Toward Discrete Geometric Models for Early Vision

Lee A. Iverson

Electrical Engineering

McGill University

Montréal

February, 1994

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

ISBN 0-315-94642-3

Abstract

Early vision is usually considered to involve the description of geometric structure in an image or sequence of images. Whether biological or artificial, the behavioural constraints on real-time visual systems typical require that this first stage of visual processing be fast, reliable, general and automatic. The design of a visual system which is general enough to handle a wide variety of tasks is thus most likely to be highly parallel, and involve distributed representations of geometric objects. In this work, we investigate some of these general principles and propose both general methodology and specific applications.

We build on a general theory of distributed, local representations which we call *thick traces.* Thick trace descriptions of continuous graphs preserve topological properties such as connectivity, and allow for the descriptions of multi-valued mappings.

Local operators for extracting image curves have been a focus of machine vision research for twenty years. Considered in the context of thick traces, however, we can reasses the goals of these operators and provide a clear description of when they should respond positively and when they should not. In order to achieve this behaviour, we develop an algebra, the Logical/Linear algebra, which incorporates features of both Boolean and linear algebra into a set of non-linear combinators. This algebra is then used to design a family of local operators which explicitly test the logical preconditions underlying the definition of an image curve.

Relaxation labelling is a highly parallel, distributed method of extracting consistent structures from a set of labels. There is a natural match between the representations used in relaxation labelling and thick traces. We exploit this connection by developing a general method for relaxing a set of potentially noisy initial estimates of thick traces (as produced by image operators) into descriptions which are thick traces of geometric models. Furthermore we show how such a system can interpolate into gaps in the traces while simultaneously respecting legitimate discontinuities and boundaries.

Finally, we apply these methods to two problems in early vision: the description of curves and texture flow fields. For image curves, the resulting descriptions of piecewise smooth curves include both local orientation and curvature information. The entire process accurately describes end-points, corners, junctions and bifurcations by allowing many consistent traces to be incident on a single point in the image.

The term texture flow is used to describe a class of static textures with locally parallel dense orientation structure (e.g. Glass or hair patterns). We derive a geometric model of these textures from a smooth non-deforming velocity field. Initial operators and a relaxation network are then defined to interpolate dense, piecewise smooth flow from sparse inputs. The resulting system produces accurate descriptions even in the presence of discontinuities, holes, and overlapping textures.

Resumé

La première étape de la perception visuelle est habituellement considérée comme nécessitant une description des structures géométriques d'une image ou d'une séquence d'images. Les contraintes typiques imposées au comportement d'un système visuel, qu'il soit biologique ou artificiel, requiert que cette première étape du processus soit rapide, fiable, générale et automatique. Ainsi, la conception d'un système visuel suffisamment général pour traîter une grande variété de tâches, se demarquera probablement par un haut degré de parallélisme et d'une représentation distribuée des objects géométriques. Dans cette thèse, nous étudions ces principes généraux et nous proposons une méthodologie générale et des applications spécifiques.

Nous nous basons sur une théorie générale des représentations locales distribuées que nous appelons *traces épaisses.* La description de la trace épaisse conserve les propriétés topologiques telles que la connectivité, et permet la description de relations à valeurs multiples.

Les opérateurs locaux pour extraire les courbes d'une image ont été un point de mire de la recherche sur la vision artificielle depuis vingt ans. Toutefois, considérés dans le contexte des traces épaisses, nous pouvons réévaluer les buts de ces opérateurs et fournir une description claire établissant quand ils doivent répondre positivement et quand ils ne le doivent pas. Afin d'obtenir ce comportement, nous développons une algèbre, l'*algèbre logique/linéaire,* qui combine les caractéristiques de l'algèbre booléenne et de l'algèbre linéaire dans un ensemble de prédicats non-linéaires. Cette algèbre sert ensuite à concevoir une famille d'opérateurs locaux qui testent explicitement les préconditions logiques sous-jacentes à la définition d'une courbe.

La méthode dite "*Relaxation Labelling*" est une méthode qui affiche un haut degré de parallélisme et qui se sert d'informations distribuées pour extraire une structure cohérente d'un ensemble d'étiquette. Les représentations de cette méthode vont de pair avec celles des traces épaisses. Pour relaxer un ensemble composé des évaluations potentiellement corrompues de la trace épaisse telles qu'obtenues par les opérateurs locaux, nous exploitons cette relation en développant une méthode générale en des descriptions qui sont des traces épaisses de modèles géométriques. De plus, nous montrons comment un tel système peut interpoler dans les trous des traces tout en respectant simultanément les discontinuités et les frontières légitimes.

Finalement, nous appliquons ces méthodes à deux problèmes en vision: la description des courbes et la description de l'aspect vectoriel des textures. Pour les courbes dans l'image, monotones par morceau, leur description se compose de l'information locale à propos de l'orientation et de la courbure. En permettant l'incidence de plusieurs traces cohérentes à un même point de l'image, ce procédé décrit avec exactitude les terminaisons, les coins, les jonctions et les bifurcations

L'aspect vectoriel des textures est utilisé pour décrire une classe de textures statiques ayant des structures parallèles denses (par ex., les cheveux et les motifs de Glass). Nous dérivons un modèle géométrique de ces textures à partir d'un champ de vélocité, monotone et qui ne se déforme pas. Des opérateurs locaux et un réseau de relaxation sont ensuite définis afin d'interpoler des champs vectoriels denses, monotones par morceau, à partir d'entrées clairsemées. Le système qui en résulte produit des descriptions exactes même en présence de discontinuités, de trous, ou de textures se chevauchant.

Acknowledgments

I would like to express my sincere gratitude to Steve Zucker, my advisor. He has created a wonderful environment for his students. His attitudes and style encourage us to always seek the most interesting problems and understand them deeply. Of all his recent students I have chosen to work on a subject that is probably closest to his heart, and yet he has not tried to force his own ideas on me (or perhaps he's just good enough at it that I don't feel pressured). For giving me the freedom and encouragement to understand, I thank you Steve.

One other thing that Steve must be given some credit for is gathering such a uniformly insightful group of students. Allan Dobbins has been a good friend for some time and has always been a source of inspiration. Michael Langer, Benoit Dubuc, and Pierre Breton have all made writing this document both more difficult and easier. In either case, they made it a much better thesis. Et Pierre, specialement pour le Resumé.

And thank you to the whole stimulating group at McRCIM, especially Jennifer, Mindle and Margaret. I'll miss you all.

To Alison, I can only say that words don't quite cover it.

Finally, I'd like to thank Bob Bolles, Yvan Leclerc, and Martin Fischler for providing sufficient incentive to actually finish this off.

Table of Contents

List of Figures

Part I

Early Vision and Representations

Chapter 1 Introduction

Vision is the process of interpreting and describing images. Early vision is the first stage in this process, taking raw measurements of light to the first *interpretations* of those images (e.g. there is a bright line here). While there is a broad consensus on some of the fundamental characteristics of this process, on certain major issues, such as the properties of the representations and descriptions involved (e.g. what a good solution looks like), there is little agreement. In this work, we will offer alternatives to some of the traditional assumptions about these representations, and will present computational methods which build on these alternatives.

It is perhaps best to begin by describing what we see as the consensus and then move on to points of disagreement. The term *early vision* is usually used to refer to those visual processes which

- are fast, automatic and unmodified by intention or motivation;
- are purely retinotopic (i.e. operate on maps of the retina); and
- describe the image in terms of general geometric properties, such as the presence or absence of one-dimensional discontinuities.

Thus, the description of curves in grey-scale images would qualify as an early vision problem, but mental rotation and object recognition would not. Some of the problems which are normally considered to be within the scope of early vision are: curve and texture description, stereo fusion, optical flow, and local shading analysis.

Disagreements arise, however, when discussions move from these general principles to their consequences and their application to real problems. For example, Marr [Mar82] suggested that the speed requirements of biological or real-time early vision precluded the possibility of considering iterative or global optimization procedures, and yet membrane [Ter84, BZ87], and regularization methods [PTK85], both of which involve global optimization, have been proposed as general theories of early vision.

Focusing in particular on the need for real-time operation or an engineering approximation to it, we make the assumption that early vision systems must be *local* and *parallel*. Moreover if they are iterative they must converge in no more than a few iterations in order to be predictably real-time. These constraints suggest the possibility of implementation with fast, special-purpose hardware [Mea89].

Given this context, we focus initially on the representations used for early vision. A *representation* is a language for delineating a class of structures and a *description* is a sentence in this language. Thus an image representation is a set of possible assertions about images. An accurate description of a particular image is one of these assertions which is true when applied to the image. Whereas some trivial descriptions are simply measurements (e.g. the intensity map of the retina), most early vision problems involve the detection and description of features in the image. In these cases, the selection of a particular description is a non-trivial assertion of the existence of certain features and the absence of others.

For example, if we wish to represent straight lines in images, our representation must allow for the description of all possible straight lines which could exist in an image. A description of a particular image would amount to the assertion that certain of these lines do exist in the image and that others do not.

To understand the effects of a choice of representation, we consider the problem of discontinuities. It has been suggested that one of the most fundamental organizing principles of a visual system is the detection and description of discontinuities. Physiologists [HW62, Orb84, HL87] have chosen to describe the organization of the mammalian visual system in terms of responses to spatially and temporally discontinuous inputs: edges, bars and points of light turning on or off. Psychologists [HR85, Ley88, Bie85] have likewise argued that much of perception is based on descriptions of discontinuities (e.g. bounding contours and their end-points, junctions and corners).

Computational vision researchers too have concentrated on the extraction of edges [MH80, Can86], corners [BA84, Lee90] and texture discontinuities [ZRD75, BPR83, Voo87, MP90] for much of the past twenty years. Yet most of these computational methods actually involve the systematic elimination or misrepresentation of certain discontinuities. Algorithms that rely on indiscriminate smoothing to combat noise are the most common culprit (e.g. [MH80, Can86]). While locating certain discontinuities (e.g. edges), they displace or destroy others (e.g. the corner points) [Lec85].

More subtle though is the problem presented by representations which make it difficult to find discontinuities. A good representation will either highlight important structural features or make them explicit, yet the usual machine vision representations for simple features often obscure these features. For example, in Canny's representation of edges [Can86], it is assumed that there is a one-to-one mapping from image points to edge directions. However, at corners and junctions there are discontinuities in edge orientation which appear as multiple edges incident on the same image point. Both the smoothing and the representation thus conspire to ensure that the resulting descriptions are *not accurate* at these points. Yet it is believed [Wal75, HR85, Bie85] that accurate location and description of these points are fundamental to the recognition of objects.

A related problem becomes apparent when we consider the case of optical flow. It is generally assumed that the problem of extracting accurate optical flow descriptions involves the assignment of a single velocity (possibly stationary) to every point in the image [Ull79, Hee87]. Yet the human visual system has no inherent difficulty in perceiving motion of partially occluded (e.g. behind a picket fence) or transparent objects. These are both cases in which there are appear to be multiple independent motions at the same point in the image [ZIH90].

In contrast to these one-value-per-pixel representations, which we call thin traces, *we will propose an alternative class of discrete representations which naturally support the description of multiple values per pixel, and thus of both discontinuities and transparencies.* This class, which we call thick traces, arises naturally from a reconsideration of how to describe a piecewise continuous graph on a discretely sampled space. We suggest that thick traces are a better choice than thin traces precisely because they facilitate the recognition of boundaries and discontinuities. The development of these thick trace representations and algorithms to extract them from images forms the body of this thesis.

The next step in our development arises from a reconsideration of the kind of reasoning needed to locate and describe curves in an image. Deciding that we will adopt a thick trace representation constrains, but does not determine the algorithms needed to extract these descriptions from images. Instead we reconsider the problem of designing a local operator to respond only when a curve with a given local geometry passes through a point in the image. We conclude that the decision implied by this goal is more complicated than can be provided by a simple threshold on operator responses, even when this is combined with local maxima selection [Can86, Har82]. The complications arise, in part, because of the need to ensure stable operator behaviour in the neighbourhood of end-points and multiple image curves.

Instead we suggest that the sign of contrast of the features being extracted be the only "threshold" involved in the decision procedure. Using linear operators as building blocks, we interpret positive responses as confirmation of associated logical hypotheses. In order to support this interpretation, it is essential to provide a logical foundation for making the decisions required (e.g. that the operator is centered on a bright line of a given orientation). We will therefore design a Logical/Linear (L/L) algebra which combines the behaviour of both Boolean and linear algebras. The operators designed with this algebra verify, rather than assume, the logical preconditions for the existence of the designed features *without incorporating an arbitrary threshold on significance.*

This leads to the final focus of this thesis, the question of what can be assumed to be a significant structure. Traditional approaches assume that some sort of threshold is applied to select points or regions of significance (e.g. [MH80, Can86, BZ87]). We suggest that this is an inappropriate method for the same reasons which lead to the introduction of the L/L algebra. There is more to this issue than just the pragmatic concerns of image curves, and in the end we question what an early vision system should do.

There seems to be an implicit assumption in the design and evaluation of vision systems that early vision processes should present to later processing stages only those features in the image which are salient and somehow "significant." The consequence of this assumption is often that a non-zero threshold on local contrast or correlation is used to separate the "wheat" from the "chaff." Yet this assumption seems to be at odds with the consensus that early vision is automatic and unintentional. If early vision were automatic, then it would have to deliver to later stages *everything that could possibly be attended to.* It would be the job of some attentional process to select which parts of this output are relevant to the task at hand. Given this assumption then, early vision systems should always operate at the boundaries of sensitivity of the visual system, and should produce descriptions of all verifiable features within their range of representation. The only non-zero thresholds in the system should depend directly on known measurement errors and noise in the system. The only fast adaptations should derive from the measurable noise or distortion in the image.

The criteria for selecting features should thus be entirely structural. If there is a predictable local geometry for certain features then it is the relationship between that geometry and the image which should act as a measure of the significance of features in early vision. For example, if a process extracts curves from the image, it should select them based on whether or not they are lines or edges, relatively brighter or darker than their backgrounds, and piecewise differentiable. It should not choose to select only those curves which exceed some arbitrary minimum contrast. In other words, it should depend only on *intrinsic* criteria and avoid the explicit or implicit incorporation of *extrinsic* criteria.

We suggest that beyond the criteria which relate the features directly to the image (e.g. whether or not they are lines or edges), the primary criterion for selection should be whether or not the local geometry of the features corresponds to some model of the image. This process thus involves the inference of connections between individual feature elements and the selection of those which appear to be part of a non-trivial structure—the thick trace of some continuous model. Thus the computational theory described in this document thus two stages:

1. Local Logical/Linear operators produce positive responses only when they can verify the existence of some feature with a specified local geometry in the image.

Because of image noise, confounding structure and operator imperfections, the responses are not guaranteed to group together into thick traces.

2. These responses are used to initialize a relaxation process which produces connected thick traces. This process is designed to converge quickly on only thick traces of piecewise smooth models.

This work is organized to follow the development above. In Part I, we analyse the discrete representation of continuous graphs on images. This discussion culminates in the adoption of a new kind of representation: the thick trace. In Part II, we design the Logical/Linear Algebra and design local image operators for extracting local descriptions of image curves. These descriptions are a first approximation to the thick traces desired. Finally, in Part III, we show how these initial estimates can be refined to incorporate intrinsic constraints from geometric models using relaxation labelling. The system relaxes to thick traces of piecewise continuous graphs on images. By starting with good approximations, the outputs of Logical/Linear operators, we ensure that the relaxation stabilizes after as few as three iterations. Finally, we demonstrate the application of the full theory for image curves and texture flow.

Throughout we rely on a number of organizing principles.

- Independence from the detailed structure of the sampling. *All analysis is applicable to both regular and random samplings of the image and image properties.*

- The avoidance of arbitrary thresholds. All decision procedures within the designed systems are based on the Logical/Linear algebra, and thus *all non-linearities depend on contrast sign.*

- The need to stabilize the location of discontinuities. Throughout, discontinuities in the image are either explicitly extracted (e.g. as edges and lines) or are stabilized so that *the locations of boundaries in the output coincide with their locations in the image.*

- Efficient exploitation of massive parallelism. All operations in the final systems are local *sums* augmented by simple, *pointwise non-linearities.* All operations *can thus be implemented with shallow feed-forward networks of simple, independent computational units.*

As a result of these principles we suggest that this work constitutes a computational theory of early vision. Although it is outside the context of this document, we also believe that these ides are relevant to a theory of the organization of processing in mammalian visual cortex.

1.1 Claims of Originality

- We define the class of early vision problems which this theory addresses as the extraction of discrete descriptions of cross-sections through fibre bundles (a generalization of graphs of functions). The class of problems covered thus encompasses static image maps, optical flow and three-dimensional vision.

- A new kind of discrete representation (thick traces) of continuous structures is defined and analysed. We conclude that this is better than traditional representations for early vision processing because it is a purely local representation which nevertheless allows for the straightforward recognition of boundaries and discontinuities, points of fundamental interest for early vision.

- An algebra for reasoning in the context of linear operators is developed. The combinators of this new algebra, the Logical/Linear algebra, exhibit both Boolean and piecewise linear properties.

- Smooth approximations to the Logical/Linear combinators are developed which allow for reasoning with uncertainty. This approximation is an interpolation between linear combination and the absolute L/L combination previously defined.

- Local L/L operators are designed which detect bright and dark lines and edges in images. These operators accurately categorize these features and operate stably in the presence of multiple curves and end-points. They are clearly good approximations to the thick traces of the underlying image curves.

- We formalize the relationship between geometric models (e.g. straight lines) and cross-sections of fibre bundles on images. The resulting analysis clarifies the criteria which may be used to recognize significant geometric structures among the outputs of image operators.

- Relaxation labelling is proposed as a theory for extracting these geometrically significant structures from image operator responses. With the analysis described above we show how to guarantee that the fixed points of a relaxation process are thick traces of geometric models. This involves the augmentation of the support network with L/L combinators.

- We demonstrate the viability of this theory by applying it to two distinct early vision problems: the extraction of image curves with orientation and curvature, and the interpolation of texture flow fields with orientation, curl and divergence. The resulting outputs are shown to be stable and accurate, and locate boundaries and discontinuities with precision.

Chapter 2 Discretization

Physical theories, or their models, are usually described in terms of continuous mathematics. When represented on a digital computer however, the data involved are represented on finite, discrete point-sets: samplings. What then, is the relationship between properties defined in the continuous spaces of our theories and discrete samplings of those spaces?

Sampling theory [Raj68] and numerical analysis [Bur89] are two fields that address this question. While they are both relevant to vision, we seek to develop a means of thinking about the process of discretization which exposes aspects of the sampling process not often considered in these fields. In particular, we wish to determine how the choice of representation can affect our ability to reason about the continuity of the sampled data. This aspect of the problem is extremely important for vision. To clarify the the development in this chapter we introduce an example relevant to the representation of image data. We will consider the question of representing a graph of the orientation of a piecewise smooth curve in a planar image. This example will then continue through the thesis, evolving with the development.

Example 2.1 Consider a piecewise differentiable, continuous plane curve $\alpha : S \to \mathbb{R}^2$ parameterized by arc-length where S is the interval $[0, \ell]$. When the tangent $\tau(s) = \alpha'(s)$ is expressed in polar coordinates we have a piecewise continuous function $\theta(s)$, the direction map (speed is constant). Corners in α will appear as step discontinuities in θ (see Fig. 2.1).

Considering the general problem of representing functions on a continuum, we start with a definition

Definition 2.1 A *sampling* of a continuum X is an ordered set of distinct values $\hat{X} = \{ x_i \in X \mid i \neq j \Rightarrow x_i \neq x_j \}$.[1]

[1]Note that the continuum $\mathbb{R}^n$ *can* be described as an ordered point-set and thus is a valid, if

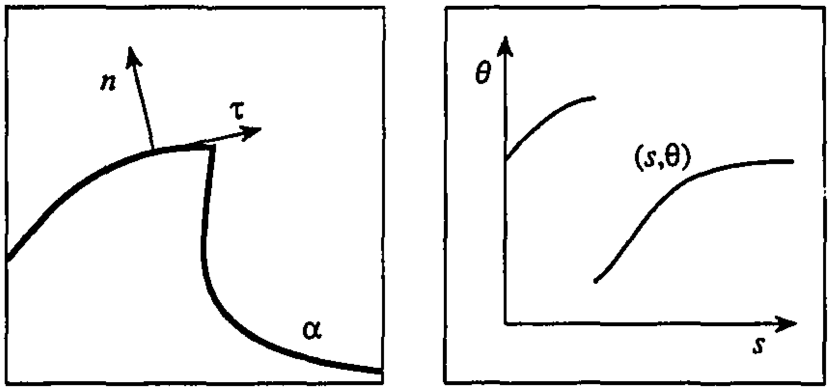


Figure 2.1: A piecewise differentiable plane curve α (a) has tangent τ and normal n. When expressed in polar coordinates, the tangent vector τ has a direction θ (b) which can be graphed over the arc-length s of the curve. Note that a corner in the curve α appears as a discontinuity in the direction θ.

Throughout the following work, we will assume that all spaces X being sampled are path-connected topological vector spaces with Riemannian metric [Spi79, Arm83].

Using such a sampling, functions are often discretized as follows.

Example 2.2 Given a function on metric spaces $f : X \to Y$, and a sampling $\hat{X}$ of the domain X, a *discretization* $\hat{f}$ of the function f is given by the ordered set of values

$$\hat{f} = \left\{ f_i = \int_X f(x)\, \Psi_i(x - x_i)\, d\mu(X) \,\middle|\, x_i \in \hat{X} \right\},$$

with $\mu(X)$ a measure on X. $\Psi_i(x)$ is known as the *point-spread function* of the discretization. If $\forall i, j\colon \Psi_i = \Psi_j$, and Ψ and $\mu(X)$ are sufficiently well-behaved then this operation is a convolution.

Perhaps the simplest such discretization is given by the *Dirac delta function* $\Psi_i(x) = \delta(x)$, which gives the discretization $f_i = f(x_i)$ by definition. In general, if the function represents some Platonic property of the real world (e.g. air tem-

counterintuitive, sampling of itself. The indexed notation (x_i) used for describing elements of the sampling set should not be taken as implying the countability of $\hat{X}$. It is simply used as a means of specifying that the point x_i is in the sampling $\hat{X}$.

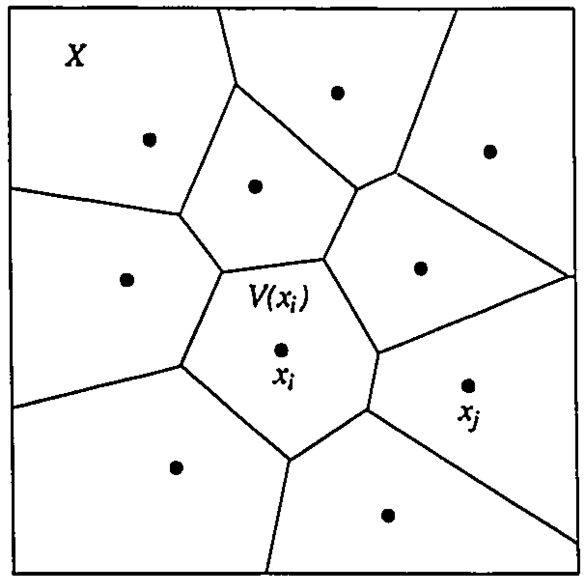


Figure 2.2: The Voronoi diagram of a two-dimensional space X is shown. The discrete points $x_i \in X$ are used to partition the space into regions which are closer to x_i than any other point x_j. There is a Voronoi cell $X_i \subset X$ associated with each point x_i.

perature), then any measurement of that property is, in effect, a spatio-temporal discretization of the function (i.e. a sample of the value at some definite point or set of points in space-time). The point-spread function captures some of the inexactness of the measurement process.

Example 2.3 *(continued)* Consider the direction $\theta(s)$ of the plane curve $\alpha(s)$ over the closed interval $S = [0, \ell]$. The integer sampling on $\mathbb{R}$ then gives the sampling $s_i = i$ for $i \in \{0, 1, \ldots, \lfloor \ell \rfloor\}$. In this case, the Dirac discretization of $\theta(s)$ is given by

$$\theta_i = \theta(s_i) = \theta(i).$$

Note that θ_i is undefined if θ is not continuous at i.

Since a sampling $\hat{X}$ is used to represent the continuum X, we assert that each point $x_i \in \hat{X}$ represents a distinct subset of the continuum. A natural partition such that each such subset is a neighbourhood of x_i is given by the Voronoi diagram of the sampling.

Definition 2.2 [Aur91, Con93] Given a sampling $\hat{X}$ of the Riemannian metric space X with distance metric $d(x,y)$, the *Voronoi diagram* (also known as the *Dirichlet tiling*) is a partition of X into disjoint subsets, the *Voronoi cells*, such that

$$X_i \;=\; \left\{ x \in X \;\middle|\; \forall x_j \in \hat{X}\colon\; x_i \neq x_j \text{ and } d(x_i,x) < d(x_j,x) \right\}.$$

We call the point x_i the *kernel* of the cell X_i.

Since there is a one-to-one mapping between kernel points x_i and Voronoi cells X_i, we can assert that x_i *represents* all of the points in X_i (and vice-versa). This relationship, in which a partition of X is described by a discrete set of points, will be fundamental to our analysis of the relationship between a sampling and the metric space being sampled.

Note that in the strict sense, the definition in Def. 2.2 does not partition X, since the boundaries between cells are not assigned to a unique cell. This detail is especially important for sparse samplings (see below). The ordering required in Def. 2.1 serves to assign these boundaries, with boundary points taken to be members of the lowest numbered Voronoi cell for which they are limit points.

Definition 2.3 [Con93] The *radius* $\rho(X_i)$ of the Voronoi cell X_i is the distance

$$\rho(X_i) \;=\; \limsup_{x \in X_i} d(x_i, x).$$

The covering radius of the sampling $\hat{X}$ is then

$$\rho(\hat{X}) \;=\; \max_{x_i \in \hat{X}} \rho(X_i).$$

We will say that a sampling is *sparse* if and only if

$$\min_{x_i \in \hat{X}} \rho(X_i) > 0.$$

Clearly, finite samplings [Raj68] are sparse, whether they are regular or random.

Example 2.4 *(continued)* Consider the domain S of α. The Voronoi cells of the sparse sampling $s_i = i$ on this space are simply the intervals

$$S_i = [i - 1/2, i + 1/2) \cap S,$$

assuming the metric $d(x, y) = |x - y|$. The radius of this sampling is then $1/2$.

Note that we have implicitly allowed the continuum to be considered as a sampling of itself. It is clear that for this sampling $\hat{X} = X$ we have

$$X_i = \{ x_i \}.$$

With this in mind, we can determine whether or not a proposed discretization of a property defined on the continuum is "reasonable."

Definition 2.4 Given a continuum X consider the relation $P(S)$ where $S \subset X$. If there exists a relation $\hat{P}(\hat{S})$ for subsets $\hat{S} \subset \hat{X}$ such that

$$\hat{X} = X \Rightarrow P(S) \Leftrightarrow \hat{P}(S)$$

then $\hat{P}$ is a *valid discretization* of P. If there exists a function $\hat{f}(\hat{S})$ for $\hat{S} \subset \hat{X}$ such that

$$\hat{X} = X \Rightarrow f(S) = \hat{f}(S)$$

then $\hat{f}$ is a valid discretization of f.

This definition clearly agrees with standard practice. For example, the Dirac discretization described in Ex. 2.2 is a valid discretization since

$$f(x_i) = \int_X f(x)\,\delta(x - x_i)\,dx.$$

by definition of the Dirac delta $\delta(x)$.

Referring to vision for a moment, the classical discretization of an image on a regular grid is simply a special case of this general formulation.

Example 2.5 For a 2-dimensional image $I : X \to Y$ defined on $X = [0,1] \times [0,1]$, a regular grid (a sampling) is given by $x_{ij} = ((2i-1)/2n, (2j-1)/2n)$ for $i,j = (1,\ldots,n)$. A digital image $\hat{I}$ is derived from the image I by sampling I on $\hat{X} = \{\, x_{ij} \in X \,\}$. The Dirac discretization of the image function is then given by $\hat{I}_{ij} = I(x_{ij})$.

In some cases, however, we cannot use this definition directly, but must instead rely on a limiting process. In that case, consider a sampling $\hat{X}$ of X. Now, consider a sequence of samplings $\hat{\mathcal{X}} = (\hat{X}_1, \hat{X}_2, \ldots)$ of X such that $\hat{X}_i \subset \hat{X}_{i+1}$ and

$$\lim_{i\to\infty} \rho(\hat{X}_i) = 0.$$

We will call such a sequence a *decreasing sequence* of samplings of X. If we consider a function $\hat{f}$ defined on a sampling, then the limit of $\hat{f}$ over a decreasing sequence will give the function f to which it is equivalent. Analogously, a relation $\hat{P}$ which selects subsets of the sampling similarly converges on subsets P of the continuum. For example, it is clear that for a decreasing sequence of discretizations $\mathcal{X}$ we can conclude that

$$\{\, x_i \,\} \;=\; \lim_{i\to\infty} X_i.$$

As a shorthand for limits taken over such a decreasing sequence of samplings, we adopt the notation $\rho(\hat{X}) \to 0$. Thus the limit above becomes, simply

$$\{\, x_i \,\} \;=\; \lim_{\rho(\hat{X})\to 0} X_i.$$

We this in mind, we note that the validity test does not carry with it a guarantee of uniqueness. In fact there may be a number of different valid discretizations of even the simplest properties. Consider again the discretization of a function on X.

Example 2.6 An alternate valid discretization of the continuous function $f : X \to Y$

on $\hat{X}$ is given by area averaging, in which

$$f_i = \frac{\int_{X_i} f(x)\, d\mu(X)}{\mathrm{Vol}(X_i)}.$$

where $\mu(X)$ is a measure on X.

To prove this for $X = Y = \mathbb{R}$ we refer to the fundamental theorem of calculus [Spi65]

$$\int_a^b f(x)\, dx = F(b) - F(a).$$

Therefore

$$\begin{aligned}
\lim_{\epsilon \to 0} f_i &= \lim_{\epsilon \to 0} \frac{\int_{b_\epsilon(x_i)} f(x)\, dx}{|b_\epsilon(x_i)|}, \\
&= \lim_{\epsilon \to 0} \frac{F(x_i + \epsilon) - F(x_i - \epsilon)}{2\epsilon}, \\
&= F'(x_i) = f(x_i).
\end{aligned}$$

The proof of the general case when X and Y are arbitrary metric spaces can be found in [Spi65].

This method has the advantage over the Dirac discretization in that it incorporates all values of the function $f(x)$ on X_i in order to form f_i. Furthermore, for this discretization the implicit point-spread function is a characteristic function of the Voronoi cell

$$\Psi_i(x - x_i) = \begin{cases} 1/\mathrm{Vol}(X_i) & \text{if } x \in X_i, \\ 0 & \text{otherwise.} \end{cases}$$

As $\rho(\hat{X}) \to 0$, this function converges on the Dirac delta $\delta(x - x_i)$. In general, any point-spread function $\Psi_i(x)$ which reduces to the Dirac delta over a decreasing sequence of samplings makes Ex. 2.2 a valid discretization.

For example, for the plane curve in Ex. 2.1, the length averaging of direction is obtained by

$$\theta_i = \int_{i-1/2}^{i+1/2} \theta(s)\, ds.$$

And for the image sampling in Ex. 2.5, the Voronoi cells X_{ij} are squares centered around x_{ij}, and a spatial averaging discretization is given by

$$\hat{I}_{ij} = \frac{1}{n^2} \iint_{X_{ij}} I(x)\, dx\, dy.$$

So far, we have only considered valid discretizations. In order to understand how the definition above restricts possible discretizations it is useful to introduce an invalid discretization.

Example 2.7 Consider a discretization of $f(x)$ based on convolution with a Gaussian kernel

$$\mathrm{G}_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{\sigma^2}}.$$

If σ is constant for all discretizations, it is easy to see that this is *not* a valid discretization since $f_i(x) = (\mathrm{G}_\sigma * f)(x_i) \neq f(x_i)$ for $\sigma \neq 0$.

If, however, instead of a *fixed* σ we vary it so that $\sigma \propto \rho(\hat{X})$, then this discretization is valid. This is, in fact, one definition of the Dirac delta, namely

$$\delta(x) = \lim_{\sigma \to 0} \mathrm{G}_\sigma(x).$$

In essence, if $\lim_{\rho(\hat{X}) \to 0} \Psi(x) = \delta(x)$ then the discretization by convolution with Ψ *is* valid. This condition is true for both the area averaging above and the $\sigma \propto \rho(\hat{X})$ case of the Gaussian convolution.

Now consider the problem of discretizing arbitrary subsets of X (i.e. choosing a subset $\hat{S} \subset \hat{X}$ to represent the subset $S \subset X$). An obvious approach would have the subset $S \subset X$ discretize to the set of points $x_i \in \hat{X}$ for which $x_i \in S$. Although this is a valid discretization, the X_i selected do not always cover S. Coverings are the fundamental building block of topology, so in order to represent or at least reason about topological properties such as connectedness it is necessary to ensure that $\hat{S}$ covers S.

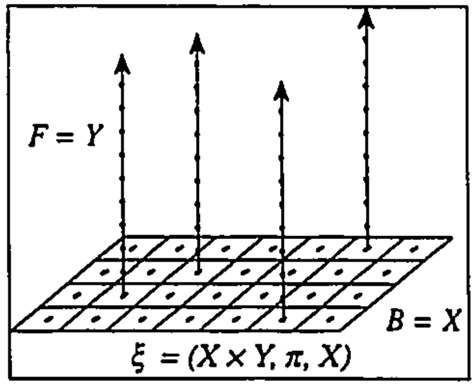


Figure 2.3: A product bundle ξ is the shown as its total space E, which is the Cartesian product of a base space B and the fibres F. This can be thought of as just a high dimensional vector space. A regular discretization of both the base space and the fibres is shown by the dots.

Definition 2.5 The closed *subset* S of X is discretized by the subset $\hat{S} \subset \hat{X}$ where

$$\hat{S} = \left\{ x_i \in \hat{X} \mid X_i \cap S \neq \emptyset \right\}.$$

The set of Voronoi cells $X_i \in \hat{S}$ forms an irreducible covering of S.

Significantly, this is also a valid discretization since for $\hat{X} = X$

$$\hat{S} = \{ x \in X \mid \{x\} \cap S \neq \emptyset \} = S.$$

2.1 Fibre Bundles

The regular sampling of continuous images presented above is the traditional domain on which early vision algorithms are formulated. However, often ignored is a second kind of "discretization" inherent in digital systems, the quantization of the values measured (e.g. image intensities). Rather than considering these as a separate processes, we introduce a formal model of the discretization of functions and images which combines both this spatial sampling and the quantization of values in a single structure. To do this we introduce fibre bundles.

Definition 2.6 [Hus66] A *fibre bundle* is a triple $\xi = (E, \pi, B)$ of a *total space* E, a *base space* B, and a *projection* $\pi : E \to B$.[2] The expressions $E(\xi)$ and $B(\xi)$ may be used to refer to the total and base spaces of ξ. For each $b \in B$, the space $F = \pi^{-1}(b)$ is called the *fibre* F over b. A bundle of the form $(B \times F, \pi, B)$ where $\pi(b, f) = b$, is known as a *product bundle.* If the base space B of a bundle is the domain X of an image $I : B \to F$ then we will refer to the bundle as an *image bundle.*

We adopt this formalism because of its descriptive advantage in dealing with the relationships between the base and fibre spaces and the natural association between cross-sections of a bundle (described below) and the problems that one faces in early vision. At times it may be simpler to think of the fibre bundles we use—product bundles of vector spaces—as simply high dimensional vector spaces.

An immediate consequence of this definition is that vector functions may be identified with certain subsets of the total space.

Definition 2.7 [Hus66] Given a vector function $f : B \to F$, we form the *natural product bundle* ξ by taking $\xi = (B \times F, \pi, B)$ and $\pi(x, y) = x$ for $(x, y) \in B \times F$. The mapping f is then a *cross-section* of the bundle over any domain $S \subset B$, where the domain of f is identified with the base space of the bundle and the range with the fibre.

For the purposes of this work, we will work with the point set which corresponds to this mapping, the *trace*

$$\mathrm{TR}(f) \;=\; \{\, (x, f(x)) \in E(\xi) \mid x \in S \,\},$$

This is also called the *graph* of f on S

Note that every function corresponds to a cross-section in some product space, and vice-versa [Hus66]. For convenience, we will therefore sometimes use the same symbol to refer to the cross-section and its equivalent function. Furthermore, if $E(\xi)$ is

[2] The bundle ξ is often identified simply as $\pi : E \to B$, since this expression contains all of the components of the bundle.

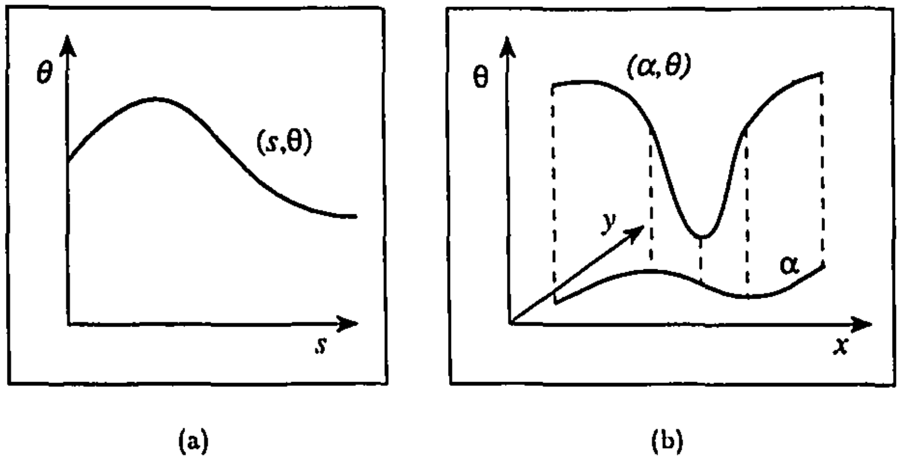


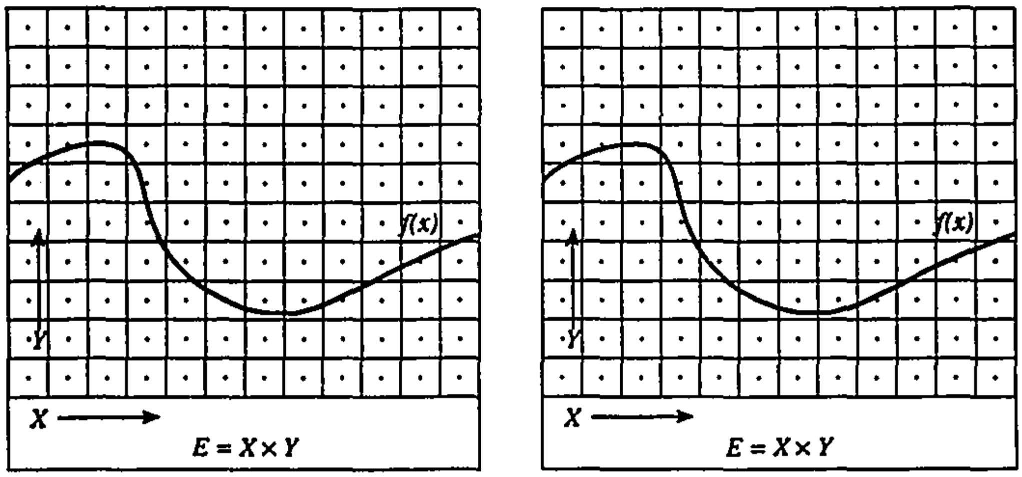
Figure 2.4: The direction map $\theta : S \to \mathbb{R}$ for a plane curve α has two different natural product bundles. (a) If we use the interval S as the base space $B = S$, then the direction map is a cross-section in the total space $S \times \mathbb{R}$. If instead, (b) we treat α as a submanifold of the base space $B = X \times Y$ then the direction map is a cross-section over $\alpha \subset B$ in the total space $B \times \mathbb{R}$.

a topological vector space with Riemannian metric, the cross-sections induced by continuous functions are also differentiable submanifolds of the total space [Lan85]. Identifying the cross-section as a manifold will be essential in some of the development below.

