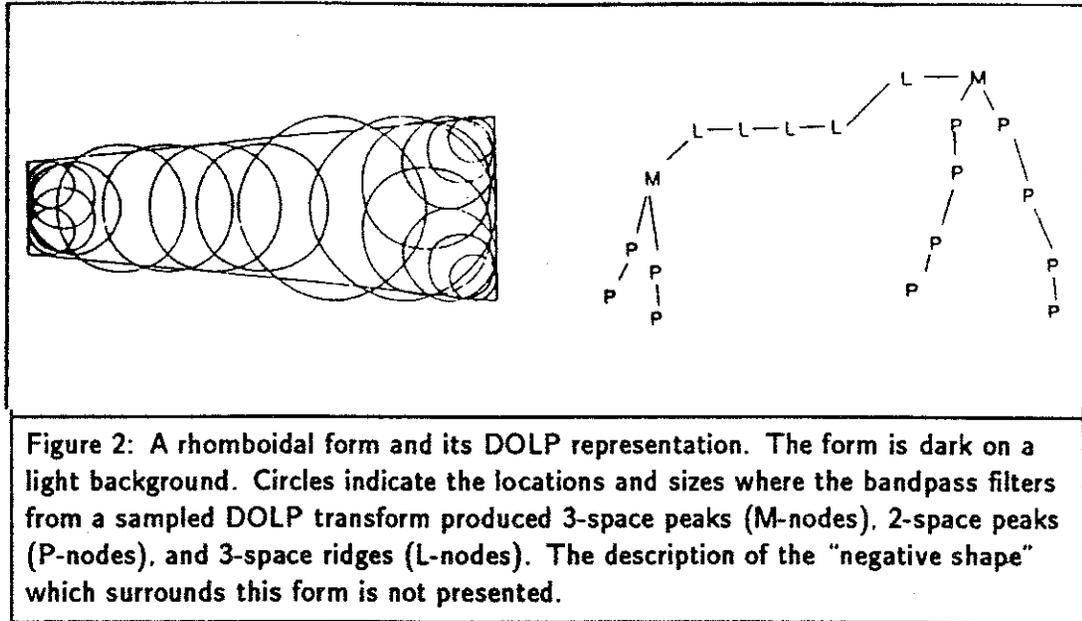
Creation of the graph from the sequence of images occurs in two phases. The first phase involves examining each image in the resolution sequence independently. The second phase examines pixels in the context of the three-dimensional space-resolution domain. Each image in the sequence is first scanned individually to identify "peaks" and "ridges". A peak is either a local maximum or a local minimum. Ridges are pixels which are a

maximum or minimum in at least one of the four possible directions, i.e., parallel to the x axis, the y axis, the $x = y$ axis or the $x = -y$ axis. A peak is called a P node, a ridge an R node. Ridges are linked to other similarly oriented ridges on the same level. This creates a ridge path between P nodes on a level. Pixels are then examined in a three-dimensional context (two spatial dimensions and the resolution dimension— they call this space the DOLP space). P nodes are connected to nearby P nodes in other levels. Following this, peaks and ridges in the DOLP space are identified. Peaks in this space are called M nodes; ridges in this space are called L nodes. A peak in the DOLP space is a local maximum in the three-dimensional space. M nodes are connected via L nodes. They claim that if a path of peak nodes is followed through DOLP space the intensity of the nodes will decrease (increase) and then increase (decrease), so there will be an M node along the path. M nodes are considered to be significant features which would play a prominent role in any possible matching strategy. The nature of the graph produced is such that

1. regions which produce a peak at one resolution can produce more than one peak at another resolution (i.e., artifacts can be created by the filtering process).
2. there is no guarantee that each peak corresponds to only one physical object or that a particular physical object will result in a single peak [Crowley and Parker, 1984; p 158]

The configuration (but not the metric information associated with the nodes) which describes a shape is invariant to the size, orientation, and position of the shape (neglecting effects due to quantization). Specific information regarding the degree to which quantization effects might affect this invariance was not provided.

The approach taken appears to be very promising. It is still too early to tell how difficult it will be to perform matching on the graph structure in day to day practice. The graph has a tree-like main skeleton which simplifies matching it to a template; while the rest of the graph structure allows for a rich representation of the global interactions. The ability of a simple object to produce many nodes (e.g., the bar example, see Figure 2, from [Crowley and Parker, 1984]) seems to be a drawback and certainly does not aid in the intuitive comprehension of how objects relate to their graph representation. Of course, if one works with these representations long enough, intuition may develop for how the two correspond.



Probabilistic Matching of DOLP Trees. Crowley and Sanderson [Crowley and Sanderson, 1984] use a tree matching technique in dealing with the trees produced by their DOLP transform. The method developed in this dissertation is designed to permit the use of their approach to tree matching and tree modeling, at least in the general concepts employed.

Each node in their prototype tree has a list of quantitative attributes. Each attribute has both an expected value and a standard deviation associated with it. It is assumed that the attribute comes from a Gaussian distribution with these parameters. To simplify the calculations, it is further assumed that all the attributes are independent of each other. Each prototype node is also assigned probability of occurrence. The probability that any particular candidate node P_j is actually an instance of a specific prototype node π_n can then be represented as $P(P_j | \pi_n)$. Bayes' law can be used to find the a posteriori probability of any particular match between two nodes

$$P(\pi_n | P_j) = \frac{P(P_j | \pi_n)P(\pi_n)}{P(P_j)}$$

In this expression $P(\pi_n)$ is the a priori probability of prototype node π_n existing. To reflect maximal uncertainty about $P(\pi_n)$, it is assumed to be the same for all prototype nodes. We are interested in picking the prototype node which maximizes the a posteriori probability. The denominator does not affect which π_n will yield the maximum a posteriori

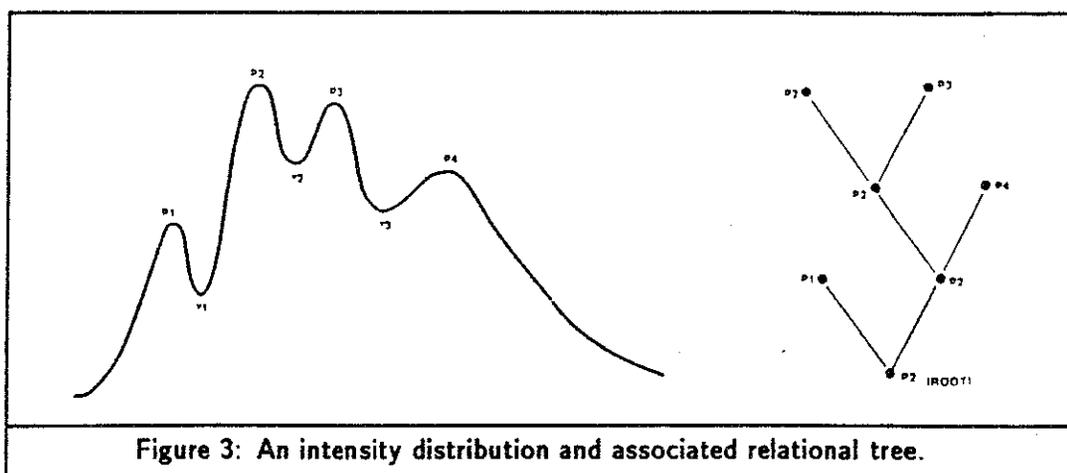
probability because it is not a function of π_n . Since we also assume independence of node attributes, $P(P_j | \pi_n)$ is just a product of Gaussians, over the attributes of the node. We can then take the logarithm of the product. Since the log is a monotonic function, we try to maximize the *log likelihood* that the observed node P_j is an instance of a prototype π_n . Crowley denotes this by $L_{j,n}$. This measure will always be negative, unless an exact match occurs, in which case it will equal zero.

It still remains to be decided whether one prototype node will be allowed to match only one candidate or several. This will depend upon the observed characteristics of our data and the sophistication of our global matching algorithm (how we match all the candidate nodes to prototype nodes). Noisy data or a matching algorithm that is not flexible enough will mean that we may match a prototype to a candidate node incorrectly and throw off all further attempts at a match for the remaining nodes. Crowley tries both approaches (i.e., matching only one node or several nodes), finding that the second is more robust.

Crowley does not search the entire state space for a global maximum; instead he uses a faster "greedy" heuristic. After matching the roots of the two trees, he examines all nodes at the next level. If there are N such nodes in the prototype and M in the candidate, then there are MN possible binary combinations (matches). The log-likelihood for each such match is calculated and a sorted list of triples $(P_m, \pi_n, L_{m,n})$ is created which is sorted with the more likely triples ahead of the less likely triples. All nodes in both trees are initially unmatched. The list is then traversed. Any time a triple is encountered which contains both an unmatched candidate node and an unmatched prototype node, the two nodes are matched to each other. Otherwise no matching occurs and the next triple is examined.

The children of a node are then matched to the children of the node it was matched to. Crowley's nodes always have a small number of children, so at this point he examines all possible combinations of matches and picks the best one. The process continues until all the nodes in the prototype tree are matched. I have omitted details relating to how one decides if the match does not meet minimal requirements and is rejected and how one normalizes to compensate for the number of nodes matched.

An additional method Crowley uses to minimize the amount of error in a match which might be produced by an object in one image being displaced slightly from its location in the other is to have many attributes of a node stored as information *relative* to



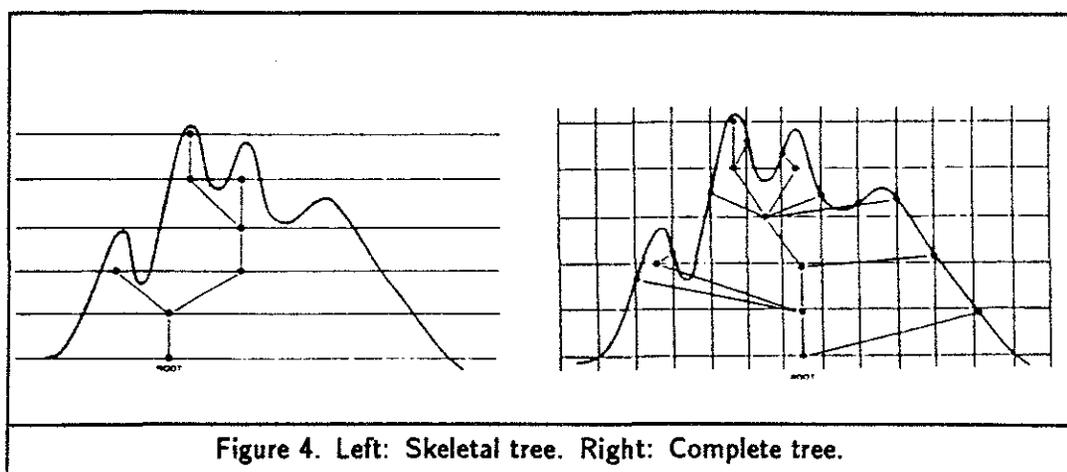
the parent node. So position, perhaps, would be stored as an offset from a parent node's position rather than as absolute coordinate values.

Crowley creates his prototype trees by using a training set of objects known to be of a certain type. Human intervention is required at times to guide the process.

4.3 Cheng and Lu

Cheng and Lu [Cheng and Lu, 1985] present a tree representation for waveforms and a technique for matching those trees. Many aspects of it are similar in spirit to the multiresolution trees that are produced by the stack method (when applied to one-dimensional images). Their domain of application is stratigraphic interpretation of seismic or well logs. They are interested in correlating two waveforms in the presence of stretching, shrinking, and missing intervals between them.

They examine three tree representations for a waveform, each containing more information than the previous one. The relational tree is formed as follows (see Figure 3, from [Cheng and Lu, 1985]). A root node is created. It is associated with the highest peak in the waveform. The waveform is then split at its lowest valley. A node is created for each subwaveform (so the root now has two children, each associated with the highest peak in its section of the waveform). Each of these can be considered a root of one subwaveform. The process is continued until no valleys remain in any subwaveform. This tree contains no information about exact heights of the peaks or about widths. This information could be added to the nodes, but that would make the matching routine more difficult.



A more informative tree is the skeletal tree (see Figure 4). It contains information about the heights of the peaks by quantizing the intensity levels in the waveform. Instead of simply associating an intensity with the node, a chain (grandparent \Rightarrow parent \Rightarrow son \Rightarrow ...) of nodes of length proportional to the intensity of the peak node is created. This is similar to the tree produced by the stack algorithm in the sense that an extremum which exists for a long time before annihilation will also be represented by a long chain of extremum nodes (the extremum path). The skeletal tree can be expanded into a complete tree (see Figure 4) by also quantizing horizontal distance and adding nodes to represent the width of peaks. One advantage of the complete tree representation is that waveform point-to-point correlation can be achieved within a peak or valley (instead of just indicating that some whole peak matches some other whole peak).

The matching algorithm used does not need to handle labeled nodes or arcs. The match is based upon the minimum number of father-son node splits or merges and brother-brother splits or merges necessary to form the match. This allows a simple interpretation of the difference between two waveforms (e.g., a father-son node split means that the peak on one waveform is a little higher than the peak on the other waveform). Under certain assumptions they claim that the tree matching algorithm has time complexity of $O(n^3)$, where n is the number of tree nodes. They present several matched waveforms as examples. The matching of two trees of approximately 150 nodes each took about 25 minutes on a VAX 11/780.

The splitting of the waveform into subwaveforms is similar in philosophy (although very different in approach) to the stack approach of looking for nested regions. The

discretizing of intensity levels to create multiple nodes is similar to the creation of stack levels. It remains to be seen if Lu's method could be extended to two dimensions because it is not clear what the appropriate method for splitting a two-dimensional wave form would be. It is also not clear if the matching algorithm is flexible enough to be applied successfully in the two-dimensional case.

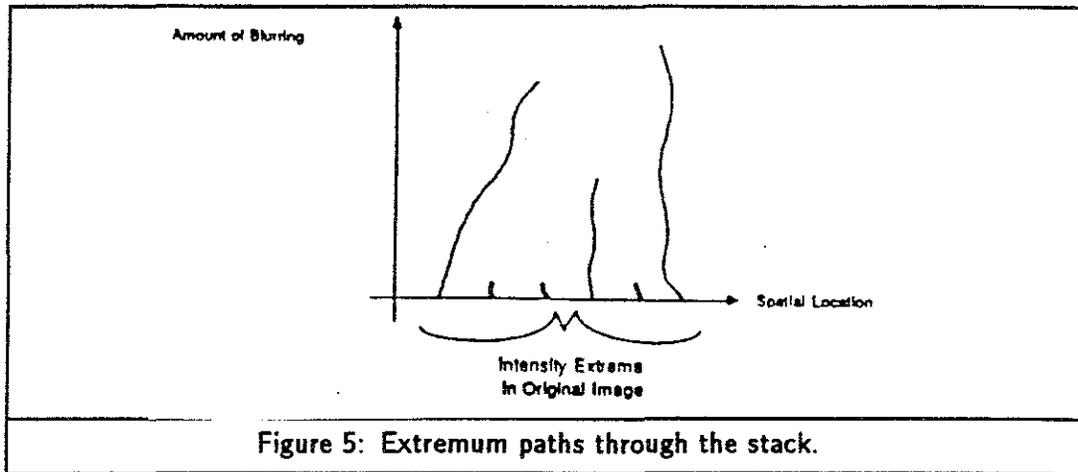
4.4 The Stack

An attractive image description which also examines images at multiple resolutions and emphasizes the role of extrema is that of Koenderink [Koenderink, 1984]. This approach focuses on decomposing the image into light and dark spots, each, except for the spot representing the whole image, contained in others. Thus a face might be described as a light spot containing a light spot (a reflection from the forehead) and three dark spots (the mouth and the regions of the two eyes). In turn the eye regions would be described as containing a dark spot (the eyebrow), a light spot (the eyelid), and a dark spot (the eye), with the latter containing a light spot (the eyeball) which itself contains a dark spot (the iris) which finally contains a yet darker spot (the pupil). We call these light and dark spots, at whatever scale, *extremal regions*, since they each include a local intensity maximum or minimum.

I choose to use the stack approach of Koenderink [Toet *et al.*, 1983; Koenderink, 1984] as the basis for producing an image description because I believe that it has several strengths:

1. it is based on considering the image simultaneously at multiple levels of resolution and can therefore use global information
2. it produces a natural tree structured extremal region description useful for matching purposes
3. it has a firm mathematical description giving investigation and extension a firm footing.

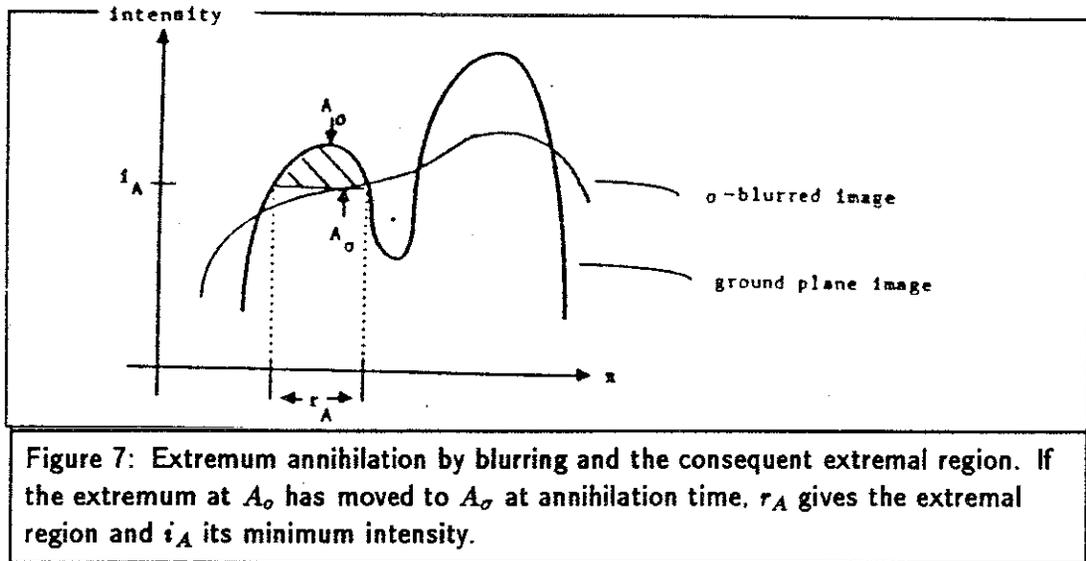
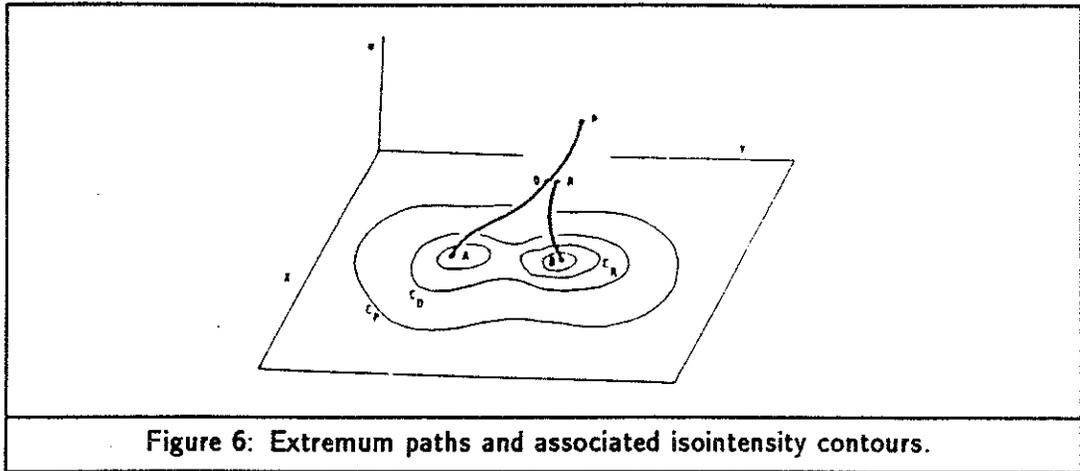
By using Gaussian blurring from level to level, creation of artificial objects by blurring can be minimized [Babaud *et al.*, 1986; Yuille and Poggio, 1986; Koenderink, 1984] and the stack of images can be modeled precisely via the heat diffusion equation [Koenderink, 1984]. The image changes between stack levels can be described with mathematical care using differential geometry and Morse theory [Poston and Stewart, 1978]. This yields a precise way to recognize a coherent region (an "object piece") as an extremal region, the levels of resolution (scale) at which an object exists, and the object containment relations. A more detailed exposition follows.



Hierarchical Descriptions from Multiresolution Processing. The image description in terms of extremal regions can be produced by following the paths of extrema in a stack of images in which each higher image is a slightly blurred version of the previous one. As illustrated in Figure 5 and explained in Koenderink [Koenderink, 1984], progressively blurring an image causes each extremum to move continuously and eventually to annihilate as it blurs into its background. An *extremum path* is formed by following the locations of an extremum across the stack of images.

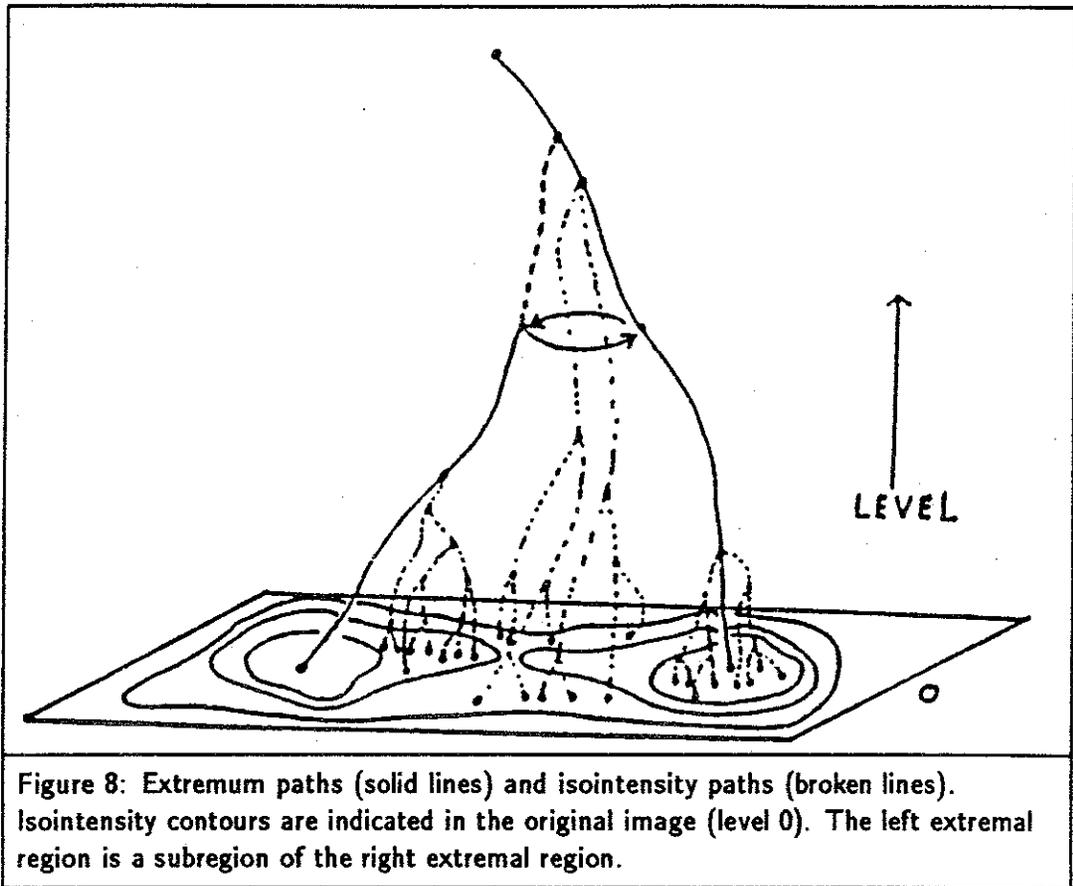
Intensity change must be monotonic (increasing for dark spots and decreasing for light spots) as one moves along an extremum path from the original image towards images of increased blurring. As illustrated in Figure 6, while following each extremum path one can associate each path point with the iso-intensity contour that is at that point's intensity and that surrounds that extremum in the original image [Koenderink, 1984]. The points (pixels) in the original image thus associated with each extremum path then form an extremal region (see Figure 7). Equivalently, each contour (non-extremum) point in the original image can be associated with its extremum path by linking the point to the closest point with its intensity level at the next level in the stack and continuing this linking through the levels until the extremum path is reached (see Figure 8). This process defines an *iso-intensity path*.

As indicated above, extrema annihilate when the blurring is sufficient to make the light or dark spot blur into an enclosing region. The amount of blurring necessary for an extremum to annihilate is a measure of the importance or scale of the extremal region, including the subregions that it contains. The intensity of the topmost point on



an extremum path is its annihilation intensity. This is the intensity of the isointensity contour that forms the boundary of the associated extremal region. The annihilation intensity bounds from below (above) the intensities in the extremal region if the associated extremum is a maximum (minimum) and if there are no extremal sub-regions enclosed.

A Tree of Extremal Regions for Image Description. As illustrated in Figure 6, when an extremum annihilates at some annihilation intensity, another region's isointensity contour at that intensity encloses the region associated with the annihilating extremum [Koenderink, 1984]. Thus, a containment relation among extremal regions is induced by the process. This set of extremal regions together with their containment relations can



be represented by an *extremal region tree* in which nodes represent extremal regions and a node is the child of another if the extremal region that it represents is immediately contained by the extremal region represented by the parent (see Figure 9). The root of the description tree represents the entire image.

Each node in the extremal region tree can be labeled with its scale, i.e. the total amount of blurring necessary for its extremum to annihilate. Furthermore, each node can be labeled with the annihilation intensity of the associated extremum. Finally, the node can be labeled with its size, shape, orientation, location, or other spatial characteristics.

It is possible that the description process described above can be beneficially preceded by some preprocessing, e.g. to enhance contrast or edges. In fact, Crowley's [Crowley and Parker, 1984] approach can be thought of as extremum following in a multiresolution pile of images that are edge-enhanced by a type of unsharp masking. We have tried such preprocessing a few times with some benefit but will not discuss it in greater detail

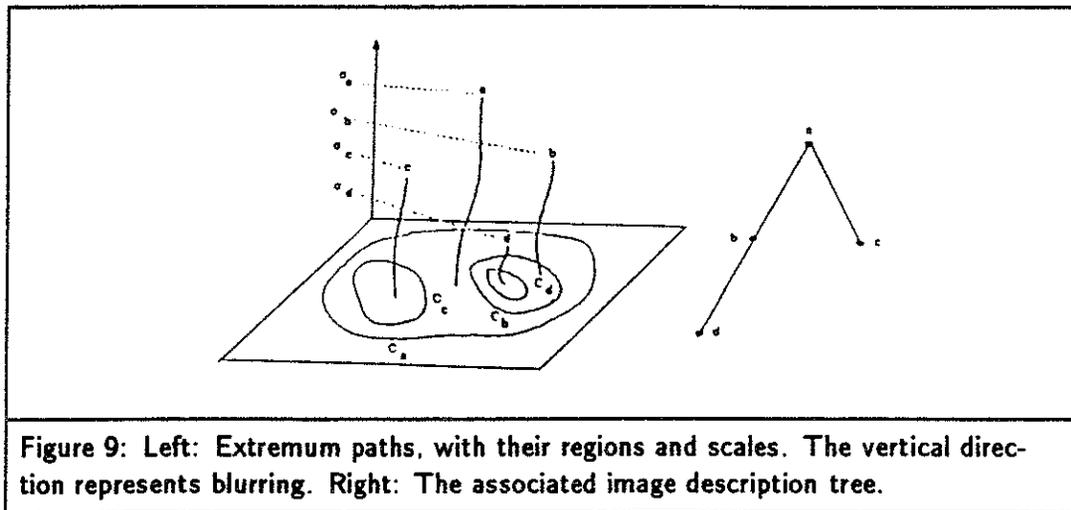


Figure 9: Left: Extremum paths, with their regions and scales. The vertical direction represents blurring. Right: The associated image description tree.

in this paper. However, it is worth noting that a noncognitive component of human visual perception may possibly be well modeled by an edge-enhancing preprocessing followed by the production of a stack-based image description.

4.5 *Helmink's Work*

Helmink [Helmink, 1984] has implemented a discrete approximation to the basic stack approach. Although the basic design is strong, the implementation does not approximate closely enough the desired linking strategy dictated by the continuous theory to lead to any definitive conclusions about the strengths and weaknesses of the stack. Shortcomings of his approach are as follows.

1. Any extremum which gets created at lower resolutions is not explicitly followed and can therefore never have an extremal region associated with it [Helmink, 1984; p 39].
2. Boundary values are merely duplicated outside the image in directions perpendicular to the boundary (e.g., all pixels at $x > xdim, 1 \leq y \leq ydim$ are given the value of the boundary pixel $(xdim, y)$).
3. If a non-extremum pixel's intensity does not fall between those of the pixels in the neighborhood directly above it, hill climbing (pit sliding) is performed until a local maximum (minimum) is reached. This is not the linking strategy which the

continuous description of the algorithm describes and can result in an isointensity path linking to the wrong extremum path.

4. "Isointensity" paths take on the intensity of the pixel they link to instead of keeping their initial intensity.
5. Paths which pass less than a pixel from each other automatically become linked together.

Chapter 3

Mathematical Properties of the Basic Stack

Several of the nice properties associated with the stack approach stem from basic results in Morse theory. One such result is that the number of minima, maxima, and saddle points cannot change (at least in the “typical” case) except through a bifurcation which causes an extremum and a saddle point to appear (or disappear). Similarly, the characteristic way in which the bifurcation occurs can be studied by examining simple “generic” cases. Morse theory precisely defines the meaning of the terms “typical” and “generic”. Some of the basic principles of Morse theory are presented below, along with a rough sketch of how some of them are derived. Differences between the canonical cases discussed in the derivation and our particular case (which has the additional restriction of having to satisfy the heat equation) have some important theoretical implications which will be presented following the canonical exposition.

1 Morse Theory Basics for the Generic Case

The following presentation of Morse theory follows Poston [Poston and Stewart, 1978]. Page references are to this book. The exposition presents in a cursory fashion some of the central theorems of Morse theory. These theorems are used to help analyze the qualitative nature of functions and families of functions. They allow us to study many seemingly different functions by instead looking at the behavior of only a handful of generic cases. With these results we will be able to examine easily all of the typical ways in which extrema can interact (appear and disappear).

Many of the theorems stated pertain to all functions related to each other by a particular class of coordinate transformations. This class of coordinate transformations usually satisfies constraints that the transformations be *smooth* and *reversible*. Technically, the change is a *diffeomorphism* [Poston and Stewart, 1978; pp 47-50]. By smooth we mean

the function possesses derivatives of arbitrary order in the entire domain. Specifically, let U, V be open sets in \mathbf{R}^n . Suppose $f(U) = V$. Then f is a diffeomorphism, provided

- a. f is smooth
- b. f has an inverse function $g : V \rightarrow \mathbf{R}^n$ s.t. $f \circ g = 1_V, g \circ f = 1_U$
- c. g is smooth

A *local diffeomorphism* at a point x is a mapping from an open set containing x which is a diffeomorphism onto its image.

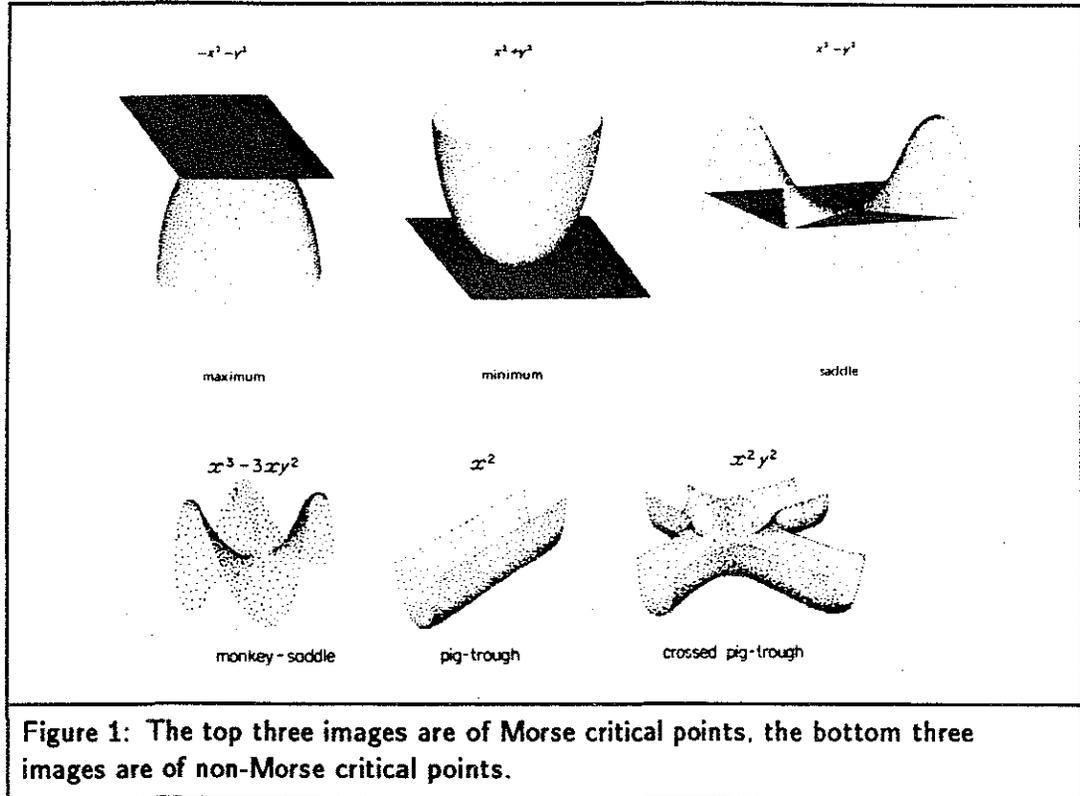
The **Inverse Function Theorem** yields a simple method for testing a mapping to see if it is a diffeomorphism. Specifically, let $f : U \rightarrow \mathbf{R}^n$ be smooth, and let $x \in U$. If the linear map $Df|_x$ is non-singular, then f is a local diffeomorphism at x . Note that $Df|_x$ is non-singular if and only if the Jacobian determinant $Jf|_x = \det Df|_x \neq 0$.

Two smooth functions $f, g : \mathbf{R}^n \rightarrow \mathbf{R}$ are said to be *equivalent* around the origin if there is a local diffeomorphism $y : \mathbf{R}^n \rightarrow \mathbf{R}^n$ around the origin and a constant γ such that, around the origin, $g(x) = f(y(x)) + \gamma$. Then y is a smooth reversible change of coordinates, and γ adjusts the value of the function at the origin, taking care of various translations of the origin (p. 58-59).

1.1 The Morse and Splitting Lemmas

Let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ be a smooth function. A point $u \in \mathbf{R}^n$ is a critical point iff $\frac{\partial f}{\partial x_i}|_u = 0, \quad \forall i \leq n$. If $n = 1$, the only such points are called minima, maxima, or inflection points. In higher dimensions the situation is not quite so simple. If $n = 2$ we have *maxima* ($z = -x^2 - y^2$), *minima* ($z = x^2 + y^2$), *saddle points* ($z = x^2 - y^2$), and others such as a *monkey saddle* ($z = x^3 - 3xy^2$), *pig-trough* ($z = x^2$), and *crossed pig-trough* ($z = x^2y^2$)— see Figure 1 (from Poston [Poston and Stewart, 1978]).

We say f has a *non-degenerate* critical point at u if the Hessian matrix $Hf|_u = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]_u$ is non-singular (i.e., $\det(Hf|_n) \neq 0$). The last three examples all have degenerate critical points at the origin. The last two do not even have *isolated* critical points; there is no sufficiently small region around a critical point which does not contain another critical point. Non-degenerate critical points are always isolated (but not the converse). A non-degenerate critical point is a *Morse* critical point. The **Morse Lemma** is the following (p. 54):



Let u be a non-degenerate critical point of the smooth function $f: \mathbf{R}^n \rightarrow \mathbf{R}$. Then there is a local coordinate system (y_1, \dots, y_n) in a neighborhood U of u , with $(\forall i) y_i(u) = 0$, $(\forall u \in U) (\exists k) \text{ s.t. } f = f(u) - y_1^2 - \dots - y_k^2 + y_{k+1}^2 + \dots + y_n^2$.

The quadratic terms above constitute a function called a Morse k -saddle. This means that every non-degenerate critical point can be transformed via a diffeomorphism to a Morse k -saddle for some k . When $k = n$, we have a maximum, and when $k = 0$, we have a minimum.

By straightforward application of the Implicit Function Theorem (which is simply derived from the Inverse Function Theorem), the Morse Lemma can be extended to degenerate critical points. This is the **Splitting Lemma**:

Let $f: \mathbf{R}^n \rightarrow \mathbf{R}$ be a smooth function whose Hessian at 0 has rank r . Then f is equivalent, around 0, to a function of the form $\pm y_1^2 \pm y_2^2 \pm \dots \pm y_r^2 + \hat{f}(y_{r+1}, \dots, y_n)$ where $\hat{f}: \mathbf{R}^{n-r} \rightarrow \mathbf{R}$ is smooth and can be thought of as the degenerate portion of f .

This means that the study of the unique characteristics of a function near a degenerate critical point can be reduced to the study of a function (\hat{f}) of only $n - r$ variables.

It is possible to prove (pp. 65-71) that a critical point is *structurally stable* if and only if it is non-degenerate. A function f is structurally stable if, for all sufficiently small smooth functions p , the critical points of f and $f + p$ are of the same type (and number); or in other words if f and $f + p$ are equivalent after a suitable translation of the origin (p. 62). For instance: $f(x) = x^2$ has a Morse critical point at $x = 0$, and no others. $\hat{f}(x) = x^2 + 2\epsilon x$ has a similar point at $x = -\epsilon$ and no others. In contrast, if $f(x) = x^3$, there exists one critical point, a degenerate one at $x = 0$. But $\hat{f}(x) = x^3 + \epsilon x$ has *no* critical points if $\epsilon > 0$ and *two* critical points if $\epsilon < 0$.

Lemmas for Families of Functions. Now let us turn our attention from a single function to a parameterized *family* of functions. For instance, $y(x, a) = x^2 + a$ is a family of functions, one function for each value of the parameter a . The set of images produced by the stack multiresolution segmentation algorithm can be interpreted as a family of functions. Each image is a function of two spatial variables, x and y . There is a different image or function for each value of the parameter which specifies spatial resolution. Hence, an understanding of the properties of families of functions should yield insight into how an image may change as its spatial resolution changes. We denote a family of functions formally as $f: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}$ for an n variable, r parameter family. For instance, $f(x, \epsilon) = x^3 + \epsilon x$ is a family of functions with one spatial variable (x) and one parameter (ϵ). The ultimate aim is to classify these families of functions in such a way that the critical point structure is not affected qualitatively by suitable coordinate changes. The following extensions of the Splitting and Morse Theorems help us along the way.

In order to extend the theorems, the concept of equivalence of functions must first be expanded to encompass equivalence of families of functions. When dealing with functions, a transformation was applied from the spatial coordinates of one to the spatial coordinates of the other. There are more coordinate transformations to deal with when a parameter space exists also. For each specific value (\mathbf{s}) from the parameter space, there is a transformation ($y_{\mathbf{s}}(\mathbf{x})$) from the spatial coordinates of the first family to the spatial coordinates of the second. There is also a mapping from the parameter space of one function (\mathbf{s}) to the parameter space of the other ($e(\mathbf{s})$). Finally, for each specific parameter value (\mathbf{s}) we allow a translation to be added ($\gamma(\mathbf{s})$). Thus, the complete mapping is defined as

follows (pp.90-91). f and g are equivalent families of functions if there exist functions e, y, γ defined in a neighborhood of $x=0$, such that $g(x, s) = f(y_s(x), e(s)) + \gamma(s)$ for all $(x, s) \in \mathbf{R}^n \times \mathbf{R}^r$ in that neighborhood, where e is a diffeomorphism $e: \mathbf{R}^r \rightarrow \mathbf{R}^r$; y is a smooth map $y: \mathbf{R}^n \times \mathbf{R}^r \rightarrow \mathbf{R}^n$ such that for each $s \in \mathbf{R}^r$ the map $y_s: \mathbf{R}^n \rightarrow \mathbf{R}^n$ with $y_s(x) \stackrel{\text{def}}{=} y(x, s)$ is a diffeomorphism; γ is a smooth map $\gamma: \mathbf{R}^r \rightarrow \mathbf{R}$.

We can now state the **Splitting Theorem for Families** (p. 95-96):

Let $F: \mathbf{R}^N \times \mathbf{R}^r \rightarrow \mathbf{R}$ be smooth. Denote a point in $\mathbf{R}^N \times \mathbf{R}^r$ by $(x, c) = (x_1, \dots, x_N, c_1, \dots, c_r)$. Suppose that the Hessian $H = [\frac{\partial^2 F}{\partial x_i \partial x_j}]_{1 \leq i, j \leq N}$ has co-rank (N minus the rank) m at $(x, c) = 0$. Then F is equivalent to a family of the form

$$\tilde{F}(y_1(x, c), \dots, y_m(x, c), c) \pm y_{m+1}^2 \pm \dots \pm y_N^2$$

where y_1 through y_m are called the *essential variables*.

The **Morse Lemma for Families** follows directly (p.97):

Let $F: \mathbf{R}^N \times \mathbf{R}^r \rightarrow \mathbf{R}$ be smooth. Suppose that the Hessian $[\frac{\partial^2 F}{\partial x_i \partial x_j}]_{1 \leq i, j \leq N}$ is non-degenerate at $(x, c) = 0$. Then F is equivalent to a family of the form

$$\pm y_1^2 \pm \dots \pm y_N^2.$$

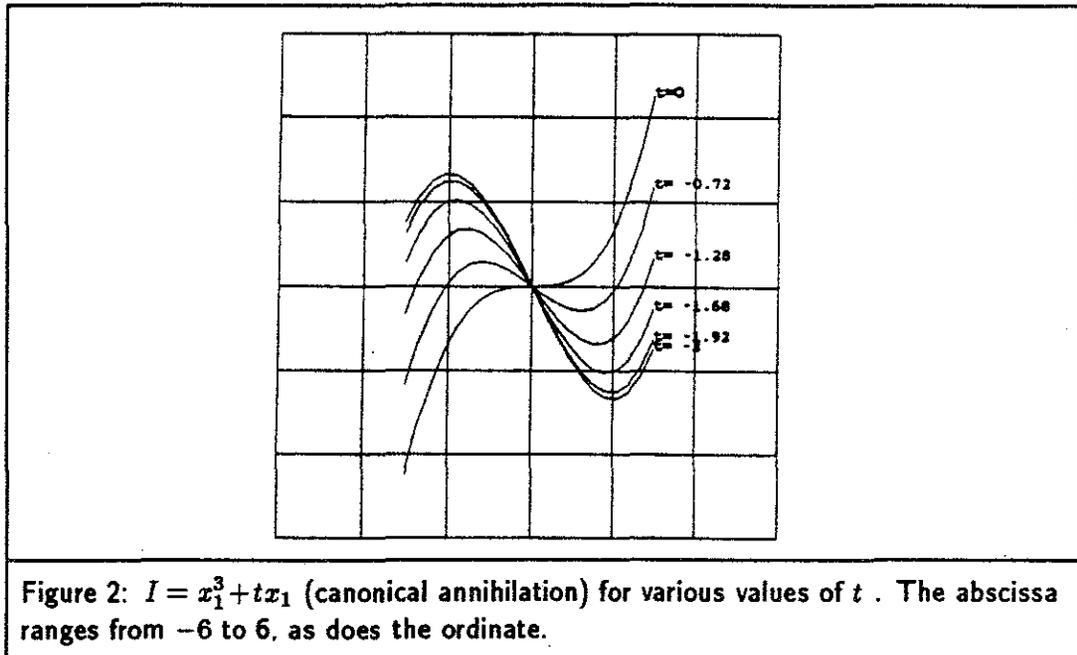
This tells us that not only does a small perturbation of a Morse function $f: \mathbf{R}^N \rightarrow \mathbf{R}$ make no qualitative difference but that any smooth *family* of perturbations $F: \mathbf{R}^N \times \mathbf{R}^r \rightarrow \mathbf{R}$, where $F|_{\mathbf{R}^N \times \{0\}}$ is just f , can also be reparameterized away (i.e., each member of the whole family still "looks like" $\pm y_1^2 \pm \dots \pm y_N^2$), yielding simply a Morse function. We can therefore conclude that *creation or annihilation of critical points does not happen as long as the critical points remain Morse* (p. 98). The collection of critical points can change *only* by one or more of them becoming non-Morse. For a two dimensional image this means that maxima, minima, and saddle points can only appear or annihilate during the blurring process by first becoming degenerate. Since this is the case, *we will focus on trying to catalog the types of degenerate points possible and how they appear and disappear.*

Typically (in a rigorous sense) a one parameter family of functions will have non-critical points, Morse-critical points, and degenerate points. The degenerate points typically will have a Hessian of corank = 1 (i.e., although all the first derivatives are zero only one dimension out of all the possible dimensions has a zero second derivative). So, although they are degenerate they have the minimum amount of degeneracy possible.

Around each of these degenerate points we can express the function (by the Splitting Theorem for Families) by $f(x_1, x_2, \dots, x_n; t) = \tilde{f}(x_1, t) \pm x_2^2 \pm \dots \pm x_n^2$. $\tilde{f}(x_1, t)$ is the degenerate portion of f . It contains just one spatial dimension since degenerate points typically have a Hessian of corank = 1. We study the behavior of f by examining $\tilde{f}(x_1, t)$. Fix t temporarily. Let $g_a(x) = \tilde{f}(x_1, t = a)$, with $x = x_1$. Now expand g_a in a Taylor series about 0: $g_a(x) \approx g_a(0) + \frac{\partial}{\partial x} g_a(0)x + \frac{\partial^2}{\partial x^2} g_a(0) \frac{x^2}{2!} + \frac{\partial^3}{\partial x^3} g_a(0) \frac{x^3}{3!} + \dots$. Here $g_a(0)$ is a constant that we don't need to be concerned with. Now $g' = 0$ (since we assumed g was the degenerate portion of f), while $g''' \neq 0$. Therefore $g_a(x) \approx kg'''x^3 + \text{higher order terms}$. There is a theorem (pp. 57-59) which says any *smooth* one-variable function has its behavior determined *locally* by its first nonzero term. Hence $g_a(x)$ is of the form x^3 .

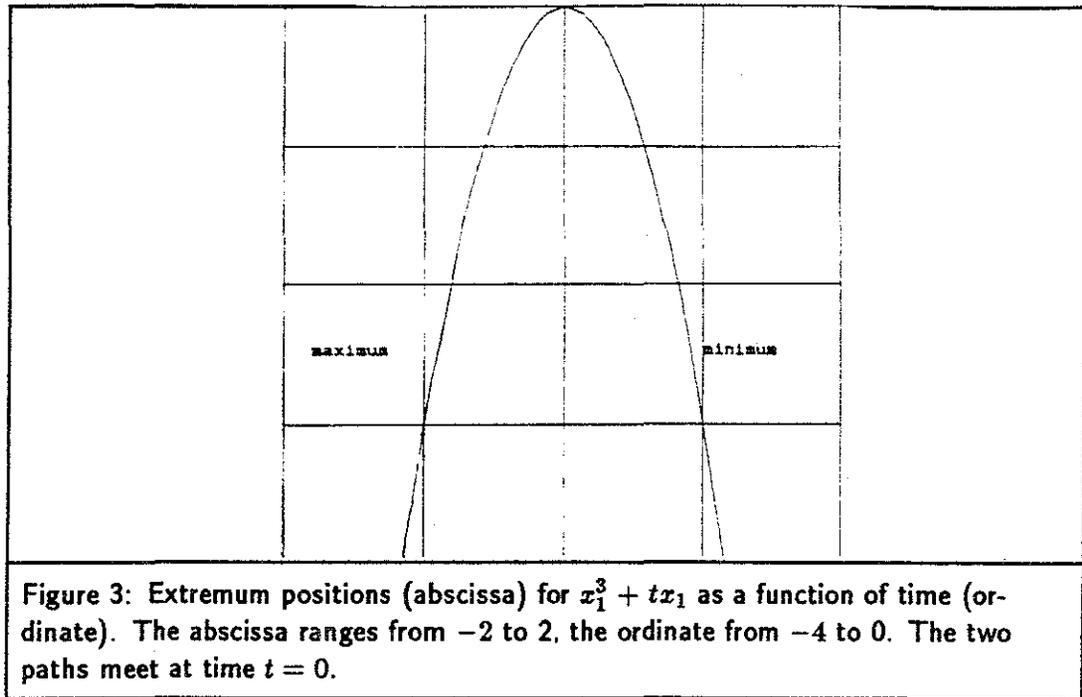
We now want to reintroduce t into the formula to "unfold" the function back into a *family* of functions. Once this is done, we will have produced a simplified degenerate family of functions with only one spatial dimension. The Splitting Theorem for Families tells us that by studying the behavior of this simpler function we can learn about the behavior of the more complex function f . But does it make a difference how t is reintroduced into g to produce the one parameter family? At first glance it seems as though the qualitative nature of the family could be greatly affected by the specific choice made. Luckily this is not the case, and most choices will result in qualitatively similar behavior. This is result of what is called the "equivalence of universal unfoldings" which means that "any transverse way through any non-Morse function (e.g., x^3) which can be met transversely by a finite-dimensional family, looks like any other" (p. 105). Readers who wish to know precisely what a "transverse way" is can read the reference; for our purposes it is enough just to know that *one* transverse way through x^3 is given by $f_t(x) = x^3 + tx$. This transverse unfolding can be transformed into any other unfolding via a mapping very similar to those allowed under the definition of equivalence of families of functions. So we can now state $f(x_1, \dots, x_n; t) = x_1^3 + tx_1 \pm x_2^2 \pm \dots \pm x_n^2$. See Figure 2 for graphs of $x_1^3 + tx_1$ at several values of t .

Let us recapitulate where we have come from and what we have concluded. Our interest is in determining how maxima and minima behave when an image is blurred. Maxima and minima are of interest because these are the basic structures to which nonextremum points link and thereby define extremal regions in the original image. We noted that blurring an image (function) creates a family of images (family of functions). By using the Morse and Splitting lemmas for families of functions we found that the only way for Morse points to appear or disappear was by first becoming non-Morse. Non-Morse



points are almost always only degenerate in one dimension. We therefore constructed a one dimensional non-Morse function and then unfolded that into a typical one dimensional family of functions which was non-Morse when the parameter (t) was zero. Then to get back to the original number of dimensions we were working with, we added back in the Morse dimensions (x_2^2 , etc.). For two dimensional images we get $f(x_1, x_2; t) = x_1^3 + tx_1 \pm x_2^2$.

We now proceed to investigate the way the Morse and non-Morse points of $f(x_1, \dots, x_n; t) = x_1^3 + tx_1 \pm x_2^2 \pm \dots \pm x_n^2$ behave. We can visualize this behavior most easily by examining $f(x_1; t) = x_1^3 + tx_1$ and remembering that in the other dimensions we have positive or negative quadratic (Morse) terms. For negative values of t both a maximum and a minimum are present in $f(x_1; t)$. These are Morse critical points. The type of Morse critical point they represent for the higher dimensional case ($f(x_1, \dots, x_n; t)$) depends upon the signs of x_2^2 through x_n^2 . If all the signs are all positive, there is a minimum (0-saddle) meeting a 1-saddle. In general we have a k -saddle meeting a $(k+1)$ -saddle. As t increases towards zero, the two Morse critical points approach each other and merge, creating a non-Morse critical point temporarily at $(x_1 = 0, t = 0)$. For positive values of t , no critical points remain. If $n = 2$, we always have minimum or maximum point annihilating with a saddle point to temporarily create a non-Morse critical point. These two cases correspond to the two possible signs that may exist in front of the x_2^2 term. If the x_2^2 term has a plus sign in front of it, the one-dimensional minimum in $f(x_1; t)$ becomes a



minimum in $f(x_1, x_2; t)$ and the maximum in $f(x_1; t)$ becomes a saddle in $f(x_1, x_2; t)$. On the other hand, if the x_2^2 term has a minus sign in front of it, the maximum in $f(x_1; t)$ becomes a two-dimensional maximum in $f(x_1, x_2; t)$ and the minimum becomes a saddle in the two-dimensional function. We therefore conclude that typically *the only way for an extremum in an image to annihilate is by meeting a saddle point*. Similarly, *if an extremum is created, a saddle point must be created also*. Figure 3 and Figure 4 show the position and intensity of the extrema from the graphs in Figure 2.

This particular unfolding of x^3 is "typical", and therefore all other unfoldings of x^3 can be induced from it (p. 146-147). Although Morse theory considers these unfoldings to be representations of the same phenomenon, the sorts of transformations allowed to induce the other unfoldings (which are very similar to those used for showing equivalence of families of functions) can be indicative of different blurring schemes and path structures in the corresponding stack tree. Thus for our purposes they may sometimes have to be considered as different functions. I will expand on the nature of this difference in the following sections.