Example 2.8 *(continued)* The natural product bundle for a particular physical or geometric mapping is not always unique. In particular, it depends on the choice of the base space used to express the function. We will describe two possible bundles, ξ_s and ξ_α, associated with the direction map of a plane curve.

For the plane curve $\alpha : S \to \mathbb{R}^2$, the direction map $\theta : S \to \mathbb{R}$ has the natural product bundle $\xi_s = (S \times \mathbb{R}, \pi_s, S)$ where $\pi_s(s, \theta) = s$ (Fig. 2.4a). The function θ is then a cross-section of this bundle.

We can also choose $B(\xi_\alpha) = \mathbb{R}^2$ as the base space of a natural product bundle for the direction map of an image curve since α is a submanifold of $\mathbb{R}^2$. This leads to a bundle in which $\xi_\alpha = (\mathbb{R}^2 \times \mathbb{R}, \pi_\alpha, \mathbb{R}^2)$ and $\pi_\alpha(x, y, \theta) = (x, y)$ (Fig. 2.4b). The map θ is then defined only over $\alpha \subset \mathbb{R}^2$.



(a) Thin trace (b) Thick trace

Figure 2.5: The trace of function $f : X \to Y$ in the natural product bundle $\xi = (X \times Y, \pi, X)$ is the cross-section $\{(x, f(x)) \mid x \in X\}$. When the bundle is discretized, with discrete total space $\hat{E} = \hat{X} \times \hat{Y}$, then the cross-section can be represented as either (a) the thin trace, in which the Voronoi cell Y_j through which f crosses x_i is chosen from each fibre, or (b) the thick trace, which includes all points e_{ij} for which the cross-section intersects E_{ij}.

2.1.1 Discretization of Graphs: Discrete Traces

We now return to consideration of the discretization of functions, but this time from the point of view of the natural product bundle. The first step is to define a sampling of the bundle.

Definition 2.8 A *discrete product bundle* $\hat{\xi}$ is given by $\hat{\xi} = (\hat{E}, \pi, \hat{B})$ where $\hat{E} = \hat{B} \times \hat{F}$ and $\hat{B}$ and $\hat{F}$ are samplings of the base space B and the fibre F respectively. When we wish to separate the samplings $\hat{B}$ and $\hat{F}$, a point in $\hat{E}$ is referred to as $e_{ij} = (x_i, y_j)$. The projection is given by $\pi_\xi(e_{ij}) = x_i$. We refer generally to the discretizations of cross-sections of discrete bundles as *discrete traces.*

Now consider the discretization of the cross-section $f : X \to Y$, a bounded, piecewise continuous function with compact support on X. Given a sampling of the total space $\hat{E} = \hat{X} \times \hat{Y}$, one discretization of the function f is given by what we refer to as the *thin trace* (see Fig. 2.1.1)

$$\text{THIN}(f) = \left\{ e_{ij} \in \hat{E} \,\middle|\, f(x_i) \in Y_j \right\}.$$

That is, for each sampled fibre $\hat{Y}$, choose the kernel point (x_i, y_j) closest to the intersection $(x_i, f(x_i))$.

Theorem 2.9 *The thin trace* $\text{THIN}(f)$ *is a valid discretization of* $\text{TR}(f)$.

This can be verified, since for $\hat{X} = X$,

$$\begin{aligned} \text{THIN}(f) &= \left\{ e_{ij} \in \hat{E} \,\middle|\, f(x_i) \in Y_j \right\} \\ &= \left\{ (x, y) \in E \,\middle|\, f(x) \in \{ y \} \right\} \\ &= \text{TR}(f). \end{aligned}$$

∎

When the sampling $\hat{Y}$ is regular, this definition is equivalent to a regular sampling of X with $x_i \in \hat{X}$ and quantization of $f(x_i)$ by rounding. This is exactly the representation of images produced by digital cameras and most often used in image processing.

Example 2.9 *(continued)* Consider the discretization of the direction function $\theta : S \to \mathbb{R}$ by the Dirac delta function so that

$$\theta_i = \theta(i).$$

This is clearly just the intersection point of θ with the fibre over i. If this value is represented in a digital computer by rounding θ_i to the nearest representable value (e.g. a floating point number), then the set of values

$$\text{THIN}(\theta) = \{ (0, \theta_0), (1, \theta_1), \ldots \}$$

is the thin trace of θ in the natural product bundle $\xi_s = (S \times \mathbb{R}, \pi_s, S)$. Since the sampling $\hat{S}$ is regular, this can be represented as just the sequence of rounded values

$(\theta_0, \theta_1, \ldots)$. Considered in this way, the floating point numbers represent a very fine sampling of the range of θ.

An alternate discretization is given by the thick trace (see Fig. 2.5b), which will become the focal point of this thesis.

Definition 2.10 The *thick trace* of the function f on $\hat{\xi}$ is given by

$$\textsc{Thick}(f) \;=\; \left\{ e_{ij} \in \hat{E} \,\middle|\, \exists x \in X\colon\; (x, f(x)) \in E_{ij} \right\}.$$

Thus, every Voronoi cell in the sampled total space $\hat{E}$ through which the cross-section $(x, f(x))$ passes is represented in the thick trace by its Voronoi kernel. An obvious consequence of this definition is that the thin trace is a subset of the thick trace.

Theorem 2.11 *The thick trace* $\textsc{Thick}(f)$ *is a valid discretization of* $\textsc{Tr}(f)$.

For $\hat{X} = X$,

$$\begin{aligned}
\textsc{Thick}(f) &= \left\{ e_{ij} \in \hat{E} \,\middle|\, \exists x \in X\colon\; (x, f(x)) \in E_{ij} \right\} \\
&= \left\{ (x, y) \in \hat{E} \,\middle|\, \exists x \in X\colon\; (x, f(x)) \in \{ (x, y) \} \right\} \\
&= \left\{ (x, y) \in E \,\middle|\, f(x) = y \right\} \\
&= \textsc{Tr}(f).
\end{aligned}$$

∎

Example 2.10 *(continued)* Since there are two distinct product bundles for the direction map, there are two distinct representations for the thick trace of the map over a particular curve.

The thick trace of θ in the bundle ξ_s is the set of Voronoi cells E_{ij} which intersect the curve $f_s = \{ (s, \theta(s)) \mid s \in S \}$. In this case, that implies that on each fibre $\pi_s^{-1}(s_i)$ we include all cells which overlap the interval $[\inf_{S_i} \theta(s), \sup_{S_i} \theta(s)]$ where $S_i = [i - 1/2, i + 1/2) \cap S$.

Alternatively, the thick trace of θ in the bundle ξ_α can also be calculated. This time $f_\alpha = \{ (\alpha(s), \theta(s)) \mid s \in S \}$, with the sampling of the total space $\hat{E}(\xi_\alpha) = \hat{\mathbb{R}}^2 \times \hat{\mathbb{R}}$

(i.e. the image domain is sampled as pixels and the directions are as above). For this bundle, the projection $\pi_\alpha(\text{THICK}(\alpha))$ is the subset discretization of α on $\hat{\mathbb{R}}^2$, while on each fibre $\pi_\alpha^{-1}(b_i)$ over this subset we again have an interval bounded by the minimum and maximum values of $\theta(s)$ for which $\alpha(s) \in B_i$.

Note that Canny's algorithm for edge detection [Can86], produces an image position to direction mapping which is single-valued for edge points in the image. The edge points are chosen by locating image points which give laterally maximal edge matches. In terms of thin and thick traces then, the Canny algorithm produces a thin trace of the direction map *over* the thick trace (in fact the discrete subset) of points in the image (the base space) which fall on the curve of lateral maxima. We will show that one consequence of this choice of representation is that algorithms such as this cannot properly distinguish either corner points or crossings. This inadequacy is an *inherent* property of the choice of representation.

In many cases there will be a clear preference for one of these representations over the other. With digital images, for example, users normally prefer thin trace representations because of the possibility of representing the base-space implicitly due to the one-to-one $B \to F$ mapping (e.g. for compact storage). Hence, from this perspective, the usual representations of images on digital storage media are all thin traces, where the indices of points are implicit and values are arranged in a rectangular array in memory.

2.1.2 Discrete Traces and Continuity

As representations, both thin and thick traces represent an equivalence class of functions which could have generated the trace. We refer to this class as the functions *underlying* a discrete trace. Thus, given a thin or thick trace, we may draw certain conclusions about the functions which might underly that trace. In many cases, a preference for one representation or another hinges not simply on the amount of memory needed to store it, but on the conclusions which can be drawn about these functions. In particular, we will show that when continuity of the functions in the

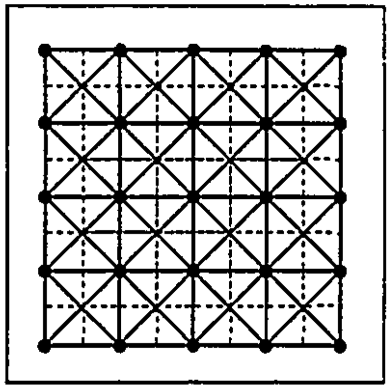


Figure 2.6: Connectivity of sampled spaces is defined in terms of contact between the Voronoi cells (boundaries delimited with dotted lines). If the contact is over an entire face, then the points are *strongly connected* (heavy lines). If the point of contact is a single point, then the points are *weakly connected* (light lines). The neighbourhood of a point consists of all of the points to which it is connected (dark or light shaded cells). For a rectangular grid, these concepts are familiar as *4-connected*ness and *8-connected*ness.

equivalence class is not assumed *a priori*, then the thick trace is a better representation since it allows one to investigate the continuity of the class in a way not possible with thin traces.

The focus of this analysis is the relationship between the connectivity of discrete traces on sparse samplings and the continuity of the underlying curves. Before this relationship can be clarified, we must state clearly what we mean by the connectivity of a sampling.

Definition 2.12 Given a sparse sampling $\hat{E}$ (i.e. $\min \rho(E_i) > 0$) the *connectivity graph* of $\hat{E}$ is a graph on the vertices $\left\{ e_i \in \hat{E} \right\}$ with the edges representing direct contact between Voronoi cells. The *strong* connectivity graph $C^s(\hat{E})$ is formed by including an edge (e_i, e_j) whenever E_i and E_j share a common face. The *weak* connectivity graph $C^w(\hat{E})$ is formed by including an edge (e_i, e_j) whenever E_i and E_j share any common limit point.

Note that the traditional notions of 4-connected and 8-connected graphs on grids are subsumed by these definitions, with 4-connectedness strong and 8-connectedness weak (see Fig. 2.6).

Interestingly, for generic (i.e. random or randomized) samplings, the strong and weak cases are equivalent. This arises from the observation that a weak connection which is not equivalent to a strong one can only occur at vertices with more than three incident edges (i.e. there are at least four points equidistant from the vertex). This occurrence is of vanishing probability for random distributions of kernel points [?]. Thus, although the strong/weak distinction is important for work on regular samplings, it is largely irrelevant for irregular samplings. For this reason, we will ignore the distinction (and distinguishing superscript) and assume the weak connectivity.

Note that the connectivity graph has only been defined for sparse samplings. Thus rather than reiying on Def. 2.4 we will instead reason directly with the connectivity graph. In particular we will prove that the connectivity of this graph is equivalent to the path-connectedness of the space on which it is defined.

Definition 2.13 The connectivity graph $C(\hat{S})$ of a sparse sampling subset $\hat{S} \subset \hat{E}$ consists of all vertices $\left\{ e_i \in \hat{S} \right\}$ and all edges $(e_i, e_j) \in C(\hat{E})$ where both $e_i, e_j \in \hat{S}$. The sparse sampling subset $\hat{S}$ is *connected* if and only if for all pairs $e_i, e_j \in \hat{S}$ a finite path exists in $C(\hat{S})$ from e_i to e_j.

Theorem 2.14 *A sparse sampling subset $\hat{S} \subset \hat{E}$ is connected if and only if the union of the Voronoi cells*

$$U = \bigcup_{E_i \in \hat{S}} E_i$$

is path-connected.

Observe that each Voronoi cell E_i is both trivially path-connected and has a connected connectivity graph. It is also clear that the union of a cell E_i and some path-connected set of cells $\{ E_j \mid i \neq j \}$ is path-connected if and only if there is some limit point in common between E_i and some E_j. This condition is equivalent to the assertion that $\exists (e_i, e_j) \in C(\hat{E})$. Thus by induction we conclude that the connectivity graph of a finite set of Voronoi cells is connected if and only if their union is path-connected.

To show that this conclusion extends to infinite subsets as well it is only necessary to consider that U is path-connected if and only if there is a path p of *finite length* between any two points in U [Arm83]. The if part is trivial, since a finite path in the connectivity graph is a path in U. To prove only if, it is necessary to consider a bounded neighbourhood N_p of the path p. Since the volume of the neighbourhood is finite and the volume of each E_i in $\hat{S}$ is non-zero, this neighbourhood has a finite, irreducible cover in $\hat{S}$. Since each E_i in this cover is path-connected and $E_i \cap N_p \neq \emptyset$, the cover is itself path-connected and thus has connected graph. Thus for every finite path p in U there is a finite cover of p in $\hat{S}$ with connected graph. ∎

This definition has an immediate and important consequence.

Theorem 2.15 *If the set $S \subset E$ is path-connected then its discretization $\hat{S} \subset \hat{E}$ is connected for any sampling*

From Def. 2.5 we know that

$$\forall E_i \in \hat{S}: \; E_i \cap S \neq \emptyset$$

and since each $E_i \in \hat{S}$ is path-connected we conclude that a finite path exists from any point $p \in \bigcup E_i$ to any $q \in S$. Thus, since S is path-connected, $\bigcup E_i$ is path-connected and by Thm. 2.14 this is equivalent to the assertion that the connectivity graph $C(\hat{S})$ is connected. ∎

Now we can explicitly state the crucial relationship between the continuity of a cross-section and connectivity of the thick trace.

Theorem 2.16 *Let $f : S \subset B \to F$ be a continuous cross-section of the total space E. If $\hat{E} = \hat{B} \times \hat{F}$ is a sampling of E, then:*

1. *The thick trace* THICK(f) *is connected;*

2. *The discretized domain $\hat{S}$ is connected;*

Each of these follow immediately from Thm. 2.15. What will be most important for vision applications is the following.

Corollary 2.17 *If a thick trace is* disconnected *then all functions underlying the trace are* discontinuous.

Moreover, the *local connectivity* of the thick trace can be related to the *local continuity* of the underlying functions. In particular, boundaries of the connectivity graph of the thick trace correspond to boundary points and points of discontinuity of piecewise continuous mappings. To see this, consider a piecewise continuous mapping f as a collection of continuous patches $\{g_1, \ldots\}$ which cover f. Then, the local connectivity of points in $\text{THICK}(g_n)$ allows one to identify the boundaries of the patch g_n.

Corollary 2.18 *Consider the connected set of points $\bar{f}_i = \text{THICK}(f) \cap \pi^{-1}(b_i)$ on the fibre $\pi^{-1}(b_i)$ for f a continuous cross-section. If $\pi^{-1}(b_k)$ is a neighbouring fibre (i.e. $(b_i, b_k) \in C(\hat{B})$) but $\bar{f}_i$ does not have a neighbour in $\pi^{-1}(b_k)$*

$$\neg\exists e_{kl} \in \pi^{-1}(b_k), e_{ij} \in \bar{f}_i : (e_{ij}, e_{kl}) \in C(\hat{E})$$

then $\bigcup_{\bar{f}_i} E_i$ contains a boundary point for every continuous patch underlying the thick trace.

Thus we have both global and local conclusions about the continuity of the equivalence class of mappings underlying a thick trace, and a simple method for recognizing when a point in a thick trace is at the boundary of its domain. Moreover, these properties depend only on the intrinsic connectivity of the trace. These are the key properties that make the thick trace important for vision.

Example 2.11 *(continued)* Consider the thick trace in $E(\xi_s)$ of a piecewise smooth curve with a corner at $s = s_d$. This results in a local discontinuity in direction $\theta(s)$ at s_d (Fig. 2.7b). Note that in the thick trace there are two disconnected points in the discrete fibre over s_d. Each of these points represents a boundary of a smooth patch covering the function θ.

In contrast, consider the thin trace of the same function (Fig. 2.7a). There are a number of points where the thin trace is not locally connected, only one of which corresponds to a discontinuity in the underlying function.

In contrast to the relationship between continuity of functions and the connectivity

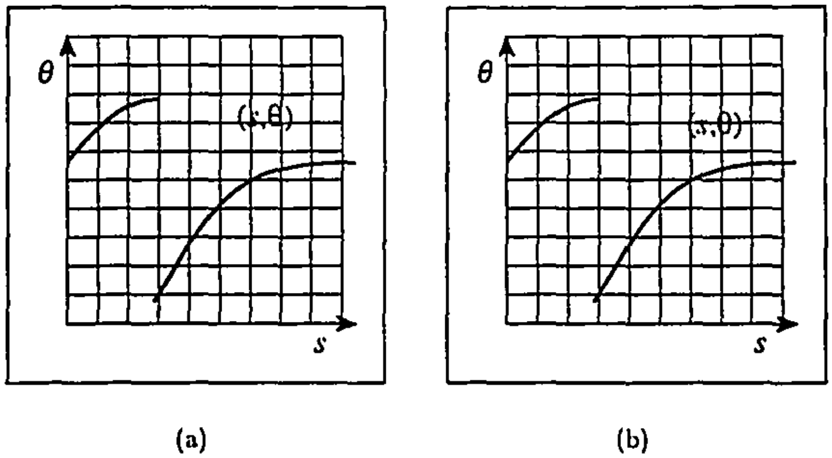


Figure 2.7: A corner in the plane curve α appears as a step discontinuity in the direction map θ. The discontinuity may appear as a disconnected point in both the THIN (a) and THICK (b) traces of the direction map, but in only the thick trace does such a discontinuity unambiguously indicate a discontinuity in θ. Notice in (a) that the thin trace is disconnected wherever the derivative is high, and not only at discontinuities.

of thick traces, is the observation that no such relationship holds for thin traces.

Proposition 2.19 *In general, a disconnected thin trace* THIN(f) *does not indicate a discontinuity in* f.

A disconnected point in the thin trace may arise from either a large derivative or a true discontinuity: these cases are indistinguishable with thin trace representations. This can be seen both in Figs. 2.1.1 and Fig. 2.7a.

In contrast with the thick trace, the continuity of f does not constrain the structure of the thin trace at all. Any thin trace (i.e. a cross-section of $\hat{E}$ such that there is exactly one $e_{ij} \in$ THIN(f) for each i), corresponds to at least one continuous function. Such a function is easily constructed by just connecting the kernel points $e_{ij} \in$ THIN(f) for all nearest neighbours on $\hat{B}$. Worse yet, *any* interpolation between these points which does not intersect any other fibres will result in the same thin trace, including those with significant wandering between the $b_i \in \hat{B}$ points. So, the thin trace THIN(f) in and of itself provides no intrinsic information about the continuity or discontinuity of those functions underlying it.

If we add additional constraints (e.g. bounds[3] on the derivatives of f or the Shannon bandwidth condition) then it is possible to draw conclusions about the continuity of a cross-section from the thin trace, but as we have seen not without such extra conditions. Thus in computational situations in which we need to verify continuity from discrete traces, a thick trace is a more useful representation than a thin trace.

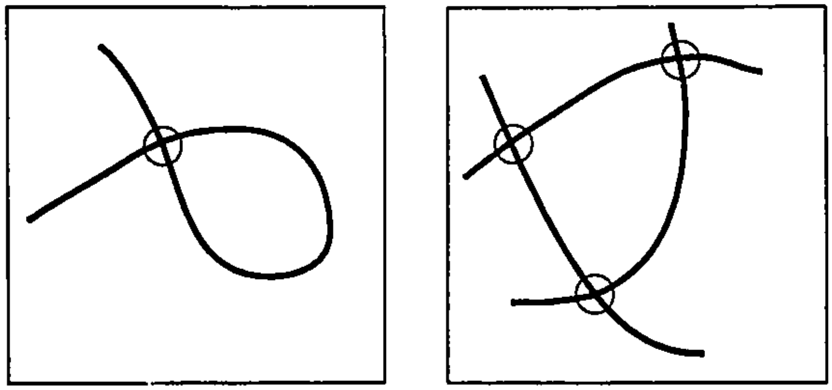
2.1.3 Other Constraints

The relationship between continuity and connectivity provides a strong argument for adopting thick traces for certain problems. Even in cases where this may not be an issue (e.g. bounds on derivatives *are* available externally), there are other reasons to prefer a thick trace.

It should be clear that computation of the thick trace on a sampling is a purely local computation. If a single processing element is associated with each sampled point in the total space, then its local decision procedure can be summarized as: does the curve intersect my Voronoi cell? The fact that the metrics we consider are strictly Riemannian guarantees that this is a purely local computation, requiring only knowledge of the position of the boundaries of the cell (fixed by the sampling). The thin trace is, however, somewhat less local in that it requires the integration of information over the entire fibre. This local selection procedure can be a bottleneck in some systems (e.g. [PM91]). We suggest that the thick trace is a better match for the representations stored in massively parallel, distributed systems such as SIMD parallel computers, neural networks or the brain.

Another advantage of of the thick trace representation is that it can represent a richer class of geometric structures than the thin trace. The fact that multiple points (connected or not) on a discrete fibre can be represented in the thick trace, allows for the accurate depiction of structures which depend on multi-valued mappings from the

[3]These bounds can be derived from the radii of the discretizations of the base and fibre spaces



(a) (b)

Figure 2.8: Both self intersections (a) and intersections of independent curves (b) can cause significant problems for representations which insist on assigning a single orientation to each pixel in an image (e.g. [Can86]). The fact that thick traces can include multiple points (connected or not) on a single fibre allows for the accurate representation of such points.

base space to the fibre. Thus a thick trace can accurately describe self-intersections and transparency in ways that the thin trace cannot. In many cases, especially in early vision, a strict assumption of single-valued mappings may lead to incomplete theories of the phenomena being examined.

A natural computer representation for thick traces is to assign one bit per point e_i in the total space $E(\xi)$ of the bundle ξ. A description of the cross-sections which exist in an image would then be given by setting all bits to zero except those which are in the thick trace the cross-sections in the image. Note that there is nothing inherent in this representation which precludes the independent activation of multiple, disjoint intervals in a single discrete fibre. As we saw above, this property has important consequences for the representation of discontinuous points in the cross-section. It is also useful in the representation of irregular curves.

Example 2.12 *(continued)* Note that nothing in the description of thick traces on ξ_α precludes the possibility that a curve may intersect the fibre Θ_i more than once. Consider a self-intersecting curve such as shown in Fig. 2.8a. At the point of self-intersection, there are *two* intersections between the curve in $E(\xi_\alpha)$ and the discrete fibre Θ_i over the point. The thick trace of this curve in the bundle ξ_α will then contain *two* separate intervals on the fibre over the crossing point. Note that neither of these intervals corresponds to a boundary in the domain of the mapping, thus this point is not a "corner." Instead, we can interpret the description in terms of the intersection of two distinct regular patches taken from a regular covering of the irregular curve [dC76].

Consider instead the representation chosen by Canny [Can86]. In this case, the algorithm finds the unique "best" local direction for an edge at every point. As we have stated before, this can be considered to be a thin trace representation of the direction map in ξ_α over the image curves. At points of intersection, Canny is forced to choose the "more significant" local direction and abandon the other, since the representation assumes at most one direction per position. In general though, the smoothing inherent in Canny's approach ensures that the direction chosen is not even one of those incident on the point, but some weighted average. This is one of the sources of the well-known problems in similar algorithms near crossings and corners.

One might argue that the storage and possibly computational expense is too great to justify the use of thick traces. To make this argument for early vision, one would have to support the hypothesis that it is sufficient to describe images in terms of one-to-one mappings from points to local structure (e.g. in regularization [PTK85] or membrane [BZ87] models for surface reconstruction). Now that we have a clear alternative though, this hypothesis seems tenuous at best. The class of representations we propose is one in which a number of different geometric structures can coexist at a point in the image. For a variety of early vision problems this seems to be a much better match than the restriction to single-valued mappings. For curves, the thick trace can clearly represent crossings; for texture flow, transparent textures; and for optical flow, transparent motions. Moreover, the discontinuities which can be expected to occur in each of these situations are not only represented in the thick trace, but easily extracted.

Finally, recall that the thick trace is a *cover* of the trace of the cross-section whereas the thin trace is not. This may be an important consideration if the topology of the underlying mapping is an issue [Arm83]. While it is certainly beyond the scope of this work to examine all uses that might be made of the algorithms developed, we can at least state that a covering is an appropriate starting point for an investigation of the topological properties of a geometric object.

2.2 Conclusions

In summary then, this analysis has constrained both the nature of the solutions we might seek and the means of achieving them. We suggest that cross-sections of fibre bundles are an appropriate framework for formulating many problems in early vision. The inherently discrete nature of digital images, however, leads one to conclude that reasoning within this framework must be implemented on discrete samplings of the bundles and discrete traces of cross-sections. For at least a few of the fundamental questions faced in early vision we have demonstrated that such reasoning is better supported on thick rather than thin traces. The body of this thesis will consist of a demonstration of means to extract and use such thick traces for some of the most basic problems of early vision.

Part II

From Image to Geometry

Chapter 3 Image Curve Traces: Logical Foundations

In Chapter 2 we explored the relationship between properties of a function and its description in discretely sampled spaces. In the rest of this thesis we apply the results of this work to machine vision by extracting thick trace descriptions from images.

The solution is divided into two stages. In Part II we show how to locate those points in the sampled total space which accurately describe local features of the image using local operators. However, since the process used is inherently local and inexact, the descriptions produced are not necessarily connected everywhere the underlying features are. In Part III we use the intrinsic geometry of the features thus found to combine those points which seem to be part of the same thick trace into connected components.

In order to provide a sustained focus for the subsequent analysis, we will investigate a classical problem which is fundamental to the development of general purpose artificial vision systems: locating and describing image curves.

In Part II, in particular, we will address the problem of designing local operators which respond only at those image points through which a curve with a particular local geometry passes. The resulting operators are similar to classical approaches [Can86] in that they are built up from linear convolutions. However, rather than relying on *ad-hoc* post-processing to select significant responses we develop an algebra (the Logical/Linear algebra) which formalizes the reasoning involved. With this algebra we are able to design operators which select specific categories of curve (e.g. bright lines but not edges) and at the same time verify local continuity.

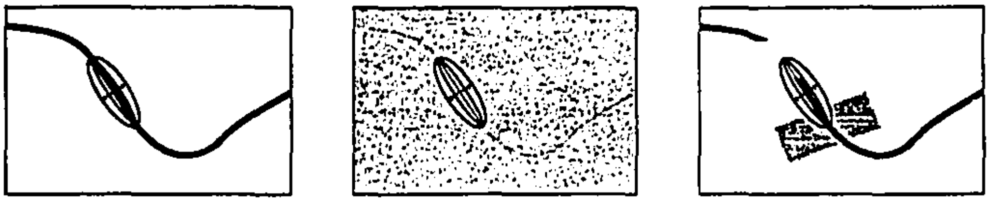
3.1 Image Curves

A typical problem in early vision is the extraction of curvilinear boundaries or edges from a static image via local operators. The operator which makes this discrimination is referred to as an "edge detector." The design of such operators is a problem which is widely considered to be fundamental to early vision.

There is no shortage of these so-called "edge detectors" and "line detectors" in computer vision. Many different designs have been proposed, based on a range of optimality criteria (e.g [Heu71, Can86]), and many of these designs exhibit properties in common with biological vision systems [JP87]. While this agreement between mathematics and physiology is encouraging, there is still dissatisfaction with these operators—despite their 'optimal' design they do not work sufficiently well to support subsequent analysis. Part of the problem is undoubtedly the myopic perspective to which such operators are restricted, suggesting the need for more global interactions [ZDDI88]. We believe that more can be done locally, and that another significant part of the problem stems from the types of models on which the operators are based and the related mathematical tools that have been invoked to derive them. In this part of the thesis we introduce an approach to operator design that differs significantly from the standard practice, and illustrate how it can be used to design non-linear operators for locating lines and edges.

The usual model used in the design of edge operators involves two components: an ideal step edge plus additive Gaussian noise. This model was proposed in one of the first edge detector designs [HB70], and has continued through the most recent [Can86, Der87]. Thus it is no surprise that the solution resembles the product of two operators, one to smooth the noise (e.g. a Gaussian) and the other to locate the edge (e.g. a derivative).

While some of the limitations of the ideal step edge model have been addressed elsewhere (e.g. [Hor77, LZ84]), a perhaps more important limitation of the operator design has not been considered. It is assumed that in viewing a small local region of the image, only a single section of one edge is being examined. This may be an appropriate simplification in some continuous limit, but it is definitely not valid in digital images. Many of the systematic problems with edge and line detectors occur when structure changes within the local support of the operator (e.g. several edges or lines coincide). Since these singularities are not dealt with by the noise component of the model either, the linear operator behaves poorly in their vicinity.



(a) (b) (c)

Figure 3.1: A set of potential image curve configurations which must be considered in the design of operators. An ideal image (a) of a black curve on a white background; a noisy image (b) of a lower-contrast version of the same curve; an obscured version (c) of the ideal image. The oval in each image represents the spatial support of a local operator. A negative contrast line operator should respond positively in all three cases.

In particular, curve-detecting operators are usually designed to respond when a certain intensity configuration occurs locally (see Figure 3.1a). A signal estimation component of the operator is then incorporated in the design to filter local noise (see Fig. 3.1b). However, significant contrast changes are rarely noise—they are more likely to be the result of a set of distinct objects whose images project to coincident image positions (see Fig. 3.1c). An operator which claims to 'detect' or 'select' a certain class of image features should continue to do so in the presence of such confounding information.

We propose that image operators should be designed to respond positively to the expected image structure, and *to not respond at all* when such structure is *not* present. Unless they meet both of these goals, they are useless for producing thick trace descriptions. Simple linear operators achieve the first of these goals, but in order to fulfill the second we must incorporate a more direct verification of the existence conditions for a given feature into the operator itself. We accomplish this by decomposing the linear operator into components which correspond to assertions of the logical preconditions for a given feature. When the expected image structure is present, a *boolean* combination of these responses produces a *linear* response, whereas if any of the conditions are violated the response will be suppressed non-linearly. Because these operators unite elements from boolean logic and linear operator theory,

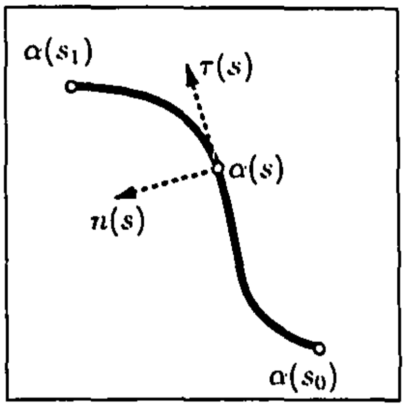


Figure 3.2: An image curve $\alpha: \mathrm{S} = (s_0, s_1) \to \mathbb{R}^2$ with the tangent $\tau(s)$ and normal $n(s)$.

we refer to them as *Logical/Linear* (L/L) operators.

3.2 Definitions and Goals

For consistency we shall adopt the following terminology. *Edges* are the curves which separate lighter and darker areas of an image—the perceived discontinuities in the intensity surface; *lines* are those curves which might have been drawn by a pen or pencil (sometimes referred to as *bars* in other work [Mar82]). *Image curves* are either of these. Two independent properties describe such image curves: their structure along the tangential and in the normal directions. Tangentially, both lines and edges are projections of space curves; it is the cross-sectional structure in the image which differentiates them.

Formally, let $I: \mathbb{R}^2 \to \mathbb{R}$ be an analytic intensity surface (an image) and $\alpha: \mathrm{S} = (s_0, s_1) \to \mathbb{R}^2$ a smooth curve parameterized by arc length (see Figure 3.2). The orientation $\theta(s)$ is the direction of the tangent $\tau(s)$, a unit vector in the direction of $\alpha'(s)$, and the normal vector $n(s)$ is a unit vector in the direction $\alpha''(s)$.

Formally, an image curve is defined by a set of *local structural conditions* on the image in the directions tangential and normal to the curve. The normal cross-section

β_s at the point $\alpha(s)$ is given by

$$\beta_s(t) = I\Big(\alpha(s) + t\, n(s)\Big), \quad s \in \mathrm{S}, t \in \mathbb{R}.$$

Definition 3.1 An *image curve* is a map $\alpha{:}\, \mathrm{S} \to I$ such that

$$\text{(Tangent)} \qquad \alpha \text{ is } C^1 \text{ continuous on } S \text{, and} \tag{3.1a}$$

$$\text{(Normal)} \qquad \text{a condition } N(\beta_s) \text{ holds for all } s \in \mathrm{S}. \tag{3.1b}$$

$N(\beta_s)$, the *normal condition*, determines the classification of the curve.

For the purposes of this analysis, we concentrate on three kinds of image curve:

1. α is an *edge*[1] in I iff α is an image curve with normal condition

$$N(\beta_s) \;\equiv\; \lim_{t\to 0^-} \beta_s(t) < \lim_{t\to 0^+} \beta_s(t). \tag{3.2a}$$

2. α is a *positive contrast line* in I iff α is an image curve with normal condition

$$N(\beta_s) \;\equiv\; \lim_{t\to 0^-} \beta_s'(t) > 0 \text{ and } \lim_{t\to 0^+} \beta_s'(t) < 0. \tag{3.2b}$$

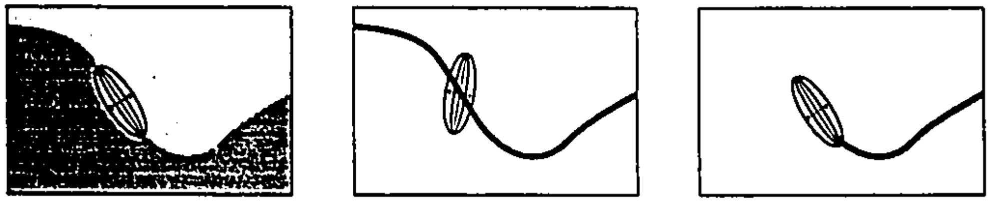
3. α is a *negative contrast line* in I iff α is an image curve with normal condition

$$N(\beta_s) \;\equiv\; \lim_{t\to 0^-} \beta_s'(t) < 0 \text{ and } \lim_{t\to 0^+} \beta_s'(t) > 0. \tag{3.2c}$$

Thus, edges are order 0 discontinuities (steps) in cross-section, while positive and negative contrast lines are order 1 discontinuities (creases) which are also maxima or minima, respectively.

Note that in contrast to traditional definitions, the tangent and normal conditions above are both *point* conditions, which must hold at every point in the trace of

[1]Note that the definition of a line is independent of curve orientation, while a *rising* edge will only be seen looking along α in one direction. Thus lines need only be parameterized over π orientations while edges require 2π orientations.



(a) (b) (c)

Figure 3.3: A set of image curve configurations which may generate false positive operator responses. An image of an contrast edge (a) should not stimulate a line operator; a improperly oriented operator (b) should not be stimulated; an operator which does not lie on the curve (c) should not be stimulated. The oval in each image represents the spatial support of the local operator. A negative contrast line operator *should not respond* positively in any of these cases.

the curve. We thus have a basis for designing purely local operators to locate and categorize such curves in images.

Linear operators do respond when these conditions are met. However, they also respond in situations in which the conditions are *not* met. These responses are referred to as *false-positives.* The current analysis will focus on a mechanism for avoiding three kinds of false-positives typical of linear operators:

1. *Confusion between lines and edges:* Lines and edges are differentiated by their cross-sections. For accurate identification the logical conditions on the cross-section must be satisfied, and in each case we will show that a linear operator tests them incompletely (Fig. 3.3a).

2. *Merging or interference between nearby curves:* The local continuity of image curves is important for resolving and separating nearby features. Linear operators interfere with testing continuity by filling in gaps between nearby features and responding significantly to curves which are far from their preferred orientations (Fig. 3.3b).

3. *Smoothing out discontinuities or failing to localize line-endings:* The locations of the discontinuities and end-points of a curve are fundamental to higher-level descriptions [Bie85, KvD76, KvD82, Koe84]. Linear operators systematically

interfere with the localization of discontinuous points by responding whenever the receptive field of the operator overlaps the curve at all (Fig. 3.3c).

To tie all of this back to the discussion of Chap. 2 we observe that the conditions above can be summarized by stating that these local operators must *in isolation* act as reasonable estimates of the stringent thick trace condition of Def. 2.10 for image curves. Thus, ideally, a perfect "edge detector" should respond positively if and only if there exists an edge in the image whose local description intersects the Voronoi cell around which the detector is tuned. Each of the three conditions above amounts to a practical translation of one aspect of the assertion that the operator be a *reasonable* estimate of the thick trace intersection.