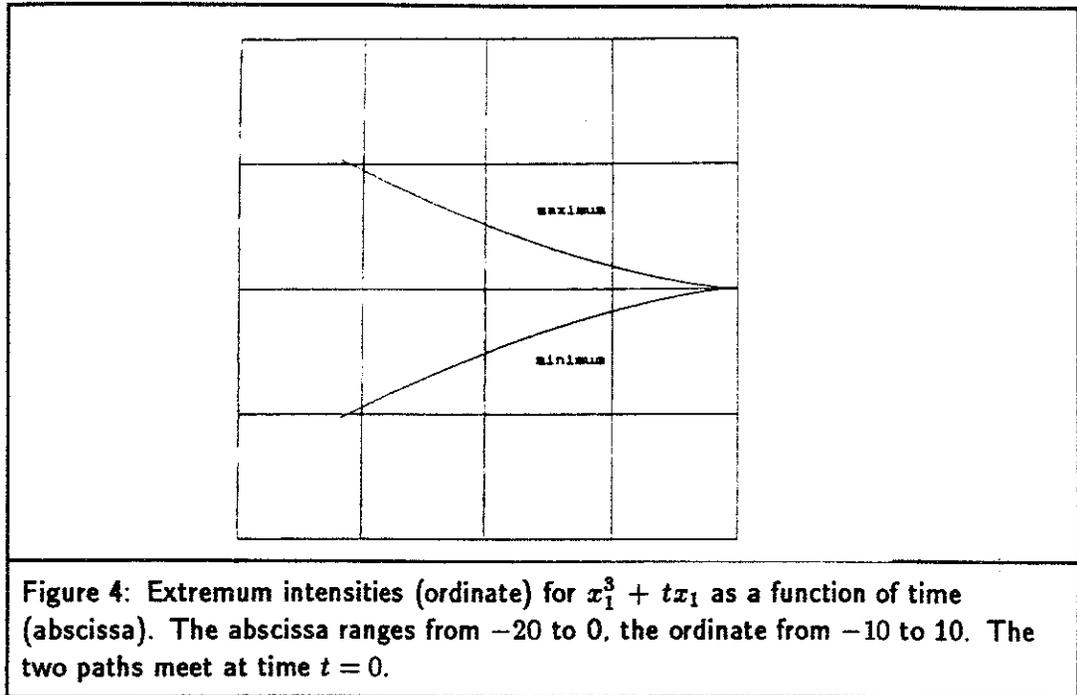


Figure 4: Extremum intensities (ordinate) for $x_1^3 + tx_1$ as a function of time (abscissa). The abscissa ranges from -20 to 0 , the ordinate from -10 to 10 . The two paths meet at time $t = 0$.

2 Embedding an Image in a Family Based Upon Gaussian Blurring

Several researchers have investigated ways of representing one image at multiple resolution scales. They all use similar criteria for deciding upon the best approach to creating lower resolution images.

The most important property is one of *causality*. This means that any *feature* present at a coarse resolution level should have a similar feature present at a finer resolution level. *Features should not spontaneously appear as we move from high to low resolution scale space*. But what, precisely, are features? Several researchers use zero-crossings of various operators (i.e., zeroes of the functions) as their features. We are concerned with extremal paths when creating the tree representation from a stack of image. Therefore our features are extrema. Ideally an extremum should not be allowed to be created (i.e., an extremal path to start) at any resolution level except the original one.

Secondly, any embedding must be *smooth*. This means that intensity changes occur in a continuous manner as scale space is traversed. This implies that extremal paths and iso-intensity paths are smooth curves in position-scale space.

Let us first examine which possible convolution kernels can be applied to the original image and satisfy the above criteria. Note that the restriction to convolutions forces the filter kernel to be shift invariant (but not necessarily radially symmetric).

Witkin [Babaud *et al.*, 1986] shows that among all possible well-behaved kernels the Gaussian is the only one which does not create *zero crossings* (or zero crossing for any derivatives). His work only applies in one-dimension and only for filters which can be expressed as convolution kernels.

Yuille and Poggio [Yuille and Poggio, 1986] give an alternative derivation of Witkin's result. They also extend the analysis to two (and higher) dimensions. They too only examine the class of filters which are shift invariant. They show that if the features of interest are zero crossings of linear derivative operations, the causality property is satisfied (for all images) if and only if the image is filtered with a two-dimensional (not necessarily rotationally symmetric) Gaussian. More generally, they show that for any differential operator L commuting with the diffusion equation (see next paragraph), then $L(F * I) = \text{constant}$ will not have solutions created if and only if F is Gaussian. Directional derivatives are linear (if in a fixed direction) and therefore this proof applies to features such as ravines and ridges. Note that this result does not pertain to zero crossings of nonlinear derivatives. Thus, the proof does *not* apply to *extrema* creation. For the specific nonlinear derivative represented by zero-crossings of the directional derivative along the gradient they were able to show that *no* (shift invariant) filter has the required behavior.

Koenderink [Koenderink, 1984] examines the specific case of intensity creation. No intensity should exist at a low resolution level which cannot be traced (in a continuous manner) to an identical intensity at a higher resolution. Applying the constraints of causality, homogeneity, and isotropy, he derives the following sufficient relationship: $I_{xx}(x, y, s) + I_{yy}(x, y, s) = I_s(x, y, s)$, where $I(x, y, s)$ is intensity, s is scale or resolution, and a subscript denotes a partial derivative. He approaches this by examining surfaces of constant intensity, $I(x, y, s) = \Lambda_0$, where Λ_0 is a constant. Since we require that the blurred image not possess intensity values which cannot be traced to higher resolution levels, $I = \Lambda_0$ should point its convex side towards the direction of decreasing resolution at the extrema locations. That is, the principal curvatures must have the same sign sense as the normal direction, (I_x, I_y, I_s) . The sign sense of this can be taken to be I_s . In

an appropriate coordinate system, the principal curvatures of the surface at the extrema locations are proportional to the λ of

$$\lambda^2 - \lambda(I_{xx} + I_{yy}) + I_{xx}I_{yy} = 0$$

with a positive scale factor [Spivak, 1970]. But at an extremum the curvature (second derivative) in any direction must have the same sign as in any other direction, so $I_{xx}I_{yy}$ must be positive. It follows that the sign of the principal curvatures, at the extrema, is the same as that of $I_{xx} + I_{yy}$. Therefore, $I_{xx} + I_{yy} = \alpha^2(x, y, s)I_s$, where α is never equal to 0. This can equivalently be written as $\nabla^2 I = \alpha^2(x, y, s)I_s$. We can conclude that it is necessary and sufficient that the family of derived images satisfy this relationship at extrema locations so that no new intensity levels are created. Going one step further, it is sufficient for the family of derived images to satisfy $\nabla^2 I = I_s$. This differential equation is the heat conduction (or diffusion) equation.

Despite the fact that no new intensity levels are created, extrema can be created in dimensions higher than one. An easy example to help in the visualization of how this can occur in two dimensions is as follows. Imagine two broad, high mountains with a deep wide valley between them. One mountain is higher than the other. Connect these two mountains with a thin ramp bridge between their tops. The heights of the mountains, the valley, and the bridge represent intensity levels. The shorter mountain is not a local maximum because the ramp connects it with the higher mountain. But as diffusion occurs on the intensity distribution represented by this geography, the intensities of points represented by the bridge will decrease, since the deep valley is on both sides of it. It will quickly turn into a bridge with a deep dip in the middle. This will turn the smaller mountain into a local maximum.

A specific example of extremum creation is shown in Figure 5 (worked in collaboration with Professor Halton at U.N.C.). The example clearly shows a maximum and a saddle appearing (and eventually annihilating) under diffusion. This is a graph of the function $f(x, y) = -.25x^4 + .5x^2 - x + 4 - 10y^2(x + 1.5)^2(x - 1)^2$. This function has a shape approximating that described above. The width of the function in the y -direction is greatest at $x = -1.5$ and $x = 1$, since the magnitude of the coefficient for the y^2 term is

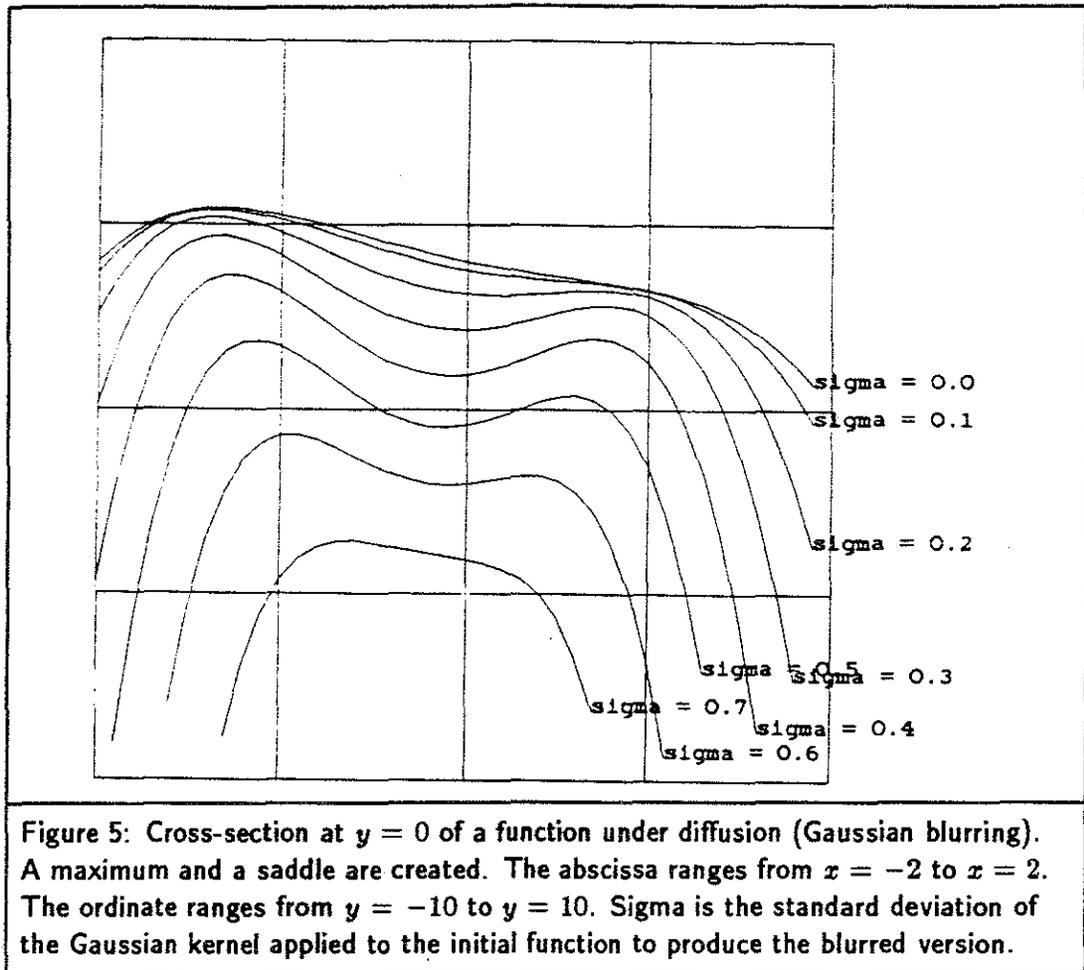


Figure 5: Cross-section at $y = 0$ of a function under diffusion (Gaussian blurring). A maximum and a saddle are created. The abscissa ranges from $x = -2$ to $x = 2$. The ordinate ranges from $y = -10$ to $y = 10$. Sigma is the standard deviation of the Gaussian kernel applied to the initial function to produce the blurred version.

zero there. The graph shows a cross section of the function ($y = 0$) after various amounts of blurring.

The diffusion (heat) equation seems to be the best alternative available to guide blurring, since it does not create new intensity levels. The ability to create extrema is undesirable; features exist at low resolution levels which do not exist at higher ones. Nevertheless, this method of extrema creation is preferable to others which may also create new intensity levels. There are several additional reasons for desiring that the embedding satisfy the heat equation everywhere. First, if the image satisfies the heat equation, the solution of the equation will depend continuously on the data (original image and boundary conditions) [Zachmanoglou and Thoe, 1976; p 333]. Second, if the image satisfies the heat equation, $I(x, y, t > 0)$ is infinitely differentiable (in x and y) even if $I(x, y, 0)$ is not.

To blur an image, one should convolve it with the filter kernel that is the Green's function for the diffusion differential equation. The solution to a differential equation depends upon its initial conditions and boundary conditions. The initial condition is $I(x, y, 0) = H(x, y)$, where $H(x, y)$ is the original image. If the original image has no boundaries (i.e., takes up all of R^2), the solution yields a Gaussian filter kernel. This essentially develops from the heat equation as follows. If only one point is set to a particular nonzero temperature (intensity) at time $t = 0$, there will be a Gaussian temperature distribution around that point at some later time t (assuming isotropy of the the medium, etc.). The variance of the Gaussian is related to t by $\sigma^2/2 = t$. The temperature (intensity) distribution at time t of an entire image is just the combination of Gaussian distributions resulting from each pixel. This is the same as the convolution of a Gaussian with the initial heat (intensity) distribution [Sokolnikoff and Redheffer, 1966; p 498].

Unfortunately, if the image is not infinite in extent, the simple result of a Gaussian filter kernel is not necessarily correct. There are several ways the original image might be modified to eliminate its boundaries, and thereby maintain a Gaussian kernel solution to the diffusion equation. Mirroring of the image at its boundaries essentially creates an infinite image. The infinite image has symmetries such that convolution of it with a Gaussian will yield the same result as performing a convolution with the finite image (while mirroring the kernel at the boundaries to handle the edge effects). This image modification has several problems. The image created has symmetries, an atypical behavior. The mirroring also creates ridges and ravines along the boundary of the image. These are non-Morse phenomena. Such a blurring scheme will also not result in just one extremum in the limit after sufficient blurring, whereas an ordinary infinite image will yield just one extremum after being blurred sufficiently. A mirrored image will yield an extremum at each corner and one in the interior of the original image, plus their reflected copies in the rest of the infinite image.

An alternative approach to eliminating boundaries is to wrap-around an image at the boundary. This means that points outside a boundary are interpreted as being inside the boundary on the opposite side of the image. This is equivalent to changing the image topology to that of a torus. This will not result in only one extremum remaining in the limit after sufficient blurring; a torus topology does not permit the existence of only one critical point. After such an image is blurred sufficiently, one minimum, one maximum, and two saddle points will be left in the image. Such a distribution can visualized by imagining a doughnut (torus) standing on its side. Let the intensity at each point on the

torus be its height above the ground. Then the highest point will be a maximum and the contact point with the ground the minimum. Opposite each of these points there will be a saddle point. Wraparound also creates discontinuities in the image initially, since the intensity at one boundary of an image is usually not the same as at the opposite boundary.

Yet a third method of handling the boundary conditions is to solve the diffusion differential equation explicitly for the finite plane with the given initial conditions (the initial image). The general solution to the finite domain heat equation is [John, 1982; p 219]

$$I(\zeta, T) = \int_{\Omega} K(\mathbf{x}, \zeta, T) I(\mathbf{x}, 0) d\mathbf{x} + \int_0^T dt \int_{\mathbf{x} \in \partial\Omega} \left(K(\mathbf{x}, \zeta, T-t) \frac{dI(\mathbf{x}, t)}{dn} - I(\mathbf{x}, t) \frac{dK(\mathbf{x}, \zeta, T-t)}{dn} \right) dS_{\mathbf{x}} \quad (1)$$

Here $I(\zeta, T)$ is the image at time T . $K(\mathbf{x}, \zeta, T)$ is a Gaussian of width T . Ω is the region the image is defined on, and $\partial\Omega$ is its boundary. $dS_{\mathbf{x}}$ is just the length element along the boundary, and $\frac{d}{dn}$ is the derivative in the direction normal to the boundary. This solution reduces to a convolution of the image with a Gaussian kernel if only the first term is nonzero. The conditions under which this is so will now be discussed.

The second term in the second integral will be equal to zero if the intensity of the image is zero along the boundary. An image which obeys the diffusion equation and has boundary values of zero can be created from the initial image (assuming the boundary conditions of the initial image are time independent). This is done by subtracting off an image which is invariant under blurring and has the identical boundary conditions. The original image satisfies the diffusion equation: $I_t(\mathbf{x}, t) - \nabla^2 I(\mathbf{x}, t) = 0$; $\mathbf{x} \in \Omega$, $t > 0$. Let the invariant part be $v(\mathbf{x})$ which is the solution to

$$\nabla^2 v = 0, \quad \mathbf{x} \in \Omega$$

$$v(\mathbf{x}) = I(\mathbf{x}, 0), \quad \mathbf{x} \in \partial\Omega.$$

Now define the time varying component as $w(\mathbf{x}, t)$ such that $I(\mathbf{x}, t) = v(\mathbf{x}) + w(\mathbf{x}, t)$. For $w(\mathbf{x}, t)$ to satisfy this equation it must be the solution to

$$w_t - \nabla^2 w = 0; \quad \mathbf{x} \in \Omega, \quad t > 0,$$

$$w(\mathbf{x}, 0) = I(\mathbf{x}, 0) - v(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

$$w(\mathbf{x}, t) = 0; \quad \mathbf{x} \in \partial\Omega, \quad t \geq 0.$$

If we use w instead of I in equation 1 (and then add v back into the result at the end if desired), the second term in the second integral will be zero.

The first term in the second integral will be nonzero unless an additional constraint is added, that the image be insulated, i.e., have $\frac{dI(\mathbf{x},t)}{dn} = 0$ at the boundary. This is not an unreasonable constraint to impose. The image is already discretely sampled in space. We can therefore always assume that the unknown intensity distribution between a boundary pixel and its neighbor in the interior is such that the constraint is satisfied without contradicting any known data or forcing the interpixel intensity distribution to be unnatural.

We can conclude that convolution of a bounded image with a Gaussian kernel (whose contribution is set equal to zero if outside the image boundary) can indeed be considered to be an appropriate solution to the diffusion equation for an *insulated* bounded image with zero intensity along its boundary. Since this approach is valid, it is the method used in this dissertation. Toet [Toet *et al.*, 1986] also uses this technique but did not show that it was in fact valid.

It is possible that a more flexible blurring strategy which does not satisfy the diffusion equation might be developed. Satisfaction of the diffusion equation is a sufficient, but not necessary, condition to prevent creation of new intensity levels. The necessary and sufficient condition for no intensity level to be created is that $\nabla^2 I = \alpha^2(x, y, s)I$, (the sign of the Laplacian of the intensity is equal to the sign of $\partial I/\partial s$) at the locations of the extrema only. In particular, nonstationary blurring schemes may be derivable using this constraint. There is probably a relatively simple way to insure that the production of lower resolution images still satisfies the necessary constraints. This would be as follows. The necessary constraints describe how an image is allowed to *change* as it evolves into a lower resolution image. We can construct a difference image which, when added to one image, will produce an image of lower resolution. This difference image can be easily constructed so that the image will evolve according to the diffusion equation around the extrema locations. Other pixels in the difference image would be free (except probably for some continuity constraints) to have whatever intensities were deemed desirable.

3 Containment for Extremal Region Paths in the Generic Case

Deciding on an embedding scheme for the original image in a family of lower resolution images is not enough. Points in the image at one resolution level must also be associated with points in the image at another resolution level. This will define a path through resolution space for each point in the original image. As mentioned in chapter 2, the way that these paths (links) join up with each other will induce a decomposition of the original image into nested regions. All those paths which link to the same extremal path define the extremal region associated with that extremal path. Criteria for linking both nonextremum points and extremum points at one level to the appropriate pixels at the next level are needed to guide the creation of this path structure. I will use the same linking criteria as Koenderink, although the derivation of the results differs somewhat.

There are two criteria a path of a nonextremum point should satisfy. First, the intensity should stay constant along the path; the path of a nonextremum is therefore also called an iso-intensity path. Second, the point should move along the iso-intensity surface (in x, y, s space) in a path of steepest ascent (where s , the resolution dimension, is up). Let the direction in which a path should move be defined as $\vec{v} = (v_1, v_2, v_3)$, where the three components are in the x, y and s directions respectively. The first constraint can now be written as $\vec{d}I \cdot \vec{v} = 0$ or $I_x v_1 + I_y v_2 + I_s v_3 = 0$, where the subscripts of x, y , and s represent partial differentiation with respect to each of those variables. The second constraint is equivalent to stating that the component of \vec{v} in the (x, y) plane should be in the same direction as the gradient, i.e. (I_x, I_y) . If this component of \vec{v} and the gradient are in the same direction, their cross product is zero. So, $(v_1, v_2, 0) \times (I_x, I_y, 0) = 0$. This yields $I_x v_2 + I_y v_1 = 0$. The results are two equations in three unknowns:

$$I_x v_1 + I_y v_2 + I_s v_3 = 0 \quad (1)$$

$$I_x v_2 + I_y v_1 = 0 \quad (2)$$

We can now solve for a direction for \vec{v} , although its magnitude will be arbitrary. This results in

$$\vec{v} = (-I_s I_x, -I_s I_y, (I_x)^2 + (I_y)^2) \quad (3)$$

This is the direction the path of a nonextremum point should take. At an extremum I_x and I_y are both equal to zero, so \vec{v} is a null vector. In other words, once a nonextremum path meets an extremum, \vec{v} no longer specifies the direction in which to proceed. Criteria for the path direction of an extremum hold from then on. The extremal region associated with

an extremum path is just all those points whose paths eventually join with the extremal path.

The path criterion for extrema is very simple. Each extremum is isolated in the typical case. That is, there are no other extrema in some neighborhood around each extremum. In addition, their positions move continuously through scale space. It is therefore always immediately evident what path each extremum should take. Contrary to isointensity path criteria, intensity along an extremal path changes. Intensity along the path of a maximum will decrease with decreasing resolution, while intensity of a minimum will increase. When an extremum annihilates, its path continues on as an isointensity path (see figure 8 of chapter 2). The path criteria for nonextremum and extremum points determine which regions in the original image become associated with each extremal path.

Let us now examine the generic case of extremum annihilation (derived in the section on Morse theory basics for the generic case) in an attempt to understand the rules guiding extremal region formation. The generic description of a saddle and a minimum annihilating was shown at the beginning of this chapter to be $f(x, y; t) = x^3 + tx + y^2$. The simplest way to visualize this (see Figure 6) is to imagine a second minimum (which will not annihilate) existing on the other side of the saddle point. The saddle exists between the two minima so that the isointensity curve through the saddle point surrounds the two minima. Annihilation of the saddle with one minimum takes place at $x = 0$, $y = 0$, $t = 0$. The annihilation intensity is zero. At some initial time $t_0 < 0$ the zero intensity contours surrounding the minima lie inside the isointensity contour of the saddle. As time progresses, the minimum which will annihilate and the saddle point move towards each other. The lobe of the saddle isointensity contour which surrounds this minimum gets smaller, as does the zero intensity contour which surrounds the minimum. The zero intensity contour remains inside the saddle intensity contour the entire time. The zero intensity contour surrounding the annihilating minimum and the lobe of the saddle contour surrounding the annihilating minimum become identical for the brief instant as annihilation occurs. To determine the nature of the extremal region such a scenario will produce, one must examine which nonextremum paths link to the annihilating extremum. Nonextremum paths are isointensity paths. Those starting out inside the zero intensity contour surrounding the extremum have intensities less than zero. They can not cross the zero intensity surface represented by the zero contour in scale space. Yet the zero intensity contour eventually collapses into the extremum point, enclosing no area at annihilation time. The only place for the nonextremum paths to go is to link up with the extremum

path. This means that all iso-intensity paths through resolution space that start off inside the zero intensity contour *must* eventually link up to the annihilating minimum. This region represents the extremal region associated with the annihilating extremum.

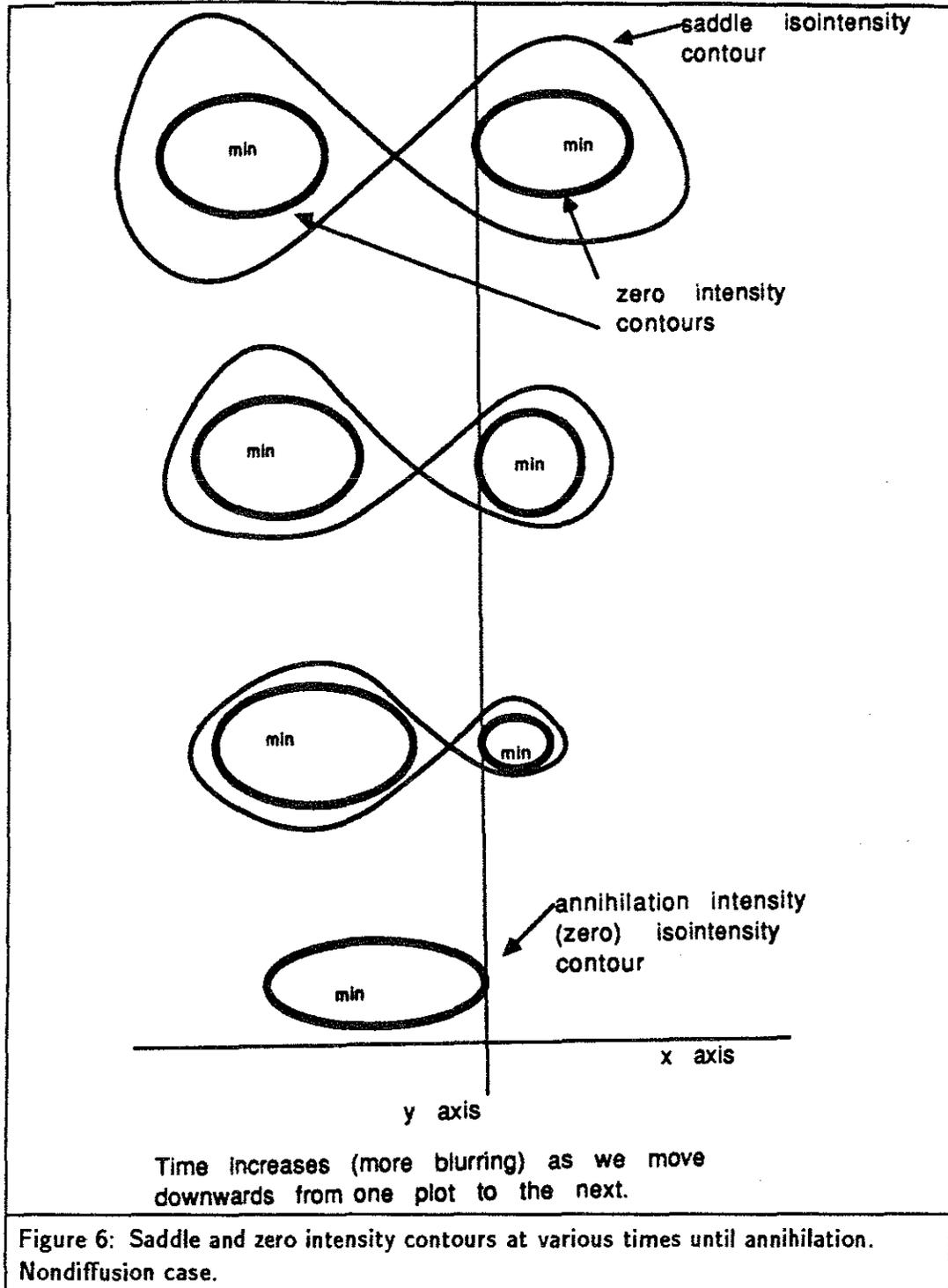
Another way to see this result is to look again at how $I(x,0,t) = x^3 + tx$ changes as time increases (see Figure 2). The point at $x = 0$ has a fixed (constant) intensity for all time. Note that all the intensities at points x less than zero fall in intensity as time increases, and vice-versa for those for which x is greater than zero (the extremal region for the annihilating minimum). This means that a point at $x > 0$ has its intensity increasing with time. So if an iso-intensity path is at a particular $x > 0$ at some \tilde{t} , at time $\tilde{t} + \delta t$ it will have to move *towards* the minimum, to compensate for the fact that all the intensities around it are rising. One can easily visualize how this phenomenon tends to cause the iso-intensity paths starting with $x > 0$ and intensity less than zero (i.e., those inside the zero intensity contour) to link to the extremal path for the minimum.

4 *Noncontainment for Extremal Region Paths in the Gaussian Case*

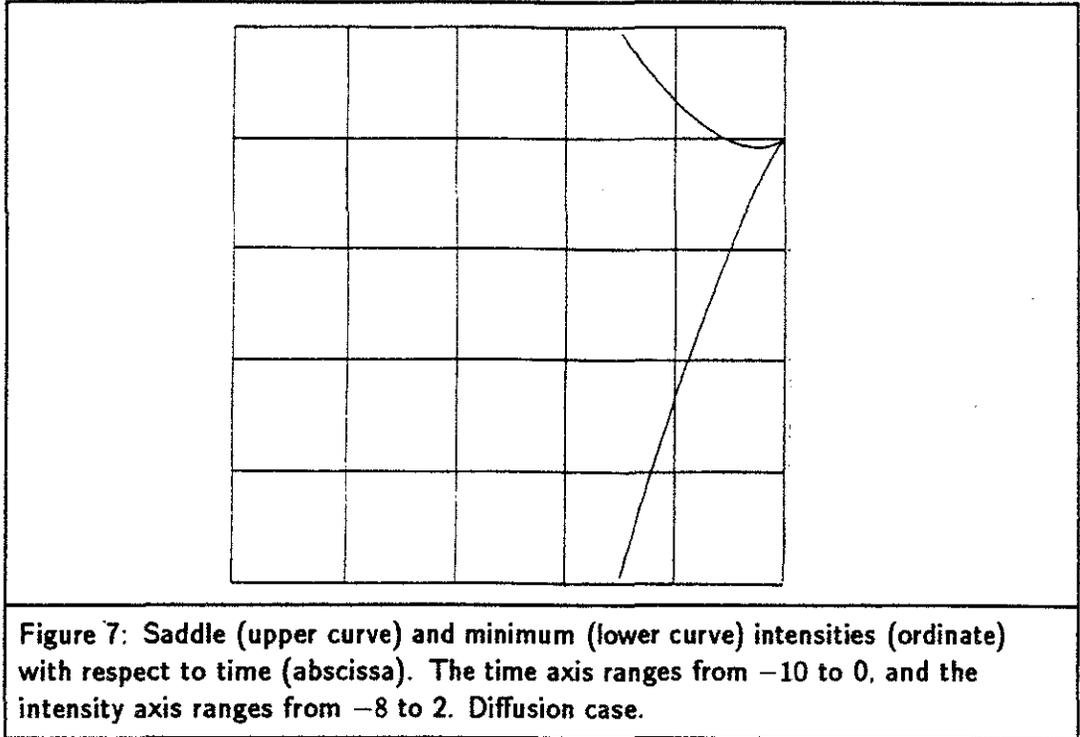
Imposing the constraint of satisfying the heat equation (i.e., convolution with a Gaussian) modifies the conclusions reached above about the manner in which extrema annihilate. Once we restrict the embeddings to a particular class of smooth embeddings, namely those satisfying the heat equation, it becomes very difficult to know which descriptions are "generic" in this restricted subclass. Nevertheless, specific cases can and will be examined which must fit into the diffusion framework. There is no reason to expect that the qualitative nature of these solutions should be different from others satisfying the constraints, although some anomalies may be introduced.

Blurring $x^3/6 + tx$ with a Gaussian produces the prototypical diffusion annihilation. In the one-dimensional case nothing changes significantly from the canonical (nondiffusion) description given above. The time parameter in the formalism gets replaced by $\sigma^2/2$, where σ is the total standard deviation of the Gaussian blurring done so far. The x^3 term gets a factor of $1/6$ in front of it so that the prototype equation satisfies the one-dimensional heat equation.

The two-dimensional annihilation via a diffusion process changes from the generic annihilation in a subtle but very important way. The annihilation equation representing the diffusion case is $I(x,y,t) = x^3/6 + tx + t + y^2/2$. We have been forced to add a term solely in t to preserve the diffusion characteristic! This has many important consequences. I will now examine this equation, which is for a saddle and a minimum annihilating, in



detail; identical conclusions can be drawn from looking at the $-y^2/2$ case which is for a saddle and a maximum.



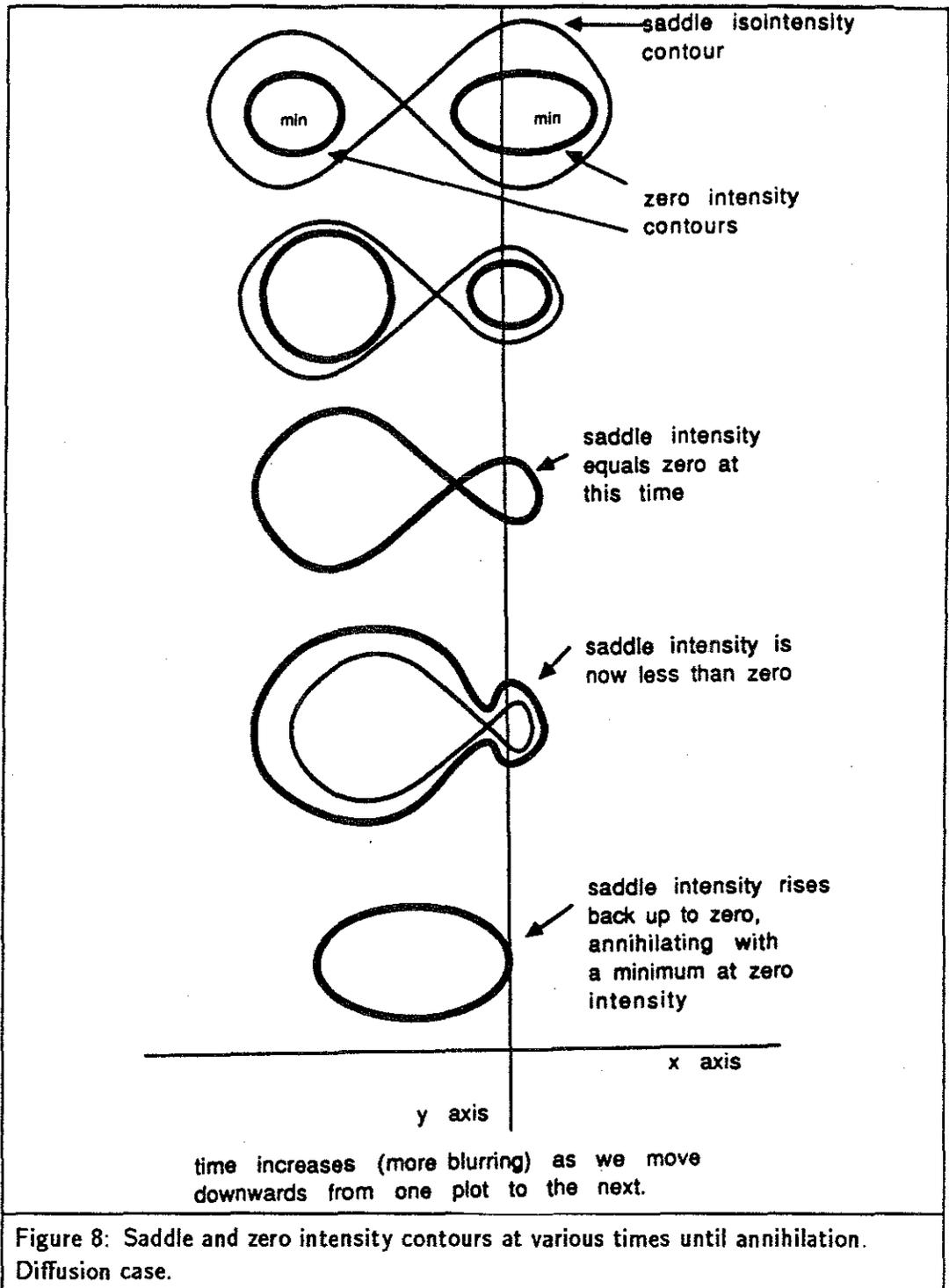
The positions of the two critical points (saddle point and minimum) remain the same as in the nondiffusion case. This is simply derived by setting $\frac{\partial I(\hat{x}, 0, t)}{\partial t} = 0$ and solving for \hat{x} , the x position of the critical point. $\hat{x}^2/2 + t = 0$, as in the previous case, so $\hat{x} = \pm\sqrt{-2t}$, $t \leq 0$. However, the intensities of these critical points differ from the nondiffusion case (see Figure 7). $I(\hat{x}, 0, t) = \pm \frac{(-2t)^{3/2}}{6} \pm t(-2t)^{1/2} + t$, where $+$ is used for $\hat{x} > 0$ ($I(\hat{x}) < 0$) whereas for $\hat{x} < 0$ ($I(\hat{x}) > 0$) $-$ is used and the square root is interpreted as being the positive root. This equation is valid for $t \leq 0$. Now examine $I(\hat{x}, 0, \hat{t}) = 0$, i.e., those times when the critical point intensity equals zero. We find that, as before, both the saddle and the minimum intensities equal zero at $t = 0$, the annihilation intensity and time. But we also now have zero intensity at $\hat{t} = -9/8$ for the saddle intensity! The saddle intensity, which starts out greater than zero (for time $< -9/8$), gradually decreases becoming less than the annihilation intensity when $t > -9/8$. This is confirmed by setting $\frac{\partial I(\hat{x}, 0, t)}{\partial t} = 0$ and noting that the saddle (but not the minimum) has a turning point in intensity at $t = -1/2$. From $t = -1/2$ to $t = 0$ the saddle intensity is increasing, as is the minimum intensity.

There are significant consequences of the saddle intensity dipping below the annihilation intensity and then rising back up to it. In the canonical case analyzed previously,

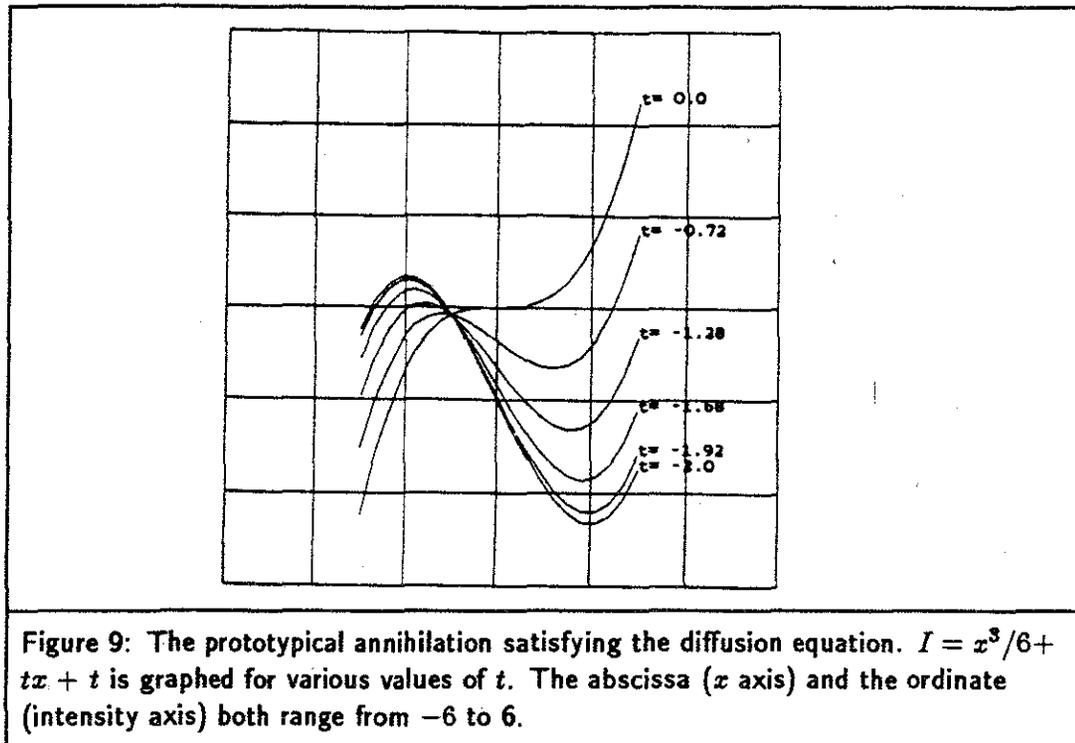
when a saddle existed between two minima, the iso-intensity curve through the saddle point surrounded the two minima (see Figure 6). The saddle intensity contour remained outside the zero intensity contour until annihilation time. This is not true for the diffusion case (see Figure 8). The zero intensity contour, which starts out inside the saddle intensity contour, ends up outside the saddle intensity contour! The saddle intensity contour and the zero intensity contour become identical at the instant the zero intensity contour surrounding the annihilating minimum joins the zero intensity contour surrounding the other minimum. After this time the zero intensity contour surrounds the saddle intensity contour and both of the minima. The "lobe" of the zero intensity contour that surrounds the minimum that will annihilate gradually contracts and "pinches off" at $x = 0, t = 0$ (see Figure 8).

The fixed point of $I(x, 0, t)$, which does not change intensity with time, is now at $x = -1$ (see Figure 9). Intensities for all $x < -1$ continuously decrease, while those for points $x > -1$ continuously increase. Hence, if an iso-intensity path is at a particular $x > -1$ at some \tilde{t} , then at time $\tilde{t} + \delta t$ it will have to move *towards* the minimum, to compensate for the fact that all the intensities around it are rising. There are points with $x < -1$ that have negative intensities. These points will move away from the minimum even though their intensities are less than the annihilation (zero) intensity! The annihilation iso-intensity surface in (x, y, t) space is still a concave cap, but now it has a hole in its side where it joins up with the other zero intensity surface; iso-intensity paths can escape from under the cap by sneaking out through this hole, so they are not forced to join the annihilating minimum's path!

It is difficult to describe mathematically the nonextremum paths which escape through this hole. The iso-intensity paths are integral curves of the vector field $\vec{v} = (-I_x I_t, -I_y I_t, I_x^2 + I_y^2)$. What needs to be done is to substitute into this the formula for $I(x, y, t)$ and solve analytically for the iso-intensity paths. Unfortunately, an analytical solution of these differential equations is intractable. Nevertheless, qualitative examination of the vector field *directions* can yield a simple proof that some nonextremum paths do escape. The proof is as follows. For the prototypical diffusion case we have $\vec{v} = (-\frac{x^2}{2} + t)(x + 1), -y(x + 1), (\frac{x^2}{2} + t)^2 + y^2)$. This means that any point with $x < -1$ and positive y coordinate will have a positive y component in \vec{v} , and such points will move away from the x axis, which is where the saddle and minimum are. Furthermore, if such a point should happen to try to move towards an x coordinate greater than -1 , it will not get there. This is because at -1 both the x and y components of \vec{v} are zero and the



point will stay fixed at $x = -1$ for all future time. By showing that all points with x coordinates less than -1 move away from the x axis (if not on it to begin with) and that



these points can never move to an x value greater than minus one, we have shown that these points (paths) can never meet either the saddle or the minimum path. Yet many of these points have intensity between that of the minimum and the annihilation intensity (zero). Therefore, some points inside the annihilation intensity contour do not link to the minimum path.

I have been able to display the entire paths of escaping nonextremum paths by simulating the blurring and linking process. In order to make the simulation as accurate as possible, an analytic model of the blurring process was created. The analytic model calculates the result of convolving a separable two-dimensional polynomial with a Gaussian. The two-dimensional polynomial represents the two-dimensional intensity field of an image. Using this technique it is possible for a computer to take small steps through three-dimensional intensity-scale space. The direction of the step is guided by the equation for the vector field (\vec{v}). The step size and direction can be dynamically modified to keep the change in intensity along the "isointensity" path within small limits. Using this simulation, I have indeed been able to identify isointensity paths which start out inside the annihilation intensity contour, and yet do not link up to the extremal path of the

annihilating minimum. Not only do paths starting at x coordinates less than -1 escape, but some paths with x greater than -1 never reach the annihilating minimum.

The existence of isointensity paths which have intensity less than the annihilation intensity yet manage not to link to the closest minimum is surprising and unsettling. One of the reasons for choosing the stack as a multiresolution scheme was the nice containment properties of the extremal regions. The situation is both worse and better than it might appear at first glance. It is worse because all of this analysis has been done for the "prototypical" case. This prototype must truly describe an arbitrary annihilation only in a small spatial and temporal neighborhood around the annihilation. In actual images the more global motions will probably be more complex, and one would expect more complex motions to yield even stranger linking relationships. On the other hand, despite the surprising theoretical result that nonextremum paths can escape from the extremal region they originate in, I have yet to observe any clear examples of escaping paths when working with actual images. This is most likely due to the fact that extremal regions are not sufficiently isolated from other regions to display the type of behavior shown in the "prototypical diffusion" annihilation case. The behavior of that case stemmed primarily from the fact that all points with x coordinate less than -1 continually decreased in intensity. In an actual image this would not occur since there would be another minimum at some x less than -1 and this would tend to cause intensities in the region to increase under blurring. In addition, our sampling resolution may be coarse enough to miss small regions. The explanation of why escaping paths are not encountered needs to be researched further.

Chapter 4

Theoretical Issues due to Discreteness

The entire theory upon which the stack algorithm is based applies to continuous Morse images embedded continuously in resolution space. Unfortunately images of this form cannot be handled by a digital computer. The continuous image must be approximated by one which is spatially discrete (i.e., made up of pixels). The smooth embedding of the image in resolution space is approximated by a stack of images each derived from the previous one by convolution with a blurring kernel of non-infinitesimal width and finite extent. Nonextremum and extremum paths, which theoretically are continuous paths, become represented by a chain of links from a pixel in one image to a pixel in the next image to a pixel in the next image and so on. The best manner in which to create these approximations and the complications such approximations create are the subjects of this chapter.

1 Linking Criteria

As mentioned in chapter 3, the nonextremum paths are theoretically integral curves of the vector field $(-I_x I_x, -I_x I_y, I_x^2 + I_y^2)$. This is equivalent to the requirement that a nonextremum path stay on an isointensity surface and move along it in the steepest path possible. This theoretical goal can only be approximated by a computer algorithm due to the discrete nature of the image representation and the non-infinitesimal amount of blurring occurring at each stage. The linking criteria employed should enable the discretely linked path to closely approximate the continuous path.

While in the theoretical case all paths are continuous and a point can always link to an arbitrarily nearby location with intensity identical to its own in an arbitrarily nearby image plane, it is not uncommon for pixels to fail to satisfy this property in the discrete case. There are three main causes for this phenomenon.

1. Areas of an image with high contrast tend to change intensity more rapidly for a fixed amount of blurring than regions of low contrast. Hence the change in the

intensities of all the pixels in the region may be too large for any good match in intensities to be found in a small local region around the current position of the linking path. In this case the local region has been blurred more than is desirable. A similar problem occurs along ridges which slope very gently along their length. A very gentle slope means that the intensity difference between two pixels which are far apart on the slope may be very small. Even a slight blurring which would cause small intensity changes along the length of the ridge would necessitate a large spatial distance between pixels in a nonextremum path.

2. The image manipulated by the computer is a sampled version of the the continuous image dealt with in theory. The samples chosen may not be identical in intensity to that of an isointensity path being followed, even though the continuous image does, in fact, include that exact intensity between the sampled pixels. In this case the desired intensity can be interpolated from nearby pixels; a slower blurring rate would not alleviate this problem.
3. Nonextremum regions very close in both intensity and position to an extremum will have no intensity to link up to (in the next level of the stack), due to the non-infinitesimal amount of blurring which has to take place. This is because the extremum intensity has passed through the desired intensity without ever having explicitly represented that intensity in one of the stack levels. These *near-extremum* points should be linked to the extremum. This problem exists independently of the rate of blurring.

Spatial discreteness caused by sampling is dealt with via a combination of two methods: use of an "original intensity" field and interpolation. Instead of searching for a parent pixel of intensity identical to the intensity of the pixel through which the nonextremum path currently passes, one is sought which has the original intensity of the nonextremum path as an interpolant of its local neighborhood. If a check for interpolation is not performed, the linking process becomes captive to the vagaries of the sampling process, and strange artifacts can occur. Particularly damaging would be the escape of a nonextremum path from the extremal region it should remain in by passing through an isointensity surface due to quantization artifacts. This would result in a nonextremum pixel becoming associated with the wrong extremal region.

The linking algorithm for a nonextremum path from one stack level up to the next (lower resolution) level is roughly as follows. First, an "original intensity" field is associated

with each nonextremum path when it starts. The intensity value of a nonextremum path does not change along its path. In particular, its intensity at a particular level is *not* the intensity of the pixel on that level with which it is currently associated. That pixel will, in general, not be of exactly the same intensity as the nonextremum path. By not associating the intensity of the pixel through which a path passes at any particular level with the path itself, the intensity value along the path is kept from changing and gradually drifting away from the desired original intensity. All tests for linkage are done by comparing pixel intensities in the new (lower resolution) plane with the original intensity field of the path. The 3 by 3 pixel region in the lower resolution image above the current position of the nonextremum path is tested to see if the desired intensity value is neither a maximum nor a minimum when compared with these 9 intensity values. If it is neither a minimum nor a maximum, its value must be able to be interpolated between the candidate pixel and one of its neighbors. If this is the case, a link is created to the candidate pixel. This allows the link to be theoretically off by almost one pixel, if the interpolated value is very close to the neighbor's value and not the candidate pixel's value, but not by more. This interpolation check is a fairly fast test to execute. If the path intensity is not an interpolant of the neighborhood around the candidate pixel, hill walking is performed. If the intensity of the nonextremum path is above (below) the intensity of the current region, the steepest path up (down) hill is traversed. Hill walking is terminated when a region is reached in which the nonextremum path intensity is an interpolant, or when a local extremum is reached. Hill walking is an attractive approach because it is directly tied to the intensity information in the image. A technique which simply enlarges the surrounding search area until it includes an interpolating region does not take this intensity information into account. Such an enlarging scheme can cause the link to skip over all types of image features (e.g., peaks and ridges) and to pass through intensity surfaces which it should not pass through.

A nonextremum link is not found if hill walking results in a local extremum being reached without reaching a neighborhood that has the original path intensity as an interpolant. If no nonextremum link can be found using the above technique, the possibility of linking to an extremum path is examined. There are three types of extremum paths which must be checked: a regular extremum path, a path which is annihilating at the present level, and a path which is appearing at the present resolution level. The local area is searched for extremum paths. The intensity along the path at the higher and lower

resolution levels is checked. If the desired intensity (original intensity) is between the intensities of the extremum path, the nonextremum path is considered to have annihilated by connecting up with the extremum. An appropriate link is then formed.

A more difficult case to recognize is that of a near-extremum when the associated extremum itself annihilates between the higher and lower resolution images. Non-infinitesimal blurring between images guarantees that the exact annihilation intensity of an extremum path is never known. This creates two problems. The first concerns the difficulty in determining how to continue the extremum path (as a nonextremum path) after it annihilates; it is not known which nonextremum path the extremum path should join. As mentioned in chapter 3, an extremum annihilates when it meets a saddle point. The intensity of the extremum and the saddle when they merge is the annihilation intensity of the extremum path. In the continuous case, there will always be a nonextremum path (at the annihilation intensity) which passes through the annihilation location and continues on. Since the exact annihilation intensity and position of the extremum is not known, it can never be certain which nonextremum path the extremum path joins. An educated guess must be made. The second problem follows from the first and affects near-extremum paths. Since the exact annihilation intensity of the extremum which the near-extrema are near is not known, there can be no accurate intensity range associated with the last link of the extremum path. Yet this range is precisely what near-extrema examine in order to make the decision for linkage to the extremum path. Incorrect links may result from an inaccurate intensity range assigned to the annihilating extremum path.

It might be thought that it is possible to reduce some of the annihilation ambiguity by searching for the saddle point with which the extremum is annihilating. Hopefully this would put some bounds on the possible positions and intensities of the annihilation point. Unfortunately there will still be some inherent ambiguity. This is due to two factors. First, depending upon the image and the blurring rate, it may be difficult to identify which saddle point is the one with which the extremum annihilates. Second, even correct identification of the saddle point does not necessarily put a bound on the intensity range within which the annihilation intensity must fall. The intensity of the saddle point is not constrained to change monotonically as blurring occurs (unlike the intensity of an extremum). This means that knowing the saddle intensity at the stack level prior to annihilation does not yield any hard information pertaining to its annihilation intensity. Heuristics might be developed to use the saddle intensity and location prior to annihilation to help handle extremum path annihilation uncertainties. Theory provides no clearly appropriate way to

handle the situation. Due to these difficulties, saddle point information is not employed in the present stack implementation. Currently the link continuing an annihilating extremum is created by hill walking on the lower resolution image until reaching a region where the extremum intensity is an interpolant of the neighborhood pixel intensities. A link is created to a nonextremum pixel in this neighborhood. Near-extrema examine the intensity range between the extremum pixel's intensity and the intensity of the nonextremum pixel to which the extremum links. If the near-extremum has an intensity in this range, it is linked to the annihilating extremum. If a decision is made to link to an annihilating extremum path, the near-extremum is linked in a manner which incorporates it into the extremal region of the annihilating extremum, and not into the larger region within which the extremum will be nesting (the nonextremum path the extremum joins will be associated with this larger region).

If, after considering these cases, a nonextremum link cannot be found, it could be assumed that this is due to overblurring between adjacent resolution levels of the region and not to sampling artifacts. This region of the image has changed so much from one blurring level to the next that it is no longer possible to correctly track iso-intensity paths. This implies that smaller blurring steps are needed in this region. One way in which I have tried to handle this is to calculate an image with resolution between that of the two that are currently being linked. This is done by blurring the higher resolution image with a kernel with a smaller variance than that used to create the lower resolution image. Linking is then attempted again. Kernels of smaller and smaller variance are applied to the higher resolution image until a satisfactory link to a pixel on an intermediate level can be found. There is then an attempt to link this intermediate pixel to the lower resolution level that was initially calculated. If a link to this level still can't be found, reblurring with a smaller kernel is again executed. The process is repeated until satisfactory links all the way up to the initially desired final resolution level can be created.

Reblurring can be a very time consuming task. If this has to be separately performed for each path which has difficulty linking, much time is lost. It is possible that a fast examination of an image may be able to yield information about the best blurring rate. One would like a scheme which is intimately related to all the implementation decisions and the current image, not one which is the result of a hypothetical model which may or may not apply for a particular image. This suggests a method based upon either results from previous blurring levels or some type of preliminary quick linking of the current level for predictive purposes. No successful scheme of this type has been found yet.