Chapter 4 A Logical/Linear Framework for Image Operators

The three qualitatively different kinds of image curves defined in §3.2 imply three distinct sets of preconditions for the existence of an image curve. As noted previously, the curve description process must respect these distinctions. Focusing for the moment on the line condition of (3.2b), we begin by adopting an oriented, linear line operator similar to the one described in [Can86].

Canny adopted the assumption of linearity to facilitate noise sensitivity analysis, and relied on post-processing to guarantee locality and selectivity of response. He arrived at a line operator whose cross-section is similar to a Gaussian second-derivative, and an edge operator similar to a Gaussian first derivative. Neurophysiologists [MTT78, SM84, JP87] and psychophysicists [SL85] have adopted such linear models to capture many of the functional properties of the early visual system, and the mathematics for analyzing them is widely known (e.g. Fourier analysis). These models are also attractive from a computational point of view because they exhibit most of the properties required of a measurement operator for image curves. However, they also exhibit the false-positive responses described above (partially shown in Figures 5.3 and 5.4).

To limit these false-positives, we will relax the assumption of linearity and test the necessary structural conditions explicitly. This is accomplished by developing an algebra of *Logical/Linear (L/L) Operators* which allow these conditions to be tested as the operator's response is being constructed. The resulting responses will appear to be linear as long as all of these conditions are fulfilled. Curiously, under these conditions they will mimic the response properties of "simple cells" in visual cortex.

4.1 Logical/Linear Combinators

As stated above, we wish to retain as much as possible of the desirable properties of the linear operator approach while allowing for the kind of structural analysis which can be used to categorize curves and verify continuity. We pursue these apparently contradictory goals by starting with an *optimal* linear operator, and then decomposing it in a way that allows for it's reconstruction, provided that the structural design conditions are verified.

In particular, we

1. Begin with a linear operator and decompose it into a set of linear component operators whose sum is identical to the initial operator.

2. These linear components represent measurement operators for the logical preconditions of the designed feature's existence.

3. The overall operator response is to be positive only if these structural preconditions are satisfied

4. For the range of inputs generating positive responses, the operator should act identically to the original linear operator.

The combination of operator responses to fulfill the third and fourth conditions above can be derived from a mapping of the real line to logical values. Assume that positive operator responses represent confirmation of a logical condition (logical TRUE) and that negative responses represent rejection (logical FALSE). To derive the numeric⇒logical mapping, we adopt the principle that *all confirming evidence should be combined if the logical condition holds, and contradictory evidence combined if the condition fails*. This leads to the following set of logical/linear combinators:

Definition 4.1 The *Logical/Linear combinators* $\Lambda\!\!\!\!\cdot$ and $\vee\!\!\!\!\cdot$ are given by

$$x \mathbin{\Lambda\!\!\!\cdot} y = \begin{cases} x + y, & \text{if } x > 0 \wedge y > 0; \\ y, & \text{if } x > 0 \wedge y \leq 0; \\ x, & \text{if } x \leq 0 \wedge y > 0; \\ x + y, & \text{if } x \leq 0 \wedge y \leq 0. \end{cases}$$

$$x \mathbin{\vee\!\!\!\cdot} y = \begin{cases} x + y, & \text{if } x > 0 \wedge y > 0; \\ x, & \text{if } x > 0 \wedge y \leq 0; \\ y, & \text{if } x \leq 0 \wedge y > 0; \\ x + y, & \text{if } x \leq 0 \wedge y \leq 0. \end{cases}$$

Before we descend into technical detail, it should be noted that these operators can be thought of as accumulating evidence for or against a particular hypothesis, with positive values being evidence 'for' and negative values evidence 'against'. Thus if an hypothesis h requires that both of two prior hypotheses (x and y) be true then consistent positive evidence from these inputs, represented by positive values, is required to produce a positive output $h = x \mathbin{\Lambda\!\!\!\cdot} y$. Should this combined hypothesis instead be rejected, all evidence for this rejection is combined. In all cases, the logical truth or falsehood of an hypothesis is contained in the *sign* of the value, while the strength of the evidence for or against the hypothesis is represented by the magnitude. It should be apparent that reasoning about the signs of derivatives (see §3.2) will be natural with this formalism.

4.2 A Logical/Linear Algebra

We now proceed to develop general properties of these L/L combinators and define a class of operators which embody these properties. With this background established, we can then move on to the development of the specialized operators we will use for early vision.

Using the combinators $\Lambda\!\!\!\!\cdot$ and $\vee\!\!\!\!\cdot$, we define a generative syntax for L/L expressions.

Definition 4.2 A *Logical/Linear operator* on the vector space X ($x \in X$) is any function $\mathrm{L} : X \rightarrow \mathbb{R}$ in the language $\mathcal{L}$ defined by the grammar:

$$\mathrm{L} \rightarrow \psi_i(x); \quad \mathrm{L} \rightarrow a_i\ \mathrm{L}; \quad \mathrm{L} \rightarrow \mathrm{L} \mathbin{\wedge\!\!\!\!\!+} \mathrm{L}; \quad \mathrm{L} \rightarrow \mathrm{L} \mathbin{\vee\!\!\!\!\!+} \mathrm{L}.$$

where each a_i is a real constant and each $\psi_i(x)$ is a bounded, real-valued, linear function.

Example 4.1 The expression

$$F(x,y) \;=\; x \mathbin{\wedge\!\!\!\!\!+} y$$

defines a L/L operator $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ which is positive only if both x and y are positive, in which case it evaluates as $F(x,y) = x + y$. An equivalent description of F as an operator is given by

$$F \;=\; \pi_1 \mathbin{\wedge\!\!\!\!\!+} \pi_2,$$

where π_i is the projection operator which selects the i^{th} dimension of X.

There are two fundamental properties which justify the use of the term Logical/Linear expressions to describe these forms: they comprise a *Boolean Algebra*, and they are *linear on certain subspaces* of their entire domain.

To show the first of these, consider the universe of vectors U in $\mathbb{R}^n$ excluding the axes[1]

$$U \;=\; \{\, x \in \mathbb{R}^n \mid x_i \neq 0 \,\} \tag{4.1}$$

and the subspaces $\{\, L(x)\,\}_+ = \{\, x \in U \mid L(x) > 0 \,\}$.

Theorem 4.3 (Logical) *For the language of L/L operators $L \in \mathcal{L}$, the set of all sets $\{\, L(x)\,\}_+$ and their complements $\overline{\{\, L(x)\,\}}_+ = U - \{\, L(x)\,\}_+$ form a* Boolean Algebra *with meet $\wedge\!\!\!\!\!+$, join $\vee\!\!\!\!\!+$ and complement $-$.*

The following equivalences can be derived directly from Definition 4.1, for all

[1] For real-valued variables, the exclusion of the axes needed to demonstrate logical equivalence is not problematic because it is a subset of measure 0.

$L_1, L_2 \in \mathcal{L}$:

$$\{ -L_1(x) \}_+ = \overline{\{ L_1(x) \}_+}, \qquad (4.2)$$

$$\{ L_1(x) \mathbin{\hat{\wedge}} L_2(x) \}_+ = \{ L_1(x) \}_+ \cap \{ L_2(x) \}_+, \qquad (4.3)$$

$$\{ L_1(x) \mathbin{\hat{\vee}} L_2(x) \}_+ = \{ L_1(x) \}_+ \cup \{ L_2(x) \}_+. \qquad (4.4)$$

It is easy to verify that these sets form a field with the help of the equivalences above (e.g. the equivalence of $\hat{\wedge}$ and $\cap$ ensures that if X and Y are members then $X \mathbin{\hat{\wedge}} Y$ is also). Furthermore, these meet, join and complement operators are clearly isomorphic to the standard set-theoretic $\cap$, $\cup$ and complement. The further observation that $\emptyset$ and U are the bounds of this field ensure that this system is a Boolean algebra. ([Sik60], p. 3) ∎

The following equivalences can also be derived directly:

$$\{ a\, L(x) \}_+ = \{ L(x) \}_+ \text{ if } a > 0 \qquad \{ a\, L(x) \}_+ = \{ -L(x) \}_+ \text{ if } a \leq 0$$

These demonstrate that the constant weights a_i in the language $\mathcal{L}$ act as either identity or complement and thus do not disturb the Boolean algebra.

Corollary 4.4 *Each L/L operator has an associated Boolean function created by substituting $\wedge$ and $\vee$ for $\hat{\wedge}$ and $\hat{\vee}$ respectively, and by replacing each a_i constant with either the identity function if positive or $\neg$ (negation) if negative. The truth value of each expression $\psi_i(x)$ is* TRUE *if $\psi_i(x) > 0$ and* FALSE *otherwise.*

Thus, continuing Ex. 4.1, the equivalent logical function $\hat{F}$ for F is

$$\hat{F}(x, y) = (x > 0) \wedge (y > 0).$$

The second fundamental property of these operators, their conditional linearity, is revealed by considering the *minimal* polynomials

$$P_j(x) \;=\; q_1(x) \mathbin{\hat{\wedge}} q_2(x) \mathbin{\hat{\wedge}} \cdots \mathbin{\hat{\wedge}} q_n(x) \tag{4.5}$$

where $q_i(x) = \psi_i(x)$ if bit i in the binary representation of j is zero, $q_i(x) = -\psi_i(x)$ if bit i is one. Then,

Theorem 4.5 (Linear) *Any L/L operator L is linear on the subspace $\{\, P_j(x) \,\}_+$ of any minimal polynomial $P_j(x)$.*

Any Boolean polynomial can be equivalently expressed as the join of minimal polynomials or the lower bound $\emptyset$ ([BM77], p. 370). Thus $\{\, L(x) \,\}_+$ can be expressed as the $\hat{\vee}$ of a group of such minimal polynomials of the $\psi_i(x)$'s (the *disjunctive canonical form* (DCF) of $L(x)$). Without loss of generality, consider a particular such polynomial $P_j(x)$. Noting that every element $\psi_i(x)$ for $x \in \{\, P_j(x) \,\}_+$ has a fixed value and thus fixed sign, Definition 4.1 guarantees that $\hat{\wedge}$ is linear on the subspace defined by $\{\, P_j(x) \,\}_+$ (for fixed sign arguments, the branch chosen in the $\hat{\wedge}$ is fixed). Thus, any minimal polynomial $P_j(x)$ is linear on $\{\, P_j(x) \,\}_+$.

Consider now the DCF of $L(x)$. We know that each $P_j(x)$ in this DCF is both linear and of constant sign on $\{\, P_j(x) \,\}_+$. By the same reasoning as for $\hat{\wedge}$ above, we can state that $\hat{\vee}$ is linear if its arguments have constant sign, and thus the DCF of $L(x)$ is a linear combination of expressions which are guaranteed linear on $\{\, P_j(x) \,\}_+$. Therefore, $L(x)$ is also linear on every $\{\, P_j(x) \,\}_+$. ∎

4.3 Logical/Linear Image Operators

By extension from the arithmetic operators, the L/L operators are applied pointwise to sequences of vectors or images. Thus, reconsidering Ex. 4.1 above, the operator F becomes

$$\forall x \in X\colon\; F(I_1, I_2)(x) = I_1(x) \mathbin{\hat{\wedge}} I_2(x), \;\; I_1, I_2 \colon X \to \mathbb{R}.$$

We are now ready to develop the class of L/L operators that we shall require to reason about images, and begin with an example.

Example 4.2 Suppose that the linear operators ψ_1 and ψ_2 provide a pointwise measure of two image properties (e.g. $\psi_1 = D_x^2$ and $\psi_2 = D_y^2$, the second directional derivatives) which are components of a more complex image property (e.g. locating convexity, the points where $D_x^2(I) < 0$ and $D_y^2(I) < 0$). If this aggregate property can be expressed as a logical combination of the signs of the linear properties, then we can build a L/L operator Ψ on the image such that

$$\Psi(I)(x) = \begin{cases} \text{positive}, & \text{if } x \text{ is a convex point;} \\ \text{negative}, & \text{otherwise.} \end{cases}$$

In this case, we would define

$$\Psi(I)(x) = (-D_x^2 * I)(x) \mathbin{\Lambda} (-D_y^2 * I)(x).$$

This example reveals a class of L/L operators appropriate for reasoning about images.

Definition 4.6 A *Logical/Linear convolution operator* Ψ is a L/L operator on an image I such that all $\psi_i(I)$ are linear convolutions of the form

$$\psi_i(I) = \psi_i * I = \int_X \psi_i(x - t)\ I(t)\, dt.$$

The operation of such an operator on an image is termed the Logical/Linear convolution of I by Ψ, and is written

$$\Psi(I) = \Psi * I.$$

Note that the linear convolution $\psi * I$ is a special case.

Returning to Ex. 4.2, we can assert that

$$\begin{aligned} \Psi(I) &= \Psi * I \\ &= (-D_x^2 * I) \mathbin{\Lambda} (-D_y^2 * I) \\ &= (-D_x^2 \mathbin{\Lambda} -D_y^2) * I \end{aligned}$$

thus justifying the notation we will use for describing L/L convolution operators:

$$\Psi \ = \ -D_x^2 \mathbin{\text{⩓}} -D_y^2.$$

This operator will produce an image whose elements are positive only for convex points of the input image.

An important relationship we will use to design image operators is that between a L/L operator and its linear reduction.

Definition 4.7 The *linear reduction* ψ of a L/L operator Ψ is that linear operator which is produced by substituting $+$ for each L/L combinator in the L/L operator description.

Corollary 4.8 *Given a the linear reduction $\psi(x)$ of a L/L operator $\Psi(x)$, a L/L convolution of $\Psi * I$ is exactly equal to the linear convolution of $\psi * I$ if the logical expression corresponding to the L/L expression is* TRUE.

Thus in fulfilling our goal of developing L/L image operators which retain some of the optimal behaviour of a particular linear operator, we will seek to design L/L operators which reduce to 'optimal' linear operators.

Before we move on to actual design, it will be important to examine a second, equivalent definition of the L/L combinators which has useful computational consequences.

Definition 4.9 The *ρ-approximate L/L combinators* are given by:

$$x \mathbin{\text{⩓}}_\rho y \ = \ x\Big(1 - \sigma_\rho(x)\,\sigma_\rho(-y)\Big) \ + \ y\Big(1 - \sigma_\rho(y)\,\sigma_\rho(-x)\Big), \ x, y \neq 0, \quad (4.6a)$$

$$x \mathbin{\text{⩔}}_\rho y \ = \ x\Big(1 - \sigma_\rho(y)\,\sigma_\rho(-x)\Big) \ + \ y\Big(1 - \sigma_\rho(x)\,\sigma_\rho(-y)\Big), \ x, y \neq 0, \quad (4.6b)$$

where

$$\sigma_\rho(x) = \frac{f(1/2+\rho x)}{f(1/2+\rho x)+f(1/2-\rho x)}, \qquad (4.7)$$

$$f(x) = \begin{cases} e^{-1/x}, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases}$$

The 'logistic' sigmoid function of [RHW86] is another option for $\sigma_\rho(x)$, but the fact that the function chosen is only non-singular (i.e. $0 < \sigma_\rho(x) < 1$) on $x \in [-1/2\rho, 1/2\rho]$ means that the "hard" logic of §4.1 still applies for values outside of this region.

The notion of ρ-approximate is clarified by the following.

Theorem 4.10

$$\lim_{\rho\to\infty} x \mathbin{\Lambda\!\!\!\!+}_\rho\, y = x \mathbin{\Lambda\!\!\!\!+} y$$

$$\lim_{\rho\to\infty} x \mathbin{V\!\!\!\!+}_\rho\, y = x \mathbin{V\!\!\!\!+} y$$

Note that

$$\lim_{\rho\to\infty} \sigma_\rho(x) = \sigma(x) = \begin{cases} 1, & \text{if } x > 0; \\ 0, & \text{otherwise.} \end{cases}$$

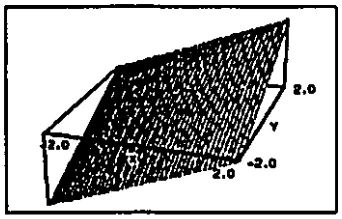
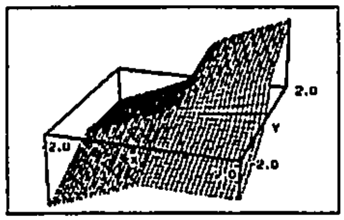
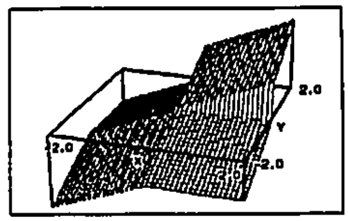
This function is a choice operator pivoting around zero, and as such it can be used to directly define the L/L combinators above. If this limit is substituted in eqs. (4.6), then they can be rephrased as

$$x \mathbin{\Lambda\!\!\!\!+} y = (x \textbf{ unless } \{x > 0 \wedge y \le 0\}) + (y \textbf{ unless } \{y > 0 \wedge x \le 0\})$$

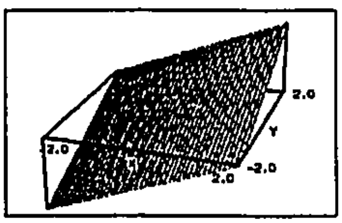
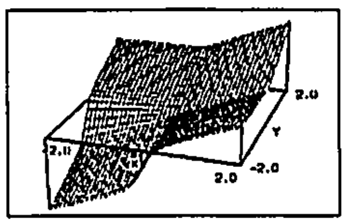
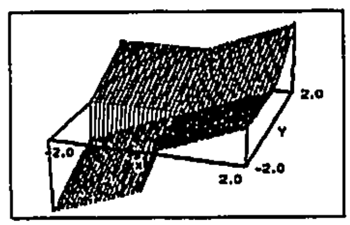
$$x \mathbin{V\!\!\!\!+} y = (x \textbf{ unless } \{x \le 0 \wedge y > 0\}) + (y \textbf{ unless } \{y \le 0 \wedge x > 0\})$$

It can be verified that these are equivalent to Def. 4.1. ∎

The approximations represented by $\mathbin{\Lambda\!\!\!\!+}_\rho$ and $\mathbin{V\!\!\!\!+}_\rho$ expose another relationship between the linear sum and the L/L combinators. Since $\sigma_0(x) = 1/2$, substitution of this value

(a) $x \wedge_\rho y$, $\rho = 0$ (b) $x \wedge_\rho y$, $\rho = 8$ (c) $x \wedge_\rho y$, $\rho = \infty$

(d) $x \vee_\rho y$, $\rho = 0$ (e) $x \vee_\rho y$, $\rho = 8$ (f) $x \vee_\rho y$, $\rho = \infty$

Figure 4.1: Graphs of the ρ-approximate L/L combinators varying ρ: (a), (b), and (c) show $x \wedge_\rho y$, (d), (e), and (f) show $x \vee_\rho y$. Note that as ρ varies between 0 and ∞, the combinators vary from purely linear to purely logical, with smooth interpolation in between.

into eq. (4.6) simplifies both L/L combinators to a linear combination

$$x \mathbin{\Lambda\!\!\!\cdot}_0\, y = 3/4\ (x+y)$$
$$x \mathbin{V\!\!\!\cdot}_0\, y = 3/4\ (x+y)$$

Thus, the ρ-approximates $\Lambda\!\!\!\cdot_\rho$ and $V\!\!\!\cdot_\rho$ form a continuous deformation from a linear combination to the absolute L/L operations as ρ goes from 0 to ∞ (see Fig. 4.1).

These ρ-approximates may be preferable to the ideal L/L operators in many practical situations. The most obvious situation in which one might prefer the smooth approximates is one in which there is some noise or uncertainty in the inputs. It could be disastrous if some small inputs are randomly positive or negative due to noise, especially if other inputs are strong and unambigous. The ρ-approximate L/L operators solve this problem by smoothing the logical transitions around zero, and thus significantly reducing the noise-sensitivity when small, potentially ambiguous inputs are introduced. The ρ-approximates also have advantages if differentiability is important (e.g. for optimization).

Of course, when we choose to use the approximate L/L combinators we are faced with the difficulty of choosing ρ. This can be problematic when the range of x and y is not precisely constrained. Since $1/2\rho$ is, in essence, a threshold on the logical significance of x or y, ρ should normally be set based on the expected range and uncertainty in x or y. Even if these are clearly defined, however, setting ρ to some fixed value can be problematic, in particular, $x \mathbin{\Lambda\!\!\!\cdot}_\rho y$ does *not* monotonically increase with increasing x for all values of y. Thus for some values of x and y an increase in x can result in a decrease of $x \mathbin{\Lambda\!\!\!\cdot}_\rho y$. This can be resolved simply by tying ρ to x and y. In particular,

$$\rho = \rho' / \max(x, y)$$

will ensure that such points of non-monotonicity do not exist. We refer to the L/L combinators with locally adaptive ρ as *adaptive ρ-approximates.*

Chapter 5 Designing L/L Operators for Image Curves

Using the framework defined above, we now proceed to derive a family of L/L image operators to locate and describe image curves as defined in Definition 3.1. We begin by observing that the conditions expressed in eqs. (3.2b,3.2c,3.2a) segregate into *independent* one-dimensional conditions in orthogonal directions—along the tangent and the normal. The normal condition selects the proper contrast cross-section to define a (positive or negative contrast) line or edge, and the tangential condition ensures local C^1 continuity of the inferred curve. Thus, our solution is a separable family of two-dimensional operators expressed as the Cartesian product of orthogonal, one-dimensional L/L operators, one normal $\mathrm{N}(y)$ and the other tangential $\mathrm{T}(x)$ to some preferred direction. With (x, y) defining a local, orthonormal coordinate system, we seek

$$\Psi(x,y) = \mathrm{T}(x) \times \mathrm{N}(y) \quad \text{or} \quad \Psi = \mathrm{T} \times \mathrm{N}.$$

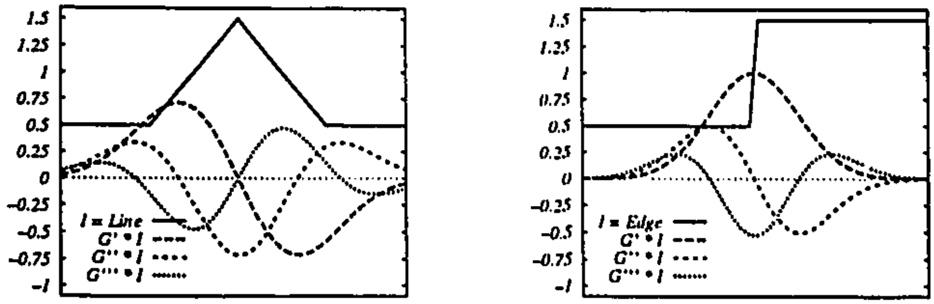
Moreover, the tangential condition (C^1 continuity), and thus the tangential operator T, is identical for all three image curve types.

Thus, we divide the design into three stages:

- First, derive a set of one-dimensional L/L operators $\{\mathrm{N}_P, \mathrm{N}_N, \mathrm{N}_E\}$ (for positive and negative lines and edges, respectively) which verify the cross-sectional (normal) conditions of Definition 3.1, while avoiding the pitfalls discussed in Chapter 3.

- Then, derive a one-dimensional L/L operator T which is capable of discriminating between locally continuous and discontinuous curves along their tangent direction.

- Finally, form a family of direction-specific two-dimensional L/L image curve operators by forming the Cartesian product of the two one-dimensional operators.



(a) (b)

Figure 5.1: Cross sections of image lines and edges. A line in an intensity image (a) is located at the peak of its cross-section. Note that this coincides with a zero in the derivative β' and a negative second derivative β''. An intensity edge (b) occurs at peaks in the derivative β' of the cross-section. The derivatives shown are derived from convolution by G$'$ and G$''$ operators with $\sigma = 3$.

5.1 The Normal Operators: Categorization

For the purpose of illustration, we will begin with the analysis of a *positive contrast line* (3.2b). The methodology developed will apply naturally to the two other image curve types.

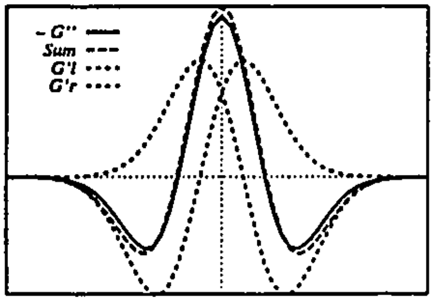
Since a necessary condition for the existence of such a line is a local extremum in intensity (fig. 5.1 is a display of typical 1D cross-sections of lines and edges), we will first consider the operator structure normal to its preferred orientation. This is just the problem of locating extrema in the cross-section β_s.

A local extremum in a one-dimensional differentiable signal $\beta(x)$ exists only at those points where

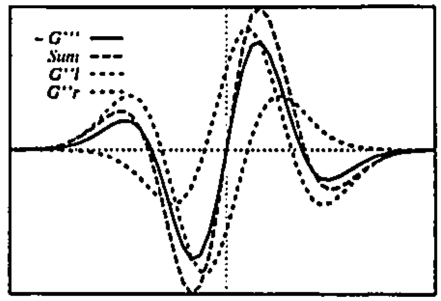
$$\beta'(x) = 0 \text{ and } \beta''(x) \neq 0. \tag{5.1}$$

Estimating the location of such zeroes in the presence of noise is normally achieved by locating zero-crossings, thus in practice these conditions become

$$\beta'(x - \epsilon) > 0 \text{ and } \beta'(x + \epsilon) < 0 \text{ and } \beta''(x) < 0 \tag{5.2}$$



(a)



(b)

Figure 5.2: Central differences suggest that an approximation to the n^{th} derivative can be obtained from a difference between two displaced $(n-1)^{th}$ derivatives. Thus in (a) the sum of two G′ operators approximate $-G''$, and in (b) the sum of two G″ operators approximate $G^{(3)}$.

for some $\epsilon > 0$. An operator which can reliably restrict its responses to only those points where these conditions hold will only respond to local maxima in a one-dimensional signal.

A set of noise-insensitive linear derivative operators (or 'fuzzy derivatives' [KvD87]) are the various derivatives of the Gaussian,

$$G_\sigma(x) \;=\; \frac{1}{\sqrt{2\pi}\sigma}\, e^{-x^2/2\sigma^2},$$

which will be expressed as $G'_\sigma(x)$, $G''_\sigma(x)$, etc. These estimators are optimal for additive, Gaussian, i.i.d. noise.

When convolved over a one-dimensional signal these give noise-insensitive estimates of the derivatives of the signal, for example

$$\beta'_\sigma(x) \;=\; \beta'(x) * G_\sigma(x) \;=\; \beta(x) * G'_\sigma(x), \tag{5.3}$$

Theorem 5.1 $\beta'_\sigma(x-\epsilon) > 0$ *and* $\beta'_\sigma(x+\epsilon) < 0$ *and* $\beta''_\sigma(x) < 0$ *are sufficient conditions* *for a local maximum in the signal* $\beta(x)$.

The identity in (5.3) shows that these conditions are necessary and sufficient for the existence of a local maximum in $\beta_\sigma(x) = \beta(x) * G_\sigma(x)$. The maximum principle for the heat equation ([PW84], p. 161) implies that convolution by a Gaussian cannot introduce new maxima. Thus the above conditions imply the existence of a maximum in $\beta(x)$. ∎

This suggests a practical method for locating maxima in a noisy one-dimensional signal. Comparing the results of convolutions by derivatives of Gaussians will allow us to determine the points where Theorem 5.1 holds. The loci of such points will form distinct intervals with widths $\leq 2\epsilon$. The parameter σ determines the amount of smoothing used to reduce noise-sensitivity.

Observe by central limits that:

$$f'(x) \;=\; \lim_{\epsilon\to 0} \Big(f(x+\epsilon) - f(x-\epsilon)\Big)/2\epsilon.$$

Thus for the derivative estimates β_σ, one would expect that

$$-\beta''_\sigma(x) \approx \Big(\beta'_\sigma(x-\epsilon) - \beta'_\sigma(x+\epsilon)\Big)/2\epsilon$$

with the accuracy of the approximation a function of ϵ. Thus the conditions in Theorem 5.1 can be verified from examination of the derivative $\beta'_\sigma(x)$—a linear combination of two points will give $-\beta''_\sigma(x)$. More specifically, we adopt the approximation $-G''_\sigma(x) \approx \Big(G'_\sigma(x+\epsilon) - G'_\sigma(x-\epsilon)\Big)/2\epsilon$, where $\epsilon/\sigma \leq 1$. Figure 5.2a shows that for $\epsilon = \sigma/2$ this is an acceptable approximation.

Thus, convolution by G' allows testing of all three conditions in Theorem 5.1 simultaneously. Using the L/L combinators of §4.1 we are now able to define a one-dimensional operator which has a positive response only within a small range of local maxima.

Operator 5.1 The one-dimensional normal operator for *local maxima* N_M is

$$N_M = n'_l \curlywedge n'_r \tag{5.4}$$

where

$$\begin{aligned} n'_l &= G'_\sigma(x+\epsilon)/2\epsilon, \\ n'_r &= -G'_\sigma(x-\epsilon)/2\epsilon. \end{aligned}$$

Clearly then, the key advantage of this N_P operator is that:

Observation 5.2 The response $N_P(\beta)(x)$ will be positive only if there is local maximum in β_σ within the region $[x-\epsilon, x+\epsilon]$.

By reference to Definition 4.1 we can see that $N_P(\beta)(x) > 0$ implies that both $n'_l(\beta)(x) > 0$ and $n'_r(\beta)(x) > 0$. Equation 5.3 then implies that

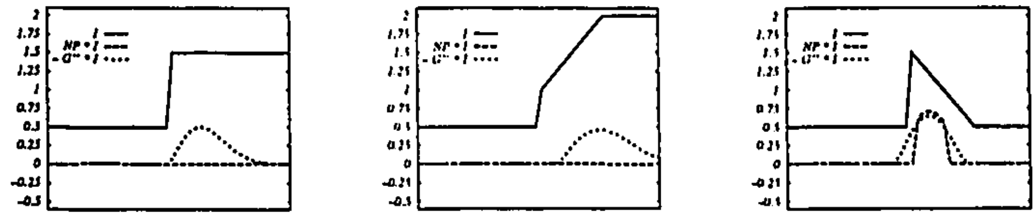
$$n'_l(x) * \beta(x) = \beta'_\sigma(x-\epsilon)/2\epsilon \tag{5.5}$$

$$n'_r(x) * \beta(x) = -\beta'_\sigma(x+\epsilon)/2\epsilon \tag{5.6}$$

Thus a positive response ensures that $\beta'_\sigma(x-\epsilon) > 0$ and $\beta'_\sigma(x+\epsilon) < 0$, which in turn imply the presence of a local maximum on β_σ between $x-\epsilon$ and $x+\epsilon$.[1] ∎

The performance improvement from introducing this non-linearity is considerable. The linear operator exhibits consistent patterns of false positive responses. The simplest example is the response near a step (see fig. 5.3). The linear operator displays a characteristic (false) peak response when the step is centered over one of the zeroes in the operator profile. The logical/linear $\curlywedge$ operation prevents this error since both G′

[1] Observe that although the local maximum in β_σ is guaranteed to fall within this region, the corresponding maximum in β is not necessarily so restricted. Qualitatively however, we can rely on the observation that the maxima for a signal will converge on the centroid of that signal under heat propagation (or as we convolve with larger and larger Gaussians). Considering the features of β in isolation then, we can state that the smoothing will cause the location of the local maximum in β_σ to shift towards the centroid of the local intensity distribution, a phenomenon observed in studies of biological visual systems (e.g. the vernier acuity studies of [WM83]).



(a) (b) (c)

Figure 5.3: Responses of L/L positive contrast line operator and the linear operator $-G''$ which it reduces to, near a step edge whose local profile varies from the ideal. The graphs show the image profiles being operated on, covering (a) a simple step edge, (b) a compound step with slope above > 0, and (c) a compound step with slope above < 0. It can be seen that the L/L operator blocks the unwanted response near a step which is not also a local maximum (a,b), but that when the edge is also a local maximum (c) it does respond. The linear $-G''$ operator, however, responds positively in each of these cases, exhibiting consistent (and erroneous) displacement of the peak response.

halves of the operator register derivatives in the same direction and so do not fulfill the conditions of (5.2). The L/L operator will respond positively only in the case that the slope above the step is negative (i.e. only when the transition point is also a local maximum).

A more specific operator can be derived by examining the implications of (3.2b) beyond the local maxima. A discontinuous peak, such as that shown Fig. 5.1a is not only a negative local minimum in β''_σ, but a positive local maximum in $\beta^{(4)}_\sigma$. Thus two addition conditions are required

$$\beta^{(3)}(x) = 0 \text{ and } \beta^{(4)}(x) > 0.$$

This can again be captured by central differences, combining two offset third-derivative estimates.

Operator 5.2 The one-dimensional normal operator for *positive contrast lines* N_P is

$$\mathrm{N}_P \quad = \quad \mathrm{n}'_l \wedge \mathrm{n}'_r \wedge \mathrm{n}^{(3)}_l \wedge \mathrm{n}^{(3)}_r \tag{5.7}$$

where

$$n_l^{(3)} = -G_\sigma^{(3)}(x+\epsilon)/2\epsilon,$$
$$n_r^{(3)} = G_\sigma^{(3)}(x-\epsilon)/2\epsilon.$$

The extension of this analysis to the other curve types in §3.2 is straightforward. The above analysis can be repeated in its entirety with a simple change of sign so as to be specific for an identical feature of the opposite contrast.

Operator 5.3 The one-dimensional operator for *negative contrast lines* N_N is

$$N_N = -n_l' \mathbin{\triangle} -n_r' \mathbin{\triangle} -n_l^{(3)} \mathbin{\triangle} -n_r^{(3)}.$$

Slightly more complicated is the case for edges. In the simplest case, a rising discontinuity is signalled by a local maxima of the first derivative, thus imposing the following conditions

$$\beta''(x) = 0 \text{ and } \beta^{(3)}(x) < 0 \tag{5.8a}$$

or

$$\beta''(x-\epsilon) > 0 \text{ and } \beta''(x+\epsilon) < 0. \tag{5.8b}$$

This condition is just the familiar zero-crossing of a second derivative, exactly the condition used by Haralick [Har82] and Canny [Can86]. Note that this operator actually selects any *inflection points* in the signal.

Mirroring the analysis above, the verification of these conditions can be realized in an L/L operator selecting inflection points N_I:

Operator 5.4 The one-dimensional operator N_I for *inflection points* is

$$N_I = n_l'' \mathbin{\triangle} n_r''.$$

where

$$n_l'' = G_\sigma''(x+\epsilon),$$

$$\mathrm{n}_r'' \;=\; -\,\mathrm{G}_\sigma''(x-\epsilon).$$

Now as with the line operators, selection of more truly edge-like features is possible by examination of other derivatives. Note that a blurred step edge has vanishing even derivatives and sign-alternating non-zero odd derivatives (see fig. 5.1). The description of an edge adopted in (5.8) is clearly consistent with this observation, but incomplete. Note also that an "edge" is the derivative of a "peak," which was used for analysing line-like images. With this additional information, we can adopt a more selective operator for image edges which requires that all of the following conditions must be verified

$$\beta'(x) > 0 \text{ and } \beta''(x) = 0 \text{ and } \beta^{(3)}(x) < 0 \text{ and } \beta^{(4)}(x) = 0 \text{ and } \beta^{(5)}(x) > 0 \qquad (5.9a)$$

or

$$\beta'(x) > 0 \text{ and } \beta''(x-\epsilon) > 0 \text{ and } \beta''(x+\epsilon) < 0 \text{ and } \beta^{(4)}(x-\epsilon) < 0 \text{ and } \beta^{(4)}(x+\epsilon) > 0 \qquad (5.9b)$$

These conditions[2] can be verified in an L/L edge operator N_E:

Operator 5.5 The one-dimensional operator N_E for *edges* is

$$\mathrm{N}_E \;=\; \mathrm{n}_c' \wedge \mathrm{n}_l'' \wedge \mathrm{n}_r'' \wedge \mathrm{n}_l^{(4)} \wedge \mathrm{n}_r^{(4)}$$

where

$$\begin{aligned}
\mathrm{n}_c' &= \mathrm{G}_\sigma'(x),\\
\mathrm{n}_l^{(4)} &= -\,\mathrm{G}_\sigma^{(4)}(x+\epsilon),\\
\mathrm{n}_r^{(4)} &= \mathrm{G}_\sigma^{(4)}(x-\epsilon).
\end{aligned}$$

This operator is significantly more selective than the 'zero-crossing of a second-

[2]The condition $\beta'(x) > 0$ is actually also implemented by Haralick [Har82] and Canny [Can86], since their lateral maxima selection is followed by a threshold $\beta'(x) > \theta$, where θ is positive.

derivative,' [MH80] which is only one of the logical preconditions on which this operator depends. One can therefore expect less of a problem of non-edge signals generating edge-like responses with this operator than with these other less specific operators.

It is important to realize that the operator family which forms the basis for this analysis is the Derivatives of Gaussian family of operators. Koenderink [Koe88, KvD90] derived this family as one orthonormal solution of the problem of deriving size-invariant spatial samplings of images. Members of this operator family can be transformed into each other via a set of simple, unitary transformations. This has definite computational advantages, since the higher derivatives and spatial offsets from pixel centers may be derived from a small canonical set of operators by linear combinations. In addition, Young [You85] has persuasively argued that this is exactly these are exactly the basis functions which are used by primate visual systems.

5.2 The Tangential Operator: Continuity

So far only the normal image structure (β_s) has been discussed. In order to extend this result to two-dimensions, we must examine the tangential (curvilinear) structure of the curves (α). By Definition 3.1 we must verify the local continuity of candidate curves. In addition, the extraction of orientation-specific measures was deemed essential for further processing. In this section, these problems will be addressed by imposing a further tangential structure on the operator. We will follow the same course as for the normal cross-section—first a linear structure will be proposed which will then be decomposed to reveal linear measurement operators for the components of the structural preconditions. The emphasis again is placed on these preconditions and their L/L combination.

Consider the image cross-section that is tangent to the image curve α at every point. Assume that the intensity variation along this curve is everywhere smooth and corrupted only by additive Gaussians noise. The local contrast along the curve as compared against its background is an acceptable measure of the curve's salience. This suggests filtering image noise with a linear Gaussians operator $\mathrm{t}(x) = \mathrm{G}_{\sigma_x}$ along

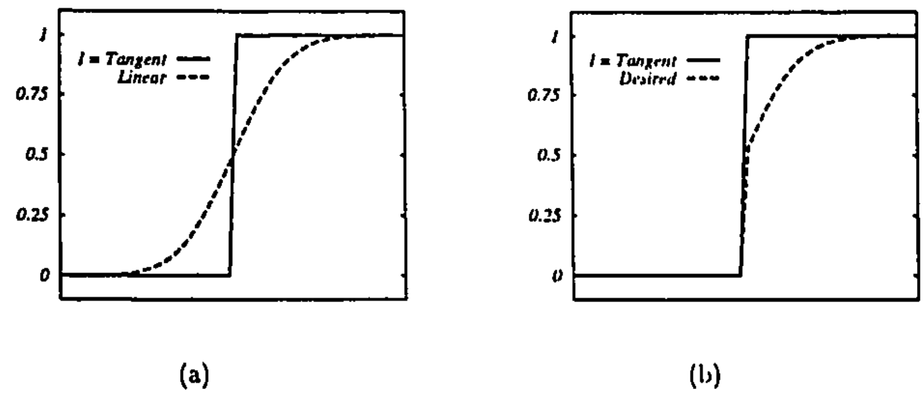


Figure 5.4: The signal is the tangential section of an image line near the discontinuous termination of the line (the endline). Note that the linear operator (a) exhibits a smooth attenuation of response around the line ending. We seek an operator (b) whose response attenuates abruptly at or near the endline discontinuity.

the tangential direction.

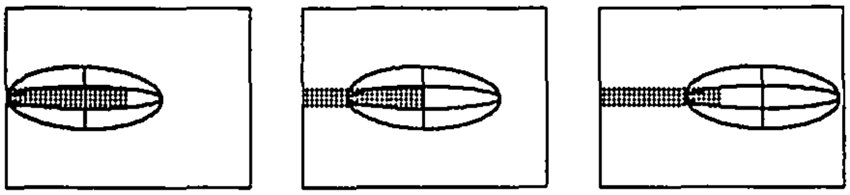
Near a curve end-point, however, the tangential section will exhibit an abrupt discontinuity (see fig. 5.4). The indiscriminate smoothing of the Gaussians will obscure this contrast discontinuity by, in effect, assuming that no discontinuity is present before it is applied, thus violating the third criterion of §3.2. *The local continuity of the curve must be verified prior to smoothing.*

To resolve this, consider a definition of the local continuity of a function. The function $f(x)$ is said to be *continuous* at x_0 iff

$$\lim_{x \to x_0 -} f(x) \;=\; \lim_{x \to x_0 +} f(x) \;=\; f(x_0). \tag{5.10}$$

For our purposes, assume that the non-linearities associated with the normal components of the L/L image curve operators are evaluated *before*[3] those in the tangential L/L operators. Then a curve termination point in the image must be signalled by

[3]With a pure linear operator expressed as the Cartesian product of normal and tangential one-dimensional operators, order-of-evaluation is unimportant, but with Logical/Linear operators order-of-evaluation can be essential.



(a) (b) (c)

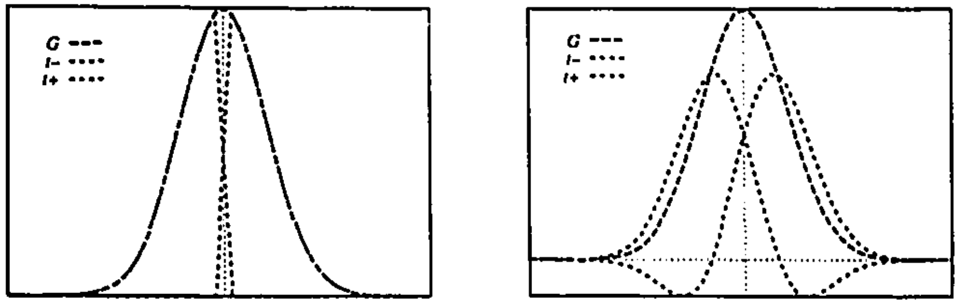
Figure 5.5: Schematic of the half-field decomposition and line endings. The elliptic regions in each figure represent the operator positions as the operator is placed beyond the end of an image line. In (a) the operator is centered on the image line and the line exists in both half-fields. In (b) the operator is centered on the end-point and whereas the line only exists in one half-field, the other half-field contains the end-point. In (c) the operator is centered off the line and the line only exists in one half-field.

a *contrast sign reversal* in the image section seen by the tangential L/L operator—a transition from a region which has been confirmed to be of the given category (positive response) to a region which has been rejected (negative response). We will call the behaviour which the tangential operator must exhibit *end-line stability.* A one-dimensional operator is end-line stable if and only if it responds positively only when centered on a uniformly positive region of the image.

Representing the intensity variation along the curve α as a function of the arc-length $I_\alpha(s)$, the worst-case line-ending (or beginning) is a step in intensity at $s = 0$. End-line stability requires that the operator's response $\mathrm{T}(I_\alpha)(s)$ be non-positive for all $s \leq \epsilon$, and positive for $s > \epsilon$.[4] Given the requirement for symmetric approach outlined in eq. (5.10), from fig. 5.5 we observe that this can be achieved by separately considering the behaviour of the curve in each tangential direction around the operator centre.

We therefore adopt a partition which divides the operator kernel into two regions along its length. Using the step function $\sigma(x)$ of eq. (4.3) a partition of $\mathrm{G}(x)$ around

[4] This property must also operate symmetrically at the other end of the curve.



(a) (b)

Figure 5.6: A one-dimensional Gaussian (representing the tangential operator t) partitioned into two regions (a) to obtain the two half-field operators defined by eq. (5.11). The addition of 'stabilizers' is shown in (b).

0 is given by

$$\mathrm{t}^-(x) \;=\; \mathrm{G}(x)\,\sigma(-x), \qquad \mathrm{t}^+(x) \;=\; \mathrm{G}(x)\,\sigma(x). \tag{5.11}$$

Operator 5.6 The one-dimensional operator for *tangential continuity* T is

$$\mathrm{T} \;=\; \mathrm{t}^- \triangle\, \mathrm{t}^+. \tag{5.12}$$

Note that $\mathrm{t}^-(x) + \mathrm{t}^+(x) = \mathrm{G}(x) = \mathrm{t}(x)$ for all x, as required. The smooth partition operator $\sigma_\rho(x)$ of eq. (4.7) can be used for a smooth, stable partition.

Observation 5.3 The operator T is *end-line stable*.

Consider the component responses in the neighbourhood of the step edge $I_a(s) = \sigma(s)$. The response of t^+ to this step is given by

$$\begin{aligned}
\mathrm{t}^+(I_a)(s) &= (\mathrm{t}^+ * \sigma)(s) \\
&= \int_{-\infty}^{\infty} \mathrm{t}^+(s-\tau)\,\sigma(\tau)\,d\tau \\
&= \int_0^{\infty} \sigma(s-\tau)\,\mathrm{G}(s-\tau)\,d\tau
\end{aligned}$$

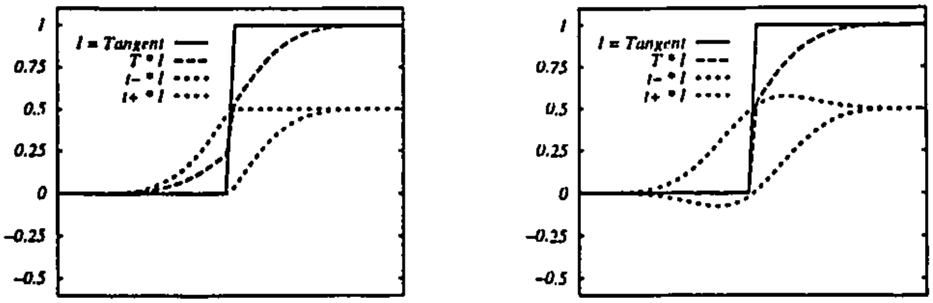


Figure 5.7: Responses (including component responses) of an unstabilized end-line operator (a) and the stabilized version (b). Note that the L/L combination of the unstabilized components (a) does not, in fact, reduce to zero beyond the end of line. This is due to the use of the L/L $\triangle_p$ approximation with $\rho < \infty$. In order to produce stable attenuation at a line ending, inhibitory regions (stabilizers) are added to the t^- and t^+ components, which have the effect of pushing the component responses 'near' but 'off' the line-ending below zero (b).

$$= \int_{-\infty}^{s} \sigma(\tau)\, G(\tau)\, d\tau$$

$$= \begin{cases} \int_0^s G(\tau)\, d\tau, & \text{if } s > 0; \\ 0, & \text{if } s \leq 0. \end{cases}$$

The L/L AND of t^+ and t^- to produce T requires that both component responses be strictly positive for a positive response, thus whenever $s \leq 0$ around the step described above, the T response is also zero. It is obvious that the same analysis applies to the symmetric t^- component and the $1 - \sigma(s)$ step edge, which describes behaviour around the other end of the line. Thus the T operator is end-line stable symmetrically around a step edge. ∎

The above proof, however, depends critically on the use of ideal L/L combinators, while in most cases we would prefer to use non-ideal combinators ($\triangle_p$ where $\rho < \infty$). When the non-ideal combinators are used, the 'end-line stable' operator described above does not properly attenuate responses beyond the line ending (see Fig. 5.7a).

In order to achieve this attenuation, it is necessary to force the component responses in the region just beyond a line ending significantly below zero.

This is achieved with the addition of the 'stabilizers' (shown in Fig. 5.6b):

$$\mathrm{t}^{-}(x) = \mathrm{G}(x)\,\sigma_a(-x) + b\mathrm{G}'(x), \tag{5.13}$$

$$\mathrm{t}^{+}(x) = \mathrm{G}(x)\,\sigma_a(x) - b\mathrm{G}'(x). \tag{5.14}$$

Thus, a smooth partition of $\mathrm{G}(x)$ by $\sigma_a(x)$ is augmented with an overshoot $-b\mathrm{G}'(x)$. The overshoot guarantees that when the center of the operator is near the line ending (see Fig. 5.7b) one component will give a negative response over the region where the operator is not centered *on* the line. Since the stabilizers are symmetric, it does not matter whether the operator is near a rising or falling line-ending—if the operator is centered over the positive region it will respond. Furthermore, since the integral of the stabilizers is zero, they will have no effect whatsoever on a locally constant signal. The candidate tangential operator is then the L/L AND of these stabilized components. The parameters a and b are chosen so that the cutoff is exactly aligned with the line ending.

An extension of this principle to multiple regions can lead to greater noise insensitivity (as suggested by Davis [DRA76]). For even n the increasing sequence of partition points $(x_1, \ldots, x_{n-1})$ can be used to partition $\mathrm{t}(x)$ into n regions where

$$\begin{aligned}
\mathrm{t}_1(x) &= \mathrm{t}(x)\,\sigma(x_1 - x) \\
\mathrm{t}_i(x) &= \mathrm{t}(x)\,(\sigma(x_i - x) + \sigma(x - x_{i-1}) - 1) \\
\mathrm{t}_n(x) &= \mathrm{t}(x)\,\sigma(x - x_{n-1}).
\end{aligned}$$

If we then constrain the x_i so that $\forall i\colon \int \mathrm{t}_i(x)\,dx = 1/n$ then this is a partition into n equal-area regions with $x_{n/2} = 0.0$. In order to guarantee end-line stability for the responses to these regions, their L/L combination must guarantee that at least one of the regions $i \leq n/2$ and at least one of those for which $i > n/2$ responds positively (i.e. there is positive support from both sides). Furthermore, if all of the regions for which $i \geq n/2$ have positive response then the overall response must be positive, since this is exactly what happens at the end point of an ideal line. Finally, Davis suggested that the majority of regions should be required to be positive.

We have examined two of the possible L/L combinations which exhibit this behaviour.

Operator 5.7 The *simple* one-dimensional combination for tangential continuity over n components is

$$\mathrm{T} \;=\; (\mathrm{t}_{n/2} \overset{+}{\wedge} \mathrm{t}_{n/2+1}) \overset{+}{\wedge} \left(\overset{+}{\bigwedge_{i<n/2}} \mathrm{t}_i \;\overset{+}{\vee}\; \overset{+}{\bigwedge_{i>n/2+1}} \mathrm{t}_i \right).$$

Thus if both central regions are positive and either of the extensions to the left or right, then the aggregate response is positive.

Operator 5.8 The *majority* one-dimensional combination for tangential continuity over n components is

$$\mathrm{T} \;=\; \overset{+}{\bigvee_{C}} \overset{+}{\bigwedge_{i\in C}} \mathrm{t}_i,$$

where C is a sequence of all choices of $n/2+1$ regions from $(1,\ldots,n)$.

There are three observations we can make from these designs:

1. Both of these reduce to eq. (5.12) for $n = 2$.
2. For $n > 2$, both these operators impose a minimal length of the positive region which generates a positive aggregate response.
3. In terms of pure L/L combinators, the implementation cost of the majority combination for $n > 4$ is much greater than the simple combination.

Henceforth, when we show the tangential combination as $\mathrm{t}^- \overset{+}{\wedge} \mathrm{t}^+$ we will assume that the decompositions and either of the L/L combinations (simple or majority) may be substituted without other modification. Finally, we note without further comment the similarity between this approach and the ANDing of LGN (lateral geniculate nucleus) afferents proposed by Marr and Hildreth [MH80]).

5.3 The Two-Dimensional Image Operators

Finally then, we can construct the two-dimensional image curve operators by taking the Cartesian product of the normal and tangential components. In order to complete the analysis, we unify this tangential continuity combination with the normal combination (see Fig. 5.8 for an example).

Operator 5.9 The *Logical/Linear image curve operators* Ψ_i (where i selects the operator category) are given by

$$\Psi_i \;=\; (\mathrm{t}^- \times \mathrm{N}_i) \mathbin{\mathbin{\wedge\!\!\!\!\wedge}} (\mathrm{t}^+ \times \mathrm{N}_i),\; i \in \{P, N, E\}$$

where

$$\begin{aligned}
\mathrm{N}_P &= \mathrm{n}'_l \mathbin{\wedge\!\!\!\!\wedge} \mathrm{n}'_r \mathbin{\wedge\!\!\!\!\wedge} \mathrm{n}^{(3)}_l \mathbin{\wedge\!\!\!\!\wedge} \mathrm{n}^{(3)}_r && \text{for Positive Contrast Lines,}\\
\mathrm{N}_N &= -\mathrm{n}'_l \mathbin{\wedge\!\!\!\!\wedge} -\mathrm{n}'_r \mathbin{\wedge\!\!\!\!\wedge} -\mathrm{n}^{(3)}_l \mathbin{\wedge\!\!\!\!\wedge} -\mathrm{n}^{(3)}_r && \text{for Negative Contrast Lines,}\\
\mathrm{N}_E &= \mathrm{n}'_c \mathbin{\wedge\!\!\!\!\wedge} \mathrm{n}''_l \mathbin{\wedge\!\!\!\!\wedge} \mathrm{n}''_r \mathbin{\wedge\!\!\!\!\wedge} \mathrm{n}^{(4)}_l \mathbin{\wedge\!\!\!\!\wedge} \mathrm{n}^{(4)}_r && \text{for Edges.}
\end{aligned}$$

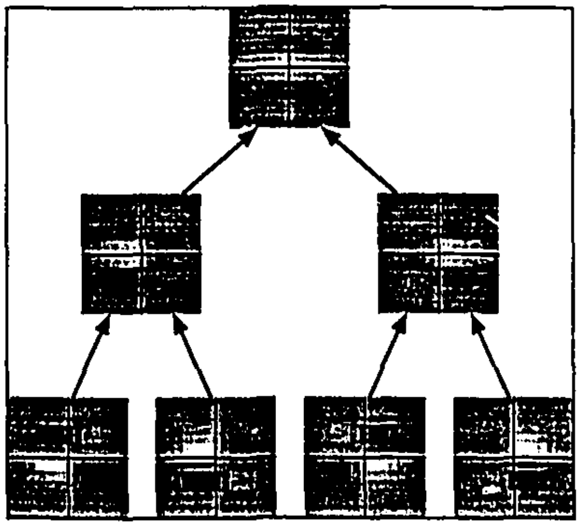


Figure 5.8: Illustration of the construction of the two-dimensional positive contrast line operator. Each of the bottom row of operators is a linear operator which is formed by one of the linear component operators $n_{l,r} \times t_{l,r}$. The middle row represents the linear reduction of the operators $t_{l,r} \times N_P$, in other words the sum of the two operators below. The operator shown at the top of the pyramid is the linear reduction of Ψ_P, the sum of the middle operators. The cross-hairs represent the centre of each operator and are provided solely for purposes of alignment.

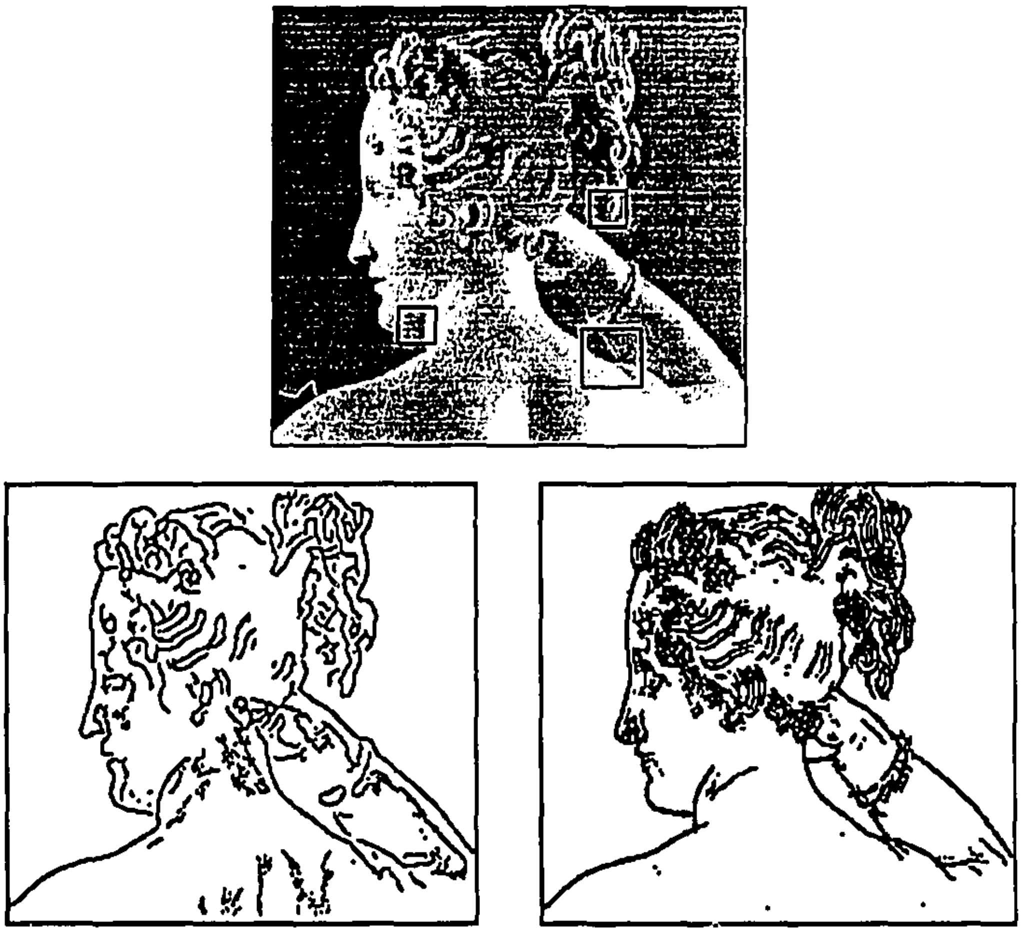
Chapter 6 Logical/Linear Results

As per the decomposition into curve types described above, we create three different classes of curve operator, for positive and negative contrast lines and for edges. The operators in the following examples all have a tangential $\sigma = 2.0$ and a lateral $\sigma = \sqrt{2}/2$ pixels. The ϵ of the the lateral operator separations is $\sqrt{2}/2$. This ensures that all curves are localized to connected regions with width $\leq \sqrt{2}$ pixels.

For the comparison images, Canny's algorithm was applied with an upper threshold of $\approx$ 15% contrast. This value was adequate for suppressing most noise, although some of the examples show that the noise has not been entirely eliminated. The low threshold was set to 1% so as to come close to matching the sensitivity of the L/L operators to very faint structures.

A natural but informal evaluation criterion for edge operators is the degree to which the 'edge map' produced corresponds to a reasonable line drawing of an image. We therefore use a test image of a statue "Paolina" not unlike the subjects in Michelangelo's drawings. This drawing is particularly suitable because, as Koenderink has pointed out, the representation of creases and folds is especially important for conveying a sense of three-dimensional structure [KvD76, KvD82]. An examination of the Canny and L/L edge images for the statue reveals a marked difference in the ability to distinguish perceptually salient edges from other kinds of intensity changes. Comparison with Michelangelo's treatment reveals clearly that the L/L operators represent more of the significant structure than the Canny operator.

Formal criteria for an image curve were established in §3.2, and these provide less subjective demonstrations of where the Canny operator fails. We stress that our goal here is not to focus on the shortcomings of the Canny operator, but rather to indicate the shortcomings of the long tradition of edge operators that consist of linear convolutions followed by thresholding, from Sobel [DH73] through Marr-Hildreth [MH80] and most recently in Canny.



(a) (b)

Figure 6.1: Image of statue (a), provided by Pietro Perona, and edge maps computed by: (b) Canny's algorithm ($h = 15\%$), and (c) L/L operators (both algorithms are run at the same scale). Compare these representations with the human expert's line drawing in Fig. 6.2, especially around the chin and neck. The Canny operator consistently signals non-salient 'edges', misses edges in complex neighbourhoods (e.g. near the T-junction of the chin and neck) and shows discontinuous orientation changes as smooth. (The boxes represent the approximate locations of the details shown in subsequent figures).

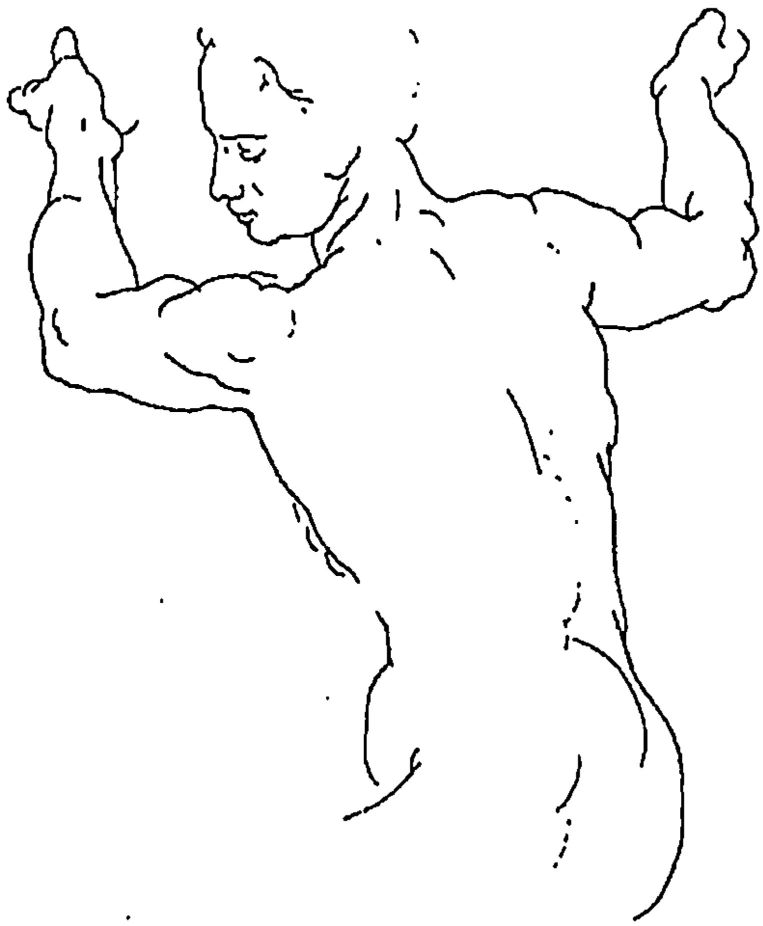


Figure 6.2: Line drawings, such as this Michelangelo, demonstrate in a clear and compelling manner the significance of image curves for the visual system. A well-executed line drawing depends critically on curvature, line terminations and junctions for its visual salience. Koenderink has stressed how the "bifurcation structures" define the arm and shoulder musculature and the manner in which the chin occludes the neck. Observe the similarity between this and the L/L operator responses, and differences with the Canny operator.

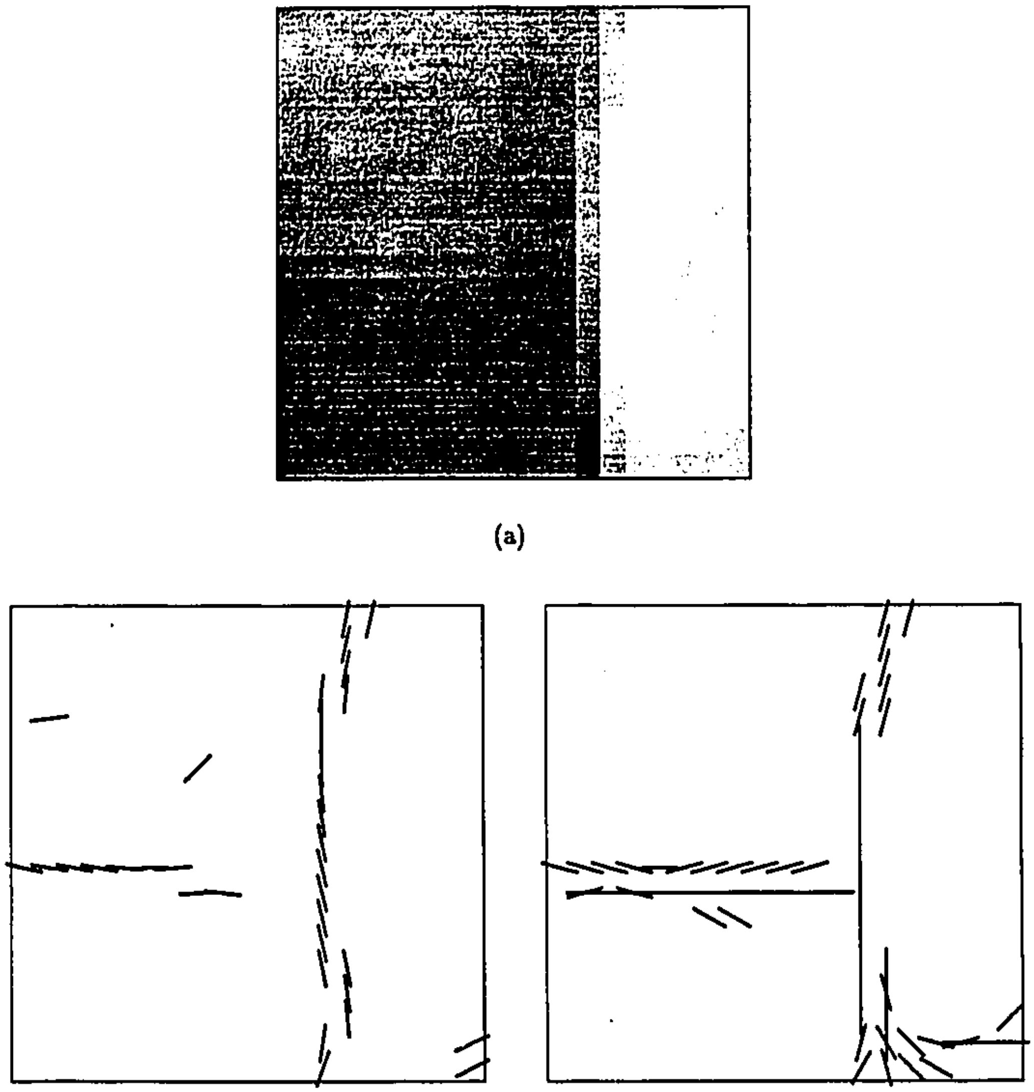
The first criterion, the need for predictable behaviour in the neighbourhood of multiple image curves is examined in each of the details from the statue image (Figs. 6.3, 6.4, and 6.5). In these circumstances, The Canny operator either leaves large gaps (Figs. 6.3b and 6.5b), or simply infers a smooth, undisturbed local contour (Fig. 6.4b). This failure disrupts the ability to reconstruct the kind of information which gives a sense of three-dimensional structure, since creases and folds involve the intersection and joining of just such multiple image curves. In the worst case, nearby curves can interfere with the Canny operator's ability to extract much meaningful structure at all (Fig. 6.7b).

This leads us to the second criterion, the need to preserve line terminations and discontinuities. In our approach to early vision, we take curve discontinuities to be represented by multiple, spatially coincident edges [LZ88, Zuc86, ZDI89]. This holds for both "corners" and "T-junctions"—such discontinuities are inadequately captured by the Canny operator. Where there are clear discontinuities and junctions in the image curves, the Canny operator either leaves gaps or gives smooth output curves (see in particular the detail in Fig. 6.4b). The L/L operators represent such curve crossings and junctions by supporting multiple independent orientations in a local neighbourhood, just the representation we require. So not only do the L/L operators respond stably in the neighbourhood of multiple coincident curves, but they are also able to adequately represent this coincidence. Preceding attempts at edge operators have relied on the *a priori* assumption (usually implicit) that only one edge need be considered in each local neighbourhood, and thus that only one edge need be represented at each point in the output image. By rejecting that assumption and ensuring that the L/L operators perform stably in the neighbourhood of edge conjunctions, we provide a stable, complete representation of these fundamental image structures.

Recently, there has been some attempt to define "steerable filters" for edge detection [FA91, PM91, Per92], and to have them provide a representation for image curve discontinuities analogous to ours (i.e., as multiple orientations at the same position). However, the linear spatial support of these operators again causes problems, in this case a "smearing" or blurring of the corner energy over a neighbourhood. An additional search process is therefore introduced to find the locations and directions of maximal response [Per92], analogous to what we called "lateral maxima selection" in earlier implementations of our system [ZDDI88]. While such search processes provide some of the necessary non-linear behaviour, they introduce additional interpretative difficulties that do not arise with the L/L decomposition. Search also further complicates parallel implementations by introducing sequential bottlenecks. Finally, the standard steerable filters still exhibit mislocalization of line endings (which led in [Per92] to the introduction of end-line detectors). The steerable filters approach is useful, however, for reasons of computational efficiency, and we suggest that they may be used as a basis set for the linear components of our L/L operators.

Finally, the third criterion, the potential confusion between lines and edges, is seen to be addressed by the L/L operator approach. This problem is acute with the Canny operator, and is deliberately confounded by the "edge energy" methods [MB88, PM91], thus necessitating a second stage of analysis *that refers back to absolute image intensities* to fully describe the local structure of the image curve. The fingerprint (Fig. 6.7) and the composite image of the statue (Fig 6.6) show the utility and richness of a representation which separates edge and line information. The fingerprint is clearly more appropriately and parsimoniously represented by the line image, while the highlights on the statue (adjacent to some of the edges) are revealed by the line image. It has been argued that most line-like structures can be revealed by looking for locally parallel edge responses, but clearly not all (e.g. the many highlights on the statue's surface). We submit that parsimonious representations will combine features from both edge and line images and interpret them as appropriate.

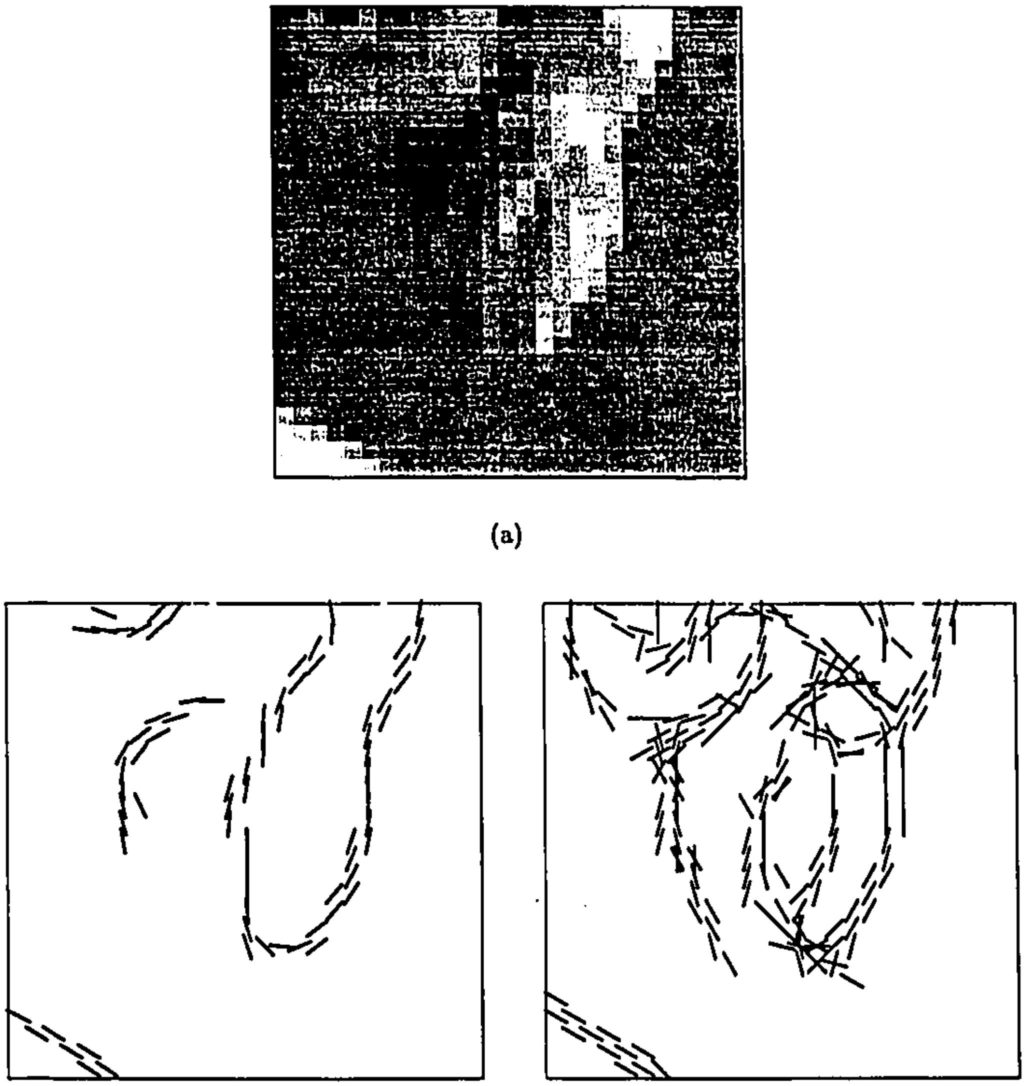
It is also important to note that computing Canny's algorithm on a parallel architecture requires a number of iterations of dilation in order to implement the 'hysteretic threshold'. Consider a planar parallel computer with one processor allocated to each image pixel. The Canny algorithm's time complexity on a such an architecture is $O(n)$, where n is the maximum length of a curve. Worst case, this is proportional to the number of pixels in the image, thus representing a significant bottleneck in an



(a)

(b) (c)

Figure 6.3: Detail of statue (a) from lower left near jaw and neck, and edge maps computed by: (b) Canny's algorithm, and (c) L/L operators (both algorithms are run at the same scale). Note that Canny's algorithm does not connect the two edges which join at the T-junction. The L/L operator responses represent the discontinuity by supporting two independent orientations in the same local neighbourhood.

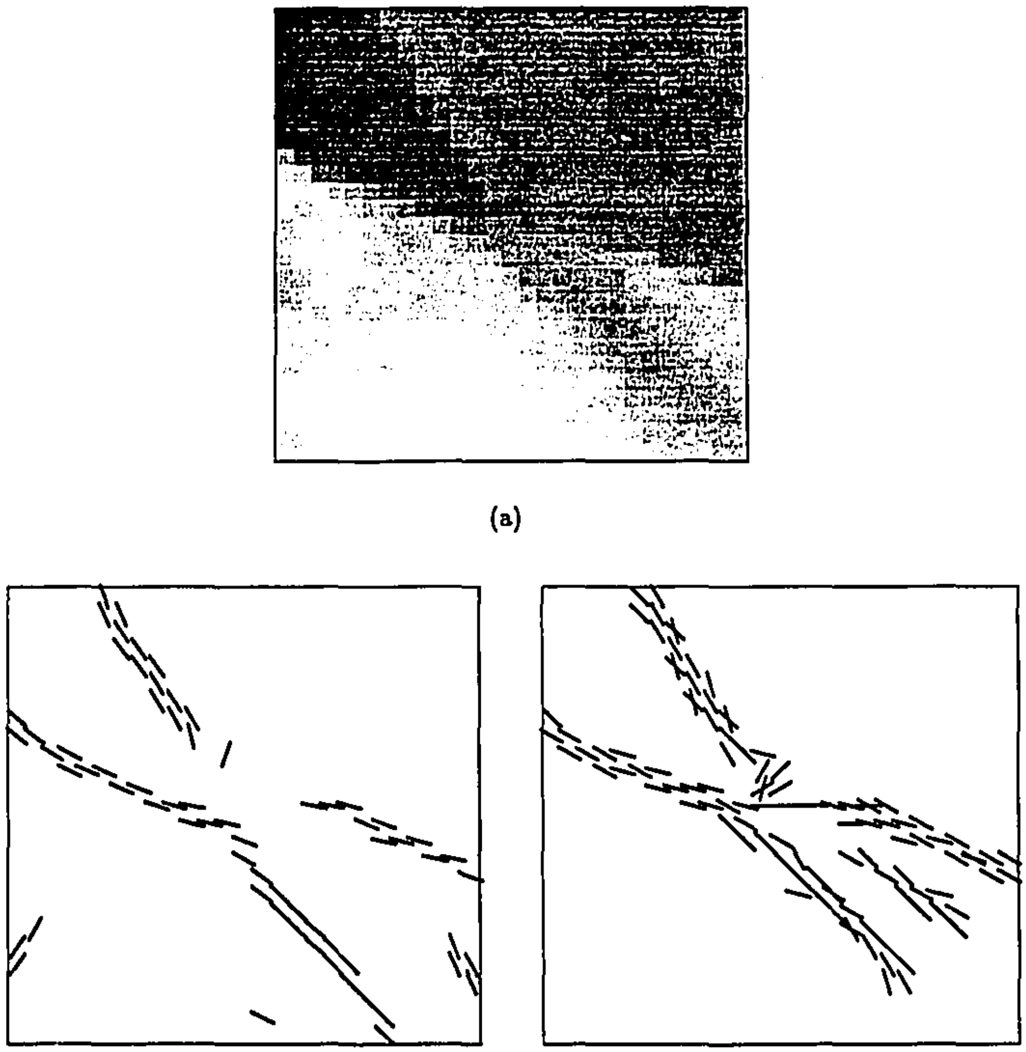


(a)

(b)

(c)

Figure 6.4: Detail of statue (a) from upper right, and edge maps computed by: (b) Canny's algorithm, and (c) L/L operators (both algorithms are run at the same scale). The Canny operator misses much of the rich structure in this small region as a result of interference between the nearby edges and the choice of high threshold. A lower threshold would have the effect of exposing more structure, but then the noisy responses seen in Fig. 6.1a would also be expanded. The L/L operator exposes this structure and also represents the discontinuities and bifurcations in the underlying edge structure.