There are other complications associated with reblurring schemes. Time can be saved by reblurring only a small piece of the image. If only a subset of the image is reblurred, the intermediate pixel values along the edges of this region will not accurately reflect the correct intermediate values (those which would exist if the entire image was reblurred). Paths which move a considerable distance during reblurring may move out of the region which is being accurately reblurred. Ridges with a gentle intensity change along their length tend to display this behavior. Pixels at one end of the ridge will move considerable distances to the other end under very slight blurring. The only way to be certain to avoid this problem is to reblur the entire image. Another problem occurs when nonstationary blurring is performed. Nonstationary blurring in two steps is not equivalent to a larger one step nonstationary blur (see chapter 5). There is no clear way to subdivide the blurring steps. In this instance, it is probably better not to attempt to exactly duplicate the image produced by the large blurring level, and just to back up to the higher resolution image and start reblurring by smaller amounts. The exact blurring schedule is not known precisely in any case.

Due to these complications I have decided that it is best to omit reblurring and simply link the nonextremum path to the closest intensity pixel in the region to which it has hill walked, despite the fact that it is not an interpolant of this region. The intensity of the nonextremum path is also modified to be this new pixel's intensity (otherwise the same problem would also occur at all future stack levels). This clearly does not correctly mimic the continuous case. I have found that a judicious initial blurring scheme usually does not produce too many such cases and that the deviations from the theory have not been too visible.

A subtle problem which must be addressed in the implementation of the linking strategy occurs when multiple nonextremum path pixels link up to the same pixel on a lower resolution image. It would introduce significant error into the nesting structure to consider all paths which link to one pixel to be the same path from there on. Each of these nonextremum paths possess their own "original intensity". These values may be quite different from each other. The pixels are all linking to the same pixel because, due to spatial quantization, their preferred interpolated positions are nearest it. Therefore, the ability is provided to allow several distinct paths to pass through one pixel, without the paths becoming merged. The paths may eventually separate, and possibly even end up in different extremal regions.

In addition to affecting the linking of nonextremum paths, discreteness also makes the linking of extremal paths more difficult. Several aspects of extremal path linking are affected. First, as mentioned above, annihilation intensity and position are difficult to determine. Second, in the continuous case extremal points always follow continuous paths and no other extremal paths exist in some finite sized neighborhood around them (unless the image is degenerate). The extremal path is therefore always immediately apparent. In the discrete case it is not always clear which of several nearby extrema in the higher resolution image should link to which extrema in the lower resolution image. If several extrema are in close proximity, when one of the extrema annihilates between the two levels it is not even always obvious which extremum was the one to annihilate. The situation can become even more complicated. In the continuous case the only way an extremal path eventually links up with a different extremal path is for one of them to first annihilate and become a nonextremum path. As a nonextremum path, it may then eventually join up with the other extremal path. But the blurring rate may be such that between two levels the annihilating extremum may well have annihilated into a nonextremum path which, by the next resolution level, should already link into one of the remaining extremal paths. Clearly, the smoother the image and slower the blurring process, the less severe this problem will be. But it has the potential to occur despite any precautions which may be taken.

The current implementation handles these problems in several steps. First, all extrema which might conceivably link to an extremum in the lower resolution image are identified. From this list the extremum which is judged most likely to be on the same path is identified. This criterion basically matches an extremum at one level with the spatially closest one at the other level, assuming they are both of the same type (e.g., both maximums) and their intensities are related correctly (e.g., the intensity of a maximum must decrease towards lower resolution). This process is performed for each extremum. Extrema which do not end up linked to another extremum are passed to the second step, which tries to link them to a nonextremum pixel. If the intensity of the extremum is found to be an interpolant among the nonextremum pixels in the region, it is linked to one of them. Otherwise it is assumed that the nonextremum path that it would have become has itself already joined an extremal path. Therefore the extremum is linked to an extremal path, but not as the main path. This is not a foolproof scheme, but due to the inherent theoretical ambiguity introduced by non-infinitesimal blurring, a better approach is not easily derivable.

2 Blurring

The blurring process itself is complicated by the discreteness of the implementation. Instead of a continuous Gaussian kernel of infinite extent a discrete approximation of finite extent must be used. Spatial Gaussians with very small variance are difficult to represent accurately. Small variance spatial Gaussians have more high frequency content than large variance spatial Gaussians. Hence sampling into them can cause significant aliasing artifacts. Even if one first analytically convolves the Gaussian with a sinc function (i.e., multiply the Gaussian by a low-pass box filter in the frequency domain to eliminate high frequencies which would alias during the sampling process) and then samples into it for the discrete representation, problems still remain. This is because even though aliasing artifacts no longer occur, a Gaussian is no longer being used.

The practical implications of this are two-fold. First, since blurring is not performed with an exact Gaussian, the theory which Koenderink derived no longer precisely applies. His analysis is for Gaussian blurring. Conclusions that new intensity levels cannot be created do not necessarily apply when convolution is not with a Gaussian. Second, since Gaussians are no longer being used, filter kernels lose many of the nice properties they would otherwise have. The convolution of two digital representations of a Gaussian is no longer necessarily equivalent to a digital representation of a third Gaussian whose variance is the sum of the first two variances. In fact, it may not be the digital representation of any Gaussian. Therefore, subdivision of the blurring into many small steps makes it difficult to know exactly what sort of large scale blurring will be performed. The total sum of the variances is no longer a completely accurate guide to when an equivalent amount of blurring has occurred. If any reblurring of an image is attempted due to poor ability to find links, this possible discrepancy between the sum of the variances and the true total blurring amount must be addressed. If it is critical for the reblurring to end in the exact lower resolution image initially attempted, then instead of blurring one of the new intermediate images in an attempt to yield the lower resolution image initially tried, one must recreate the lower resolution image via the exact blurring process initially used on the initial higher resolution image. That is, links should be attempted to the previously calculated lower resolution image instead of one calculated based upon one of the intermediate resolution images calculated during the reblurring phase. It is also desirable that none of the intermediate images created during the reblurring process are of lower resolution than the originally attempted low resolution image. The intensity of several pixels can be tracked from intermediate level to intermediate level to help insure

that overblurring does not occur. If the intensity of a pixel moves outside the intensity range represented by the intensity at the initial higher resolution image and the initial lower resolution image, overblurring may be occurring.

At the other end of the spectrum are Gaussians with large spatial variance. The aliasing artifacts are not nearly so severe for these. However, to accurately represent them one would want samples out to at least two standard deviations from the center. Such a large convolution matrix can lead to substantially longer convolution times with such a kernel. There is a crossover point where, at a certain size convolution matrix, it is faster to perform the blurring operation in the frequency domain. (If a nonstationary blurring scheme is used – see chapter 5, a frequency domain approach to blurring will not be possible.) Another way to speed up the convolution times is to only apply the convolution matrix to, say, alternate pixels. This amounts to sampling into the image at a lower rate. Doing so would normally be expected to create aliasing problems but, if large blurring steps are only desired after a significant amount of blurring has already occurred, most high frequency components in the image should already be gone. Therefore sampling alternate pixels for blurring purposes should cause a minimal amount of aliasing. These time considerations become even more important when working with three-dimensional images. The current implementation has the ability to perform the blurring in either the spatial or frequency domain. The blurring rate is limited so that a blurring step is never attempted which would necessitate too large a matrix to represent it (i.e., one which would take too long to apply).

3 Blurring Rate

The amount of blurring in each step needs to be controlled so that confusion in following extremal paths and associated contour levels across stack levels is avoided, while limiting the number of steps so that reasonable efficiency is achieved. When there are many extremal paths, this criterion may be interpreted to imply an inter-level blurring that is just large enough to ensure that real changes dominate changes due to arithmetic error. When there are few extremal paths, an inter-level blurring is chosen that produces faster progress toward annihilation of one of the paths. It is safe to perform faster blurring at this point since there are no longer any features left which will change quickly due to blurring.

For resolution levels before the point where efficiency considerations lead to faster blurring (when only few extrema remain and they are widely separated), there is the need

to interpret the phrase "real changes dominate changes due to arithmetic error" more precisely. Koenderink [Koenderink, 1984] and Pizer [van Os, 1984] have both interpreted this to mean that the attenuation of the height of some basic function is a small integer multiple of the arithmetic error, but these two investigators have chosen a different basic function. Koenderink chose a sinusoid at the Nyquist frequency associated with the total amount of blurring at any given level, while Pizer chose a Gaussian which was a spike in the scene (but not the original image, which already is a blurred version of the scene) on a flat background, where the ratio of the height of the spike to the background has some value chosen as a parameter.

Koenderink's choice leads to

$$\frac{\delta\sigma^2}{\sigma^2} = k \frac{\sqrt{2\rho}}{-\log_e \rho},$$

where σ^2 is the variance of the total blurring in the image at the present resolution level, $\delta\sigma^2$ is the variance of the blurring to be applied to that image, ρ is the bound on the relative error in the computer representation of intensity, and k is a small integer. This has the attractive property that the amount of additional blurring ($\delta\sigma^2$) is proportional to the total blurring done to create this level.

Pizer's choice leads to

$$\frac{\delta\sigma^2}{\sigma^2} = \frac{k'\rho(1 - \beta\frac{\sigma_0^2}{\sigma^2})}{1 - k'\rho\beta\frac{\sigma_0^2}{\sigma^2}},$$

where ρ is as before, k' is a small integer, β is the ratio of background to peak height in the scene, and σ_0^2 is the blurring due to imaging. Eventually the peak height relative to the background becomes so small that no blurring can reduce it by the criterion degree. At this point the algorithm for determining the blurring rate at each step is changed. For all lower resolution levels the blurring rate is set to allow a decrease in spatial sampling of 2 in each dimension, a common approach in multiresolution methods. To achieve this goal, $\delta\sigma^2$ must be proportional to σ^2 with a constant of proportionality of 3; the result is that the total blurring standard deviation increases by 2 at each step. Studies by van Os and Pizer [van Os, 1984] suggest that Pizer's choice leads to fewer levels of blurring with

no major loss in the quality of the result, when the blurring used in the very first step of the two approaches is the same.

In practice, the blurring rate should be based upon the actual image. As mentioned above, one such approach, reblurring, has been tried. Exact blurring rates are not critical, as long as the rate is not significantly too fast. In particular, I have found that the blurring rate suggested by Pizer is reasonable until the criterion switches to using three times the current total variance as the variance of the kernel for each step. Such a large increase causes large changes in the image from level to level. This makes it difficult to follow iso-intensity and extremum paths and to determine the annihilation intensity for extremum paths. Instead I increase the standard deviation of the blurring kernel by ten percent for each level. This is a slightly faster rate than what Pizer's criterion yields before it switches to using three times the total variance for the current variance. It is fairly easy to get a feel for the rate to blur certain classes of images (i.e., similar region sizes and noise characteristics) based upon experience with a few images.

4 Spatial Sampling

The sampling in space should, by normal sampling practice, decrease as you move up the stack, that is, as the amount of blurring increases. More precisely, the inter-pixel distance should be proportional to the standard deviation of the total amount of blurring due to imaging and blurring to create the resolution level image. Using an argument based on the aliasing error at the Nyquist frequency, Pizer [Pizer, 1983] suggests a proportionality constant of approximately $\pi/\sqrt{1-\ln\rho}$. However, changing the spatial sampling at lower resolution levels in the stack complicates the extremum and iso-intensity path following processes. As a result the spatial sampling in all resolution levels has been left the same as in the original image. Part of the efficiency of resampling is still achieved since only extremum paths and iso-intensity paths that have not yet joined an extremum path are followed to the next resolution level.

5 Extrema Creation

Several mechanisms can create extrema during blurring. One mechanism was mentioned earlier. If the neighborhood around a nonextremum point decreases (increases) in intensity faster than the point itself does, the nonextremum point becomes a maximum (minimum). This cannot happen in the one-dimensional case under Gaussian blurring. In higher dimensions it can occur.

Another possible cause of extrema creation is the non-Morse nature of an original image. Plateaus in intensity oftentimes exist in the initial image. These plateaus are non-Morse and hence nongeneric. Any slight perturbation of the image (e.g., blurring) will cause these plateaus to break up. In the process many extrema will get created. The most common time this occurs is during the initial blurring of the image. This phenomenon is minimized by blurring the initial image slightly before analyzing it. This new image (which would be Morse if the blurring kernel was of infinite extent, and has fewer plateaus in the finite kernel case) is the one which the stack algorithm is applied to. The intensity values of the original image are still used for display purposes. In addition, ridges and other non-Morse features may get created if certain types of non-Gaussian blurring are employed (see chapter 5). Extrema creation will inevitably follow as soon as the non-Morse image is blurred.

A third way extrema can be "created" is due to the spatial sampling. If an extremum is of very small spatial extent, the extremum may be missed completely at one resolution level and sampled into at another.

Two precautions are taken in order to mitigate the consequences of extremum creation. First, nonextremum paths are allowed to turn into (as opposed to link to) an extremum path. Second, extremum paths which appear but are not on any path which started in the original image plane are still tracked. Since the extremum exists, nonextremum paths may very well eventually need to link into this extremum.

6 *Artifacts Introduced by Discrete Implementation*

A single isolated pixel can link to a distant object in instances in which it would not be isolated in the spatially continuous case. This occurs because the region that would also link to the object, and keep this isolated pixel from becoming isolated, is not represented in any pixel. An example will make this clearer. Suppose we have a group of pixels with initial intensities as follows:

100	100	100
100	75	100
50	50	50

Now suppose there is a nearby maximum with an initial intensity of 90 and an annihilation intensity of 60. Then the only pixel in this group which is between 60 and 90 and thus may link to this maximum is the center one. It might do this because as lower resolution images are created an isointensity path to the maximum might appear. If this is the case, it will be the only pixel in this three by three neighborhood to link to this maximum. If the image was not discretely sampled, this pixel would not be an isolated region linking up to the maximum. Between the pixel at 100 on the center-right and the pixel at 50 on the bottom-right there would be a thin region at an intensity such that it would link to the same maximum as the center pixel. Since original intensities are kept for each nonextremum path and regions are always examined to determine if the original intensity is an interpolant, the path linking algorithm would have no problem correctly linking the isolated pixel (the one with intensity 75) to the correct extremum path.

Chapter 5

Extensions to the Stack - Embedding Schemes

The basic stack algorithm does not always segment an image in the most preferred manner. It can take pixels that, visually, all belong to one region and link them to different extremum paths (and hence different regions). It may also join pixels together in one region that visually are very similar but semantically should not be joined (i.e., they are in different organs which happen to abut each other). The algorithm works well, but not always well enough. It therefore seems reasonable to try to adjust it in a way which would improve its performance. The aforementioned problem is one of accuracy in segmentation. An accurate tree description implies the existence of subtree structures whose leaves represent completely an area of the image which we subjectively determine to be meaningful, and only that region (e.g., it would include all pixels in the liver, and none not in the liver).

The question is how the stack algorithm should be modified to produce a more accurate image segmentation. There are two modifications to the algorithm which come immediately to mind. One is to alter the way the original image is embedded in the multiresolution stack. A different embedding should cause different extremal regions to form. The second approach is to alter the linking criteria. Modification of the linking criteria has not been investigated but would be an area for promising research in the future. Of course, both of these modifications could be applied concurrently.

Intuitively one would expect that modifying the shape of the blurring kernel to reflect the shapes of the regions of interest would yield better segmentation results than always performing stationary, isotropic Gaussian blurring. Any blurring scheme adopted should still be required not to create any new intensity levels as blurring proceeds. Koenderink's main criterion for this is that $I_{xx} + I_{yy} = \alpha^2(x, y, t)I_t$ at the extrema. Of course, the embedding should also remain smooth. This allows for considerable flexibility in choosing a blurring strategy. Let us first examine the effects of stationary, but anisotropic, blurring, and then those of nonstationary blurring.

1 Anisotropic Stationary Blurring

It is fairly easy to show that for anisotropic Gaussians with $\sigma_{xy} = 0$, convolution of that Gaussian with a polynomial is equivalent to a rescaling of the coordinate axes, isotropic blurring, and then rescaling the coordinates back. This result has validity over a wider set of cases than might be originally thought. Since axes can always be chosen for any Gaussian so that $\sigma_{xy} = 0$, the result applies to any anisotropic convolution of a Gaussian with a polynomial. But even the restriction to polynomials is less constraining than it appears. Since we can approximate any continuous function arbitrarily closely with a polynomial [Kolmogorov and Fomin, 1970; p48], any image of interest can be represented by a polynomial. Thus, convolution of any continuous image with any anisotropic Gaussian can be represented equivalently as a coordinate scale change followed by isotropic blurring followed by a coordinate change back. But a smooth coordinate change cannot create or eliminate local extrema from an image. This means that the properties of isotropic Gaussian blurring of images discussed in chapter 3 (no creation of new intensity levels, theoretical ability of paths to leave their original extremal region) should apply to the anisotropic case also. It does *not* mean that the extremal regions formed are identical in the isotropic and anisotropic cases.

A way to study the extremal regions formed is to investigate the analytical form of an anisotropic Gaussian convolved with the generic $\frac{x^2}{6} + tx + t + \frac{y^2}{2}$ polynomial. A two-dimensional Gaussian is represented by

$$G(x, y) = \frac{1}{2\pi|\Sigma|^{1/2}} e^{-\frac{1}{2}(x,y)^T \Sigma^{-1}(x,y)}$$

where

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{pmatrix}$$

and

$$\Sigma^{-1} = \begin{pmatrix} \sigma_y^2 & -\sigma_{xy}^2 \\ -\sigma_{xy}^2 & \sigma_x^2 \end{pmatrix} \times \frac{1}{\sigma_x^2 \sigma_y^2 - \sigma_{xy}^4}$$

Examination of the simpler case of a Gaussian oriented along one of the coordinate axes yields the separable case of

$$G(x, y) = \left(\frac{1}{\sqrt{2\pi\sigma_x}} e^{-\frac{x^2}{2\sigma_x^2}} \right) \left(\frac{1}{\sqrt{2\pi\sigma_y}} e^{-\frac{y^2}{2\sigma_y^2}} \right)$$

In this case a convolution yields

$$a_n x^n * G(x, y) = \frac{1}{\sqrt{2\pi}} \left[a_n \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k} x^{n-2k} \Gamma(k+1/2) 2^{k+1/2} \sigma_x^{2k} \right]$$

Similar results are obtained from $b_m y^m * G(x, y)$. For $c x^n y^m * G(x, y)$ the result is just $c \times (x^n * G) \times (y^m * G)$. Using these results, we find that

$$\left(\frac{x^3}{6} + t_{ox}x + t_{oy} + y^2/2\right) * G(x, y)$$

is equal to

$$x^3/6 + (t_{ox} + \sigma_x^2/2)x + (t_{oy} + \sigma_y^2/2) + y^2/2. \quad (1)$$

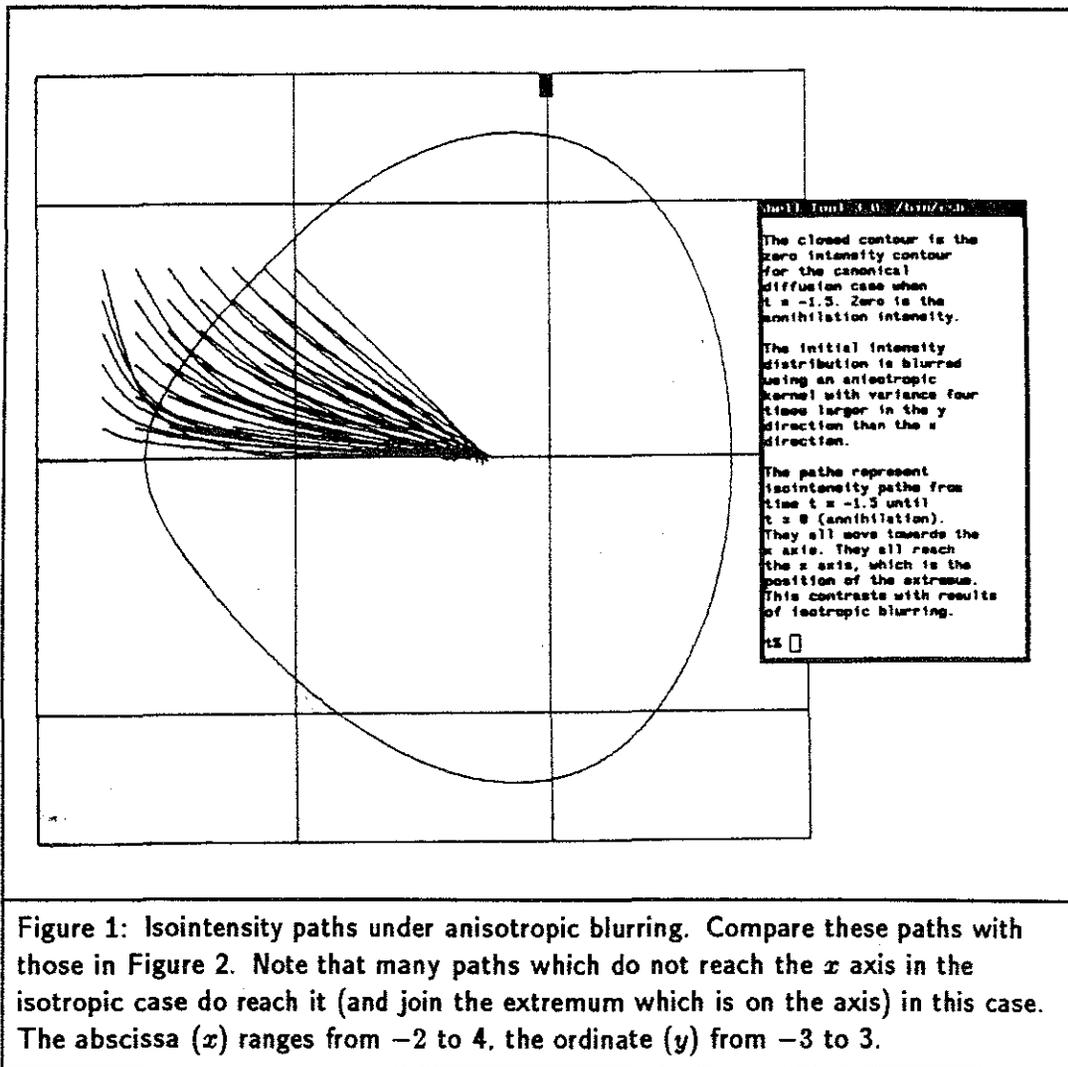
Let us now investigate how this new formula changes which pixels are associated with the extremal region as compared with those associated in the isotropic case. Nonextremum pixels link to form nonextremum paths by following an iso-intensity path through resolution space. The vector field, \vec{v} , defining the direction of the iso-intensity paths changes because of differences between equation 1 and that for the isotropic case. Recall that the defining equation for \vec{v} is $(-I_t I_x, -I_t I_y, I_x^2 + I_y^2)$. The isotropic case of $I(x, y, t) = x^3/6 + tx + t + y^2/2$ produced $I_x = x^2/2 + t$, $I_y = y$, and $I_t = x + 1$. Before calculating the equivalent quantities in the anisotropic case, a decision must be made as to what the "time" parameter is in this case. The coefficient for the x term in equation 1 changes at a different rate than the one for the low order term (x^0 term). For the simple case when $\sigma_x(\text{image}_i) = c\sigma_y(\text{image}_i)$ for all images, the choice between $\frac{\sigma_x^2}{2}$ and $\frac{\sigma_y^2}{2}$ as the time parameter is not significant since the rates are related by a constant. This is the case we will analyze. Let the "time" parameter be chosen as $t_{ox} + \sigma_x^2/2$ with $\sigma_x = c\sigma_y$ for all images. In this case I_x and I_y are the same as for the isotropic case but now $I_t = x + 1/c^2$. This has several implications. The first is that the point which does not change intensity over time (fixed point) is different from the isotropic case. Before it was at $x = -1, y = 0$ (because c was equal to one and the solution is for that x for which $I_t = 0$). Now the fixed point is at $x = -1/c^2$. All points with x coordinate less than the fixed point decrease in intensity continuously and all points with x coordinate greater than the fixed point increase in intensity continuously. The second implication is that the annihilation intensity has also changed. This can be seen as follows. Since annihilation still takes place at the origin, we can see from equation 1 that the annihilation intensity is $t_{oy} + \sigma_y^2(t_{ann})/2$, where t_{ann} is t at the annihilation time. A simpler form for this expression is found by rewriting $\sigma_y^2(t_{ann})/2$. We know that at annihilation time $t = 0$. Therefore, $t_{ox} + \sigma_x^2(t_{ann})/2 = 0$, i.e. $\sigma_x^2(t_{ann}) = -2t_{ox}$. But since $\sigma_x(t) = c\sigma_y(t)$, this is the same as $\sigma_y^2(t_{ann}) = \frac{-2t_{ox}}{c^2}$. Substituting in the previous expression for $\sigma_y^2(t_{ann})$ gives an annihilation intensity of $t_{oy} - t_{ox}/c^2$. This can be simplified even further if we let

$t_{oy} = t_{oz} = t_o$, which was done for the isotropic case. The annihilation intensity becomes $t_o(1 - 1/c^2)$. Thus, the annihilation intensity is *no longer zero*.

Changes in \bar{v} cause changes in the isointensity paths. Since I_x and I_y have not changed, the third or time component ($I_x^2 + I_y^2$) of \bar{v} is as before. Also the ratio of the x and y components to each other remains the same. Thus the orientation of the x, y component of \bar{v} has not changed. At first this seems to imply that the paths have not changed. This contradicts our intuitive belief that changes in the relative amounts of blurring in the x and y directions should cause the isointensity paths to move more in one direction than the other. We would expect I_x to grow relative to I_y if $\sigma_x = c\sigma_y$ for $c > 1$. Our intuition has not failed us, but the mathematics which expresses the change in path direction is more subtle than expected. Even though the x and y components have not changed relative to each other, they have changed relative to the time component, since I_t has changed from the isotropic case and the time component does not contain I_t as a factor. This means that the path taken will *not* be the same as in the isotropic case. For a given point in an image, both a path for the isotropic case and a path for the anisotropic case will start off in the same (x, y) direction, but they will climb in the resolution direction at different rates. So, for example, after some δt the two paths will be at different (x, y) positions in the image - even though they started out at the same position. From now on their (x, y) components don't even agree since the intensity distribution about each point will be different.