(a) (b) (c)

Figure 6.5: Detail of statue (a) from lower right near shoulder, and edge maps computed by: (b) Canny's algorithm, and (c) L/L operators (both algorithms are run at the same scale). Again the Canny operator does not represent the conjunction of edges in this neighbourhood, while the L/L operators show the edge bifurcation clearly.

Figure 6.6: The statue as represented by the three categories of L/L operators. The black lines show the edge responses while the white and grey lines show the bright and dark lines respectively. Note that some features, such as the bottom of the palm of the hand, are only clearly represented by the line images.

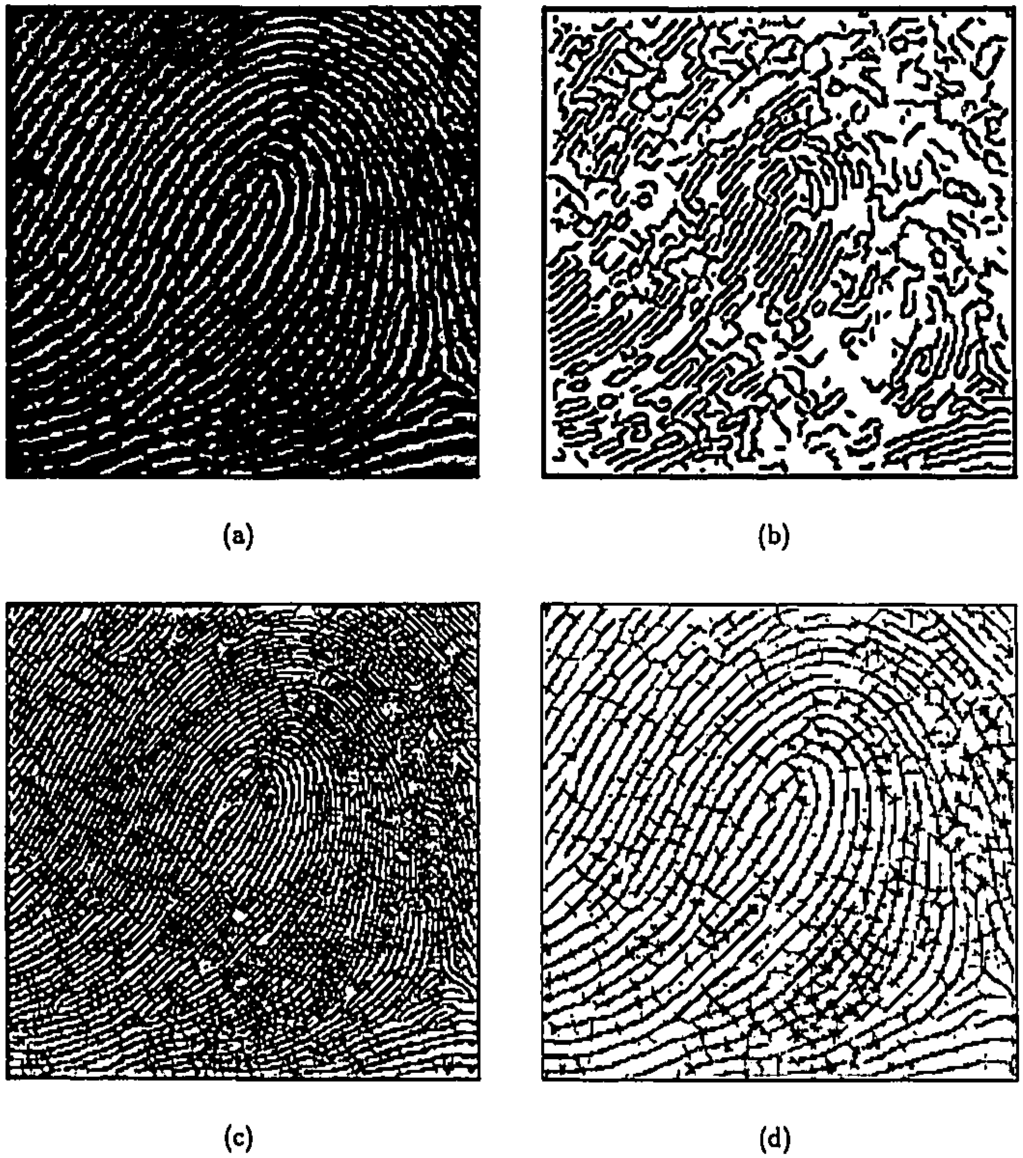


Figure 6.7: Fingerprint image (a), and edge maps computed by (b) Canny's algorithm, and (c) L/L edge operators. The most appropriate representation (d) is the L/L positive contrast line operator. The complexity of display and the proximity between nearby image features are the most significant contributors to the breakdown of Canny's algorithm in this case. These problems are dealt with in the L/L operators by the explicit testing of local consistency before combining component inputs. This serves to isolate features even when other nearby structures exist within the spatial support of the operator.

otherwise parallelizable algorithm. In contrast, the L/L operator implementation has $O(1)$ time complexity for such an architecture.

Part III

Intrinsic Geometry

Chapter 7 Relaxation Labelling

An image curve has both structure with respect to the image (i.e. it is an edge), and structure intrinsic to its geometry (i.e. continuous curvature). No matter how well local image operators work, they are restricted to direct analyses of the image and thus are subject to disturbances. Part II described methods for limiting both false positive and false negative responses, but the operators obtained are not entirely free from the effects of noise and confounding stimuli. Moreover, there is no guarantee that their responses represent connected thick traces of continuous curves. If we hope to produce descriptions of image curves which are verifiably thick traces we need to reason about the continuity and consistency of the image curves considered as geometric objects.

So our immediate problem is to select from amongst the responses of these initial operators those which are most likely to form thick traces according to some local geometry. To reason with thick traces directly, however, we are faced with a potentially enormous explosion in the amount of data to be processed. For example, if we represent 16 possible orientations of lines for each point in our image, then a sampling of the total space of a 256×256 pixel image will contain over 1 million sites. For a system to manage this data quickly and verifiably we will have to exploit as much parallelism as possible. One well-defined system for solving exactly this kind of problem (extracting consistent structures from discrete representations) is *relaxation labelling.*

Designing a relaxation network for a specific problem can be difficult, however, because of the problem of relating the pragmatic goals of a particular situation to the fixed summative networks of relaxation labelling.

The contribution of this chapter is to define explicit criteria by which a set of operator responses which constitute a noisy description of a thick trace may be transformed into a verifiable thick trace. This will involve a formal description of the kinds of geometric models which form these thick traces and then a translation of these models into support networks for relaxation labelling. The resulting labelling assignments are verifiably accurate approximations to the thick traces allowed by the model.

We found that in order to design initial operators which respected the structure of images, we were forced to reason using an explicit hypothesis-testing framework. In this chapter, we conclude that in order to reason about the intrinsic properties of the curves themselves we must do the same. Moreover, we find that the L/L algebra of Chap. 4 is a perfect match for the kinds of reasoning required in these new situations. The resulting networks incorporate L/L combinators into the classical relaxation framework, and are reliable, stable and converge very quickly (3 to 4 iterations). We thus believe that the methodology described below is of general use for early vision.

7.1 Definitions

Relaxation labelling [HZ83] is a computational method for finding consistent structures within a network of hypotheses. Closely related to popular neural network methods [MZ92], it involves representing an assignment problem as a set of labelled nodes I each with a set of associated labels Λ_i. Each $\lambda \in \Lambda_i$ is interpreted as a possible value to be assigned to node i. The scalar $p_i(\lambda), i \in I, \lambda \in \Lambda_i$ is the confidence that λ should be assigned to node i. There are two restrictions on these confidences

$$0 \leq p_i(\lambda) \leq 1 \quad \text{and} \quad \sum_{\Lambda_i} p_i(\lambda) = 1.$$

If these restrictions hold for all $i \in I$ and $\lambda \in \Lambda_i$, then the triple of (I, Λ, p) is called a *labelling assignment*. The space of such labelling assignments is K. A simple interpretation of such an assignment is that $p_i(\lambda)$ is the *confidence* that the label λ should be assigned to node i.

The goal of relaxation labelling is to solve an assignment problem: to choose a labelling assignment which maximizes some measure of consistency. In order to do this, we augment the labelling with a matrix $r_{ij}(\lambda, \lambda')$, which is a measure of the compatibility between label λ at i and label λ' at node j.

Definition 7.1 The *support* for a label λ at i is

$$s_i(\lambda; \bar{p}) = \sum_{j \in I} \sum_{\lambda' \in \Lambda_j} r_{ij}(\lambda, \lambda')\, p_j(\lambda').$$

An *unambiguous labelling* is a labelling assignment such that

$$\forall i \in I,\ \lambda \in \Lambda_i\colon\ p_i(\lambda) \in \{\, 0, 1 \,\}$$

which defines a mapping $i \to \lambda$ if and only if $p_i(\lambda) = 1$. We say that λ is assigned to the node i. A *consistent labelling* is a labelling assignment which fulfills the condition

$$\forall i \in I, \bar{v} \in K\colon \sum_{\lambda} p_i(\lambda) s_i(\lambda; \bar{p}) \geq \sum_{\lambda} v_i(\lambda) s_i(\lambda; \bar{p}),$$

where K is the space of all labelling assignments.

Formally, relaxation labelling solves the problem of finding a consistent labelling given an initial description (I, Λ, p) and the compatibility matrix r_{ij}. If *average local consistency* (for symmetric compatibilities) is given by

$$A(p) = \sum_{i \in I} \sum_{\lambda \in \Lambda_i} p_i(\lambda) s_i(\lambda; \bar{p}),$$

then the algorithm in Fig. 7.1 constitutes a gradient ascent on average local consistency which terminates at a consistent labelling [HZ83].

For the early vision problems we consider here, the general case can be simplified considerably. Remember that we are attempting to extract cross-sections through image bundles. We assume that one node is associated with every discrete point in a image bundle and that the labels at each point are $\Lambda_i = \{\,\textsc{True}, \textsc{False}\,\}$. Thus, a node i represents the hypothesis that there is a cross-section which intersects the Voronoi cell E_i. We refer to this special case as *two label relaxation labelling* [PZ85].

The two-label representation, and update and projection steps are considerably

1. Compute an initial estimate of $p = \{ p_i(\lambda) \}$ which constitutes a labelling assignment. Call this p^0.
2. Repeat starting with $n = 0$ until p^n is consistent:
   (a) Repeat for all $i \in I$:
      i. Compute $p_i^* = p_i^n + \delta s_i$.
      ii. Project p_i^* onto a valid labelling assignment. This new assignment is p_i^{n+1}.
   (b) Set $n = n + 1$.
3. Generate the mapping $i \rightarrow \lambda$.

Figure 7.1: The Hummel-Zucker algorithm for relaxation labelling [HZ83].

simplified. In terms of the representation, we need only evaluate and store $p_i(\text{TRUE})$ since $p_i(\text{FALSE}) = 1 - p_i(\text{TRUE})$. In addition, if we impose the design condition that $s_i(\text{FALSE}) = -s_i(\text{TRUE})$, then any evidence for a hypothesis is naturally evidence opposing its converse. This amounts to a condition on the structure of the compatibilities r_{ij}, and is easily realizable in practice, with

$$r_{ij}(\text{TRUE}, \text{TRUE}) = -r_{ij}(\text{TRUE}, \text{FALSE}) = -r_{ij}(\text{FALSE}, \text{TRUE}) = r_{ij}(\text{FALSE}, \text{FALSE}).$$

With these restrictions, we can simplify the notation, using p_i to refer to $p_i(\text{TRUE})$ and s_i to refer to $s_i(\text{TRUE})$. The support then simplifies to

$$s_i = \sum_{j \in I} r_{ij}\, p_j, \tag{7.1}$$

and the update rule becomes

$$p_i^{n+1} = [p_i^n + \delta s_i]_0^1,$$

where $[x]_0^1$ is x truncated to the interval $[0, 1]$.

Gradient ascent procedures are necessarily vulnerable to the presence of local minima. One of the requirements of a problem definition using a relaxation labelling paradigm is thus to ensure that the combination of the initial estimate p^0 and the compatibility matrix r_{ij} produces only *meaningful* local minima. One way of thinking of this is to require that both the initial estimate and the update function preserve whichever features are deemed to be essential for a viable solution (i.e. the support s_i is only positive when these features can be verified locally). These requirements must be treated as design preconditions on the calculation of the initial estimates and the derivation of the compatibilities.

7.2 Geometric Compatibility

As we have suggested, the problem of extracting geometrically "consistent" structure from an image depends on the definition of consistency. If the *representation* is in terms of a sampled fibre bundle, then the *description* of a consistent structure will be the thick trace of some cross-section of the bundle. As we showed above, the selection of the points in such a cross-section can be formulated as a two-label assignment problem. However, selecting these points will depend critically on the local geometry of these cross-sections; that is, on the compatibility structure chosen.

In this section we develop a framework for relating an image geometry to the geometry of cross-sections in a fibre bundle. The resulting *models* match the structure of the fibres to a description of the local image geometry, and constrain the cross-sections. With this framework in place, we can determine if a pair of points in the total space are "compatible" with the desired image geometry and, if they are not, how "incompatible" they are. By then extending this investigation to a sampling of the total space we will lay the groundwork for the design of a relaxation labelling network whose fixed points are thick traces model cross-sections.

To clarify the reasoning and results, the derivations will be accompanied by an elaborated example. For simplicity, we will consider the problem of finding continuous straight lines in images.

7.2.1 Continuous Spaces

We begin by introducing geometric models on fibre bundles. A geometric model is a complete description of a geometric form on the base space of an image bundle.

Definition 7.2 A *model* $\mathcal{M}$ on the bundle ξ is a set of differentiable cross-sections $M : B \rightarrow E$ of ξ which cover the total space E. If $\mathcal{M}$ is a partition of E (i.e. each point $p \in E(\xi)$ has a unique cross-section $M \in \mathcal{M}$ such that $p \in M$) then we say that the model $\mathcal{M}$ is *minimal*, and we can refer to the cross-section selected by p as M_p.

For example, the set of all straight lines in the plane $\mathbb{R}^2$ and their orientations θ is a model on the bundle $\xi : \theta \rightarrow \mathbb{R}^2$. Note that a minimal model can be expressed as a mapping $\mathcal{M} : E \times B \rightarrow E$ where $\mathcal{M}(p, q) = M_p(q)$. We restrict these mappings $\mathcal{M}$ to be differentiable.

To use this abstract construction for the representation of geometric structure, we assume that each cross-section M is the instantiation of some geometric primitive. We refer to this kind of model as a *geometric model*. If this model is minimal, then each point $p \in E$ is a complete description of the local geometry.

Whether a model is minimal or not, it can be used to group together all of the points which share membership in one of its cross-sections. We say that such points are compatible with each other.

Definition 7.3 For a model $\mathcal{M}$, we define a *compatibility* relation $C_{\mathcal{M}}(p, q)$ between points $p, q \in E(\xi)$ where

$$C_{\mathcal{M}}(p,q) \;\Leftrightarrow\; \exists M \in \mathcal{M}\colon\ p \in M \text{ and } q \in M.$$

For minimal models this is clearly an equivalence relation. Moreover, for minimal models this definition simplifies to the requirement that $q \in M_p$, but since q uniquely selects the cross-section M_q

$$C_{\mathcal{M}}(p,q) \;\Leftrightarrow\; q \in M_p \text{ or } p \in M_q.$$

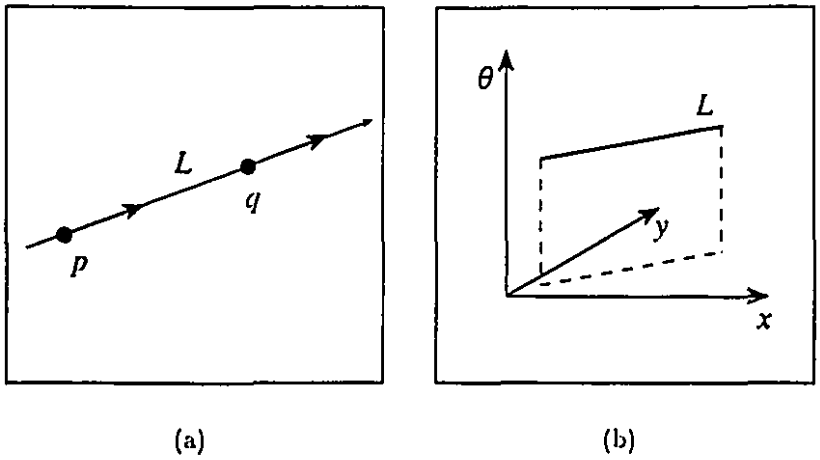


Figure 7.2: A diagram of a model of straight lines in planar images. In (a) is shown the geometric description of lines in images. Compatible local models (e.g. at points p and q) generate the same line. The model determines a family of cross-sections in the natural product bundle $\xi = X \times \Theta$ (b) which themselves form straight lines. The geometry of lines in the plane thus induces a geometry of cross-sections in the total space.

It is clear from the above results that minimal models give much more structure to the compatibility relation than general models. It is for that reason that we will henceforth restrict our analysis to minimal models. Much of the work below will apply equally well to general models, but attempting to delineate the differences would likely obscure some of the results.

Example 7.1 We know that a straight line in a plane is completely described by a position and a direction. Thus, a model $\mathcal{L}$ of straight lines in the plane image $I : X \to Y$ where $X \subset \mathbb{R}^2$ is defined on the bundle $\xi_{\mathcal{L}} = (X \times \Theta, \pi, X)$, where $\Theta = [0, \pi)$, by the cross-sections:

$$L(x, \theta)(s) = (x + s(\cos\theta, \sin\theta), \theta),$$

for $x \in X$ and $\theta \in \Theta$. We will refer to the line generated by $p = (x, \theta)$ as

$$L(x, \theta) = \{ L(x, \theta)(s) \mid (x, \theta) \in X \times \Theta \text{ and } s \in \mathbb{R} \}.$$

It is important to note at this point that topologically the total space $E_{\mathcal{L}}$ is cylindrical, since the orientation θ is the same as $\theta + n\pi$ for integral n.

Proposition 7.4 *The model $\mathcal{L}$ is minimal.*

The fact that this is a minimal model is clear from a purely geometric point of view, since a point and an orientation uniquely determine a straight line. It is still useful to explicitly prove minimality in terms of the model. We must show that all $L(q)$ intersecting the point $p = (x, \theta)$ are identical. For any $q' \in L(q)$, we have $q' = L(q)(s') = L(p)(s + s')$. Thus $q' \in L(q) \Rightarrow q' \in L(p)$ and therefore $L(q) \subset L(p)$. But this is also true for p and q reversed, so $L(p) \subset L(q)$ and thus $L(p) = L(q)$. ∎

The straight line model illustrates a method for defining models by parameterization.

Definition 7.5 A *parameterized model* is a model $\mathcal{M}$ on ξ described by a family of cross-sections

$$\mathcal{M} = \{\, M(p) : S \to E \mid p \in E \,\}.$$

where S, the *parameterization*, is a neighbourhood of the origin in $\mathbb{R}^n$ and $M(p)(0) = p$.

Observe that since all M are cross-sections of ξ, $n \leq \dim(B_{\mathcal{M}})$.

Corollary 7.6 *A parameterized model $\mathcal{M}$ is minimal if and only if for all $q \in M(p)$, $M(q)$ is a reparameterization of $M(p)$.*

We have so far ignored the fact that locality is often an issue when applying models to real situations. For example if the model is derived from a truncated Taylor expansion of a local neighbourhood, then we can only be confident that it is accurate within some bounded neighbourhood. The extent of this neighbourhood is determined by the distance travelled along $\pi(M) = \{\, \pi(e) \mid e \in M \,\}$, the projection of M into the base space. To see why this is so, it is sufficient to realize that since M is a cross-section, motion on M is completely proscribed by motion in the base space B (i.e. there is a one-to-one mapping from B to M).

As a first step in understanding local compatibility then, we need to define a means of relating both compatible and incompatible pairs of points in E to the model $\mathcal{M}$. Initially, consider only compatible pairs.

Definition 7.7 The *transport distance* $d^t_{\mathcal{M}}(p,q)$ between two compatible points $p,q \in M$ is the length of the shortest path between $\pi(p)$ and $\pi(q)$ in $\pi(M)$.

Since M is connected, such a path does exist, and with the Riemannian metric on $\pi(M)$ inherited from B, the length of such a path is well-defined. In fact, since $\pi(M)$ is a submanifold of B, we can deduce that the minimal path from p to q is a geodesic on $\pi(M)$ (thus justifying the use of the term "transport"). Given this, $d^t(p,q)$ is clearly an induced metric on $\pi(M)$ and thus also on M.

To extend this to arbitrary pairs of points, we refer to the tubular neighbourhood of M. This is a generalization of the perpendicular projection operator over M in E which is a basic building block of modern differential geometry. We refer to the tubular neighbourhood since it may help readers familiar with fibre bundle manipulations to understand subsequent derivations. Other readers need only note that the tubular map π_M is simply the perpendicular projection from E onto M.

Treating the model cross-sections M as differentiable submanifolds of E, we can identify the tangent bundles of the total space $\mathcal{T}(E)$ and the submanifold $\mathcal{T}(M)$.

Definition 7.8 [Lan85, Kos93] The *normal bundle* νM of the submanifold $M \subset E$ is the quotient bundle $\mathcal{T}_M(E)/\mathcal{T}(M)$. There exists a bundle $\pi_M : T_M \rightarrow M$, unique under isotropies, equivalent to the normal bundle νM such that T_M is a neighbourhood of M in E and the zero section of π_M is M. The neighbourhood T_M is the *tubular neighbourhood* of M and π_M the *tubular map*. In $\mathbb{R}^n$ the tubular map is a perpendicular projection from T_M onto M. If $T_M = E$, then the tubular neighbourhood is said to be *total*. Finally, we note that the subspace of E normal to the manifold M at $e \in M$ is just the inverse of the tubular map $\pi_M^{-1}(e)$.

Now consider two points p and q in the total space associated with the model $\mathcal{M}$. If p and q are compatible under $\mathcal{M}$, then there exists some $M \in \mathcal{M}$ such that $p,q \in M$. Otherwise, if p and q are in the tubular neighbourhood of some cross-section

M^*, then $\pi_{M^*}(p)$ and $\pi_{M^*}(q)$ are on M^* and thus compatible under $\mathcal{M}$.

Definition 7.9 We define the *incompatibility* of p and q with the model $\mathcal{M}$ as

$$d^c(p,q) = \inf_{M^* \in \mathcal{M}} d(p,p^*) + d(q,q^*). \tag{7.2}$$

where $p^* = \pi_{M^*}(p)$ and $q^* = \pi_{M^*}(q)$.

In essence, $d^c(p,q)$ is a measure of the minimal perturbation of p and q which will produce a compatible pair. We refer to the triple (M^*, p^*, q^*) as the *minimal projection* of p and q onto $\mathcal{M}$.

The minimal projection is used to extend the definition of transport distance to arbitrary points.

Definition 7.10 The *transport distance* between points $p, q \in E$ is

$$d^t_{\mathcal{M}}(p,q) = d^t(p^*, q^*)$$

where (M^*, p^*, q^*) is the minimal projection of p and q onto $\mathcal{M}$.

These two measures of compatibility between p and q will form the basis for all of our subsequent calculations with compatibilities.

Example 7.2 *(continued)* Consider two points $p = (x_p, y_p, \theta_p)$ and $q = (x_q, y_q, \theta_q)$ in $E(\mathcal{L}) = X \times \Theta$ and a cross-section $M^*(s) = \ell + s(\cos\theta^*, \sin\theta^*, 0)$ where $\ell = (x^*, y^*, \theta^*)$. Now, let $p^* = M^*(s_i)$ and $q^* = M^*(s_j)$, and since $(p^* - \ell) \cdot (p^* - p) = 0$ and $(q^* - \ell) \cdot (q^* - q) = 0$ we can express p^* and q^* as functions of ℓ. Therefore if $p^* = M^*(s_p)$ and $q^* = M^*(s_q)$

$$\begin{aligned} d^t(p,q) &= |s_p - s_q|, \\ d^c(p,q) &= \min_{\ell \in E} d(p^*(\ell), p) + d(q^*(\ell), q). \end{aligned}$$

The projections p^* and q^* are shown in Fig. 7.3a.

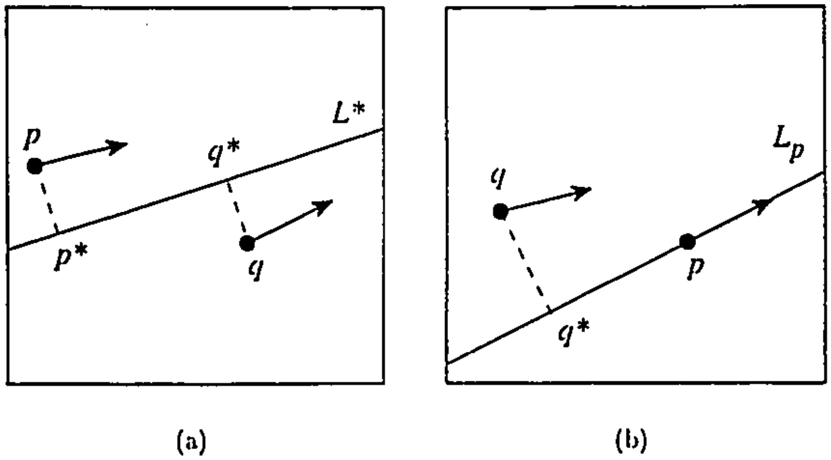


Figure 7.3: The minimal projection of two points p and q onto the straight line model $\mathcal{L}$. The sum of the distances $d(p, p^*)$ and $d(q, q^*)$ is a measure of the incompatibility $d^c(p, q)$ between the lines p and q, while the distance $d(\pi(p^*), \pi(q^*))$ is the transport distance $d^t(p, q)$. When p and q are compatible, then $d^c(p, q) = 0$. Of the two diagrams, (a) shows the symmetric minimal projection, while (b) show the projection associated with the asymmetric compatibility of Def. 7.15.

7.2.2 Discrete Spaces

Before we can understand the impact of these measures on the derivation of relaxation labelling compatibilities we need to investigate their translation into sampled spaces.

We start by defining a relation on discrete points in the total space of a geometric model. This relation is a discretization of the compatibility defined in Def. 7.3.

Definition 7.11 Given a model $\mathcal{M}$ on ξ and a discretization $\hat{E}(\xi)$ of ξ, we define a *discrete compatibility* relation $\hat{C}_{\mathcal{M}}$ between points $e_i, e_j \in \hat{E}(\xi)$ where

$$\hat{C}_{\mathcal{M}}(e_i, e_j) \;\Leftrightarrow\; \exists p \in E_i,\ q \in E_j,\ M \in \mathcal{M}\colon\ p \in M \text{ and } q \in M.$$

That is, there is some cross-section in the model $\mathcal{M}$ whose thick trace includes both points e_i and e_j. Note that, in contrast with the continuous definition, while this relation is commutative, it is in general *not* transitive, even if the model is minimal.

Theorem 7.12 *The discrete compatibility relation* $\hat{C}_{\mathcal{M}}(e_i, e_j)$ *for* $e_i, e_j \in \hat{E}$ *is a valid* *discretization of* $\mathrm{C}_{\mathcal{M}}(e_i, e_j)$.

For $\hat{E} = E$, the Voronoi cells E_i and E_j contract to the singletons $\{e_i\}$ and $\{e_j\}$, and the equation above becomes

$$\hat{\mathrm{C}}_{\mathcal{M}}(e_i, e_j) \Leftrightarrow \exists M \in \mathcal{M}\colon\ e_i \in M \text{ and } e_j \in M,$$

exactly the continuous definition of Def. 7.3. ∎

As we noted above, this relation can be expressed in terms of thick traces. For a minimal model, the thick trace of a cross-section provides a convenient verification of compatibility since

Corollary 7.13 *If* $e_j \in \textsc{Thick}(M_{e_i})$ *or* $e_i \in \textsc{Thick}(M_{e_j})$ *then* e_i *and* e_j *are compatible under the model* $\mathcal{M}$.

Furthermore, comparing the two definitions (2.10 and 7.11) immediately reveals that

Corollary 7.14 *For a given point* e_j, *the thick trace of the cross-section* M_{e_j} *is a subset of the set of points compatible with* e_j: $\left\{ e_i \in \hat{E} \mid \hat{\mathrm{C}}_{\mathcal{M}}(e_i, e_j) \right\}$.

These observations lead immediately to a second discretization of the compatibility relation.

Definition 7.15 Given a minimal model $\mathcal{M}$ on ξ and a discretization $\hat{E}(\xi)$ of ξ, we define the *asymmetric compatibility* relation $\hat{\mathrm{C}}'_{\mathcal{M}}(e_i, e_j)$ between points $e_i, e_j \in \hat{E}(\xi)$ where

$$\hat{\mathrm{C}}'_{\mathcal{M}}(e_i, e_j) \Leftrightarrow E_i \cap M_{e_j} \neq \emptyset.$$

Again, the reduction to continuous compatibility is obvious.

Corollary 7.16 *The discrete compatibility relation* $\hat{\mathrm{C}}'_{\mathcal{M}}(e_i, e_j)$ *for* $e_i, e_j \in \hat{E}$ *is a valid discretization of* $\mathrm{C}_{\mathcal{M}}(e_i, e_j)$.

With this relation, the minimal projection becomes (M_{e_j}, e_i^*, e_j) where

$$e_i^* = \pi_{M_{e_j}}(e_i),$$

and the compatibility metric, transport distance and incompatibility are derived from this new projection. However, this relation is neither commutative nor transitive (hence *asymmetric*). We will see, however, that it provides the basis for mapping geometric models onto relaxation labelling supports.

Example 7.3 *(continued)* Consider two points $e_i = (x_i, y_i, \theta_i)$ and $e_j = (x_j, y_j, \theta_j)$ in $E(\mathcal{L}) = X \times \Theta$. Let $e_i^* = e_j + s(\cos\theta_j, \sin\theta_j, 0)$. The perpendicular projection of e_i onto L_{e_j} is then found by solving $(e_i^* - e_j) \cdot (e_i^* - e_i) = 0$ for s, which gives

$$s = (x_i - x_j)\cos\theta_j + (y_i - y_j)\sin\theta_j.$$

The projection to e_i^* is shown in Fig. 7.3b.

With this projection, we can explicitly calculate the transport distance and incompatibility between any two line elements e_i and e_j.

$$d^t_{\mathcal{L}}(e_i, e_j) = |s|. \tag{7.3}$$

$$d^c_{\mathcal{L}}(e_i, e_j) = |e_i^* - e_i|. \tag{7.4}$$

These definitions provide a basis for understanding compatibility relationships in discretizations of geometric models, but they leave a number of computational questions open—most significantly the question of how to determine whether or not a particular cross-section M intersects the Voronoi cell E_i. Since the E_i is a convex polyhedron, it can be defined by a set of linear inequalities of the form $(e - e_i) \cdot n \leq r$, one for each face. The problem of testing intersection between the cross-section M and the cell E_i is thus formalized as

$$M \cap E_i \neq \emptyset \;\Leftrightarrow\; \forall f \in F_i : \exists e \in M : (e - e_i) \cdot n_f \leq r_f, \tag{7.5}$$

where F is the set of faces of E_i. While this is not an expensive computation for a single cross-section M and Voronoi cell E_i, when the tests number in the millions, it may be prohibitive.

As an alternative, we suggest an approximation which may be significantly less expensive to compute. If E_i is nearly spherical (i.e. we use a close-packed tesselation or E is of high dimensionality), then it is clear that

$$|\pi_M(e_i) - e_i| \leq \rho(E_i)$$

is an approximate test for intersection between M and E_i. The only circumstance in which this is true but $M \cap E_i = \emptyset$ is when $\pi_M(e_i) \in (b_{\rho(E_i)}(e_i) \cap \bar{E}_i)$. As long as E_i is nearly spherical $\text{Vol}(E_i) > \text{Vol}(b_{\rho(E_i)}(e_i))/2$, thus for the *majority* of points e_i for which this is true $M \cap E_i \neq \emptyset$.

The above approximation depends on the assumption that the samplings are *unbiased*, that is that the E_i are all very nearly spherical and of the same radius (i.e. $\forall i\colon \rho(E_i) \approx \rho(\hat{E})$). The usual regular samplings (e.g. square and hexagonal grids) are unbiased. In general any sampling for which the sample points are the centers of a spherical close packing [Con93] is also. We will assume below that all samplings are unbiased.

7.3 Relaxation Labelling Support

We now have the machinery to answer the key question: what is the relationship between compatibility as defined in §7.2.2, and the matrix r_{ij} in relaxation labelling?

We can set up the labelling assignment problem by assuming that each point $e_i \in \hat{E}(\xi)$ has an associated node i in the labelling. Furthermore associate with node i a confidence p_i that for some point $e \in E_i$, the projection $\pi(e)$ in the image has the local geometry $M_e \in \mathcal{M}$. Thus $p_i = 1$ implies that e_i is in the thick trace of some cross-section M in the model $\mathcal{M}$. Since these cross-sections are continuous, an unambiguous labelling for such a geometric problem consists of a set of connected components of the discrete total space each of which is associated with a single model cross-section. We will call such a connected component a ***compatible subset*** of the labelling.

The design problem is thus to develop a support function so that the fixed points of the relaxation are labelling assignments which consist only of non-degenerate compatible subsets. Each assigned node will then represent confirmation of the existence of a patch of the selected geometry in the associated image.

There are three goals which constrain the designing of labelling supports for such a system:

- The labelling assignment in which all confidences are zero except those in the thick trace of a model cross-section must be a fixed point of the relaxation. The closest fixed point to such an assignment should be the unambiguous assignment in which all of these labels have confidence 1. Therefore, nodes in such a thick trace should receive positive support.

- The thick traces extracted should be either disjoint or only connected at single points. We would normally expect them to be disjoint, but allowing point connectivity, we shall see, will allow for the possibility of representing *bifurcations* in certain geometric objects. Overall, this requirement for disjointness can be seen as implying that there be an empty region surrounding each thick trace in which all nodes have zero confidence. In terms of support then, nodes which are near but not *on* a thick trace of assigned labels should receive negative support.

- Isolated labels should receive non-positive support.

Ideally, *support for a label should be positive if and only if the label represents a point which is on a model cross-section defined by some set of neighbouring labels.* If these goals are met, then the effect of relaxation on an initial estimate of label confidences should be the selection of mutually supporting collections of labels which represent continuous patches of model cross-sections. All other labels should be suppressed and eliminated. A diagram of this plan is provided in Fig. 7.4—it will be useful to keep it in mind as we develop the solution.

Keeping these goals in mind, we seek to define a support function s_i such that the thick traces of model curves are fixed points of the relaxation. From the point of view of a single node, the support for a point e_i should be positive only if it can be verified to be in such a thick trace. One method of calculating support would thus

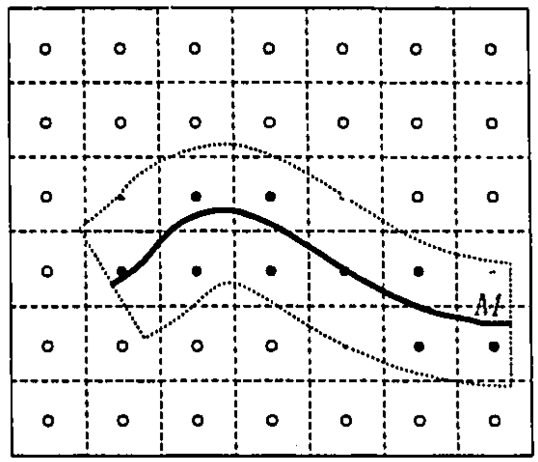


Figure 7.4: The fixed point of a geometric relaxation problem is shown. The cross-section M is shown in the sampled total space $\hat{E}$ with the points in its thick trace $e_i \in \text{THICK}(M)$ in solid black. If these points are to be a fixed-point of relaxation labelling, then a labelling in which only the black points have $p_i > 0$ must produce a support function in which $s_i > 0$ for only those points. All points $e_j \notin \text{THICK}(M)$ (open circles) receive non-positive support. Instead we design a support function by reconstructing the cross-section M underlying $\text{THICK}(M)$. We then define a new trace $\text{THICK}_\rho(M)$ (including the gray points as well) such that $d(e_i, M) < \rho$ for some $\rho \geq \rho(\hat{E})$. The support function is then positive only if $e_i \in \text{THICK}_\rho(M)$. In this case, the trace $\text{THICK}_\rho(M)$ is the fixed point of the relaxation.

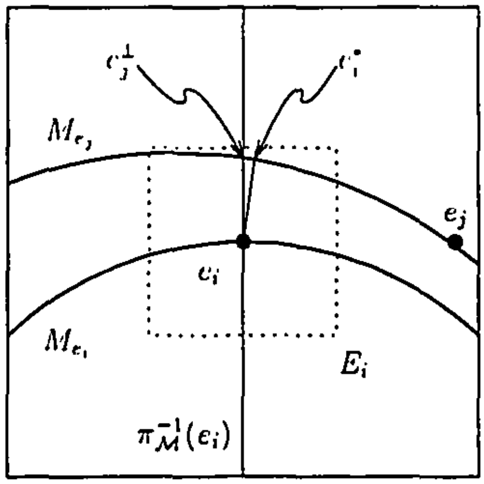


Figure 7.5: The model cross-sections in the neighbourhood of a point e_i. Assuming that e_i and e_j are compatible, then the cross-section M_{e_j} intersects E_i and $e_i^* \in E_i$. Show are $c_i^\perp = M_{e_j} \cap \pi_{\mathcal{M}}^{-1}(e_i)$ and $e_i^* = \pi_{M_{e_j}}(e_i)$. Since $\mathcal{M}$ is smooth, M_{e_j} and M_{e_i} are approximately parallel locally and thus $d(e_i, c_j^\perp) \approx d(e_i, e_i^*)$.

be to construct a smooth cross-section $M \in \mathcal{M}$ from a labelling assignment and then update p_i based on whether or not $e_i \in \text{THICK}(M)$. Section 7.2.2 concluded with an examination of how to make this selection efficiently. Therefore, we seek a method of combining the labels in a thick trace to form a smooth underlying cross-section $M \in \mathcal{M}$.

Consider the neighbourhood of the point e_i and the associated cross-section M_{e_i}. If our reconstruction method is sound, then the smooth cross-section constructed from $\text{THICK}(M_{e_i})$ must pass through E_i. For the moment focus on the points $e_j \in \text{THICK}(M_{e_i})$. By Def. 7.11 we know that

$$e_j \in \text{THICK}(M_{e_i}) \;\Rightarrow\; \hat{\mathrm{C}}(e_i, e_j).$$

and thus, since $e_i^* = \pi_{M_{e_j}}(e_i)$ that

$$e_i^* \in b_{\rho(E_i)}(e_i).$$

Now, on the fibre $\pi_{M_{e_i}}^{-1}(e_i)$ of the tubular bundle of M_{e_i} at e_i (which we shorten to $\pi_{\mathcal{M}}^{-1}(e_i)$), we can construct the point of intersection

$$e_j^{\perp} = M_{e_j} \cap \pi_{\mathcal{M}}^{-1}(e_i).$$

Assuming that the minimal model $\mathcal{M}$ considered as a mapping $\mathcal{M} : E \times B \to E$ is smooth, then for $e \in E_i$,

$$d(e_i, e_j^{\perp}) \approx d(e_i, \pi_{M_{e_j}}(e_i))$$

(see Fig. 7.5). Thus, since $e_i^* = \pi_{M_{e_j}}^{-1}(e_i)$ we can conclude that for some radius $\rho \approx \rho(E_i)$

$$e_i \in \text{THICK}(M_{e_j}) \Rightarrow (\forall j : e_j^{\perp} \in b_\rho(e_i)).$$

Thus from the point of view of the fibre of the tubular bundle over e_i, reconstructing a smooth cross-section underlying the trace may be achieved by combining the points $e_j^{\perp}$ into a single point $e \in \pi_{\mathcal{M}}^{-1}(e_i)$. The point e_i is then in the thick trace of this cross-section if and only if $e \in E_i$.

A possible solution is suggested in [DZ90]. We restate their observation in terms of sampled bundles and cross-sections. The sampling of E implies that a point e_j in $\hat{E}$ represents an equivalence class of cross-sections which intersect E_j. If we represent this equivalence class by a Weiner measure over the cross-sections, then for any point $e_j' \in M_{e_j}$ the Weiner measure restricted to $\pi_{M_{e_j}}^{-1}(e_j')$ (the subspace normal to M_{e_j} at e_j') is approximately Gaussian. The Weiner measure arises because the class of continuous functions is equivalent to the sample functions of a Brownian motion [Doo84]. Because of this, the central limit theorem implies that the distribution of points $e_j^{\perp} = M_e \cap \pi_{M_{e_i}}^{-1}(e_j')$ for fixed e_j' and $e \in E_i$ is approximately Gaussian. This observation can be used to show that in the plane, the linear combination of a set of Gaussians around the sample points on the thick trace of a curve forms a potential field, the valleys of which are curves with the same thick trace!

We can apply similar reasoning to the more general problem of averaging values on the fibres of a tubular bundle. First consider the combination of discrete values in $\mathbb{R}^n$.

Theorem 7.17 *If $P : \mathbb{R}^n \rightarrow \mathbb{R}$ is a weighted sum of Gaussians of the form*

$$P(x) = \sum_i a_i \, \mathrm{G}_{\sigma_i}(|x - x_i|),$$

for some collection of $x_i \in \mathbb{R}^n, a_i > 0$, and $\sigma_i > 0$, then all local maxima of $P(x)$ are within the convex hull of $\{ x_i \}$. Moreover, there is at least one such local maximum.

Consider the face F of the convex hull of $\{ x_i \}$. Define the half-plane

$$H_F = \{ x \mid (x - x_F) \cdot \gamma \leq 0 \}$$

where $x_F \in F$ and γ is the inward facing unit normal to F. Taking the directional derivative of P with respect to γ gives

$$P_\gamma(x) = \sum_i a_i G^i_\gamma(x)$$

where

$$G^i(x) = \mathrm{G}_\sigma(|x - x_i|).$$

Since $G^i(x)$ is monotonically decreasing with increasing $|x - x_i|$, we see that

$$(x - x_i) \cdot \gamma > 0 \Leftrightarrow G^i_\gamma(x) > 0.$$

Therefore

$$\begin{aligned} x \notin H_F &\Rightarrow (x - x_i) \cdot \gamma > 0 \\ &\Rightarrow G^i_\gamma(x) > 0 \\ &\Rightarrow P_\gamma(x) > 0. \end{aligned}$$

Since local extrema coincide with a sign change in all directional derivatives, all local extrema of $P(x)$ must be in H_F. This conclusion holds for each face of the convex hull, therefore the local maxima in $f(x)$ must be in the intersection of all such half-planes—the convex hull. To see that such a local maximum exists we need only refer to the maximum principle of the heat equation ([PW84], p. 161) which guarantees that for finite time (i.e. finite σ) a global maximum (and thus a local maximum) always exists if the initial value is non-constant. ∎

Since $\pi_{\mathcal{M}}^{-1}(e_i)$ is isomorphic with $\mathbb{R}^n$ where $n = \dim(E) - \dim(M)$, we can construct a cross-section underlying the thick trace $\text{THICK}(M)$ of M by constructing the function

$$P(e) \;=\; \sum_j p_j \; \mathrm{G}_{\sigma_t}(d^t(e,e_j)) \; \mathrm{G}_{\sigma_c}(d^c(e,e_j)),$$

where $e_j \in \text{THICK}(M)$. Since the restriction of this function to $\pi_M^{-1}(e_i)$ is of the form described in Thm. 7.17 with

$$\begin{aligned} a_i &= p_j \; \mathrm{G}_{\sigma_t}(d^t(e,e_j)) \\ x_i &= e_j^{\perp} \end{aligned}$$

the local maxima of P on $\pi_{\mathcal{M}}^{-1}(e_i)$ are all within the convex hull of $e_j^{\perp}$. The distances $|e_j^{\perp} - e_i|$ are bounded above by some constant ρ (since $e_j^{\perp} \approx e_i^{*}$). Therefore, all local maxima of P on $\pi_{\mathcal{M}}^{-1}(e_i)$ are in the ball $b_\rho(e_i)$ which is approximately the Voronoi cell E_i. Without proving that there is a unique such maximum (largely irrelevant since the sampling will combine them), the discussion at the end of §7.2.2 suggests that we can define the ρ-trace

$$\text{THICK}_\rho(M) \;=\; \{\, e_i \mid d(\pi_M(e_i), e_i) \le \rho \,\}$$

which is an approximation to $\text{THICK}(M)$ such that

$$\rho \ge \rho(\hat{X}) \;\Rightarrow\; \text{THICK}(M) \subset \text{THICK}_\rho(M).$$

Moreover, $\text{THICK}_\rho(M)$ is clearly connected for smooth M if $\rho \ge \rho(\hat{X})$, thus the relationship between the path-connectedness of M and the connectedness of THICK(M) is preserved in $\text{THICK}_\rho(M)$. So, if we start with a labelling assignment of the form

$$p_i = \begin{cases} 1 & \text{if } e_i \in \text{THICK}_\rho(M); \\ 0 & \text{otherwise,} \end{cases}$$

and a support function s_i which is positive if and only if there is a local maximum of $P(e_i)$ in $b_\rho(e_i) \cap \pi^{-1}_{M_{e_i}}(e_i)$, then s_i is positive if and only if $e_i \in \text{THICK}_\rho(M)$.

Corollary 7.18 *If s_i is positive exactly when there is a local maximum of $P(e)$ in $b_\rho(e_i) \cap \pi^{-1}_{\mathcal{M}}(e_i)$ then the thick trace $\text{THICK}_\rho(M)$ for $M \in \mathcal{M}$ is a fixed point of relaxation labelling.*

With the Logical/Linear Algebra of Chap. 3, we have a direct means of testing this condition locally.

Observation 7.19 If $x \in X$ is a local maximum of $P : X \to \mathbb{R}$ in some neighbourhood $b_\rho(x)$, then x is a local maximum of P in every neighbourhood $b_\rho(x) \cap \alpha$ where $\alpha : \mathbb{R} \to X$ is a differentiable curve with tangent $\gamma = \alpha'(x)$.

In order to locate local maxima in P we need only identify regions within which the directional derivatives P_γ change sign and the second directional derivative $P_{\gamma\gamma} < 0$.

Definition 7.20 Let $\Gamma = (\gamma_1, \ldots, \gamma_n)$ be an orthonormal basis for $\pi^{-1}_{\mathcal{M}}(e_i)$. The local *geometric support* for e_i is given by

$$S_i = \bigwedge_{k \leq n} P_{\gamma_k}(e_i - \rho\gamma_k) \wedge -P_{\gamma_k}(e_i + \rho\gamma_k).$$

where P_{γ_k} is the directional derivative of P in the direction γ_k.

This support S_i is guaranteed to be positive if there is a local maxima in $b_\rho(e_i) \cap \pi^{-1}_{\mathcal{M}}(e_i)$.

Theorem 7.21 *Given that $\rho > \rho(E_i)$ the geometric support S_i is positive if there is a local maximum of $P(e_i)$ on the subspace of E normal to M_{e_i} in the Voronoi cell E_i.*

Referring to Thm. 5.1, we see that

$$P_{\gamma_k}(e_i - \rho\gamma_k) > 0 \;\text{ and }\; P_{\gamma_k}(e_i + \rho\gamma_k) < 0 \;\text{ and }\; P_{\gamma_k\gamma_k}(e_i) < 0$$

guarantee the existence of a local maximum of P on the line from $e_i - \rho\gamma_k$ to $e_i + \rho\gamma_k$. Thus, when there is a local maximum of P in $b_\rho(e_i)$, this condition is true for all directions $\gamma \in \Gamma$. Therefore each of the expressions $P_{\gamma_k}(e_i - \rho\gamma_k) \mathbin{\hat{\wedge}} -P_{\gamma_k}(e_i + \rho\gamma_k)$ is positive when there is a local maximum within E_i.

By Def. 4.1 we know that the L/L AND of positive values is necessarily positive. Thus the support S_i is positive if there is a local maximum of $P(e_i)$ on $\pi_{\mathcal{M}}^{-1}(e_i)$ within the ball $b_\rho(e_i)$. Since $\rho > \rho(E_i)$, this ball encloses the Voronoi cell E_i. ∎

Choosing an orthonormal basis for $\pi_{\mathcal{M}}^{-1}(e_i)$ is not difficult for the geometric problems we have investigated. Because the cross-section has a unique value on each fibre, any orthonormal basis for the fibre F will be at a proper subset of a basis for $\pi_{\mathcal{M}}^{-1}(e_i)$. If S, the domain of M, is of the same dimensionality as B then this basis is complete since it is orthonormal, normal to M and of the required dimensionality. Otherwise, we need only augment this basis with an orthonormal basis for the subspace normal to S in B.

Finally we can relate the geometric support which we have developed to the relaxation compatibilities of §7.1. Recall that in the two label case, the relaxation support simplifies to

$$s_i \;=\; \sum_j p_j\, r_{ij}.$$

Now, the local geometric support for e_i is of the form

$$S_i \;=\; \mathop{\hat{\bigwedge}}_k \pm P_{\gamma_{k/2}}(e_i \mp \rho\gamma_{k/2}).$$

where the linearity of convolution and the derivative operator imply that

$$P_{\gamma_{k/2}}(e_i \mp \rho\gamma_{k/2}) \;=\; \sum_j p_j\, G^2_{\gamma_{k/2}}(e_i \mp \rho\gamma_{k/2}).$$

where

$$G^{ij}(e_i) \;=\; G_{\sigma_t}(d^t(e_i, e_j))\; G_{\sigma_c}(d^c(e_i, e_j)).$$

Thus the geometric compatibility can be rewritten in the form

$$S_i \;=\; \bigwedge_k S_i^k, \tag{7.6}$$

where

$$S_i^k \;=\; \sum_j p_j\, r_{ij}^k$$

and

$$r_{ij}^k \;=\; \pm G^{ij}_{\gamma_{k/2}}(e_i \mp \rho\gamma_{k/2}).$$

Note that because of the partial derivatives in normal directions, those points which are laterally displaced from the inferred manifold by greater than ρ and are parallel to it are inhibited. Thus the S_i^k components are referred to as the *lateral inhibition components* of the support or simply the *lateral components.* The analogy with traditional lateral inhibition methods [Rat65] is obvious.

Thus we have a relaxation labelling algorithm with guaranteed fixed-points, the thick traces of cross-sections $M \in \mathcal{M}$ of some geometric model. And while it is true that the L/L non-linearities obscure the relationship between the relaxation and a gradient ascent, the behaviour of this relaxation meets the goals set out above. We have already observed the fixed-point behaviour, and the lateral inhibition guarantees that a label near M but not in $\text{THICK}_\rho(M)$ will be actively suppressed.

A number of free parameters arise in this calculation: the values of ρ, σ_c and σ_t. Significantly though, they are tightly constrained. As we have seen ρ in Def. 7.20 is currently constrained by $\rho > \rho(E_i)$. We have already shown that if this is true, then the points selected by $S_i > 0$ are a superset of those ideally selected. It is clear that $\rho \approx \rho(E_i)$ for $\text{THICK}_\rho(M)$ to be a reasonable approximation to $\text{THICK}(M)$, and must certainly be constrained above by $\rho < 2\rho(E_i)$. Furthermore, P on $\pi_{\mathcal{M}}^{-1}(e_i)$ is a weighted sum of Gaussians $G_{\sigma_c}(|e_i - e_j^\perp|)$ where $|e_i - e_j^\perp| < \rho$. Thus, for the L/L combination of Def. 7.20 to provide a good estimate of the second derivative of this field we require that $\sigma_c > \rho$. However, σ_c cannot be much greater than this without causing inappropriate merging of nearby independent traces [DZ90]. This discussion is summarized by the conditions:

$$\begin{aligned} \rho(E_i) \; < \; \rho \; < \; 2\rho(E_i), \\ \rho \; < \sigma_c < \; 2\rho(E_i). \end{aligned}$$

Note that these constraints are *independent* of the particular geometric problem being solved.

The last of the free parameters σ_t is the least constrained. It is certainly obvious that $\sigma_t > \sigma_c$ since we *want* to combine points on the same cross-section even at some distance. However, the upper bound of σ_t is constrained only by the accuracy of the model in reflecting the local structure of the image. The significant question is over what range we can safely assume that the model is an accurate description of the image. For example using the line model, if we know that the extracted image curves have curvatures which vary significantly from zero, then the straight line model may only be accurate for as little as two or three pixels around any point. This would then require that $\sigma_t \approx 1.0$. Clearly this is not much context to contribute to the construction of global curves. This is one reason that our actual image curve models incorporate curvature information directly.

Example 7.4 *(continued)* To define the local geometric support for our line model we must determine a basis for the subspace normal to a model cross-section. Clearly the vector $\gamma_1 = (0,0,1)$ constitutes a basis for the fibre Θ over any point in $B(\mathcal{L})$. Since the co-dimension of S in $B(\mathcal{L})$ is one, we need only augment this basis with the vector $\gamma_2 = (\sin\theta_i, -\cos\theta_i, 0)$ at $e_i = (x_i, y_i, \theta_i)$. $\Gamma = \{\gamma_1, \gamma_2\}$ thus forms a complete basis of the subspace of $E(\mathcal{L})$ normal to L_{e_i} at e_i.

Given this basis, we can calculate the geometric support for e_i as

$$S_i \;=\; P_{\gamma_1}(e_i - \rho\gamma_1) \curlywedge -P_{\gamma_1}(e_i + \rho\gamma_1) \curlywedge P_{\gamma_2}(e_i - \rho\gamma_2) \curlywedge -P_{\gamma_2}(e_i + \rho\gamma_2). \qquad (7.7)$$

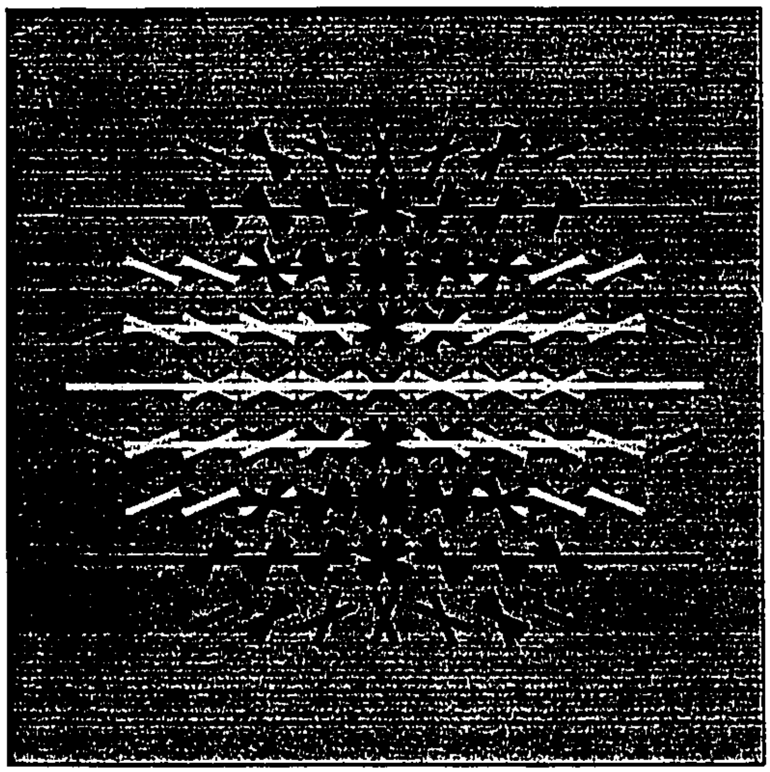
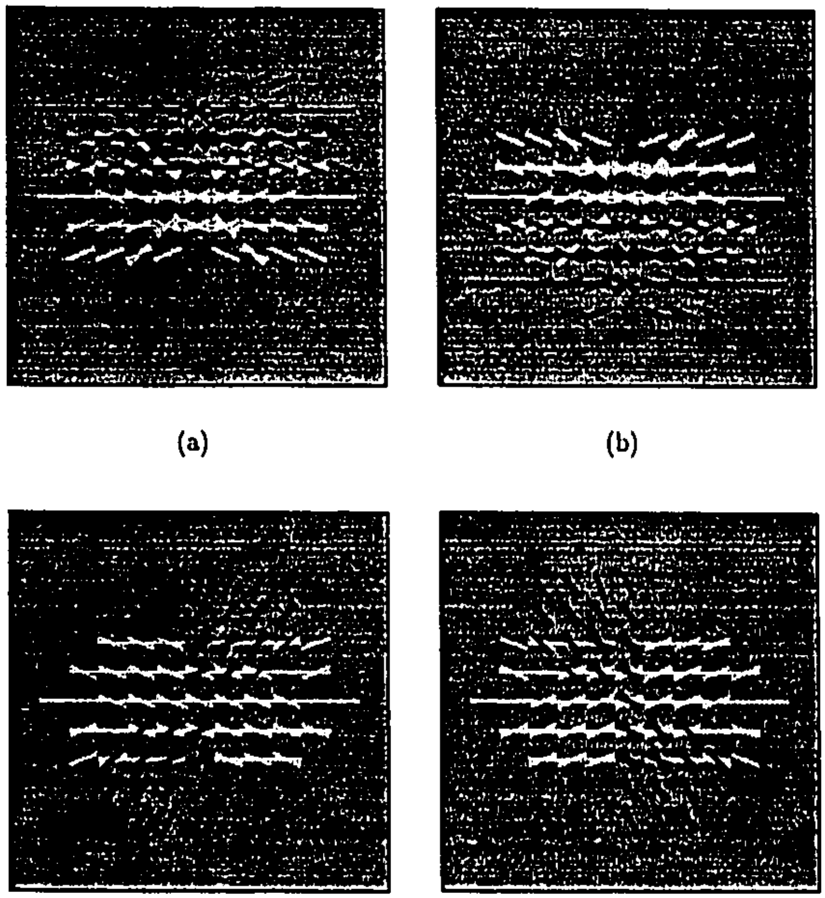


Figure 7.6: The linear reduction of the L/L support for the horizontal line label. Shown are the relative positions and orientations of compatible (white) and incompatible (black) labels in the neighbourhood of a horizontal line label.



(a) (b)

(c) (d)

Figure 7.7: The four component support networks for the horizontal line label. The four networks represent (a) $P_{\gamma_1}(e_i - \rho\gamma_1)$, (b) $-P_{\gamma_1}(e_i + \rho\gamma_1)$, (c) $P_{\gamma_2}(e_i - \rho\gamma_2)$, and (d) $-P_{\gamma_2}(e_i + \rho\gamma_2)$. The local support for the horizontal label is only positive if the inner products of each of these fields with the local confidences is positive. The networks (a) and (b) ensure that the horizontal line is at a local maxima in position normal to its orientation, while (c) and (d) ensure that it is at a local maxima in orientation.

The four linear components of this support are instantiated in the four support networks shown in Fig. 7.6.

7.3.1 Boundary Stability

Boundary stability is the requirement that the boundaries of cross-sections must neither dilate nor contract through iterations of the relaxation. This same property was referred to as end-line stability in §5.2 in the context of image curve operators. It is easy to appreciate that boundary stability is a generalization of this concept, since the boundaries of lines are their end-points.

To see that this is a problem in the design of relaxation support we need only refer to the analysis above. The support function as defined in Def. 7.20 involves the selection of local maxima in directions perpendicular to a manifold, constructed by a smooth combination of the confidence measurements on a thick trace. This method ensures that regions of positive support will not dilate perpendicular to this manifold, but it deliberately encourages dilation along the manifold. Thus if the set of selected labels is some *subset* of the thick trace of a model cross-section, then the *entire* thick trace will receive positive support. This is a significant problem if the target cross-sections are bounded, proper subsets of the model cross-sections.

Example 7.5 *(continued)* Each line L in the model $\mathcal{L}$ is bounded if and only if the base space X is bounded. However, an actual image of straight lines will contain a number of line segments with arbitrary bounds. Assume that one such image consists of a single line segment ℓ and that an initial labelling correctly selects the thick trace of this line. Thus, only those points in the thick trace have non-zero confidence. Examining eq. (7.7) reveals that every point in the thick trace of the model line L for which $\ell \subset L$ will receive positive support (see Fig. 7.8). Thus the fixed point of the relaxation labelling will be the thick trace of the model line L and not the segment ℓ.

This example suggests that the solution to the boundary stability problem may be developed as an extension of the design of end-line stable image operators. Recall the eventual statement of the continuity problem for image curves in §5.2: *a one-*

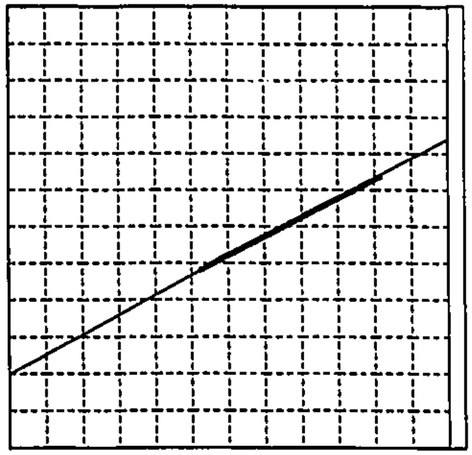


Figure 7.8: Without incorporating end-line stability conditions, the region of positive support (shaded) around the thick trace of a bounded line segment (the heavy line) covers the entire model line. In order to ensure that support select only the the thick trace of the segment, we must impose an additional condition referred to as boundary stability.

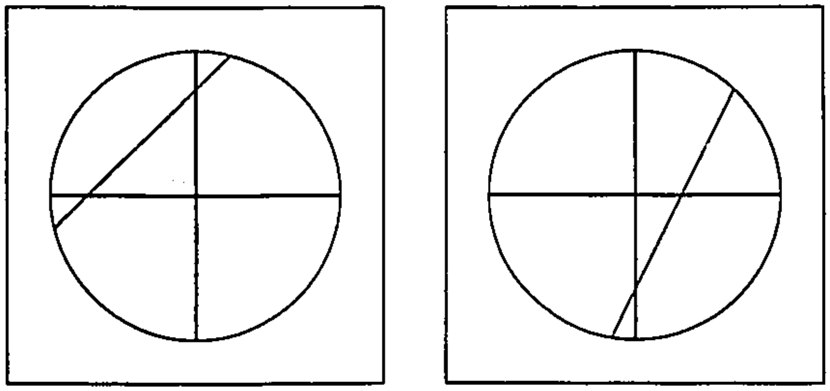
dimensional operator is end-line stable if and only if it responds positively if and only if its centre is in a uniformly positive region of responses. We can restate this condition for support on manifolds of arbitrary dimension as: *geometric support is boundary stable if and only if it is positive exactly when centered on a uniformly positive region of the labelling map.*

The problem thus becomes one of deciding when e_i is within a region of positive support defined by the labelling map.

Observation 7.22 Consider a closed, connected submanifold M_0 of M. If M_0 is of the same dimensionality as M, then for all regular curves $\alpha : \mathbb{R} \to M$ such that $\alpha(0) = e_i$

$$e_i \in M_0 \iff \exists \rho > 0, \forall s \in [0, \rho) : \alpha(s) \in M_0.$$

In essence, $e_i \in M_0$ if and only if the M_0 *surrounds* e_i. In geometric terms, cast a ray out from e_i in all directions on M and if each such ray intersects M_0 in a neighbourhood which includes e_i then e_i is surrounded. There may be other methods of verifying whether or not a point is in the interior of a region, but this can be incorporated naturally into the calculation of support defined above. Since the support field S_i at a point e_i is the sum of contributions from points in the neighbourhood



(a) (b)

Figure 7.9: A decomposition of the neighbourhood of a point into quadrants allows one to verify whether or not the point is "inside" some convex region (in gray) by verifying that all quadrants intersect the region.

of e_i, if we need to verify that positive support is coming from all directions we can decompose the support calculation into a small set of direction ranges and ensure that there is positive support from all of these ranges. If we assume that locally, the boundary of the region of positive support is linear or convex, then there is a simple local decomposition which will verify this kind of "surround."

Definition 7.23 Let $Z = \{ \zeta_1, \ldots, \zeta_n \}$ be an orthonormal basis for the manifold M. We can then define the n half-planes around $m \in M$ as

$$H_i(m) \;=\; \{ x \in M \mid (x - m) \cdot (\zeta_i - m) > 0 \}\,.$$

Define a partition of M around $m \in M$ into the 2^n regions $Q_l(m)$, which we call the *generalized quadrants* of M around m, such that

$$Q_l(m) \;=\; \{ x \in M \mid \forall i \in (1, \ldots, n) \colon x \in H_i(m) \Leftrightarrow \textsc{Bin}(l, i) = 1 \}.$$

where $\textsc{Bin}(l, i)$ is the i^{th} digit in the binary representation of l.

For example, if M is the real line, then the $Q_l(m)$ are the half-fields greater than or less than m; if M is the plane, then the $Q_l(m)$ are quadrants around m, and so on. Each such quadrant is spanned by a compact, connected set of directions around m. They thus form an ideal starting point for the discrimination described above since:

Theorem 7.24 *If N is a convex, path-connected submanifold of M with the same dimensionality as M, then*

$$(\forall l\colon N \cap Q_l(m) \neq \emptyset) \;\Rightarrow\; m \in N.$$

If $\forall i\colon N \cap Q_l(m) \neq \emptyset$ then we can select one point x_l from each quadrant $Q_l(m)$ such that $x_l \in N$. The convex hull of these points $H(x_1, \ldots, x_{n^2})$ contains m. Since all these x_l are in N and N is convex, then $H(x_1, \ldots, x_{n^2}) \subset N$ and thus $m \in N$. ∎

The consequence of this result on the design of boundary stable support functions is now clear. Since the boundaries on the model section M project to unique boundaries on the base space $B(\mathcal{M})$, we are concerned that $\pi(e_i)$ be inside a region of positive support. The support network for e_i is decomposed into the 2^n regions specified by the quadrants $Q_l(\pi(e_i))$ around $\pi(e_i)$ on $\pi(M_{e_i})$ where n is the dimensionality of M_{e_i}. The geometric support will now be positive if and only if the support in each of the 2^n quadrants is positive. Assuming then that $Z = \{\zeta_1, \ldots, \zeta_n\}$ is an orthonormal basis for B, we define the quadrants

$$Q_l(e_i) \;=\; \{\, e \in B \mid \forall j \in (1, \ldots, n)\colon e \in H_j(e_i) \Leftrightarrow \textsc{Bin}(l, j) = 1 \,\},$$

around e_i where

$$H_j(e_i) \;=\; \{\, e \in B \mid (e - e_i) \cdot (\zeta_j - e_i) > 0 \,\}.$$

Then the geometric compatibility can be rewritten in the form

$$\begin{aligned} S_i &= \bigwedge_l S_i^l \\ S_i^l &= \bigwedge_k S_i^{kl} \end{aligned}$$

$$S_i^{kl} = \sum_j p_j\, r_{ij}^{kl}$$

where

$$r_{ij}^{kl} = \begin{cases} \pm G^{ij}_{\gamma_{k/2}}(e_i \mp \rho\gamma_{k/2}) & \text{if } e_j \in Q_l; \\ 0 & \text{otherwise.} \end{cases}$$

This decomposition of the support network into *support regions* guarantees that the geometric support is stable near boundaries—positive inside and non-positive outside.

Referring back to the implementation of end-line stability for image curve operators in §5.2, we can see that there is an alternate definition for this. A characteristic function of the quadrant $Q_l(e_i)$ is given by

$$q_l(e_i, e_j) = \prod_\iota \sigma_\iota^l(\zeta_\iota \cdot (e_j - e_i)),$$

where

$$\sigma_\iota^l(x) = \begin{cases} \sigma(x) & \text{if } \mathrm{BIN}(l, \iota) = 1; \\ \sigma(-x) & \text{otherwise.} \end{cases}$$

Thus

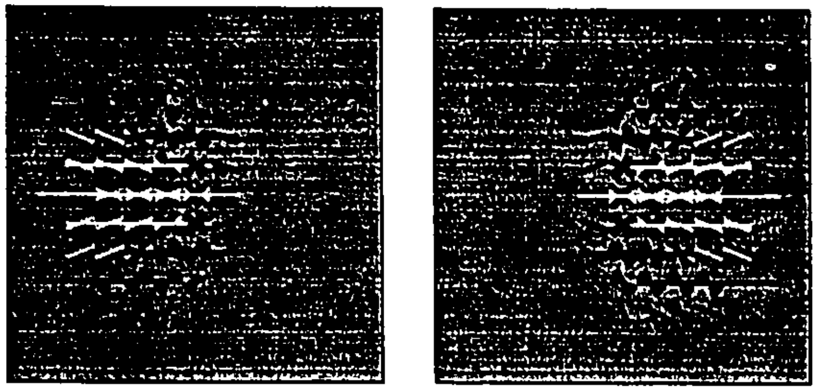
$$r_{ij}^{kl} = \pm G^{j}_{\gamma_{k/2}}(e_i \mp \rho\gamma_{k/2})\; q_l(e_i, e_j).$$

Example 7.6 *(continued)* The straight line of orientation θ has a basis in the plane consisting of the vector $\zeta = (\cos\theta, \sin\theta)$. Thus, as with the image curve operator, to produce a boundary stable support we decompose into two regions along the line. The half-field partitions are defined as

$$q_1(e_i, e_j) = \sigma(\zeta \cdot (e_j - e_i)),$$
$$q_2(e_i, e_j) = \sigma(\zeta \cdot (e_i - e_j)).$$

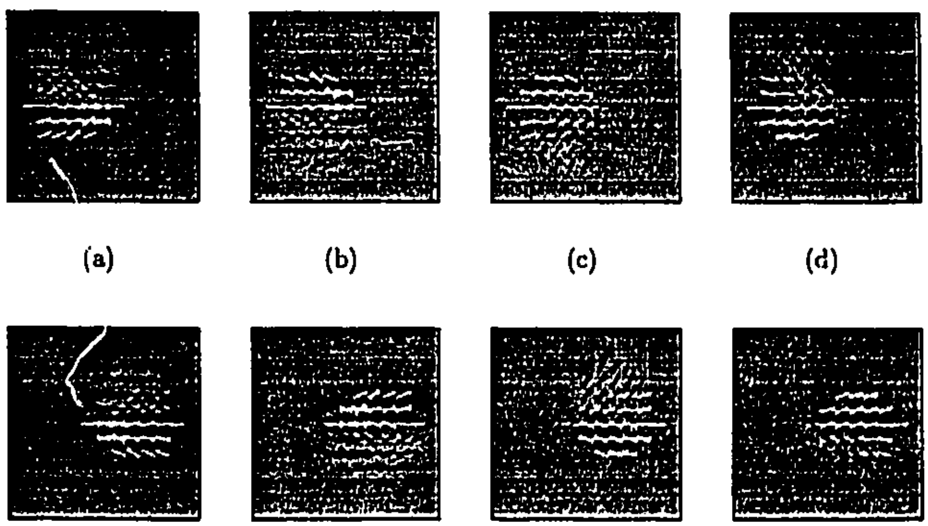
Thus the geometric support of the line label e_i is given by

$$S_i = \bigwedge_{l,k} \sum_j p_j\, r_{ij}^{lk},$$



(a) (b)

Figure 7.10: The linear reductions of (a) the left and (b) the right hand sides of the boundary stability decomposition of the support network for the horizontal line label.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7.11: The eight component support networks for the horizontal line label. Each field shows the relative positions and orientations of compatible (white) and incompatible (black) labels in the neighbourhood of a horizontal line label. The networks show are: (a) $r_{ij}^{1,1}$, (b) $r_{ij}^{2,1}$, (c) $r_{ij}^{3,1}$, (d) $r_{ij}^{4,1}$, (e) $r_{ij}^{1,2}$, (f) $r_{ij}^{2,2}$, (g) $r_{ij}^{3,2}$, and (h) $r_{ij}^{4,2}$. All eight of these networks are instantiated for each discrete position in the image and each discrete orientation.

and

$$r_{ij}^{lk} = \pm G^{j}_{\gamma_{k/2}}(e_i \mp \rho\gamma_{k/2})\ q_l(e_i, e_j),$$

where

$$G^j(e_i) = \mathrm{G}_{\sigma_t}(d^t_{\mathcal{C}}(e_i, e_j))\ \mathrm{G}_{\sigma}(d^c_{\mathcal{C}}(e_i, e_j)).$$

The resulting support networks are shown in Fig. 7.11.

7.4 Summary

In review, we have shown how to construct relaxation networks which solve geometric problems. After constraining the class of geometric models to consider, we defined a general method for translating these models into fixed relaxation labelling networks. These networks produce good approximations to thick traces of model cross-sections. Unlike previous approaches, this method formally identifies the fixed points of the relaxation with particular cross-sections in the models. Moreover, by designing the relaxation for stability in the presence of boundaries, we have ensured that the only effect of the relaxation is to select and fill in thick traces where they previously existed.

The one concern that may remain in applying these methods to early vision is the iterative nature of the relaxation method. Marr [Mar82] claimed that the speed demanded of the early vision system precluded the use of relaxation or global optimization methods which typically require tens or hundreds of iterations to converge on stable solutions. However, we will show in the following chapters that this geometric relaxation method typically converges in as few as three or four iterations, thus rehabilitating it as a theory of early vision.

Chapter 8 Image Curves

Reliable descriptions of *image curves* as piecewise smooth plane curves in images are fundamental for much of high-level processing, especially recognition. In this chapter we will reconsider the problem of designing a reliable system for extracting thick trace descriptions of those curves in light of the results of Chap. 7.

In Chap. 3, an image curve was defined as the locus of one-dimensional discontinuities in the intensity surface. We will build on this definition and the consequences outlined in §3.2. In keeping with the focus of the previous chapter, we will now incorporate an explicit representation of the local geometry of the curves into a geometric model, and design a relaxation labelling system to extract thick traces of curves from Logical/Linear operator responses.

8.1 Representation

The Fundamental Theorem of the Local Theory of Curves ([dC76] pp. 19) asserts that the combination of local orientation and curvature maps defines a plane curve uniquely under translation and rotation. It is not surprising then that curvature is a fundamental building block of modern theories of contour shape [Ley88, Kim91, KTZ92]. Clearly then, a visual system must make curvature measurements explicit for whichever higher-level processes construct descriptions of shapes prior to their recognition. The only issue then is *when curvature is made explicit.*

In Chap. 3 we demonstrated that orientation must be made explicit in order to simply detect the presence image curves. Two pieces of evidence suggest that curvature too should be made explicit at the earliest stages of curve description. Psychophysical analysis of dotted lines suggests that purely *local* curvature information can strongly bias the ability to reliably locate curve discontinuities [LZ88]. Neurophysiologically, curvature tuned neurons have been observed in primary visual cortex [DZC87].

From a purely empirical viewpoint, we suggest that a model with explicit cur-

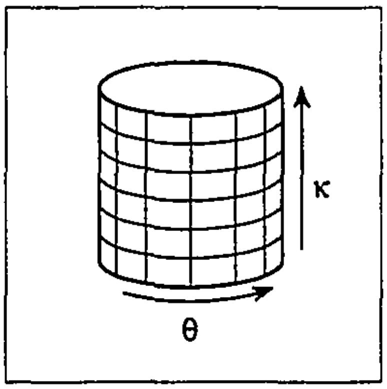


Figure 8.1: The fibre space $\Theta \times K$ for image curve representation is a cylindrical space, with orientation/curvature pairs identified with points on the surface of the cylinder.

vature will allow for faster, more accurate relaxation. Considering the orientation and curvature as the first two terms in a local Taylor expansion of the curve, we can assert that a description which includes both will be accurate (to within sampling uncertainties) over a larger neighbourhood than one which includes only orientation. We will show that a relaxation system which incorporates local curvature explicitly can thus integrate more local information in a single iteration, and therefore converge very quickly.

Thus we choose an image bundle which explicitly represents local orientation and curvature measurements on the fibre $F = \Theta \times K$ (see Fig. 8.1) over each point $x \in X$ for the image $I : X \to \mathbb{R}$. Given that both the base space X and fibre F will be discretized by sampling, the image curves are represented in this sampled total space as thick traces of the actual curves in the image.