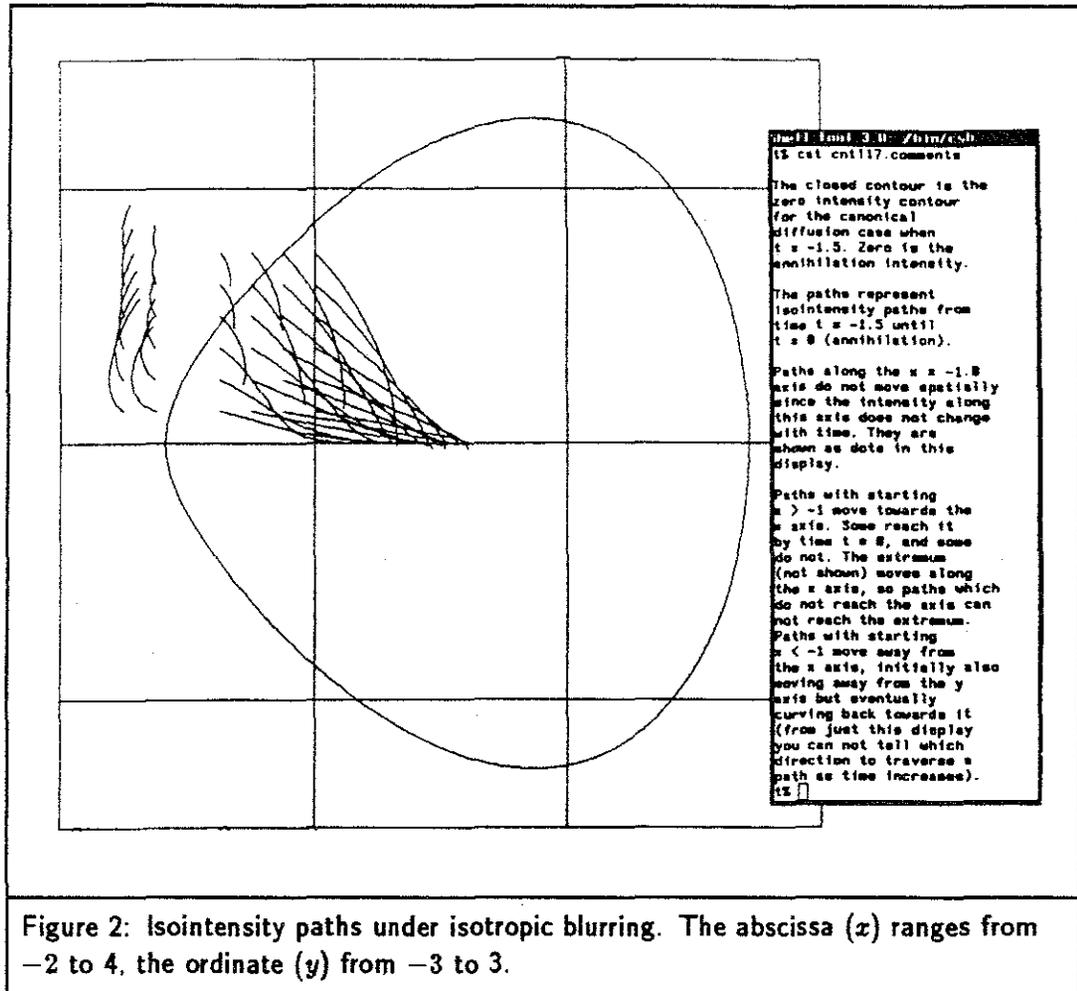
Modification of the simulation program (see chapter 3) to perform anisotropic convolutions has permitted simulation of isointensity paths for anisotropic Gaussian blurring. Comparisons of Figure 1 (anisotropic paths) to Figure 2 (isotropic paths) clearly show that the isointensity paths are able to move more quickly in the y direction when $c < 1$. In addition, paths which before headed *away* from the minimum (which has $x > 0$ and $y = 0$ and moves towards $x = 0$ as time progresses) now head *towards* it. The extremal region associated with the minimum is clearly different in the anisotropic case from the isotropic blurring case.

Analysis of the more complicated case of stationary, anisotropic blurring when the ratio $c(\text{image}_i)$ of σ_x to σ_y is not held constant yields similar conclusions about the ability to change extremal regions. The following results are easy to prove for such blurring applied to the canonical formula $x^3/6 + tx + t + y^2/2$. The positions of the two critical points remain symmetric with respect to each other; that is, a plot of both the saddle and minimum positions versus time will have the time axis as a symmetry axis. The specific



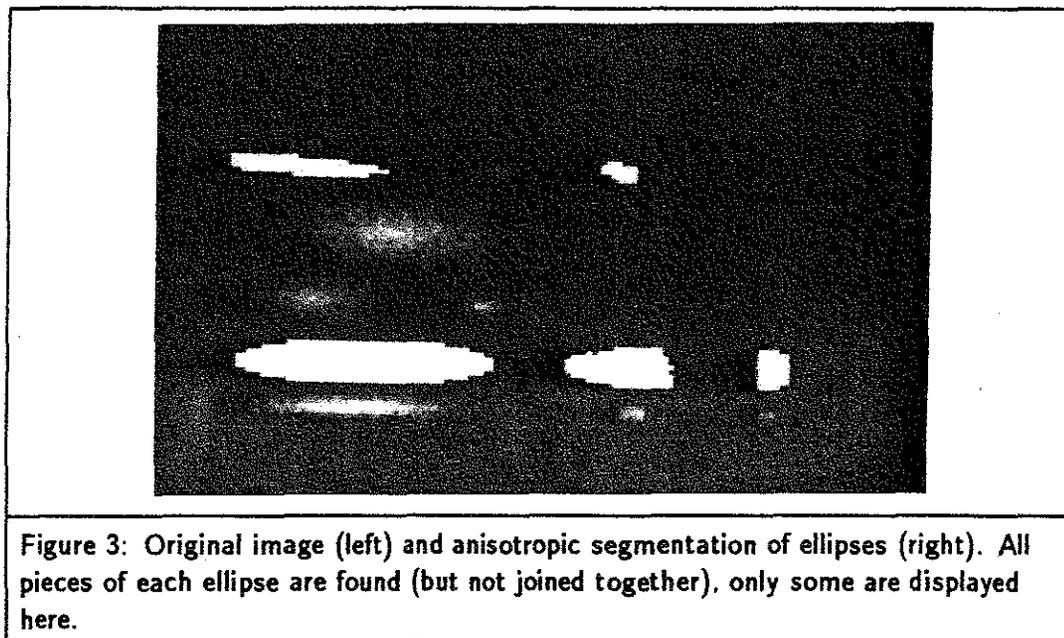
plots are not, however, the same as in the isotropic and constant $c(\text{image}_i)$ anisotropic cases. There will be no point along the x axis with fixed intensity (since " c " is now a function of time) for all time. The saddle may follow a complex series of increases and decreases in intensity as blurring proceeds, depending upon the function c . Nevertheless, it can still be shown that the intensity of the saddle and the intensity of the extremum will never cross each other. This last result agrees with physical intuition. For instance, a saddle (pass) between two peaks clearly must always be lower in intensity than either of the peaks it is separating. These results are not very surprising.

These results may all also apply in the stationary isotropic case when the domain of investigation shifts from the canonical description (which is a *local* description induced



from the initial image via suitable coordinate transformations) to more global untransformed descriptions of the blurring phenomenon. In other words, all extremum annihilations do not necessarily behave quantitatively the same as the canonical $x^3/6 + tx + t$ case. However, after suitable coordinate transformations (based upon equivalence relationships), all the annihilations *will* be quantitatively identical to the canonical case (at least for some time close to the annihilation time). If any particular extremum annihilation is studied without transforming to the canonical description, the intensity of the saddle may increase and decrease as in the anisotropic canonical case when c is not held constant.

Figure 3 shows the effect of isotropic and anisotropic blurring schemes on a synthetic image of ellipses. This demonstrates that segmentation results vary depending upon the orientation of the Gaussian kernel used to blur the image. Convolution of the image



with an isotropic Gaussian finds each ellipse as an extremal region. When this image is convolved with a Gaussian which has four times the standard deviation in the y direction (up-down axis on the page) as in the x direction, each ellipse is not found as a single extremal region. As one would expect, the results are worse since the ellipses are oriented at a 90 degree angle to the Gaussian kernel. The algorithm is robust enough not to merge pixels from several neighboring ellipses into one region, but it also does not link all the pixels in one ellipse into one region. Each ellipse is represented by several regions, each of which independently links up to the last remaining extremum (instead of to each other). The second image in Figure 3 shows several of these regions. Despite this fragmentation of ellipses into multiple regions, the post-processing discussed in chapter 6 (“displaying a union of subtrees”) would allow us to view each ellipse in its entirety.

2 Nonstationary Blurring

Spatially variant blurring based upon local image content should improve the correspondence between the extremal regions and the semantically meaningful regions in the image. This statement is motivated by the human visual model of Cohen and Grossberg [Cohen and Grossberg, 1984], which indicates that object perception is performed via an intensity diffusion process (i.e., blurring) which is moderated by edge strength measures in the image (i.e., it is nonstationary). The precise way that local image content should guide nonstationary blurring has not yet been resolved. Below I discuss the results of a preliminary investigation into various aspects of nonstationary blurring.

2.1 Mathematics of Nonstationary Blurring

Nonstationary blurring is multiplication and integration of an image with a kernel whose shape varies depending upon its current position. In one dimension this is represented as $g(x) = \int_{\mu} f(\mu)h(x-\mu; x) d\mu$. Here $h(x-\mu; x)$ depends not only on $x-\mu$, but also on x . Unlike the anisotropic stationary case, nonstationary blurring is not equivalent to a simpler blurring scheme applied in some other coordinate system. This can be shown for a simple case as follows. Imagine a one-dimensional case where the nonstationary blurring of $I(x)$ is with a Gaussian with σ_1 for all $x < 0$ and with a Gaussian with $\sigma_2 = k\sigma_1$ for all $x \geq 0$. If there is a coordinate system in which an equivalent stationary blurring can be performed, it must be a rescaling of the positive coordinate axis by a factor of $1/k$. This is clear since in the limit as σ_1 and σ_2 go to zero, the image values at positive and negative x coordinates interact with each other less and less and therefore the two parts of the x axis can be treated separately. To show that the nonstationary blurring is not equivalent to stationary blurring, it is enough to show that at least one point does not have the same intensity after the nonstationary blurring as it does when stationary blurring is performed in the rescaled coordinate system.

Let us examine the intensity at a specific point $x_1 > 0$. Does its intensity after convolution with a Gaussian with σ_2 equal its intensity after convolution with a Gaussian with σ_1 and requisite coordinate transformations? In other words, does

$$\frac{1}{\sqrt{2\pi\sigma_2}} \int_{-\infty}^{\infty} I(x)e^{-(x-x_1)^2/2\sigma_2^2} dx \stackrel{?}{=} \frac{1}{\sqrt{2\pi\sigma_1}} \int_{-\infty}^0 I(x)e^{-(x-x_1/k)^2/2\sigma_1^2} dx + \frac{1}{\sqrt{2\pi\sigma_1}} \int_0^{\infty} I(xk)e^{-(x-x_1/k)^2/2\sigma_1^2} dx \quad (2)$$

The left hand side of this equation represents the results of nonstationary blurring. It results from the convolution of a Gaussian with a standard deviation of σ_2 with the image (convolution with a Gaussian with standard deviation σ_1 occurs for points $x < 0$, but that need not concern us here). For a specific point this is just the integral of the product of a Gaussian centered at that point with the image itself. The right hand side of the equation represents the results at point x_1 of convolving a rescaled image with a Gaussian with standard deviation of σ_1 (convolution with a Gaussian of this standard deviation occurs for all points). We have rescaled the positive half of the x -axis by the factor $1/k$ (the ratio of the two σ 's). This rescaling moves the point of interest from x_1 to x_1/k . Therefore, the result of the convolution at this point is again just the integral of the product of a Gaussian (with $\sigma = \sigma_1$) centered at x_1/k with the rescaled image. This is broken up into

two parts. The image for $x < 0$ has not been rescaled. Its contribution to the integral is the first term on the right hand side of equation 2. The second term is the contribution from the rescaled part of the image. It is identical to the first except that $I(xk)$ is used instead of $I(x)$. $I(xk)$ is the intensity of the rescaled image at x ; this can be understood as follows. Since the coordinate system was scaled by $1/k$, a point in the rescaled image at x/k has the intensity of the point in the original image at x . Therefore a point in the rescaled image at x has the intensity of the point from the original image at xk . Now rewrite the right hand side by performing the variable substitution $z = xk$. This yields

$$\frac{1}{\sqrt{2\pi\sigma_1}} \int_{-\infty}^0 I(z/k) e^{-(z/k - x_1/k)^2 / 2\sigma_1^2} dz/k + \frac{1}{\sqrt{2\pi\sigma_1}} \int_0^{\infty} I(z) e^{-(z/k - x_1/k)^2 / 2\sigma_1^2} dz/k.$$

By substituting σ_2 for $k\sigma_1$ this can be rewritten as

$$\frac{1}{\sqrt{2\pi\sigma_2}} \int_{-\infty}^0 I(z/k) e^{-(z-x_1)^2 / 2\sigma_2^2} dz + \frac{1}{\sqrt{2\pi\sigma_2}} \int_0^{\infty} I(z) e^{-(z-x_1)^2 / 2\sigma_2^2} dz.$$

Since z is a dummy variable of integration, the second term is now identical to the left hand side of equation number 2, except for the limits of integration. Subtracting this from both sides yields

$$\frac{1}{\sqrt{2\pi\sigma_2}} \int_{-\infty}^0 I(x) e^{-(x-x_1)^2 / 2\sigma_2^2} dx \stackrel{?}{=} \frac{1}{\sqrt{2\pi\sigma_2}} \int_{-\infty}^0 I(z/k) e^{-(z-x_1)^2 / 2\sigma_2^2} dz.$$

Since, in general, $\int f(x)g(x) dx \neq \int f(x/k)g(x) dx$, nonstationary blurring is *not* the same as stationary blurring applied to a transformed coordinate system.

One might also wonder whether nonstationary blurring can be considered as a non-planar slice through an image-resolution space generated via stationary blurring. In other words, can each output pixel be considered to be a Gaussian (centered at the output pixel) weighted average of the surrounding input pixels, with each Gaussian possibly having a different variance? Certainly for any nonstationary, isotropic Gaussian blurring applied once to an image the answer is yes. This is clear since the conditions set forth in the question precisely describe the definition of a nonstationary blur of an image. The answer is quite different if one is concerned about whether the output of several successive nonstationary blurrings can be considered as a nonplanar slice through an image-resolution space produced by stationary blurring. The most important fact for analyzing this situation is that two successive nonstationary blurrings with Gaussian kernels is, in general, not equivalent to any one nonstationary blurring with any other Gaussian kernel. This can

be demonstrated quite simply in one dimension. Suppose the two separate nonstationary blurrings are $g_1(x) = \int_{\mu} f(\mu)h(x - \mu; x) d\mu$ and $g_2(x) = \int_{\nu} g_1(\nu)h(x - \nu; x) d\nu$. Then,

$$\begin{aligned} g_2(x) &= \int_{\nu} \int_{\mu} f(\mu)h(\nu - \mu; \nu) d\mu h(x - \nu; x) d\nu \\ &= \int_{\mu} f(\mu) \left[\int_{\nu} h(\nu - \mu; \nu)h(x - \nu; x) d\nu \right] d\mu \end{aligned}$$

Therefore the two successive blurrings are equivalent to one convolution with the kernel given in brackets. This function is not a Gaussian even when $h()$ is.

2.2 Blurring Strategies

There is no simple relationship between embeddings produced by nonstationary blurring schemes and stationary blurring schemes. Nevertheless, many nonstationary schemes can satisfy the causality and smooth embedding constraints. Which nonstationary approach should be used to best insure a close relationship between extremal regions produced and the semantic regions present in an image? Let us first examine the unrealistic, but simple and hopefully enlightening, case when exact object locations and boundaries are known. The problem is to determine the best way to use this information to guide the nonstationary blurring. Any blurring scheme must address several questions. The scheme must not only be able to decide how much to blur inside an object, but also how much to blur between objects. A related topic is how to switch between objects of different scales. A pixel is in many objects, each of which is a subobject of a larger object. How does one decide which of these object dimensions to use as a guide for blurring? Several approaches seem reasonable at first glance. They are discussed below.

1. We want to make it more difficult for isointensity paths to cross out of one object and into another. One way to do this is to "slow time down" for paths near the boundaries. In other words, blur at a slower rate near the boundaries of regions and at a higher rate near the center. Since the rate of blurring is slower along the edge, it will take longer for a path which would normally escape to do so. The amount of blur at the center may be based upon the size of the region.

A significant problem with this approach is that the local extrema (due to noise, perhaps) which exist near the boundaries of regions tend to remain for a long time, since minimal blurring is performed near the boundaries. The center of the region becomes smooth with one extremum while the region remains ringed with other extrema.

2. A different line of reasoning leads to opposite conclusions as to what to do. Pixels near the edge of a region must move the most before they link up with the extremum (on the average, assuming the extremum is near the center). Therefore edge regions should be blurred the fastest so that all pixels in the region reach the extremum in approximately the same time frame.

This would speed up the entire stack process, but it is not clear it would improve the correspondence of extremal regions with meaningful objects. It may even create more problems, since stronger blurring near the edges of a region will tend to cause it to merge with neighboring regions quicker.

3. A third approach is to move away from Gaussian blurring by forcing those parts of the filter kernel falling outside the region to be zero. In other words, use a "Gaussian" kernel but set the contribution to the convolution equal to zero for any part of the kernel which falls outside the region of the pixel it is centered at. This allows a rapid rate of blurring near the edges, alleviating the problem with the first approach above, while still minimizing the probability that a isointensity path will leave the specified region, thus handling the problem with the second technique.

The way to view this approach inside the diffusion framework is to think of each region (Ω) as completely isolated (this isolation might change at a later point in time when the decision is made to blur based upon a larger, containing region or if the paradigm is changed to allow some blurring across borders). The total intensity within a region will not change. Eventually the region will be of one intensity, its initial average intensity. This form of blurring is equivalent to changing the boundary conditions on the diffusion equation to read $\frac{\partial I}{\partial n}(x, y, t) = 0$; $(x, y) \in \partial\Omega$, $t \geq 0$, where $\frac{\partial I}{\partial n}$ denotes the directional derivative of I in the direction of the normal \mathbf{n} to $\partial\Omega$. That is, the border is insulated so no intensity flows across it. If the initial boundary intensity of the region is also held fixed, the solution to such a diffusion (heat) equation problem is precisely that which we used on the image as a whole when performing the standard stationary, isotropic blurring. It is just blurring with a Gaussian kernel inside the boundary while setting contributions from outside the region to zero.

This approach presents its own problems. Since it uses a very precise cutoff at the boundary, once it is applied to applications in which the region boundaries are not precisely known it may be very unforgiving of errors. Also, it tends to

create images with sharp changes in intensity across the region boundaries, and very uniform intensity distributions inside the regions. These are very non-Morse images, and hence the differential topology theorems do not apply to these. On a specific practical note, when blurring is finally allowed across a boundary, many extrema will be created because the ridge of constant intensity along the boundary will get worn away unevenly, creating pits and peaks. However, realistically the region boundary would not be precisely known. Therefore, the insulation of the region would be partial and consequently extrema creation would be minimized.

Chapter 6

Use of the Stack Program

1 Implementation Performance

The implementation of the stack and display algorithms has not been optimized for speed of execution or minimization of space requirements. Some indications of the speed and size of the implementation are nevertheless in order. The stack program has been applied to about 15 two-dimensional CT images of the upper abdomen. Approximately five one-dimensional and three-dimensional images have been analyzed also. To provide faster results for testing purposes, most of the CT images have been reduced from an initial size of 512 by 512 pixels to 64 by 64 pixels. Running on a moderately loaded VAX 780, it takes approximately 30 to 45 seconds to create each level in the stack, together with all of its associated data structures. The 64 by 64 images tend to need about 30 levels of blurring before only one extremum remains. Thus the program runs about 20 minutes. The image description tree created is approximately 250 kbytes.

All of the above numbers scale approximately linearly with image area. Of course, an image with a lot of noise or very many objects will tend to take longer and create a larger data structure. If it is known in advance that the structures of interest in an image are of small scale, the processing may be terminated before only one extremum remains, saving time.

The table below lists typical subroutine execution times when applied to a 128 by 128 pixel image.

% time*	total secs	# of calls	ms/call	name
42.2	6,695.52	59	113,483.40	blur image
18.8	9,675.96	1	2,980.44	calc. invariant image
13.9	11,876.58	1,294,414	varies	linking overhead
8.1	13,169.04	59	21,906.16	diagnostics
6.4	14,182.78	592,108	1.71	link nonextrema
4.2	14,848.05	60	11,157.89	find extrema
1.7	15,117.32	65,248	4.12	link extrema
0.1	15,135.14	5,273	3.38	annihilate extrema

* Not all routines are shown, so times may not add to 100%

Note that the vast bulk of the time is spent performing the blurring. A large percentage of time is also spent calculating the image that is invariant under Gaussian blurring and has the same boundary values as the initial image. This algorithm was chosen for ease of implementation without regard to execution speed. I believe that the stack program execution time can be decreased by at least a third to a half without the use of any special hardware.

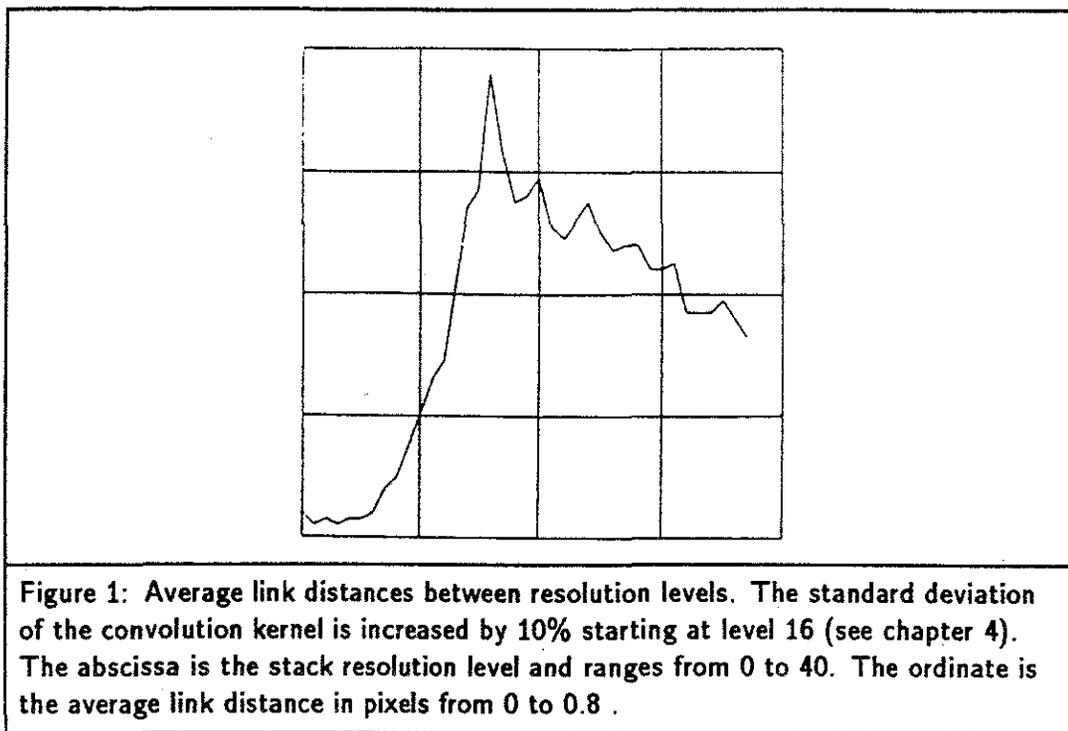
Typical relative frequencies of link types and typical distances for several blurring levels are shown below (not all link types are shown so numbers may not add exactly).

Level 5 of 38 levels for a 64 by 64 image			
link type	# of links	avg dist	max dist
nonextremum to extremum	48 = 1%	1.29	2.00
extremum continuing	456 = 11%	0.01	2.00
extremum annihilating	1 = 0%	2.00	2.00
nonextr. to annihilating extr.	2 = 0%	1.50	2.00
nonextremum continuing	3,477 = 86%	0.01	3.00
total of all links	4,001 = 100%	0.03	3.00

Level 15 of 38 levels for a 64 by 64 image			
link type	# of links	avg dist	max dist
nonextremum to extremum	147 = 4%	3.22	11.00
extremum continuing	170 = 5%	0.12	3.00
extremum annihilating	6 = 0%	4.83	9.00
nonextr. to appearing extr.	24 = 0%	1.25	2.00
nonextremum continuing	2,555 = 86%	0.42	9.00
total of all links	2,946 = 100%	0.56	11.00

Level 35 of 38 levels for a 64 by 64 image			
link type	# of links	avg dist	max dist
nonextremum to extremum	17 = 0%	4.24	7.00
extremum continuing	129* = 6%	0.01	1.00
extremum annihilating	0 = 0%	0.00	0.00
nonextr. to appearing extr.	0 = 0%	0.00	0.00
nonextremum continuing	1,821 = 92%	0.36	5.00
total of all links	1,967 = 100%	0.37	7.00

* All but one extremum are along the edge and will not annihilate

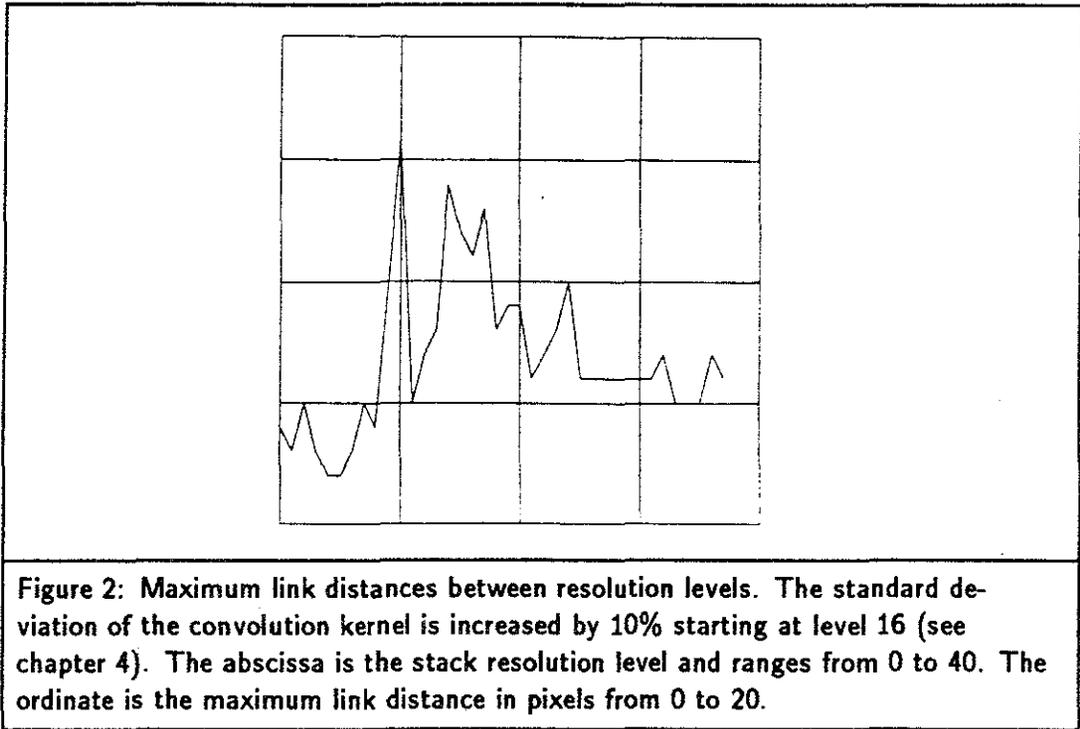


The path link types maintain approximately the same relative frequencies through a wide resolution range. Average link distances for nonextremum paths of less than one pixel indicate that the blurring rate is slow enough to minimize errors in following isointensity paths.