Following the analysis of Chap. 2, we will first consider the sampling of the total space of this bundle. A regular sampling of the base space is naturally provided by the pixelization of the image. Thus for each pixel in the image we have a discrete set of orientation/curvature pairs which represent possible local geometries for curves passing through that pixel. Orientation is sampled regularly over either π or 2π radians depending on whether the image curve has direction or simply orientation (we will return to this with specific examples). Thus for n discrete orientations, we sample at the points $\theta_i = i\pi/n$ for $i \in \{0, \ldots, n-1\}$. Curvature is also sampled regularly over the range $K = [-\kappa_{\max}, \kappa_{\max}]$, where $\kappa_m ax$ is chosen appropriate to the scale (we will return to this below). With m curvatures we sample at $\kappa_j = \kappa_{\max}(-1+2j/(m-1))$ for $j \in \{0, \ldots, m-1\}$. The simplest Riemannian metric which ensures that this is an unbiased sampling is an L_2 metric such that the distance between adjacent sample points in each dimension is always 1. This is achieved with the metrics

$$\begin{aligned} d(\theta_i, \theta_j) &= \frac{|(\theta_i - \theta_j) \bmod_c r|}{\pi/n}, \\ d(\kappa_i, \kappa_j) &= \frac{|\kappa_i - \kappa_j|}{2\kappa_{\max}/(m-1)}, \end{aligned}$$

and $d(e_i, e_j)$ an L_2 combination of these. The $x \bmod_c y$ operation is a centered modulus, with the output values restricted to the interval $(-y/2, y/2]$. Since for lines only relative orientation is significant $r = \pi$, whereas for edges $r = 2\pi$. Thus for each pixel in the image we have $m \times n$ discrete local geometries.

Referring to these samples of the fibre as local geometries depends, of course, on the assignment of a model C to this image bundle. The single assumption needed to develop such a model is the assumption of (locally) constant curvature. Each triple of a point, orientation and curvature in the total space of the model then delineates a unique circle. These circles can be described succinctly by a parameterized model, which is equivalent to *cocircularity* as defined in [PZ85].

Consider first a circle passing through the origin with orientation 0 and curvature κ. An arc-length parameterization of the position, orientation and curvature of this circle is given by the following vector

$$C_\kappa(s) = \begin{pmatrix} \sin(\kappa s)/\kappa \\ (1-\cos(\kappa s))/\kappa \\ \kappa s \\ \kappa \end{pmatrix}.$$

Now, since all circles with the same curvature are simply rotations and translations of such a circle, we can parameterize the image curve model over the entire total space

E_C by rotating and translating these circles.

Identifying a point in the total space as (x, y, θ, κ) the image curve model $C = \{ C_{xy\theta\kappa} \}$ is parameterized by the function

$$C_{xy\theta\kappa}(s) \;=\; \mathrm{T}_{x,y}\, \mathrm{R}_{\theta} \begin{pmatrix} \sin(\kappa s)/\kappa \\ (1-\cos(\kappa s))/\kappa \\ \kappa s \\ \kappa \end{pmatrix},$$

where $\mathrm{T}_{x,y}$ is translation by (x, y) and R_θ is rotation by θ around the origin. Note that $C_{xy\theta\kappa}$ is trivially re-parameterized around $(x', y', \theta', \kappa') = C_{xy\theta\kappa}(s)$ by translation $\mathrm{T}_{x'-x,y'-y}$ and rotation $\mathrm{R}_{\theta'-\theta}$, both of which are invertible.

Corollary 8.1 *C is a minimal model.*

8.2 Initial Estimates

The first practical problem in extracting the thick traces of image curves is of course to define how those curves are instantiated in an image, and how to design local operators tuned for this instantiation and a particular point in the fibre. The discussion and motivation surrounding the development of the Logical/Linear operators of Part II clearly establishes them as candidates for this task. They classify image curves into bright and dark lines and edges, their response profiles cover the Voronoi cells for the zero-curvature subspace of our total space, and they represent a stable, logically well-founded approximation of the intersection condition which forms the foundation of the definition of the thick trace (Def. 2.10) at least for isolated curves. Significantly too, the graded responses from these operators can be interpreted as a "strength of agreement" between the abstract model and the image, a clear foundation for their use as initial estimates in a relaxation. The only difficulty is that these operators are uniquely tuned for straight lines, exhibiting a monotonic decrease in response with increase in curvature.

There is however a clear path around this impasse. Dobbins has developed a theory of end-stopping in visual cortex which equates this phenomenon with curvature tuning of simple cells [DZC88, Dob92]. In the simplest form, an end-stopped operator is constructed by taking the difference between the responses of an excitatory component R_+ and an inhibitory component R_- via the formula

$$R_{ES} = \phi(\phi(R_+) - \phi(R_-)),$$

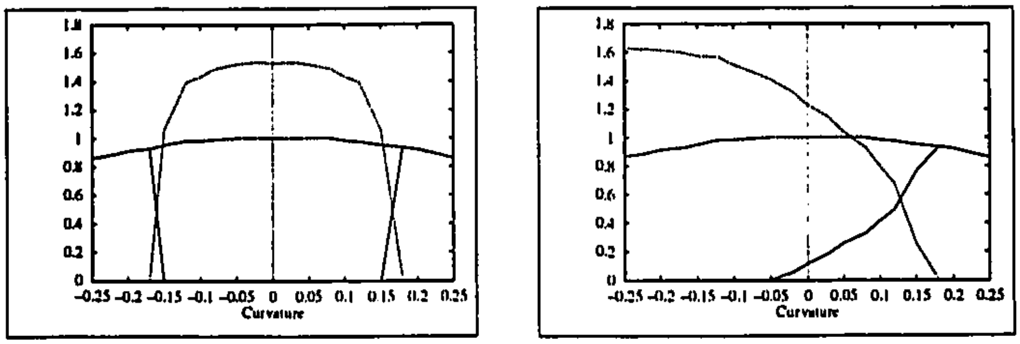
where

$$\phi(x) = \begin{cases} x & \text{if } x \geq 0; \\ 0 & \text{otherwise.} \end{cases}$$

The effect of the rectification operator ϕ is to prevent a negative response to the inhibitory component from contributing positively to the aggregate operator. In Dobbins' work this is justified neurophysiologically by reference to the low spontaneous firing frequency of neurons in primary visual cortex. For our work, the pragmatic effect is more significant.

If both component operators are matched in position, cross-section and orientation tuning, then any differences in their sensitivity to variation in curvature will result in an aggregate response which is maximal for some specific, possibly non-zero, curvature. In particular assuming that both components have response maxima at zero curvature (e.g. the L/L operators of Part II), then if the excitatory component is broadly curvature tuned and the inhibitory operator is tightly tuned, the aggregate response will be maximal for some non-zero curvature. The behaviour can be seen in Fig. 8.2.

Dobbins' work and analysis was built on the assumption that the component operators were linear, so there may be some hesitation in applying it unmodified to the L/L operators we use. Firstly, we can can question whether the analysis used to support the model applies to these operators. Secondly, we can ask how the rectifying non-linearity compares with the L/L nonlinearities? The first concern is at least partly dealt with by referring to the linear part of the L/L paradigm. Within the region of input space for which the L/L operators give positive responses (a restriction imposed by the rectifying operator) we can assert that we are within a



(a) (b)

Figure 8.2: Tuning profiles of end-stopped components can produce an aggregate operator tuned for non-zero curvature even when both components are tuned for zero curvature. In both profiles, the responses of both excitatory and inhibitory components are shown dotted and their rectified difference is shown solid. In the case (a) where the curvature responses of both the excitatory and inhibitory components of an end-stopped operator are symmetric, the aggregate response is also symmetric and thus tuned for the magnitude of curvature. When (b) the inhibitory component has an asymmetric curvature response profile the aggregate operator can be tuned for both magnitude and *sign* of curvature.

single linear subspace of the input space. Thus at least some of Dobbins' extensive linear analysis can be rehabilitated for our non-linear operators. To deal with the second, we need only observe that the rectification operator $\phi(x)$ is, at least in the ideal sense, a Logical/Linear combinator itself! Consider the equation

$$\phi(x) = 0 \mathbin{\text{△}} x.$$

So there should be little conceptual difficulty in applying this "end-stopping equals curvature" methodology to the design of L/L operators.

The one final concern is that the simple Dobbins' operator selects only curvature magnitude, whereas it is essential to discriminate both sign and magnitude of curvature. As seen in Fig. 8.2b this *can* be achieved with an asymmetric inhibitory response. Significantly, this asymmetry can be obtained by creating an inhibitory operator formed from a subset of the normal components which form the excitatory operator. For example, for the positive contrast line operator we produce the inhibitory component by selecting the normal components which are on the side *away* from the preferred sign of curvature. Referring back to eq. (5.7) we thus describe the normal cross-section of the inhibitory operator as

$$\begin{aligned} \mathrm{N}^{+}_{P-} &= \mathrm{n}'_l \mathbin{\text{△}} \mathrm{n}^{(3)}_l, \text{ or} \\ \mathrm{N}^{-}_{P-} &= \mathrm{n}'_r \mathbin{\text{△}} \mathrm{n}^{(3)}_r, \end{aligned}$$

depending on whether we are designing an operator for positive or negative curvature.

Thus we can extend our design from Chap. 5 to support curvature tuning.

Operator 8.1 The *Logical/Linear image curve operators* Ψ_i tuned for non-zero curvatures are given by

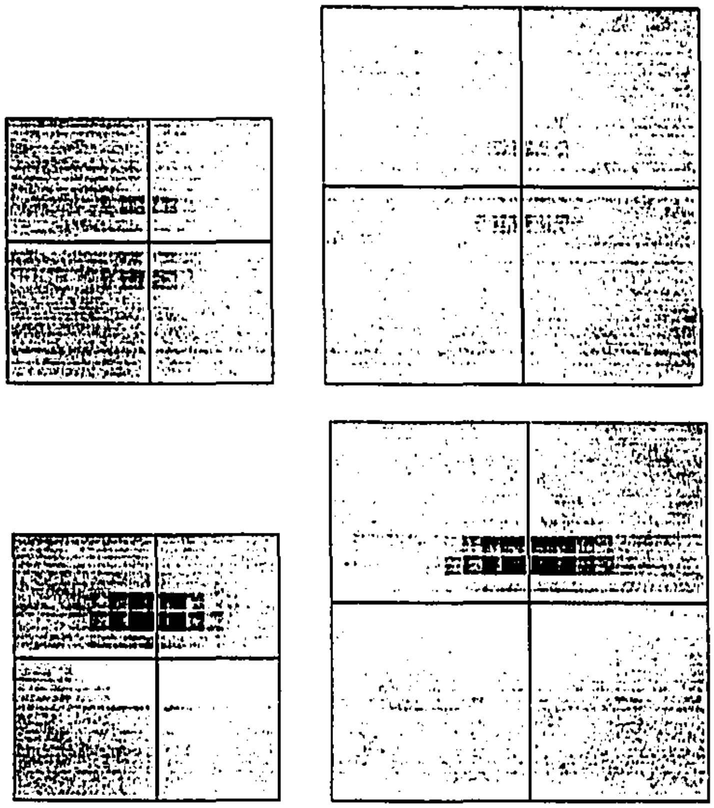
$$\begin{aligned} \Psi_i &= w_+\Psi_{i+} - \phi(w_-\Psi_{i-}), \\ \Psi_{i+} &= (\mathrm{t}^{-}_{\sigma_{i+}} \times \mathrm{N}_{i+}) \mathbin{\text{△}} (\mathrm{t}^{+}_{\sigma_{i+}} \times \mathrm{N}_{i+}), \\ \Psi^{\pm}_{i-} &= (\mathrm{t}^{-}_{\sigma_{i-}} \times \mathrm{N}^{\pm}_{i-}) \mathbin{\text{△}} (\mathrm{t}^{+}_{\sigma_{i-}} \times \mathrm{N}^{\pm}_{i-}), \end{aligned}$$

where w_+ and w_- are weights, $i \in \{P, N, E\}$ and $\sigma_{t-} > \sigma_{t+}$ and

$$
\begin{array}{lcll}
\mathrm{N}_{P+} & = & \mathrm{n}'_l \wedge \mathrm{n}'_r \wedge \mathrm{n}^{(3)}_l \wedge \mathrm{n}^{(3)}_r & \\
\mathrm{N}^+_{P-} & = & \mathrm{n}'_l \wedge \mathrm{n}^{(3)}_l & \\
\mathrm{N}^-_{P-} & = & \mathrm{n}'_r \wedge \mathrm{n}^{(3)}_r & \text{for Positive Contrast Lines;} \\
\mathrm{N}_{N+} & = & -\mathrm{n}'_l \wedge -\mathrm{n}'_r \wedge -\mathrm{n}^{(3)}_l \wedge -\mathrm{n}^{(3)}_r & \\
\mathrm{N}^+_{N-} & = & -\mathrm{n}'_l \wedge -\mathrm{n}^{(3)}_l & \\
\mathrm{N}^-_{N-} & = & -\mathrm{n}'_r \wedge -\mathrm{n}^{(3)}_r & \text{for Negative Contrast Lines;} \\
\mathrm{N}_{E+} & = & \mathrm{n}'_c \wedge \mathrm{n}''_l \wedge \mathrm{n}''_r \wedge \mathrm{n}^{(4)}_l \wedge \mathrm{n}^{(4)}_r & \\
\mathrm{N}^+_{E-} & = & \mathrm{n}''_l \wedge \mathrm{n}^{(4)}_l & \\
\mathrm{N}^-_{E-} & = & \mathrm{n}''_r \wedge \mathrm{n}^{(4)}_r & \text{for Edges.}
\end{array}
$$

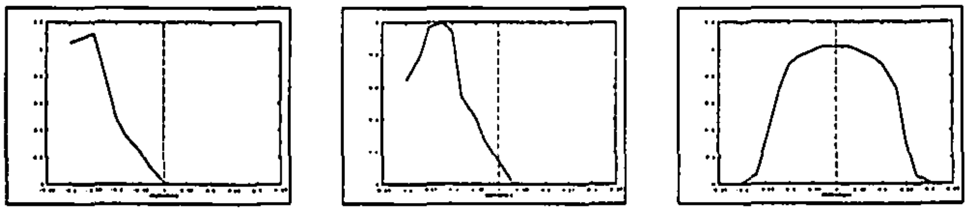
Examples of these operators for bright lines (Ψ_P) are show in Fig. 8.3. The curvature responses of these two operators are shown in Fig. 8.4. Note that normal to the preferred orientation, each operator smooths with a Gaussian with $\sigma_n = \sqrt{2}/2$ and localizes maxima to a region $\sqrt{2}$ pixels wide. This produces a family of operators tuned for perhaps the smallest scale possible while still reliably eliminating noise. At this scale, 5 pixels is close to the minimum radius of a circle which can be reliably distinguished from a blob and simultaneously categorized into either a line or edge-like discontinuity. This is the source of the limit, $\kappa_{max} = 0.2$.

Having seen that these operators are tuned for position, orientation and curvature (the basis functions of the total space), the final piece of data needed to justify their use as estimators for the thick trace of image curves is some match between their sensitivity and the Voronoi cells of the sampling. Consider a response map over the points e_i in the total space E_C for an ideal image formed from the model curve C_{e_i}. In order to ensure that there are no blind-spots in the operators (ideal curves which no operator will respond to), we must ensure that the total space is covered by the positive responses of the operators to these ideal stimuli. Locally, this means that if we map the positive responses of an individual operator varying the position, orientation, and curvature of the stimulus curve—the sensitivity region must *cover* the respective Voronoi cell.



(a) (b)

Figure 8.3: Excitatory and inhibitory parts of curvature tuned positive contrast line operators. For each of the operators shown, the aggregate response is the difference between the response to the excitatory operator (top) and the inhibitory operator (bottom) is the aggregate response. According to the results of Dobbins [Dob92] this response should be tuned to a particular combination of orientation and curvature, dependent on the relative lengths of the excitatory and inhibitory operators. Two end-stopped bright line operators are shown: (a) tuned for curvature $\kappa = -0.2$, and (b) $\kappa = -0.1$.



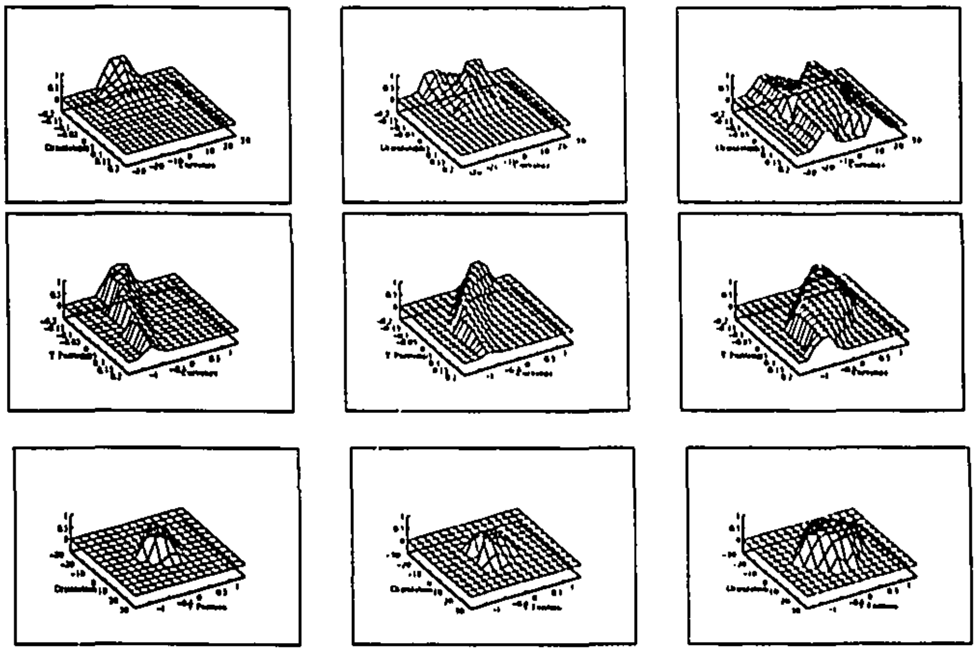
(a) (b) (c)

Figure 8.4: Curvature responses for the two curved line operators of Fig. 8.3 and a straight (zero-curvature) operator. These are obtained by examining the response to an ideal curve of width 2 pixels. The responses are for operators tuned for (a) $\kappa = -0.2$, (b) $\kappa = -0.1$, and (c) $\kappa = 0$.

Lines	Excitatory			Inhibitory		
	σ_{t+}	w_+	n_+	σ_{t-}	w_-	n_-
$\kappa = 0.0$	2.80	1.1	4	—	—	—
$\kappa = \pm 0.1$	2.40	1.2	4	3.2	2.2	4
$\kappa = \pm 0.2$	1.67	1.3	2	2.3	2.8	4

Edges	Excitatory			Inhibitory		
	σ_{t+}	w_+	n_+	σ_{t-}	w_-	n_-
$\kappa = 0.0$	2.80	1.50	4	—	—	—
$\kappa = \pm 0.1$	2.40	1.60	4	3.2	3.0	4
$\kappa = \pm 0.2$	1.67	1.75	2	2.3	3.5	4

Table 8.1: Parameters for curvature tuned line and edge operators. All operators used have the same normal parameters $\sigma_n = \sqrt{2}$ and $\epsilon = \sqrt{2}/2$, thus restricting responses to a spatial region within the radius of a square pixel around the curve. The parameters which are varied for curvature tuning are the tangential extent σ_t, number of tangential regions n, and the relative weights w of excitatory (+) and inhibitory (−) components.



(a) (b) (c)

Figure 8.5: The surfaces shown here are the responses of the curved line operators as we systematically vary model position in the total space E_C of our image curve model C. These are obtained by generating ideal model curves (positive contrast circles) parameterized around the model points shown as axes of the graphs. The responses are then obtained by simply computing the L/L response to the operator examined. The responses are organized in columns with (a) $\kappa = -0.2$, (b) $\kappa = -0.1$, and (c) $\kappa = 0.0$. Because of the equivalence of the operators under rotation and translation, we examine only the operator response at the origin and zero orientation.

We examine this by producing a series of maps of cross-sections through the operator's sensitivity map. The relevant maps for each of the bright line operators described in Table 8.1 are shown in Fig. 8.5. The non-zero responses in these maps represent ideal curves to which the operator responds—these non-zero responses should cover the Voronoi cell for which the operator is tuned. We can see that as long as spatial sampling has radius less than ≈ 1 pixel, orientation sampling has radius less than $\approx 15^\circ$, and curvature sampling has radius less than ≈ 0.15, then these operators will cover the total space.

8.3 Relaxation

We can see from both the idealized probe stimuli and the empirical tests with real images that the L/L operators do not produce fully consistent thick traces of the image curves. The response maps for even ideal curves are not perfect matches for the Voronoi cells. Significantly too, the operators are not entirely insensitive to noise and other variations from the ideal, which cause both gaps and extraneous, noisy responses. To separate the signal from the noise and fill in these gaps, we use the local geometric information which the responses represent to construct smooth, connected thick traces. This means developing a relaxation labelling network which verifies membership in such a trace, using the confidences from the L/L operators as a starting point, and relaxing to equilibrium.

Following the analysis in §7.3 we use the asymmetric compatibility to define relaxation support. As we can see from Fig. 8.6b, this is accomplished by choosing e_i^* as the perpendicular projection of e_i onto the circle generated by e_j. Assume for a moment that $e_j = (0, 0, 0, \kappa_j)$, and $e_i = (x_i, y_i, \theta_i, \kappa_i)$ then we can conclude that[1]

$$\phi = \arg(x_i, y_i - 1/\kappa_j).$$

[1]The function $\arg(x, y)$ is the angle of the ray from the origin to (x, y).

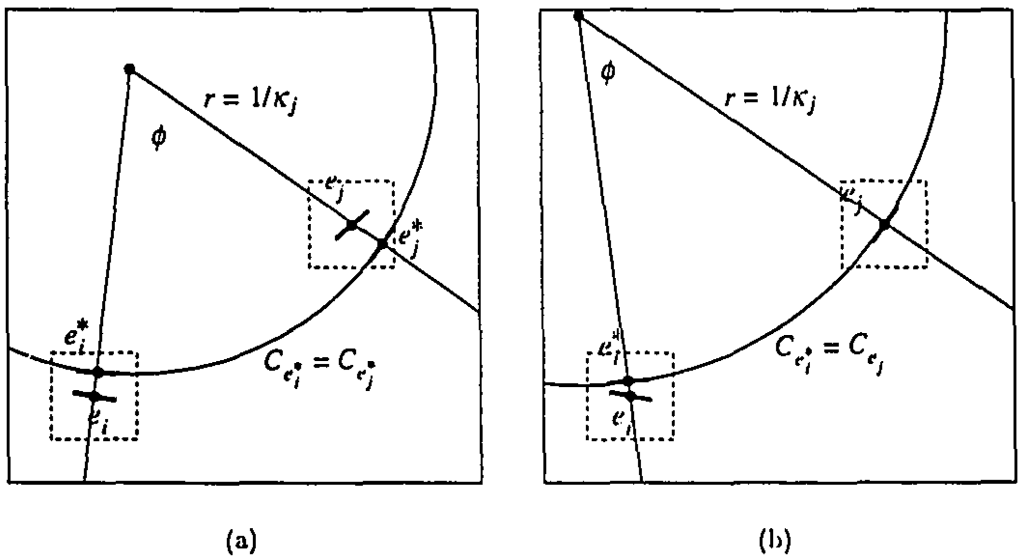


Figure 8.6: The calculation of compatibility for the image curve model C starts with the solution to the minimization problem in eq. (7.2). In (a) is shown the geometric solution of this problem. In (b) is shown the geometric solution of the related asymmetric problem, which is solvable algebraicly (see eq. (8.1)). The asymmetric compatibility is the basis for the support calculations.

then

$$e_i^* = \begin{pmatrix} \sin(\phi)/\kappa_j \\ (1-\cos(\phi))/\kappa_j \\ \phi+\pi/2 \\ \kappa_j \end{pmatrix},$$

unless $\kappa_j = 0$, in which case it becomes

$$e_i^* = (x_i, 0, 0, 0).$$

Thus in the general case $e_j = (x_j, y_j, \theta_j, \kappa_j)$

$$\phi = \arg(\mathrm{T}_{x_j,y_j}\mathrm{R}_{\theta_j}(x_i, y_i) - (0, 1/\kappa_j))$$

$$e_i^* = \mathrm{T}_{x_j,y_j}\,\mathrm{R}_{\theta_j} \begin{pmatrix} \sin(\phi)/\kappa_j \\ (1-\cos(\phi))/\kappa_j \\ \phi+\pi/2 \\ \kappa_j \end{pmatrix}. \tag{8.1}$$

Note that this perpendicular projection (the tubular map) is unique for every point in the total space except the center of the circle. Thus the tubular neighbourhood is total except for the singularity.

From this, we can immediately define the structure of the geometric support around the point $e_i = (x_i, y_i, \theta_i, \kappa_i)$ in the sampled total space.

$$d_C^c(e_i, e_j) = |e_i^* - e_i|,$$
$$d_C^l(e_i, e_j) = |\phi/\kappa_j|.$$

The basis functions for the tubular neighbourhood γ_i are most naturally:

$$\gamma_1 = (-\sin\theta_i, cos\theta_i, 0, 0)$$

$$\gamma_2 = (0, 0, 1, 0)$$

$$\gamma_3 = (0, 0, 0, 1)$$

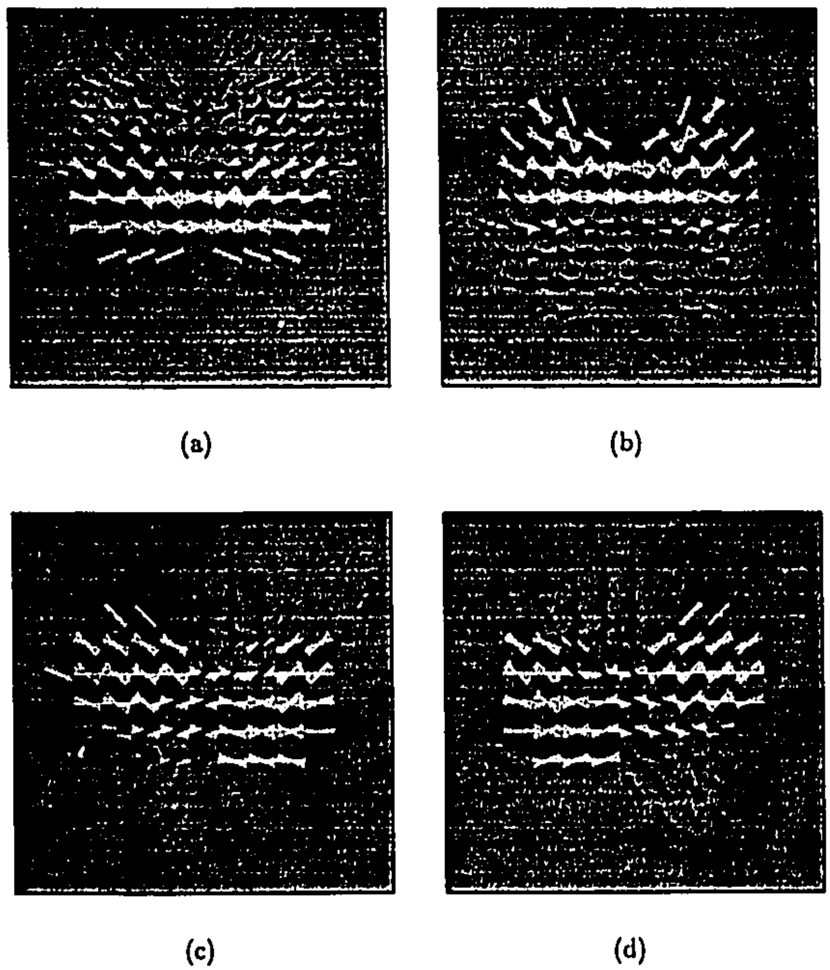
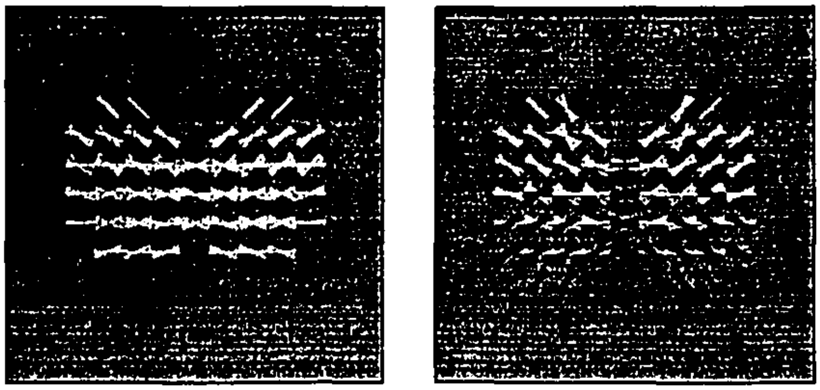


Figure 8.7: Shown are the lateral inhibition component networks of the full support network for a curve label with $\theta_i = 0.0^\circ$ and $\kappa_i = 0.1$. The interactions shown are (a) r^1_{ij}, (b) r^2_{ij}, (c) r^3_{ij}, (d) r^4_{ij}. Note that (a) and (b) together select local maxima in position, (c) and (d) in orientation.



(a) (b)

Figure 8.7: *(continued)* Shown are the lateral inhibition component networks of the full support network for a curve label with $\theta_i = 0.0^\circ$ and $\kappa_i = 0.1$. The interactions shown are (e) r^5_{ij}, and (f) r^6_{ij}. Note that (e) and (f) together select local maxima in curvature.

Thus we have six components which make up the support around e_i:

$$\begin{aligned} r^1_{ij} &= G^j_{\gamma_1}(e_i - \rho\gamma_1), & r^2_{ij} &= -G^j_{\gamma_1}(e_i + \rho\gamma_1), \\ r^3_{ij} &= G^j_{\theta_i}(e_i - \rho\gamma_2), & r^4_{ij} &= -G^j_{\theta_i}(e_i + \rho\gamma_2), \\ r^5_{ij} &= G^j_{\kappa_i}(e_i - \rho\gamma_3), & r^6_{ij} &= -G^j_{\kappa_i}(e_i + \rho\gamma_3), \end{aligned}$$

where

$$\mathrm{G}^j(e_i) = \mathrm{G}_{\sigma_t}(d^t_C(e_i, e_j))\ \mathrm{G}_{\sigma_c}(d^c_C(e_i, e_j)).$$

From §7.3 we have the constraints $\sigma_t \geq \sigma_c \geq \rho > 0.5$ and $\sigma_c < 1$. We use the values $\sigma_c = \rho = \sqrt{2}/2$ and $\sigma_t = 2.5$. The support components calculated with these values are shown in Fig. 8.9.

The boundary stability partition is straightforward. Since the model is one-dimensional, we can use the partition developed for lines

$$\begin{aligned} \zeta &= (\cos\theta_i, \sin\theta_i, 0, 0), \\ s(e_i, e_j) &= \zeta \cdot (e_j - e_i), \end{aligned}$$

$$q_1(e_i, e_j) = \sigma(s(e_i, e_j)),$$
$$q_2(e_i, e_j) = \sigma(-s(e_i, e_j)).$$

In the same way that the end-line extensions described in §5.2, increased the reliability of the initial operators, we may increase the specificity of the support if we partition into more than two regions and then combine using a stronger combination condition. Partitioning the support network into more than two regions can be achieved by the same means as for the curve operators, since the curve model is parameterized by arc-length $s(e_i, e_j)$. If $(s_1, \ldots, s_n)$ is an increasing sequence of partition points, then we can partition by the characteristic functions q_l:

$$q_1(e_i, e_j) = \sigma(s_1 - s(e_i, e_j))$$
$$q_l(e_i, e_j) = \sigma(s_l - s(e_i, e_j)) + \sigma(s(e_i, e_j) - s_{l-1}) - 1$$
$$q_n(e_i, e_j) = \sigma(s(e_i, e_j) - s_{n-1}).$$

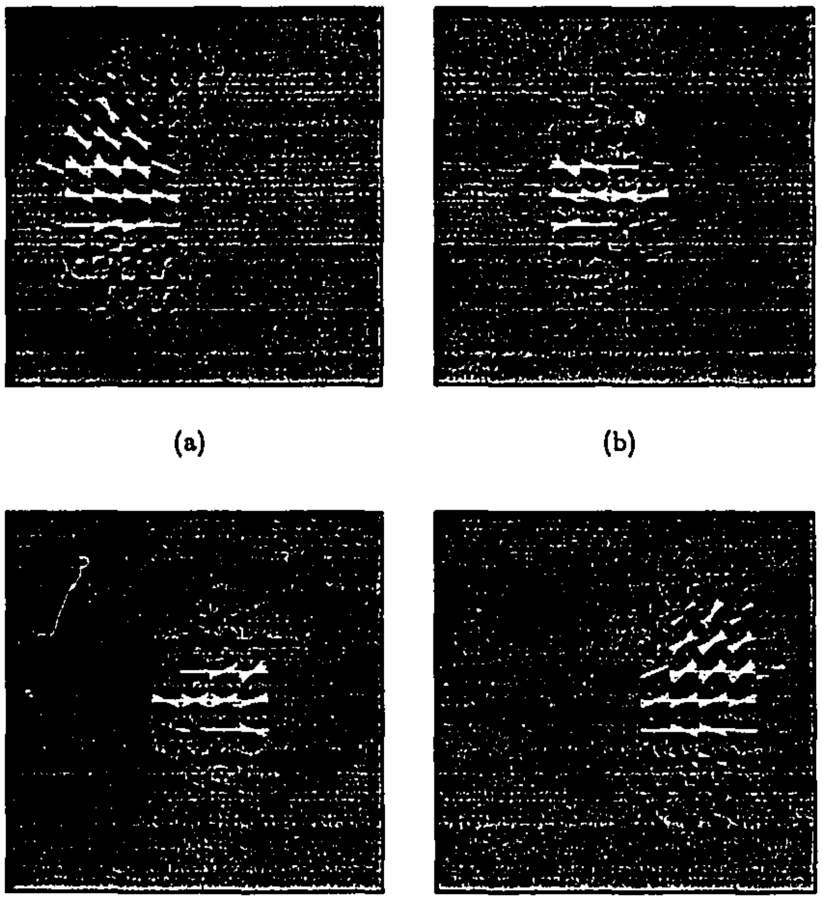
such that

$$\int_{E(\mathcal{L})} G^j(e_i)\, q_l(e_i, e_j)\, de_j = 1/n.$$

The tangential components of this support network are then

$$S_i^l = \bigwedge_k \sum_j p_j\, r_{ij}^{lk}.$$

To combine these component responses we could adopt either the simple or the majority combination from §5.2. In this case, one of the goals of the relaxation is to interpolate between nearby compatible curves. The "simple" combination rule has a strong veto for the central regions, so it is unlikely to achieve this interpolation. The "majority" rule is more lenient, but complicated to implement in terms of the basic L/L combinators. Instead we seek an L/L combination which will interpolate positive support into gaps only when there is no local negative support. Of course, this combination must also be end-line stable as outlined in §5.2. The solution we adopt is derived from an extension of the principle of "surround" introduced in §7.3.1.



(a) (b)

(c) (d)

Figure 8.8: Decomposition into regions of the support network for a curve label with $\theta_i = 0.0^\circ$ and $\kappa_i = 0.1$. Shown are the linear reductions of the networks for four regions divided such that the ideal support from each region is equal.

We start with an analogy. Imagine you are standing at a point and must determine whether you are surrounded by gunmen. You look around. If the first person you see in all directions has a gun, then you are surrounded. Any unarmed person between you and the gunmen is evidence that you may not be surrounded. Applying this principle to the partition of the support network above, "all directions" simply means $s < 0$ and $s > 0$, while "the first person" in each direction is the first *unambiguous* response from one of the tangential support components! To embody this principle then, the "surround" combinator should combine responses by selecting the nearest unambiguous response and then adding to it those responses from further regions which agree in sign. So if the nearest unambiguous is positive then we add together all positive responses, and vice versa.

This process can be formalized by using the ρ-approximate L/L combinators with $\rho < \infty$. In that case, the response x is ambiguous when $x \in [-1/2\rho, 1/2\rho]$, or when

$$\sigma_\rho(x)\,\sigma_\rho(-x) \;>\; 0.$$

The responses x and y have the same sign when

$$\sigma_\rho(x)\,\sigma_\rho(y) > 0 \quad \text{or} \quad \sigma_\rho(-x)\,\sigma_\rho(-y) > 0.$$

Definition 8.2 For four regions $\{S_i^1, S_i^2, S_i^3, S_i^4\}$ the *surround* combination is given by

$$S_i \;=\; S_i^- \mathbin{\Delta} S_i^+,$$

where

$$\begin{aligned}
S_i^- &= S_i^2 + S_i^1 \left(\sigma_\rho(S_i^2)\,\sigma_\rho(S_i^1) + \sigma_\rho(-S_i^2)\,\sigma_\rho(-S_i^1) + 2\sigma_\rho(S_i^2)\,\sigma_\rho(-S_i^2)\right)\\
S_i^+ &= S_i^3 + S_i^4 \left(\sigma_\rho(S_i^3)\,\sigma_\rho(S_i^4) + \sigma_\rho(-S_i^3)\,\sigma_\rho(-S_i^4) + 2\sigma_\rho(S_i^3)\,\sigma_\rho(-S_i^3)\right).
\end{aligned}$$

Thus if $S_i^2 = 0$, then $S_i^- = S_i^1$. For $\rho = \infty$, we have

$$S_i^- = S_i^2 \text{ and } S_i^+ = S_i^3$$
$$\text{therefore } S_i = S_i^2 \mathbin{\Delta} S_i^3.$$

So it is fair to consider this to be a kind of Logical/Linear combinator. It is equivalent to a simple Boolean combination of the signs of the inputs and the output is always a linear combination of the *unambiguous* component responses. It is, however, only well-defined for $\rho < \infty$.

The identities above also clearly show that this combination will be end-line stable, since the combination $S_i^2 \mathbin{\Delta} S_i^3$ would be. Thus for this combination of support, the point e_i will receive positive support as long as there are unambiguous support regions surrounding the point e_i. Thus, the surround combinator interpolates into gaps only when surrounded and when local supports do not contradict the interpolation. This is the behaviour we set out to design above.