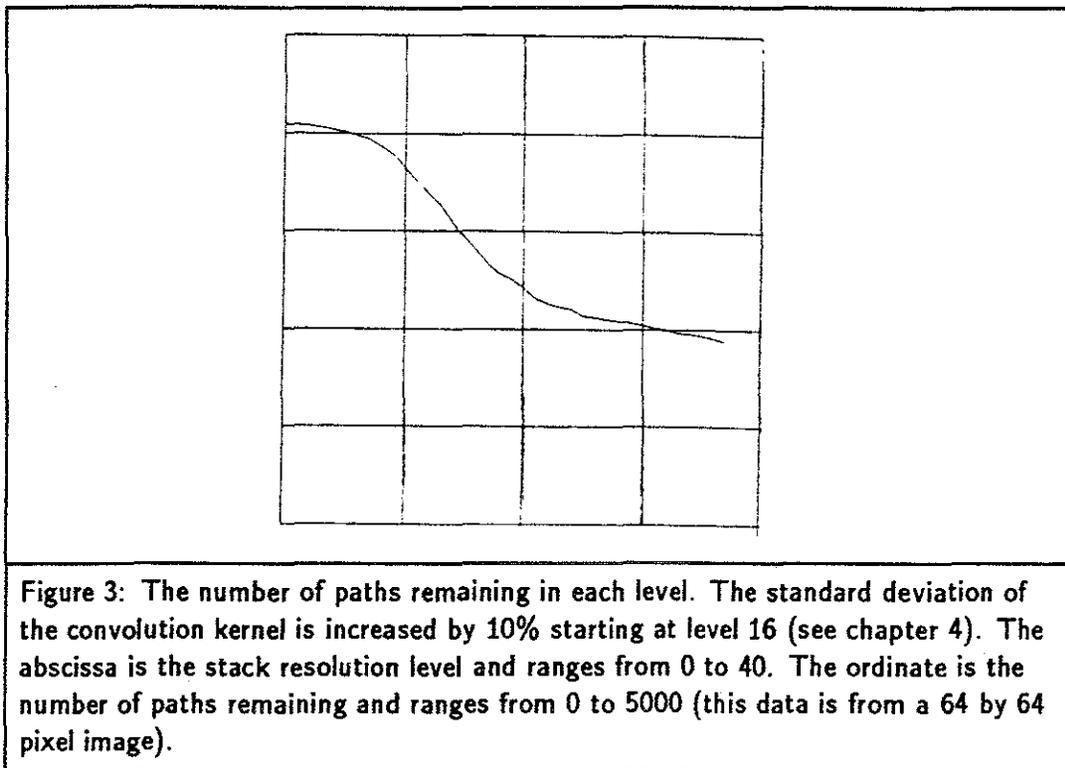
Typical average and maximum link distances as a function of resolution level are shown in Figure 1 and Figure 2. The link distances are averaged over all link types. Figure 3 is a graph of the number of paths remaining at each level in the stack (this number decreases as isointensity paths link up to extremum paths).

2 Interactive Display Based on the Image Description Tree

I have investigated several different interaction methods with the data structure produced by the stack algorithm. The display techniques developed permit easy specification of the region of interest to be examined and also yield insight into why certain subregions nest the way they do. The display program initially reads in the image description tree from a file produced by the stack algorithm. A user can then interactively control which regions in the image (subtrees in the data structure) are displayed. Several methods for specifying these regions are provided. These methods are discussed at length below.



The image is displayed on an Adage 3000 raster graphics system. A vector representation of the tree of extremum paths (along with the nonextremum paths they turn into) can be displayed simultaneously with the raster image. The vector image is displayed in three dimensions (x, y, t) on an Evans and Sutherland Picture System 300. This is a color vector graphics system. Paths of minima and maxima are displayed in different colors (see Figure 4). This tree can be interactively rotated using knobs to specify the rotation about each axis. This display gives the user a very good feel for the way the extrema in the image have moved and merged together during the blurring process. This tree can be interrogated by picking any branch using a light pen. The extremal region associated with this branch will then be displayed on the raster display device. This has turned out to be a very powerful tool for visualizing the relationship between the tree structure produced and the original image. Frequently a major organ can be displayed quickly by using a light pen to choose the major tree branch (i.e., a branch that exists until very low resolution levels) in the location of interest. A visual examination of the tree structure sometimes focuses attention on high interest areas. For instance, noting that the extremum path representing the stomach region has a smaller branch (subregion) associated with it can focus attention on that subregion, which may be a tumorous mass.



A different visualization of the tree can be presented on the vector display. This version represents each region by a cube whose side area is proportional to the area of the region represented. Each cube is displayed at a depth position related to the resolution level at which its extremum path annihilated, and at the (x,y) position of the annihilation point. A region which is nested inside another region is shown as a cube connected by a line (branch) to the other cube (which will be larger and at a greater depth). This tree representation shows the nesting relationship between regions very clearly. Figure 5 is an example of this type of display.

The tree data structure can be examined more globally via various A/D devices. Two sliders are used to specify a scale range of objects which should be displayed. The two sliders specify the low and high scale limits of the range of interest. The scale of a region is defined to be the blurring level at which its associated extremum path annihilates. All extremum paths which annihilate within the scale range specified by the sliders have their associated extremal regions displayed on the raster display. This method of choosing regions has been very useful. Organs in the CT images can frequently all be chosen simultaneously simply by setting the sliders to display only regions of large scale. The

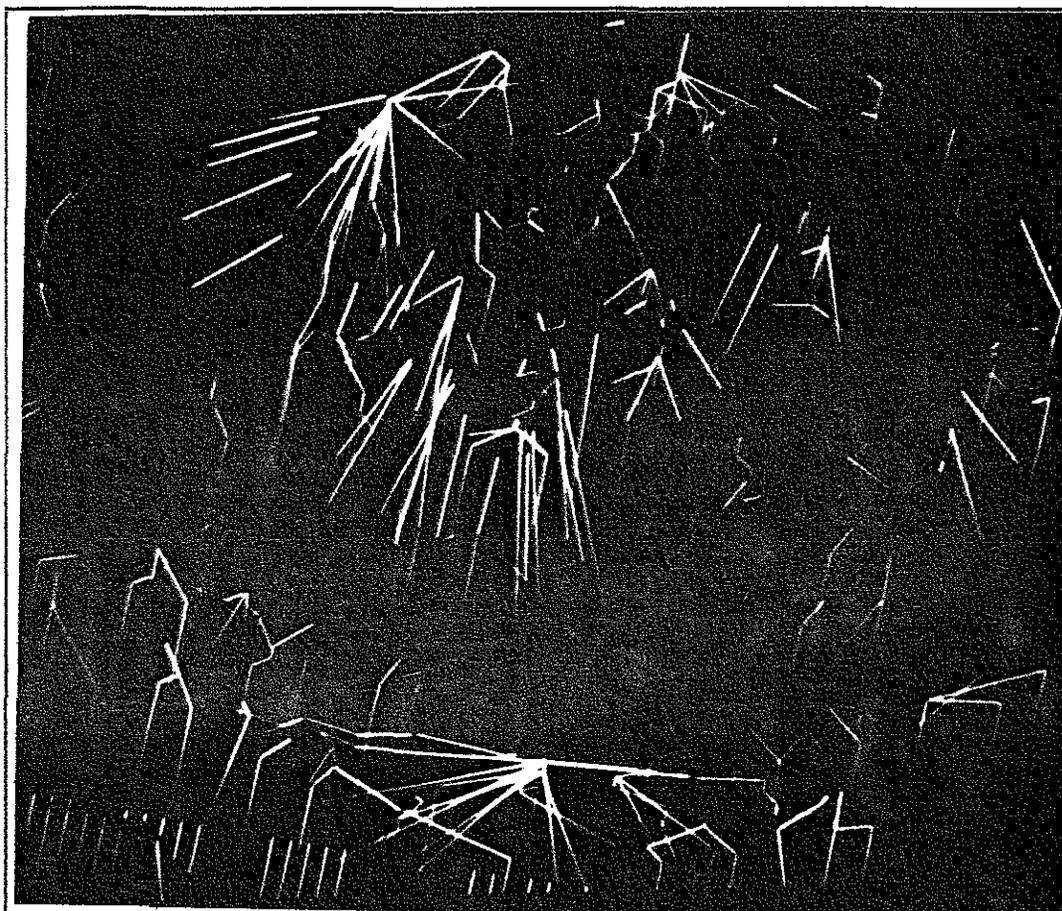
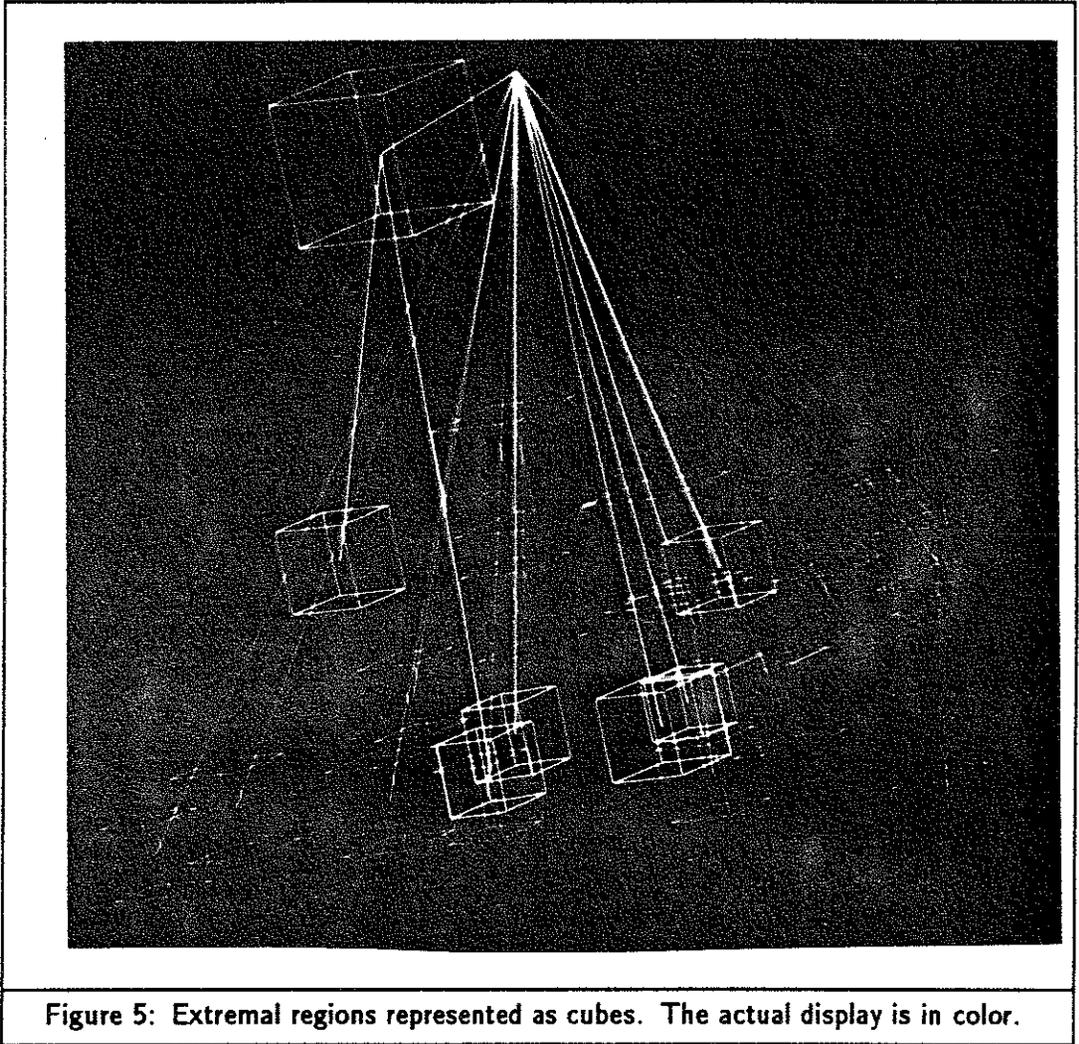


Figure 4: Paths of maxima and minima. Roughly half of these paths are minima and half maxima. The actual display is in color; this clearly differentiates between the two types.

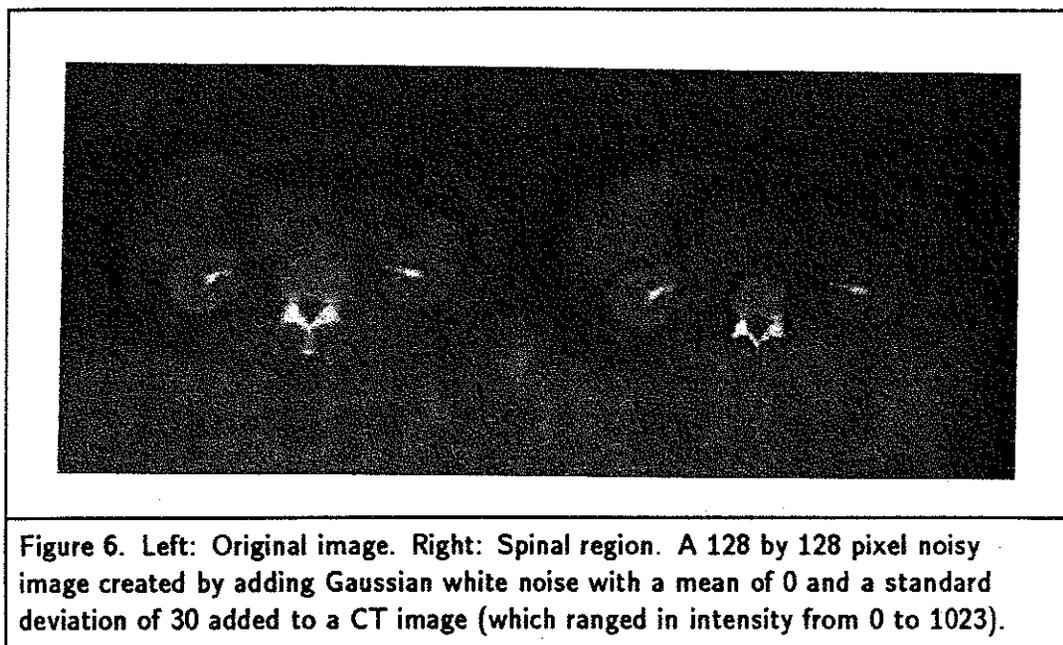
length of time it takes to display the specified regions depends upon the number of regions specified. Typical display times range from about two to five seconds.

Similarly, two other sliders specify the intensity range of objects to be displayed. Even if the annihilation level of an extremum path lies within the specified scale range, it is only displayed if the average intensity of its associated region lies within the intensity window selected by the intensity sliders. Intensity windowing is useful when the regions of interest (or regions of disinterest) are most easily distinguished based upon their intensity. This would be used, for example, to select bright objects like the spinal column in the abdominal CT images or eliminate dark regions such as bowel gas. The display of extremal regions can also be constrained based upon the (x, y) position of the annihilation point



of the associated extremum path. Four knobs are used to specify the range of spatial locations (maximum and minimum x and y coordinates) within which extremum paths must annihilate in order to be candidates for display. The chosen extremum paths can also be highlighted on the Picture System 300 display for better visualization. Intensity and spatial windowing have not been used extensively. It is not yet clear how useful they will be in a production (e.g., clinical) setting.

There is a difficulty in displaying some of the lowest resolution (largest scale) structures in an image. This is easiest to explain by use of an example. The spinal column and its associated musculature are oftentimes the largest object in an abdominal CT scan. In this case an intensity maximum in the spinal column will become the extremum path in the image which lasts the longest under blurring. If this extremum path is picked with the



light pen and its associated extremal region is displayed, the entire image appears. This is so because all pixels in the image eventually link up to the last remaining extremum path. There is no subtree which explicitly represents *just* the spinal column region. I have attempted to deal with this problem by providing more flexibility in the display mechanism. Entire regions associated with an extremum path do not have to be displayed. Subsections can be specified. Instead of following down all links from an extremum path, the user can specify that only links which join up before a certain blurring level be traversed. By picking this level low enough (high enough resolution), much of the image can be eliminated from the display. This often allows the isolation of the region of interest (e.g., the spinal column). The spinal region shown in Figure 6 was specified in this manner (Figure 9 is a schematic of organ positions in an abdominal CT scan). This method works since the true object of interest is usually spatially the closest to the extremum which forms the longest extremum path. As such, pixels in this region usually join the extremum path sooner than the other pixels in the image. Alternatively, if a resolution level is specified which is not as high as one used for the spinal region, the entire body in the CT image may be displayed without any of the surrounding image (e.g., the table the person is resting on).

Instead of displaying regions, the user has the option of displaying edges of regions superimposed over the original image. This is often useful since the interpretation of an isolated region displayed out of context can be difficult. Superimposing the edge of a region

over the original image takes slightly more time than simply calculating the region itself. The time taken is highly variable depending upon the number of objects in the image, the number of objects specified to be displayed, and system load. The slower update rate does not significantly hamper interaction unless a large number of regions is specified.

Perhaps the most natural region specification method is via a cursor on the raster graphics console. By moving the puck on a data tablet, the user can position the cursor over a pixel in a region of interest (perhaps a pixel in the kidney). When a button on the puck is depressed, the smallest extremal region this pixel is in is displayed. Upon display, information about the region is shown on the user's terminal. This includes region size in pixels and average intensity of pixels in the region. The next larger extremal region is displayed when another button is pressed. Each successively larger region can be displayed until the root node of the entire tree is reached. This display method is the easiest to use, and will probably dovetail well with the future post-processing techniques.

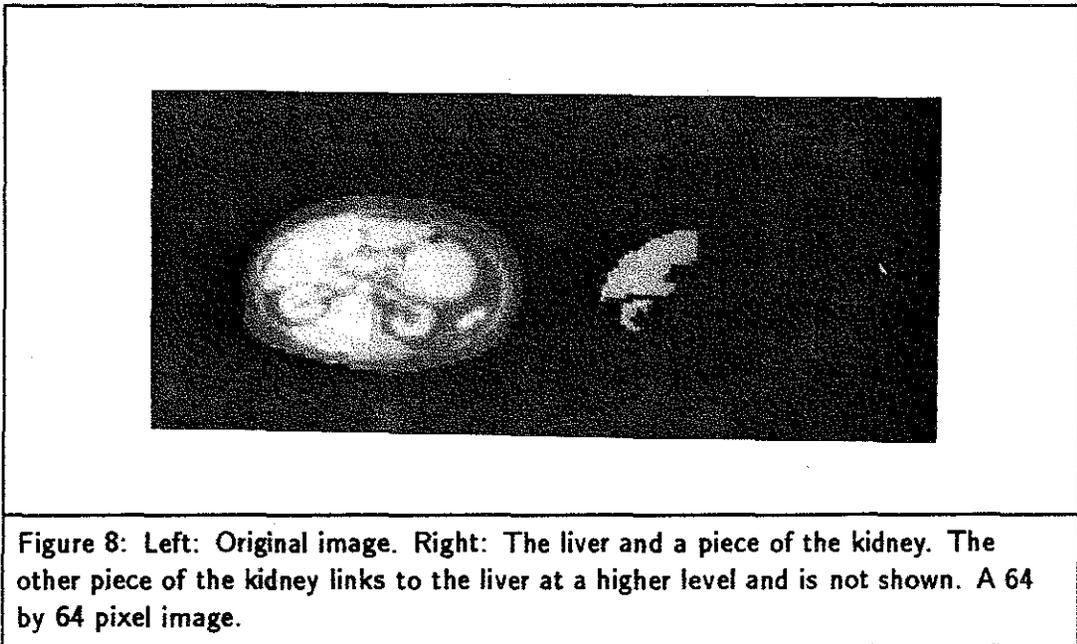
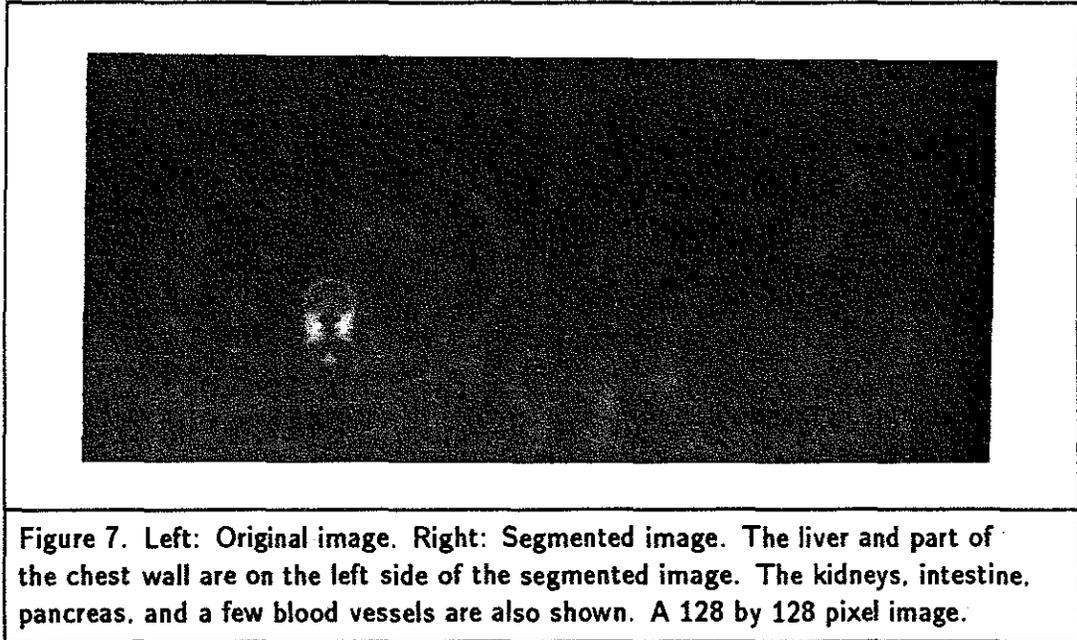
3 Future Post-Processing Techniques

The difficulties with the algorithm as it stands now are:

1. A region of interest might not be precisely represented by an extremal region.
2. A region of interest does not always show up as one explicit subtree in the tree structure. It may be two subtrees with no common root except the last extremum.

An example of the first case might be an extremal region which includes the liver, but also includes part of the chest wall near the liver (see Figure 7). An example of the second type might be the kidney. Due to its shape, the two halves of the kidney may be represented as separate extremal regions. If the kidney is far enough from other organs, these two region will join together (with one extremum path remaining) and be represented as one subtree. But proximity to the liver may cause each subtree to separately link to the liver extremum path instead of to each other (see Figure 8). If this is the case, there is no single subtree which will display just the kidney.

One way of dealing with these problems is to modify the basic stack algorithm. Various different blurring strategies based upon a priori and edge information seem promising (see chapter 5). An interactive post-processing step may be a simpler way to handle these difficulties. Various post-processing capabilities might be provided. A few of the most promising ones are discussed below.



Difficulties presented by the first case listed above could be mitigated by use of a simple pixel editor. This editor would allow use of the cursor to delete or add pixels to a region displayed on the console. If the region displayed is accurate except for a few pixels, it would be a simple matter to delete or add those pixels to the display. If desired, it

would also be simple to actually change the corresponding links in the data structure so that the data structure remains in accord with the display.

Problems arising from the second case (multiple disconnected subtrees) could be minimized by building a simple graphical editor for the tree data structure which is displayed on the Picture System 300. Using the light pen as a dragging device, the branch representing one of the kidney subtrees could be picked and dragged over to the other subtree and graphically joined to it. This graphical operation could then be used to guide a similar change in the actual data structure. Alternatively, disconnected subtrees can be dealt with by finding connected regions in the displayed image (as opposed to in the data structure). Even if the kidney is represented as two subregions which do not link to each other in any way, the capability currently exists to display both subregions simultaneously. This is done by picking each subregion separately while specifying that the previously picked region remain on the display. Once the entire kidney is displayed, it would be a simple matter to automatically find all the pixels which are in the one connected region representing the kidney. As long as the pixels in the region of interest can be displayed on the console as a connected, isolated region, that region could be easily be defined to be one object and information about it calculated.

4 Results

The results obtained so far have been encouraging. Many correct image segmentations have been produced. Presented below are examples from several different images; the images on the left side of each photograph are the originals. CT scans are oriented so that the view presented is as if the viewer is standing at the foot of a table that the patient is lying on (face up), looking toward the patient's head. Thus the left side of the image is the right side of the person and the spinal column is towards the bottom of the image. High density objects (such as bone) show up as higher intensity (whiter) regions in the image. A schematic of the organ locations is shown in Figure 9 (from [Lee *et al.*, 1983]). Which organs are actually present in a particular CT slice, and their size, depends upon the axial position of the slice and whether disease is present.

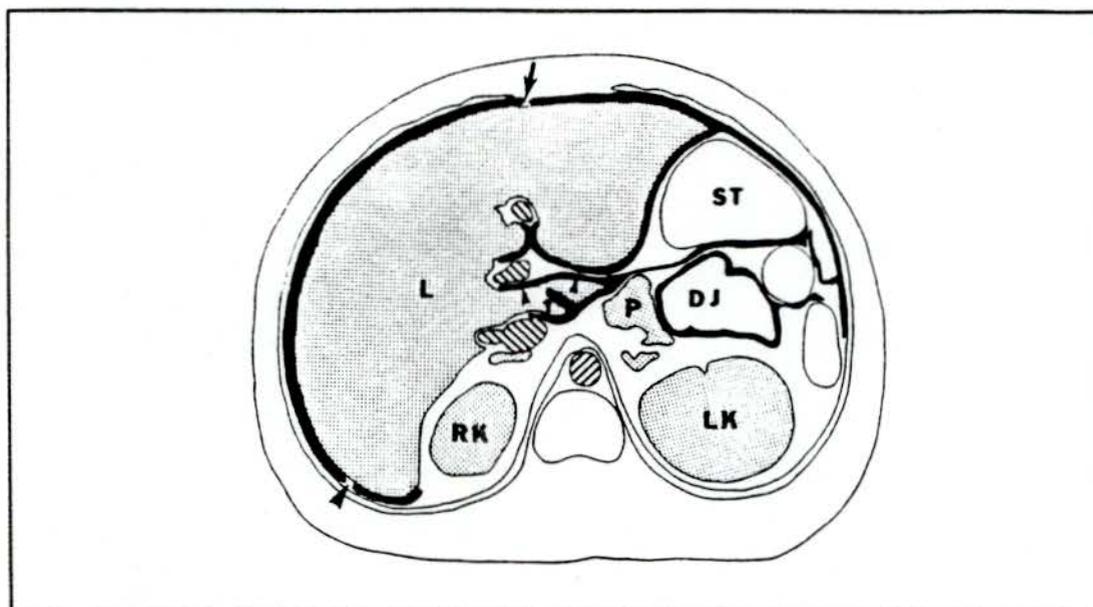


Figure 9: An anatomical model. RK-right kidney. LK-left kidney. L-liver. ST-stomach. P-pancreas. DJ-duodenojejunal flexure (part of the intestine). The white object in lower center is the spinal column. The cross-hatched region above it is the aorta. This figure is from Lee [Lee *et al.*, 1983].



Figure 10: The kidneys, liver, and some blood vessels (the aorta and inferior vena cava, running perpendicular to the image plane near the center of the image). The dark region near the kidney and the liver is a part of the intestine, as is the region near the left kidney (right side of image). The region extending out from the top of the left kidney is a large tumor, as is the very small region extending out from the bottom right part of the right kidney. A 128 by 128 pixel image.

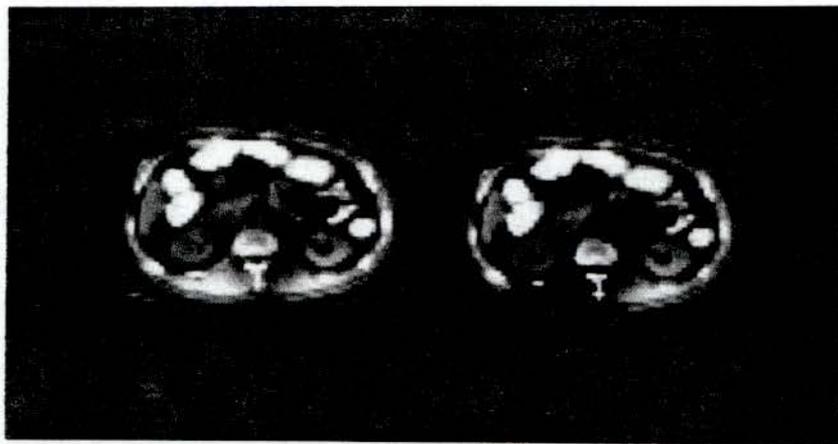


Figure 11: The intestines (white region running horizontally along the top), liver, kidneys, and spinal region. The pancreas is the triangular shaped region near the right kidney (left side of image). The image on the right does not have the dark regions interior to the body displayed (this is much clearer on the color display which is actually used). A 64 by 64 pixel image.



Figure 12: The kidneys, liver, spinal region, aorta, inferior vena cava, and pancreas. The two small light regions near the top of the pancreas are the superior mesenteric vein and artery. A 128 by 128 pixel image.



Figure 13: The kidneys, liver, aorta and inferior vena cava. This image is the image in Figure 6.12 with Gaussian white noise of mean 0 and standard deviation 30 added to it. The original image ranged in intensity from 0 to 1023.

Chapter 7

Conclusions and Future Directions

1 Conclusions

Stack-based image segmentation correctly isolates anatomical structures in abdominal CT images. Both small vessels running perpendicular to the image plane and large organs are successfully identified in many instances. The inaccurate segmentations are frequently close enough to the desired result so that a simple interactive post-processing step might produce a correct segmentation. Post-processing is necessary because sometimes two nearby objects are represented as one extremal region, or a few pixels are missing from (or added to) a desired object. It remains to be seen whether a system with simple post-processing abilities will be accurate enough, often enough, to be employed on a routine basis.

The main problem remaining is the incorporation of additional, probably imprecise, knowledge about the image into the stack segmentation algorithm. The knowledge might be edge strengths from the image itself, or perhaps model-based information. The challenge is to use this information in a manner which produces a more accurate segmentation without forcing the result to mimic the input knowledge.

2 Future Directions

As suggested in chapter 5, it would be useful to investigate various different blurring strategies. One approach might be to use edge strength information to guide the diffusion process in a manner similar to Grossberg [Cohen and Grossberg, 1984]. Post-processing graphical editing of pixel-to-region identification (as discussed in chapter 6) may also be extremely useful in insuring that region identification is accurate enough for quantitative information derived about a region to be useful.

It may be desirable to match the tree produced by the stack algorithm with a template tree for labeling purposes, identification of abnormalities (e.g., tumors), and guidance of the blurring process. Many tree matching algorithms exist. The properties

of the trees being matched determine which algorithm to use. Therefore, properties of the tree produced by the stack algorithm should be examined. Of primary interest are the stability and nesting characteristics of extremal regions. The exact nature of these characteristics will help determine how to model families of trees, how to modify the stack blurring algorithm to change the extremal region structure, and how to predict changes in the tree due to the presence of noise. One might also want to study under exactly which circumstances paths really do escape from their initial extremal region. A discussion of some of these issues follows.

7.2.1 Modeling Tree Stability and Deformations

Equivalence Relations. Only once an *equivalence relation* between trees is defined can there be a well defined answer to the question of whether two trees are the same and how much a tree has changed. The relation definition incorporates our notion of which changes are significant and which ones are not. One common equivalence measure between images is *topological equivalence*.

An image is a scalar function of points in the plane. Suppose there exist two images, that is, two functions $f, g: \mathbf{R}^2 \rightarrow \mathbf{R}$. We can define these two functions to be topologically equivalent if there is a homeomorphism $h: \mathbf{R}^2 \rightarrow \mathbf{R}^2$ such that $f = gh$ [Blicher, 1984; p 170]. Therefore, if f and g are topologically equivalent, h takes level sets (isointensity contours) of f into level sets of g . A type of tree called a *level set tree* can be constructed such that if f and g are topologically equivalent, the level set trees produced from the two images will be the same. Unfortunately, a level set tree is not the same as a stack tree.

A level set tree is formed from an image quite simply. Each extremum in the image is represented as a leaf node. Each isointensity contour through a saddle point in the image is represented as an interior node. If an extremum is surrounded by the isointensity contour through a saddle, its leaf node becomes a child of the interior node. Figure 1 (from [Blicher, 1984]) shows a hypothetical image, and Figure 2 (from [Blicher, 1984]) shows two slightly different level set tree descriptions of that image.

All topologically equivalent images can be represented by the same level set tree, but they may not be represented by the same stack tree. This can be visualized quite easily. A topologically equivalent image can be created from an initial image by dilating (expanding) the region around an extremum point. But there are *two* parameters which determine the range of scales for which a given feature (extremum path) in a Gaussian stack will exist. They are the geometric extent of the region *and* the intensity of the extremal region

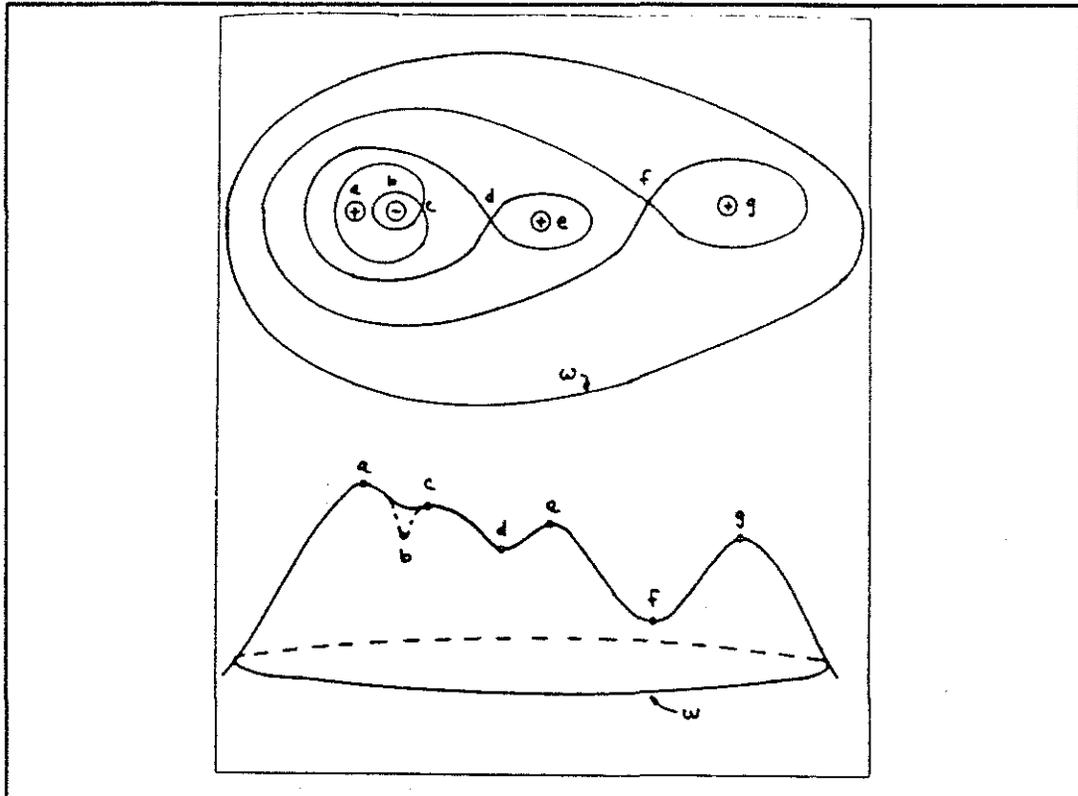


Figure 1: Isointensity contours through the saddle points of an image and a side view of the image. + marks a local maximum in intensity, - a local minimum.

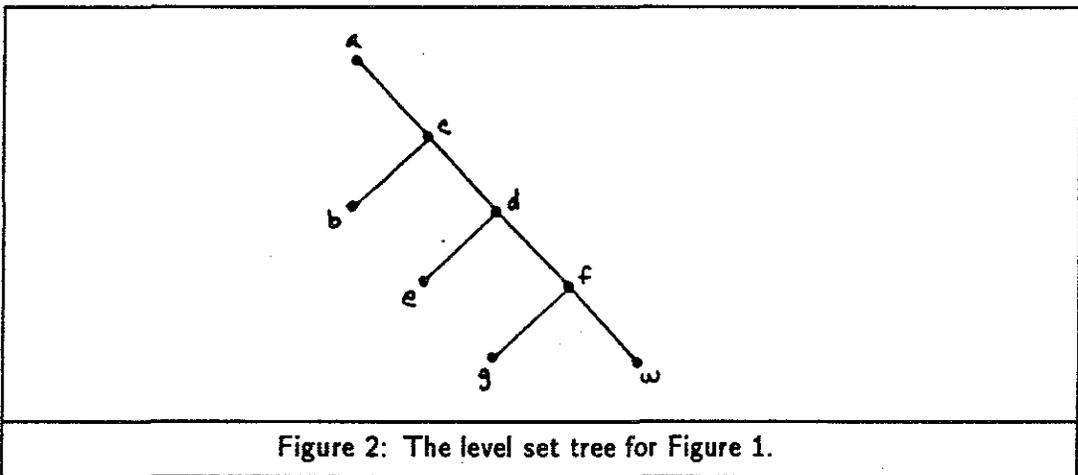
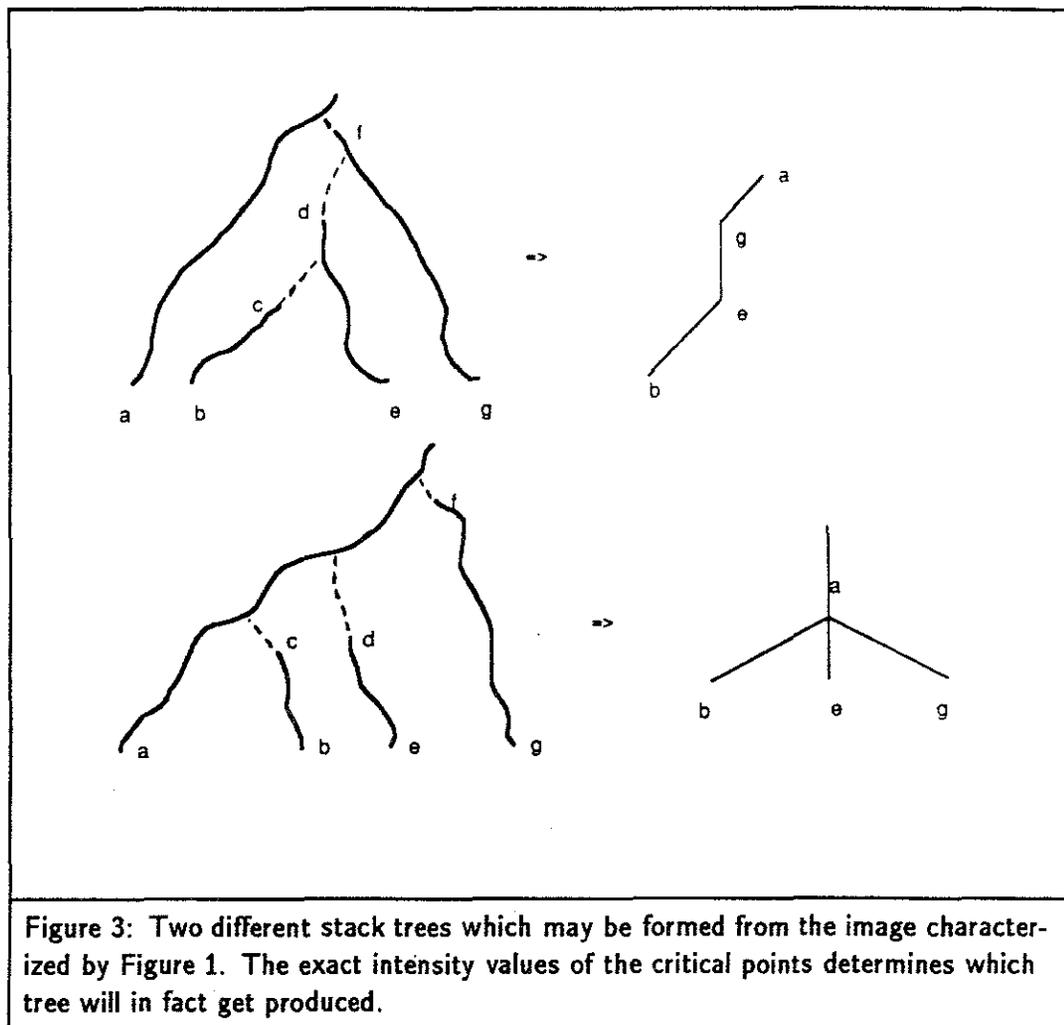


Figure 2: The level set tree for Figure 1.

relative to its neighborhood. Both of these factors are important perceptual clues as to the significance of a region in an image. Yet, combining them (by blurring) to determine



region importance prohibits any direct relationship between the *topology* of the level sets of an initial image and the stack tree which represents the image (see Figure 3). Since topologically equivalent images can yield different stack trees, more sensitive equivalences must be developed.

Dynamical Systems and the Stability of Paths. One way to analyze the stability of the stack tree under noise and blurring modifications is to rephrase the problem in terms of a *dynamical system*. The behavior of dynamical systems has been well studied. Any system of simultaneous first order differential equations can be thought of as dynamical system. The description of the iso-intensity path of a point in an image is a set of such equations: $\vec{v} = (-I_t I_x, -I_t I_y, I_x^2 + I_y^2)$. The path of the point is just the integral curve of this three-dimensional vector field. Integral curves of a vector field are just curves

which are always tangent to the vector field. The integral curves are also known as orbits or trajectories. The integral curves are solutions of a three-dimensional autonomous (time independent) system, namely, $\frac{dx}{dt} = -I_r I_x$, $\frac{dy}{dt} = -I_r I_y$, $\frac{dr}{dt} = I_x^2 + I_y^2$, where r is a variable introduced to represent the resolution dimension. This formulation is time independent since none of the right hand sides depend upon t . Alternatively, an equivalent two-dimensional nonautonomous formulation (time dependent) is $\frac{dx}{dt} = -I_t I_x$ and $\frac{dy}{dt} = -I_t I_y$.

Three main types of stability have been identified in dynamical systems work. They are as follows:

- 1 . Liapunov Stability: this is concerned with how much the path of a point changes if you perturb its starting point.
- 2 . Structural (Andronov-Pontryagin) Stability: this describes the qualitative changes in the structure of the orbit fields if the vector field is perturbed slightly.
- 3 . Bifurcation Theory: this investigates changes in the orbit structure due to changes in the form of the equations defining the vector field (not just perturbation in the coefficients as in structural stability).

Several different classes of vector fields with different characteristics for their orbits have been identified. Conservative fields are fields which preserve some quantity along each orbit. The isointensity curves of the stack preserve intensity! Therefore analysis of the paths should probably center on the study of the qualities of conservative fields. Most analyses are for fields with isolated sink points (locations where the differential equation is zero). In the three-dimensional autonomous representation for the stack each extremum path is a path of sink points. Therefore the sink points of the dynamical system representing the stack are not isolated. This makes the analysis more difficult. The recognition that there is still a well defined path for a point once it reaches an extremum (i.e., the extremum path) may solve this difficulty. If the definition of the extremum path can be added to the dynamical system description in a tractable way, it would have no sinks, but it would no longer be a conservative field.

Unfortunately, many of the results from a dynamical systems stability analysis may not be applicable to our problem. Most of the theorems are phrased in terms of ϵ (very small) perturbations and whether the resulting field is diffeomorphic to the original field. Thus, most of the analysis is for perturbations which are smaller than interest us, and the

definition of equivalence of orbits is too liberal to be applied to an analysis of structural stability of stack trees. Nevertheless, the sophistication and insight gained by studying dynamical systems may be useful. In addition, some of the qualitative results may be of direct practical value.

2.2 Tree Matching

As mentioned above, labeling regions in an image, detecting abnormalities, and guiding future blurring may all be possible by matching the tree produced by the stack algorithm to a prototype tree. Various prototypes may exist, one for each class of "objects" (e.g., different regions in the body). The goal in this case may be to decide to which of these classes of objects the candidate tree actually belongs. Alternatively, there may be only one prototype tree which we attempt to match with the candidate tree in the "best" possible way.

The specific type of match chosen will have an important effect. For instance, a symmetrical matching algorithm would not be a wise choice for our application. In a symmetrical match, if tree A matches tree B, then B matches A. At first thought this might seem to be a necessity for any "reasonable" matching scheme, but this is not so. For example, a prototype tree may contain only branches relating to large regions of interest in an image. Any calculated candidate tree would have not only these branches but also many more representing less significant image features. A match from a candidate with extra branches to a prototype with fewer branches would not be a cause for much concern; a match from a candidate to a prototype with *more* branches would be considered highly suspect. If we also want to be able to pick a *best* match, we need more than a set of rules deciding if two trees match; a *goodness of fit* measure (figure of merit, cost function, distance) is needed.

There has been much mathematical research done in the more general area of graph matching. The most exact match defined is a *relational isomorphism*. This is a symmetrical match. The least restrictive match, which allows many nodes in one graph to match the same node in the other, is a *relational homomorphism*. Even this match is probably not flexible enough for our purposes; it cannot deal with matching nodes which have real valued attributes associated with them (e.g., region size). It also is not able to recognize if a match is close, or pick the "best" of several matches. Shapiro [Shapiro and Haralick, 1981] generalizes the matching notion to an ϵ -*homomorphism*. This definition weights each relation in a graph. If that relation is not represented in the graph being

matched to, then a cost proportional to the weight is incurred. Similarly for attribute values associated with the nodes, if differences between the prototype and candidate are not within some acceptable threshold, a cost is incurred. Any attempted correspondence whose total cost exceeds some epsilon threshold is disallowed. If the cost is less than the threshold, a "match" is found. Haralick and Shapiro just use this information to determine whether an ϵ - *homomorphism* exists, not to rank matches.

Haralick and Shapiro present another interesting paradigm for deciding which of many prototypes a candidate graph matches best. We assume that the candidate is created by random fluctuations in one prototype due to noise and other influences. We can then calculate the probability that the candidate is an instance of each of the prototypes. A likelihood ratio can be calculated with the numerator being the probability that the candidate is an instance of a particular prototype and the denominator the probability that the candidate arose from a completely random structural description. The prototype match which maximizes the likelihood ratio is then chosen.

Wong and You [Wong and You, 1985] spell out the details of a philosophically similar approach. They define the notion of a *random graph*. In a random graph, each vertex and arc is a random variable. Each particular outcome of a random graph has a probability associated with it. Synthesis of a random graph from a set of observed graphs is also possible. By combining in this manner the tree structures derived from several similar images we could create a "random graph" prototype tree. Wong and You also define a distance measure between two random graphs based upon an entropy measure. Using this method both supervised and unsupervised classification of graphs is possible. Their technique is also capable of the automatic detection of subclasses in a set of observed graphs. This means that a series of images that differ due to some disease process might be automatically recognized as a subclass.

The relative strengths and weaknesses of the above mechanisms for creating prototypes and matching them to a candidate tree must be evaluated. As mentioned in chapter 2, Crowley's approach [Crowley and Sanderson, 1984] also seems very promising. Perhaps the strengths from several of these algorithms can be combined.

2.3 Additional Modifications to the Blurring Algorithm

Another avenue for investigation is the use of model or template tree structures to guide the blurring. A model tree might be produced manually, it might be the result of applying the basic stack algorithm to a model image, or it might be a composite tree (e.g., random graph) produced from applying the stack algorithm to many sample images of similar scenes. The present implementation of the stack algorithm can read in such a model tree and consult the model tree to guide the blurring at each pixel and level of the process. The question remains how to best use this information to guide the blurring process.

2.4 Linking Strategies

Segmentation results can be modified not only by changing the blurring algorithm, but also by changing the linking algorithm used to create the isointensity paths. Linking strategies which do not require paths to stay on an isointensity surface may prove more flexible and accurate in identifying semantically meaningful objects. An object which changes intensity significantly from one of its sides to the other (perhaps a blood vessel with varying amounts of contrast material down its length) may be more likely to be accurately delineated if isointensity contours are not the sole determining linking criterion. For example, links may be restricted from crossing over regions which in the original image have high edge strength values.

References

- Babaud, J., A. P. Witkin, M. Baudin, and R. O. Duda. January, 1986. "Uniqueness of the Gaussian Kernel for Scale-Space Filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1).
- Blicher, A. Peter 1984. Edge Detection and Geometric Methods in Computer Vision, STAN-CS-85-1041, Stanford University.
- Burt, Peter J., Tsai-Hong Hong, and Azriel Rosenfeld. 1981. "Segmentation and Estimation of Image Region Properties through Cooperative Hierarchical Computation," *IEEE Transactions on Systems, Man, and Cybernetics*, 11(12), 802-809.
- Chen, P. C. and T. Pavlidis. 1980. "Image Segmentation as an Estimation Problem," *Computer Graphics and Image Processing*, 12(2), 153-172.
- Cheng, Yao-Chou and Shin-Yee Lu. May, 1985. "Waveform Correlation by Tree Matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(3), 299-305.
- Cohen, Michael A. and Stephen Grossberg. 1984. "Neural Dynamics of Brightness Perception: Features, Boundaries, Diffusion, and Resonance," *Perception and Psychophysics*, 36(5), 428-456.
- Crowley, James L. and Arthur C. Sanderson. December 19, 1984. Multiple Resolution Representation and Probabilistic Matching of 2-D Gray-Scale Shape, CMU Technical Report - The Robotics Institute No.CMU-RI-TR-85-2.
- Crowley, J. L. and A. C. Parker. March, 1984. "A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2), 156-169.
- Feldman, Jerome A. and Yoram Yakimovsky. 1974. "Decision Theory and Artificial Intelligence: I. A Semantics-Based Region Analyzer," *Artificial Intelligence*, 5, 349-371.
- Fu, K. S. and T. S. Yu. 1980. *Statistical Pattern Classification using Contextual Information*, John Wiley & Sons, England.
- Haralick, R. M. July, 1983. "Decision Making in Context," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4), 417-428.
- Helmink, Leen 1984. "Extremum Following", Department of Medical and Physiological Physics, State University of Utrecht, Netherlands.
- Hong, Tsai-Hong, K. A. Narayanan, Shmuel Peleg, Azriel Rosenfeld, and Teresa Silberberg. 1982. "Image Smoothing and Segmentation by Multiresolution Pixel Linking: Further Experiments and Extensions," *IEEE Transactions on Systems, Man, and Cybernetics*, 12(5), 611-622.
- John, Fritz 1982. *Partial Differential Equations*, Springer-Verlag, New York.
- Koenderink, Jan J. 1984. "The Structure of Images," *Biological Cybernetics*, 50, 363-370.

- Kolmogorov, A.N. and S.V. Fomin. 1970. *Introductory Real Analysis*, Dover Publications, Inc., New York.
- Lee, J.K.T., S. S. Sagel, and R. Stanley. 1983. *Computed Body Tomography*, Raven Press, New York.
- Pizer, Stephen M. June, 1983. "Stack Sampling in the Resolution Direction", Personal notes, The University of North Carolina at Chapel Hill.
- Poston, Tom and Ian Stewart. 1978. *Catastrophe Theory and its Applications*, Fearon-Pitman Publishers Inc., San Francisco.
- Rosenfeld, A. 1984. *Multiresolution Image Processing and Analysis*, Springer-Verlag, Berlin.
- Shapiro, Linda G. and Robert M. Haralick. September, 1981. "Structural Descriptions and Inexact Matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(5), 504-519.
- Sokolnikoff and Redheffer. 1966. *Mathematics of Physics and Modern Engineering*, McGraw-Hill, New York.
- Spivak, Michael 1970. *A Comprehensive Introduction to Differential Geometry*, Publish or Perish, Inc., Berkeley.
- Toet, Lex, Jan. J. Koenderink, Cornelius N. deGraaf, and Peter Zuidema. 1986. "The Treatment of Image Boundaries in the Case of Progressive Defocussing", Internal Report, The Department of Medical and Physiological Physics, The State University of Utrecht, Netherlands.
- Toet, A., J. J. Koenderink, P. Zuidema, and C. N. Graaf. 1983. "Image Analysis - Topological Methods," *Information Processing in Medical Imaging: Proceedings of the VIIIth International Conference on Image Processing in Medical Imaging, Brussels, 1983*.
- Toussaint, Godfried T. 1978. "The Use of Context in Pattern Recognition," *Pattern Recognition*, 10(3), 189-204.
- van Os, Ron July, 1984. "The Inca-Pyramid Algorithm", Internal Report, The Department of Medical and Physiological Physics, The State University of Utrecht, Netherlands.
- Wong, A. K. C. and M. You. September, 1985. "Entropy and Distance of Random Graphs with Application to Structural Pattern Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(5), 599-609.
- Yuille, A. L. and T. A. Poggio. January, 1986. "Scaling Theorems for Zero Crossings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1).
- Zachmanoglou, E.C. and D.W. Thoe. 1976. *Introduction to Partial Differential Equations with Applications*, Williams and Wilkins Co., Baltimore.