8.4 Results

We will reserve most of the comments on specific results to the figure captions, and concentrate only on general points. Both the line and edge compatibilities were calculated with the same set of parameters. Curvature was sampled into five classes with values $\{-0.2, -0.1, 0.0, 0.1, 0.2\}$, and direction was sampled into either 8 (lines) or 16 (edges) discrete direction classes. In both cases, the difference between adjacent directions was 22.5°. The lateral components of the support were implemented with $\sigma_c = \epsilon = \sqrt{2}/2$, normalized to the distance metric. The tangential extent was determined by setting $\sigma_t = 2.5$ and dividing into four regions for end-line stability. The compatibilities were normalized so that the maximum possible support for a label is 1 and only those compatibilities greater than 5% of maximum were used. The relaxation was performed with a step-size of $\delta = 1$. Finally, the initial measurements used the ρ-approximate L/L operators with $\rho = 16$, and the relaxation used the adaptive

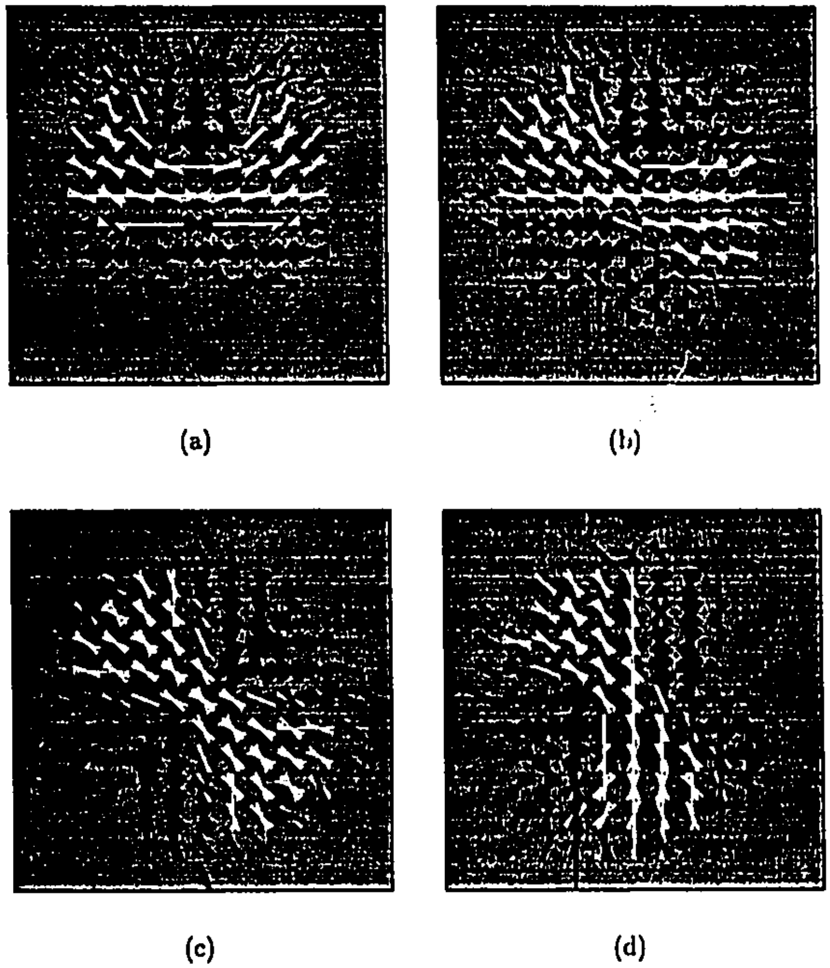


Figure 8.9: A selection of the support networks for the image curve model C. The networks shown are the linear reductions of the full support networks for a sampling with 8 orientations and 5 curvatures $\{-0.2, -0.1, 0.0, 0.1, 0.2\}$. Shown are the networks supporting the labels (a) $\theta_i = 0^\circ$ and $\kappa_i = -0.2$, (b) $\theta_i = 22.5^\circ$ and $\kappa_i = -0.1$, (c) $\theta_i = 45^\circ$ and $\kappa_i = 0.0$, (d) $\theta_i = 67.5^\circ$ and $\kappa_i = 0.1$.

ρ-approximates with $\rho' = 4$.

To this point, we have not mentioned implementation at all. The system itself was developed using a general image processing package developed by the author for the MasPar MP-1. A SIMD parallel machine, the MasPar has between 1024 and 16384 4-bit processors arranged in a planar array. Since all of the computations in this system are either pointwise or involve only local communication, it was straightforward to map the processing elements to both L/L operators and individual nodes in the relaxation. The resulting system will run on both the MasPar and uniprocessor machines.

In examining the results we observe a number of general principles:

- In no case is structure "created" by the relaxation. The only interpolation performed is *into* regions surrounded by consistent structure. Thus ***small gaps in curves are filled in***, but large gaps result in distinct traces.

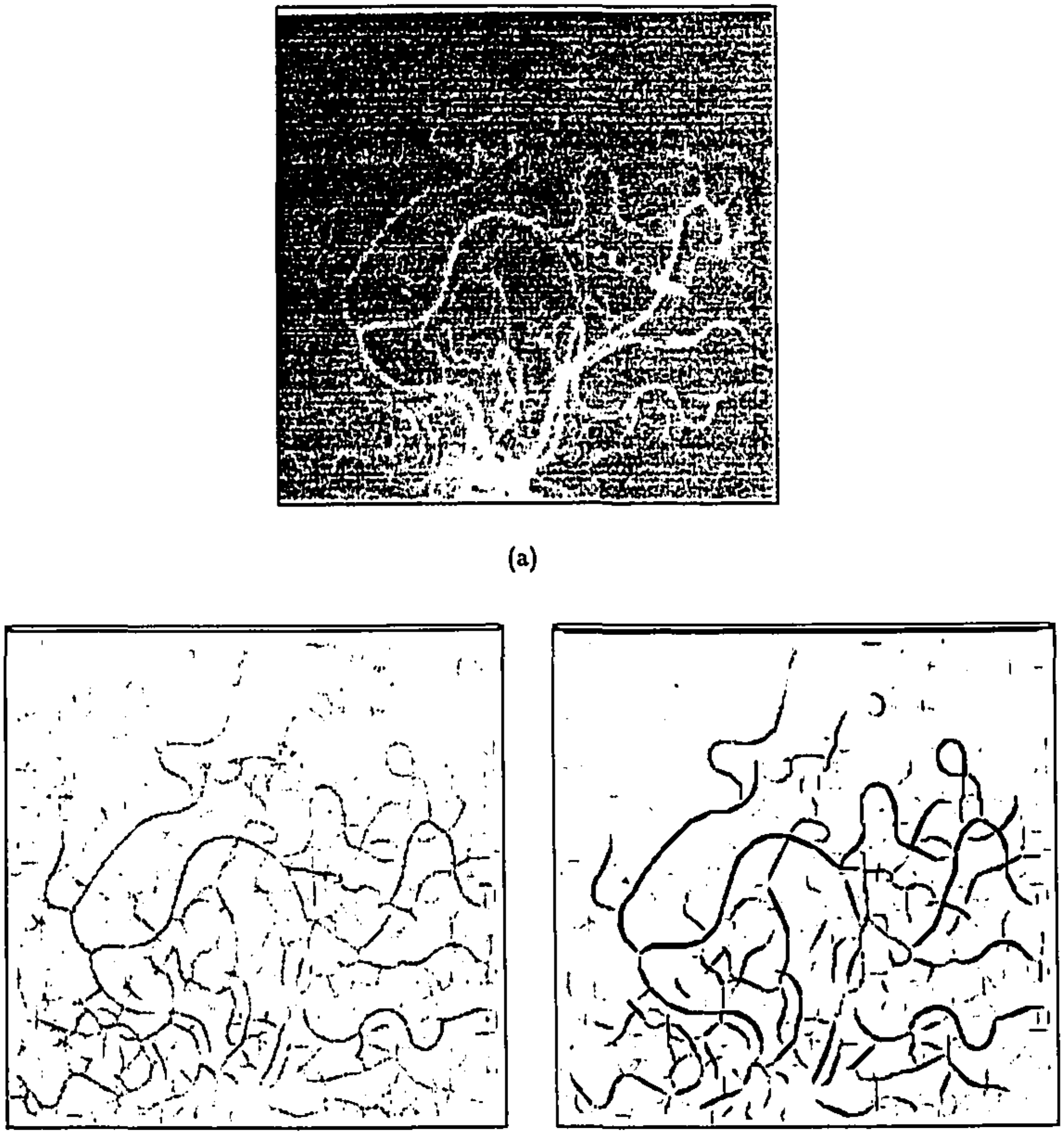
- The end-points of curves and lines are apparently stable, even when a number of independent curves have coincident end-points. These *coincident endpoints* are the building blocks for much higher level processing as they signal the presence of *corners, and junctions.*

- *Curves do not interfere with each other when they cross.* This independence is a function of their relative orientations at the crossing point. In general, two crossing curves will not interfere as long as the difference between their orientations is greater than the difference between adjacent orientations in the sampling. With orientation sampled at 22.5° intervals there will be no interference for incident angles greater than 45°. A finer sampling will decrease this threshold.

- In certain cases, *bifurcations* in curves are clearly described at a point by a single trace approaching from one side and multiple traces approaching from the other.

- Isolated points are not eliminated, but their confidences remain uniformly low, since they receive positive support only from themselves. Thus they may be "seen" or noticed only at the boundaries of the system's own discriminability.

- There is no need to adapt the initial operators or the relaxation compatibilities to different situations. The images used vary greatly in contrast and noisiness, yet the results are uniformly good. This is not to say that there is no need for adaptability or contrast renormalization in the initial operator responses—their responses may improve from these enhancements. However, note that the relaxation network is completely independent of the actual image, depending only on the geometry of curves and the representation we have chosen for that geometry. So the only thing that such changes might effect would be the quality of the starting point for the relaxation.

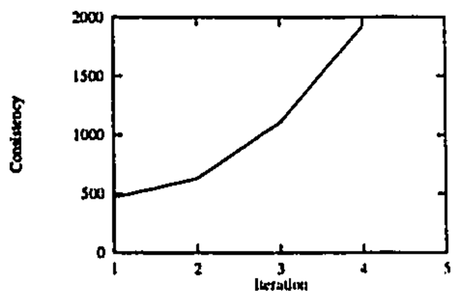


(a)

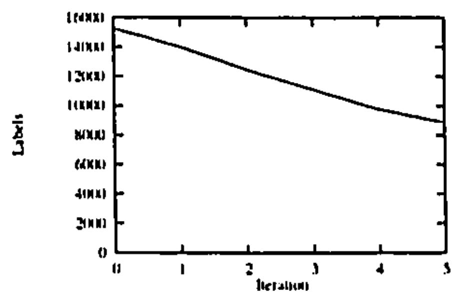
(b)

(c)

Figure 8.10: One of the basic tasks in interpreting a cerebral angiogram (a) is to recognize and describe the blood vessels, which show up as bright lines. The images are typically noisy and of low contrast. Shown are the results of applying initial bright line operators (b), followed by 5 iterations of geometric relaxation (c). The darkness of the lines displayed is proportional to the label confidence. There are a number of features to note here: the sharpening of corners, the enhancement of long faint curves, and the filling in of short gaps.



(a)



(b)

Figure 8.11: Two graphs showing the progress of the angiogram relaxation through five iterations. In (a) is shown the total consistency of the labelling after each iteration. We see that increases uniformly and quickly. In (b) is shown the number of labels with non-zero confidence after each iteration. Clearly the relaxation is quite selective, eliminating half of the labels after 5 iterations.

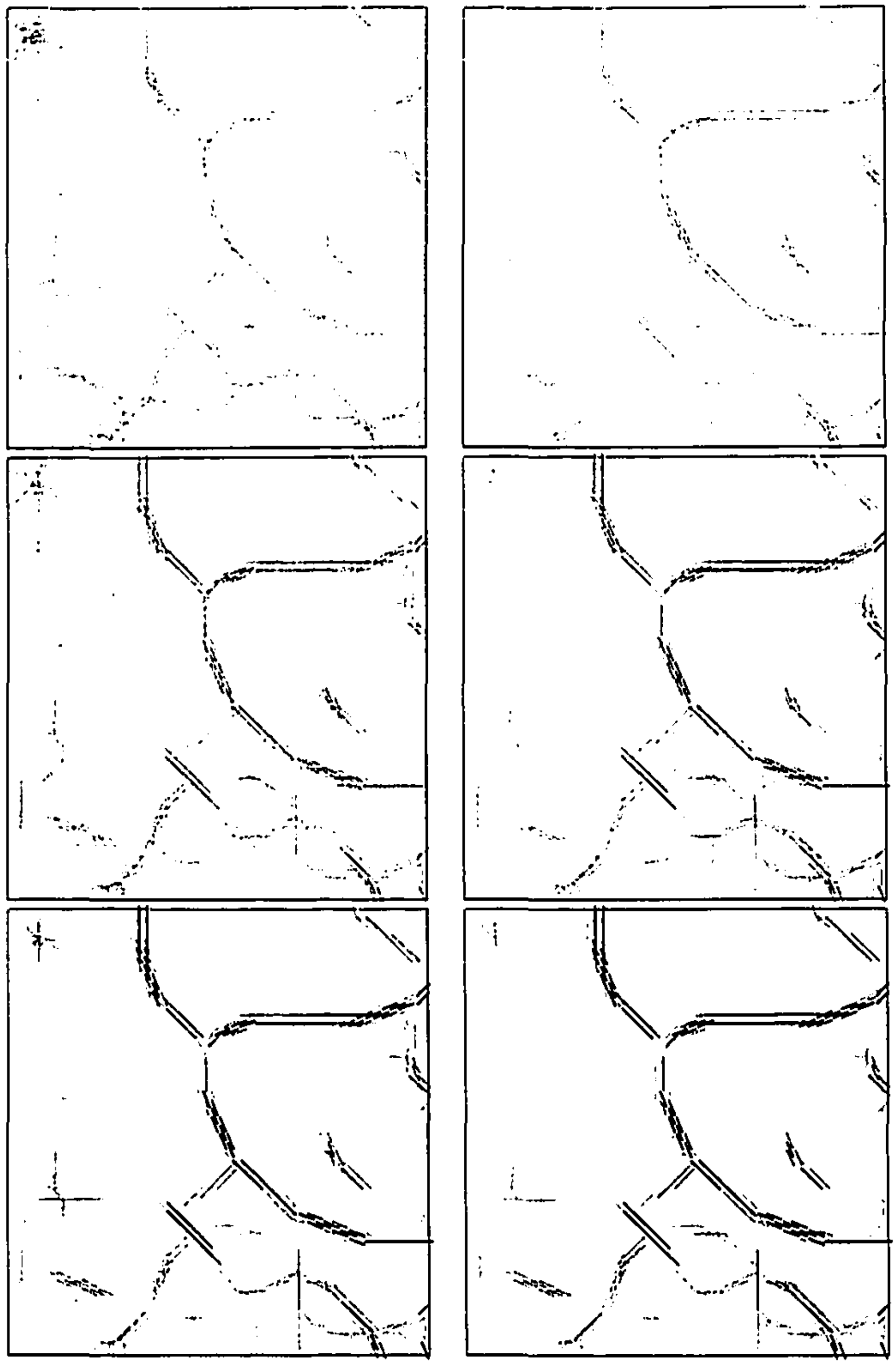
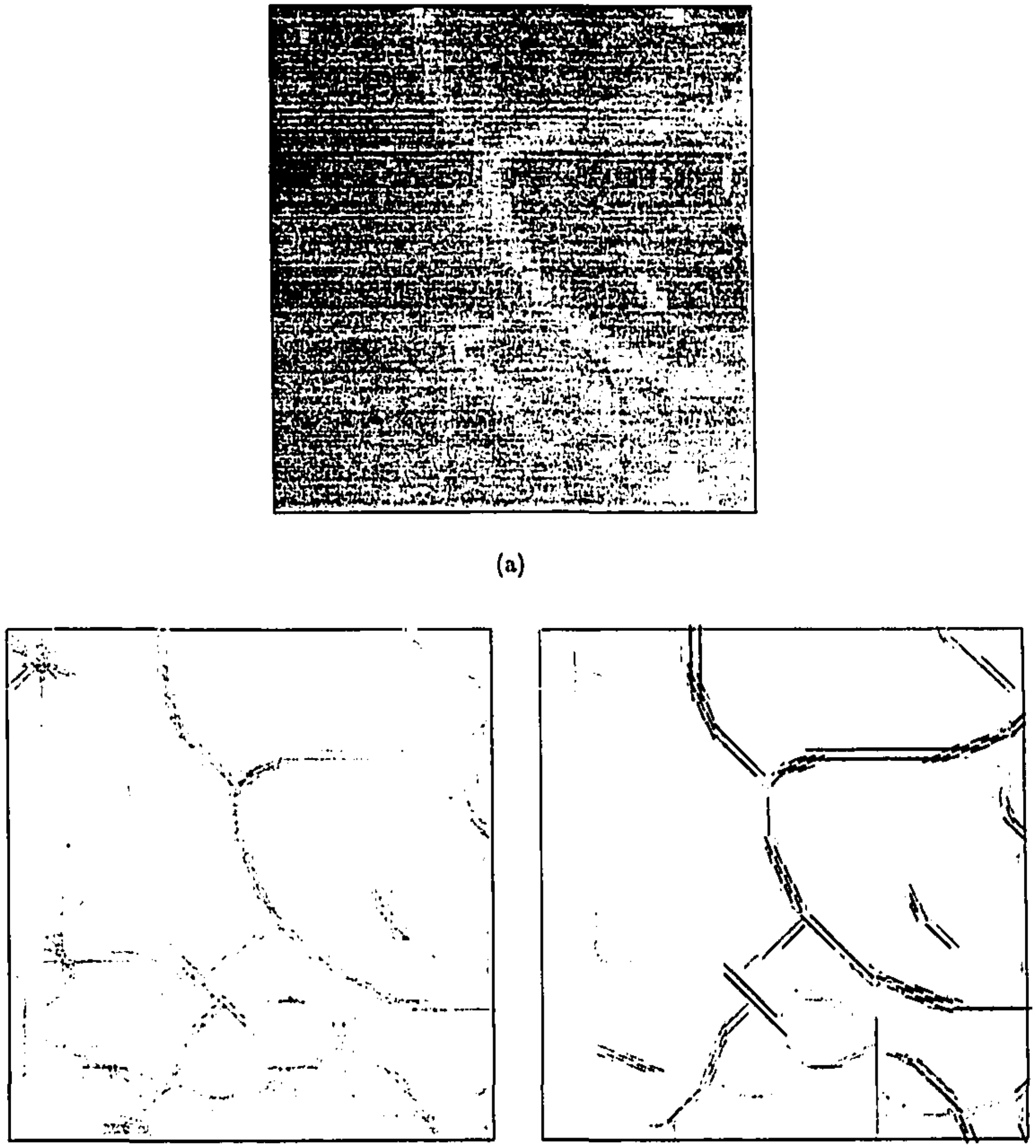


Figure 8.12: The state of the relaxation of the bright lines in the cerebral angiogram after initial operators (top left), and through five iterations (across and down).

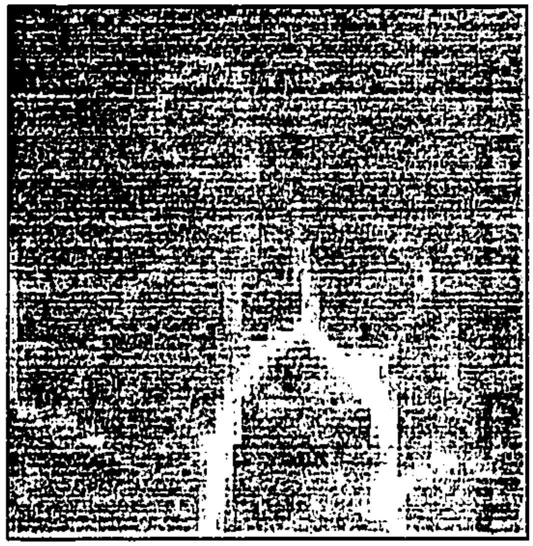


(a)

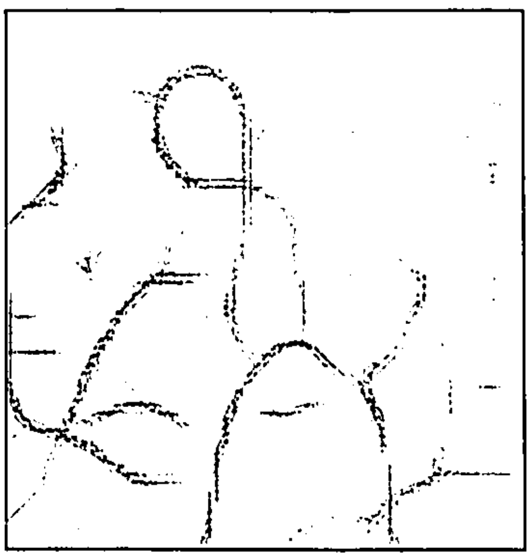
(b)

(c)

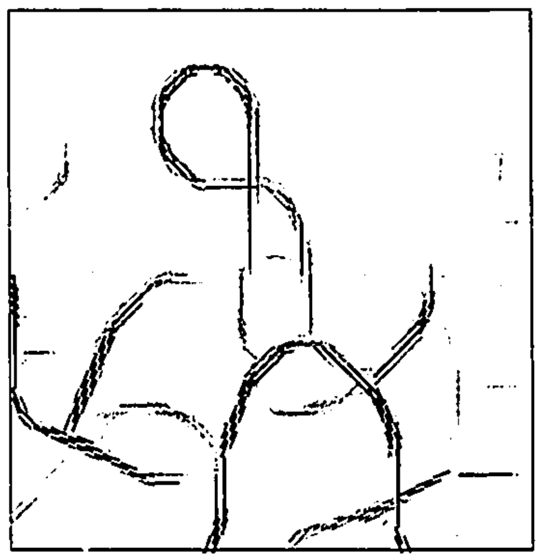
Figure 8.13: A detail of the angiogram showing an area from the lower left (a), initial operator responses (b), and the results after 5 iterations (c). Note the removal of the "hair" around line endings, the increase in accuracy of the local curvature estimates (shown by the curvature of the segments), the stability of junctions and corners, and the filling in of short gaps.



(a)

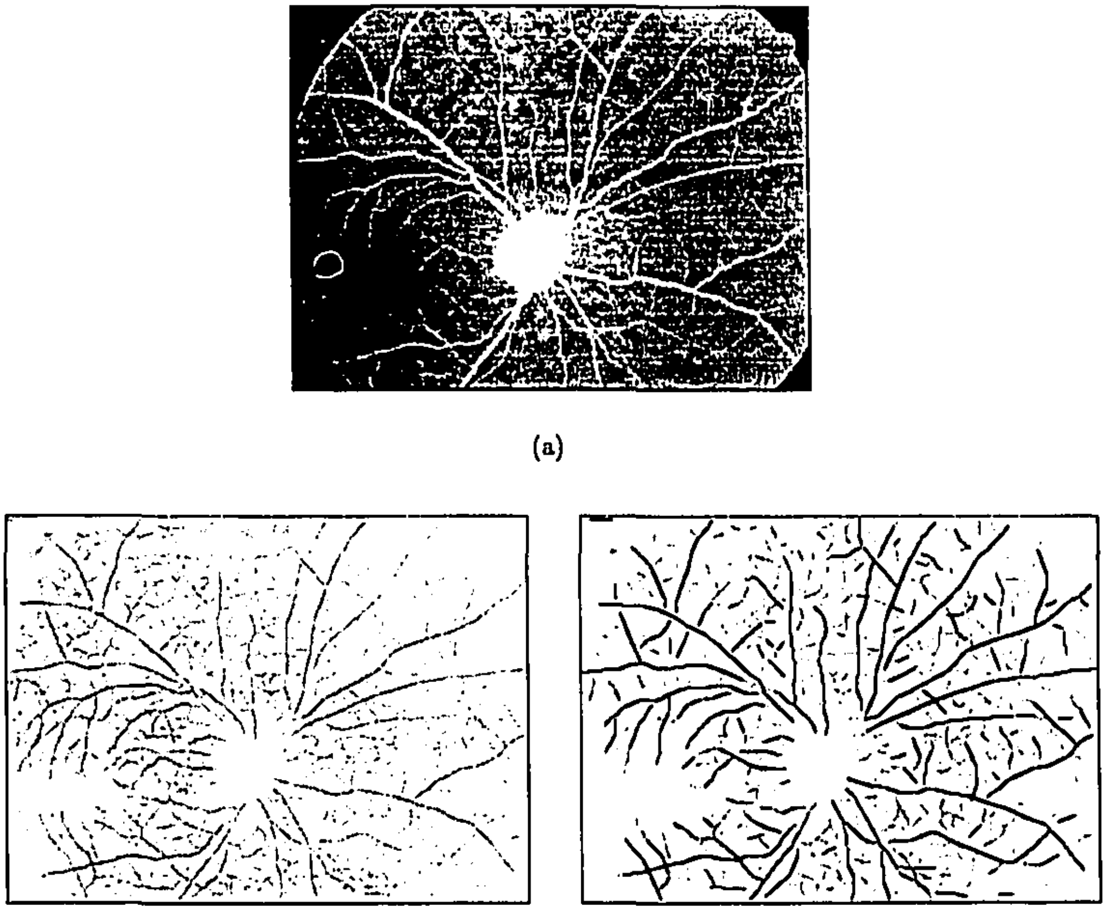


(b)



(c)

Figure 8.14: A detail of the angiogram showing a loop in the upper right (a), initial operator responses (b), and the results after 5 iterations (c). Note the increasing accuracy of the curvature estimates, the stability of the crossings and end-points, the filling in of short gaps. Of particular interest is the clear description of the bifurcation which seems to occur in the lower part of the image.



(a)

(b)

(c)

Figure 8.15: Retinal microgram showing blood vessels on the retina (a), and the results of applying initial bright line operators (b), followed by 5 iterations of geometric relaxation (c). The major and most of the minor blood vessels show up clearly. The tree structure is readily apparent although in many places the branch points of the tree seem disconnected. Where these gaps are observed is exactly where the major vessels are more than two pixels wide, thus the minor vessels generally abut on the *boundary* of these vessels.

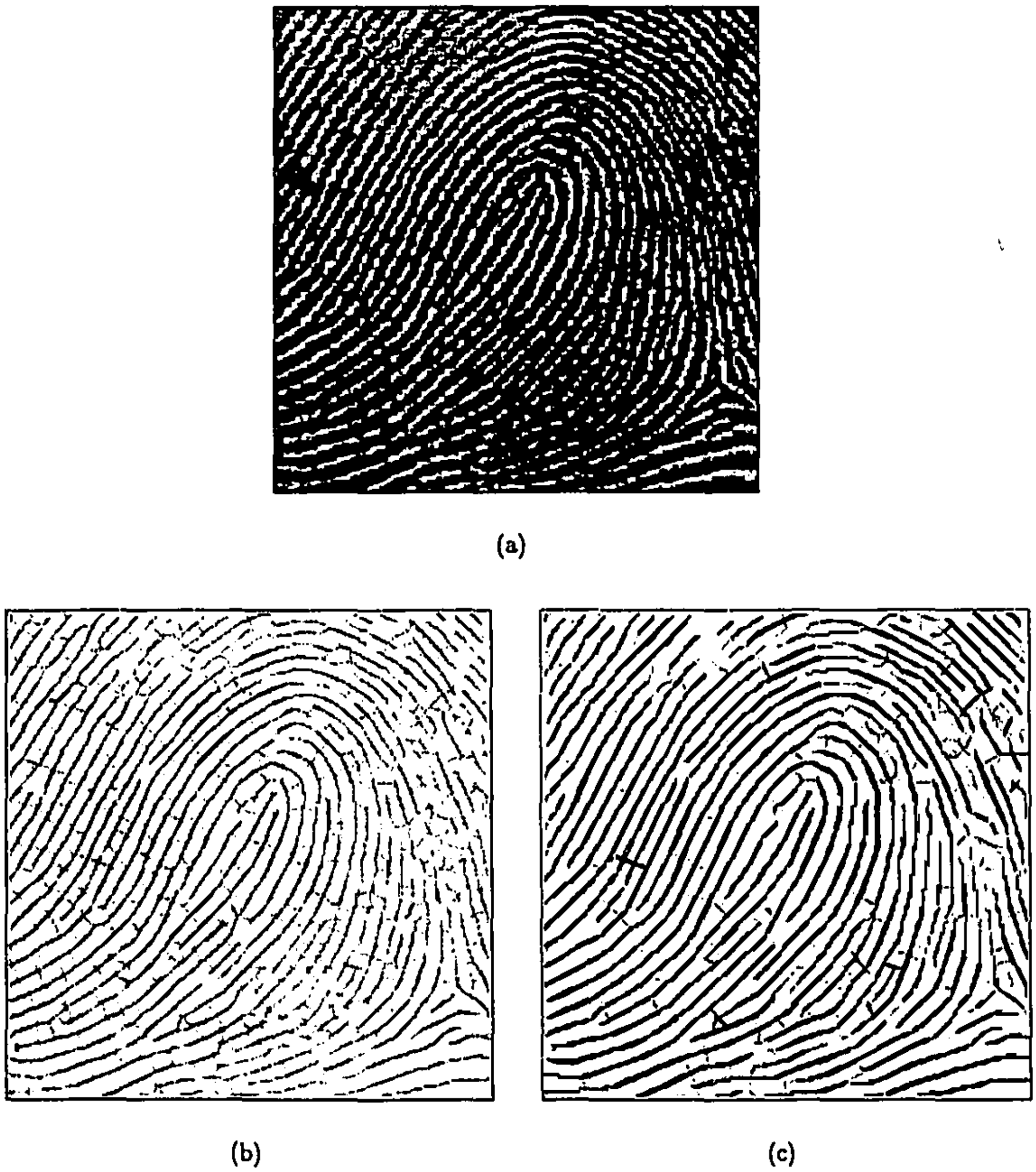


Figure 8.16: Fingerprint (a), and the results of applying initial bright line operators (b), followed by 5 iterations of geometric relaxation (c). There is virtually no interference between nearby curves even when they are parallel.

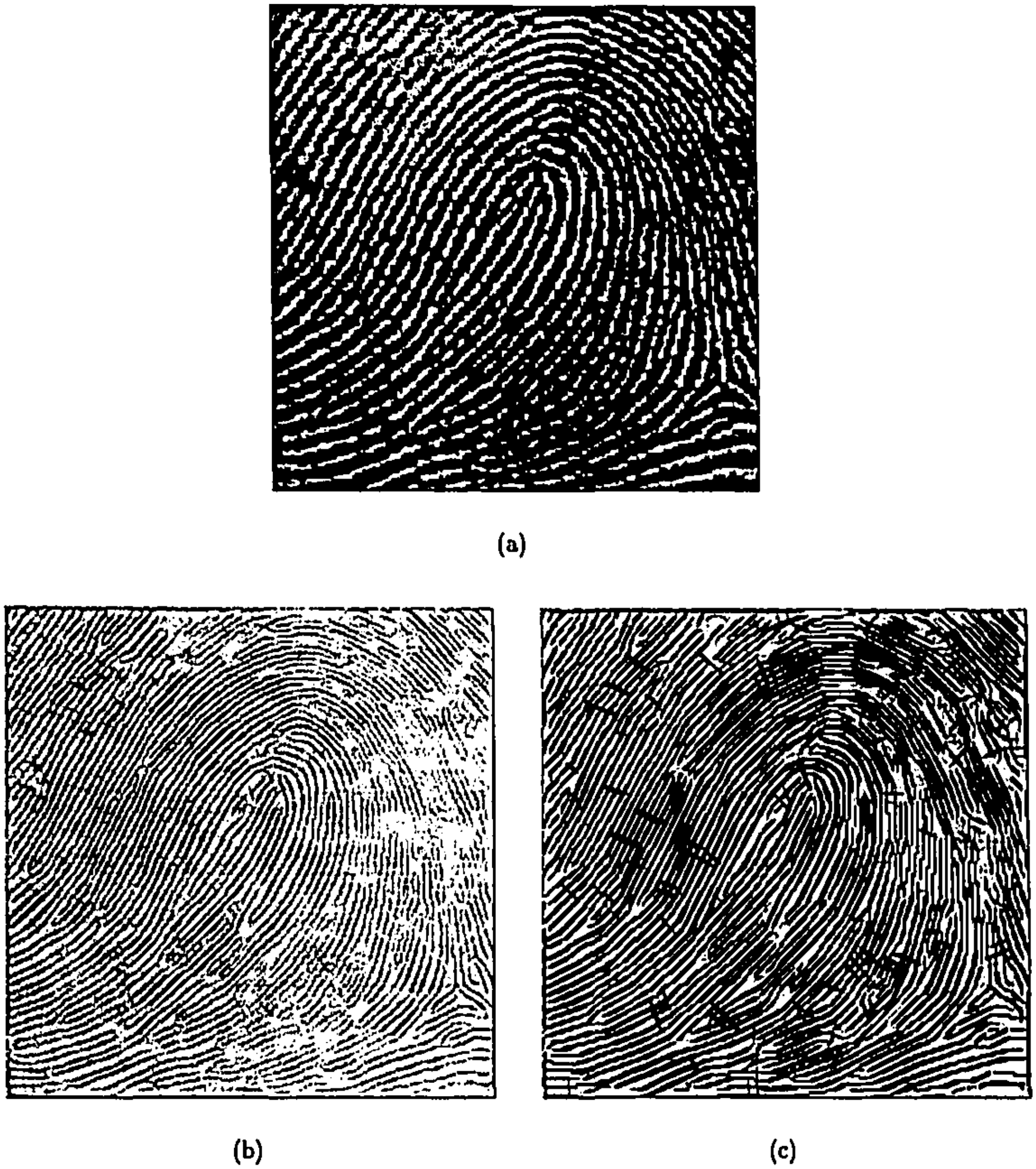
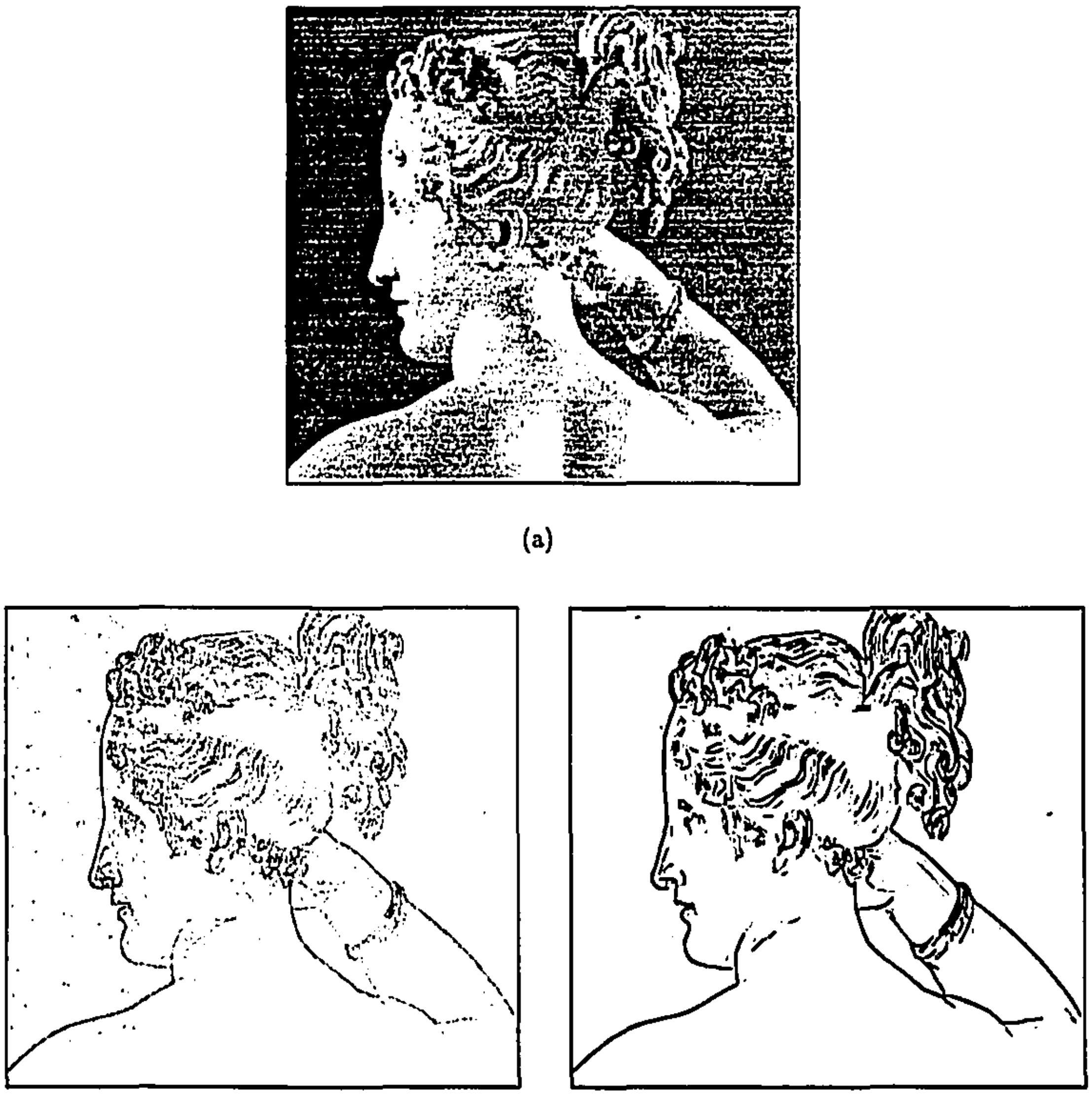


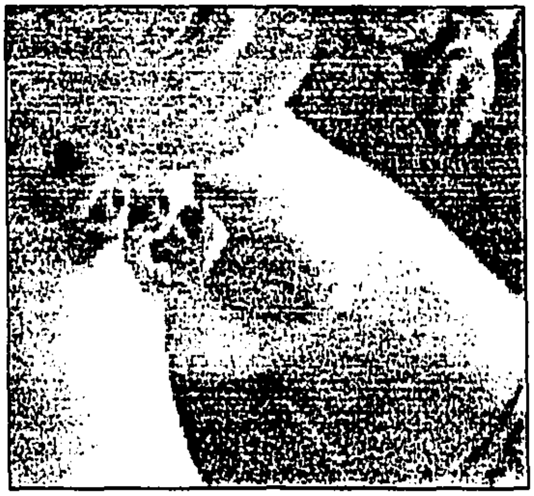
Figure 8.17: Fingerprint (a), and the results of applying initial edge operators (b), followed by 5 iterations of geometric relaxation (c). Note the correspondence between the edge terminations and discontinuities, and between bifurcation points. The flaws in the fingerprint also show up clearly. And again there seems to be little or no interference between nearby edges even though in some cases they are separated by as little as two or three pixels.



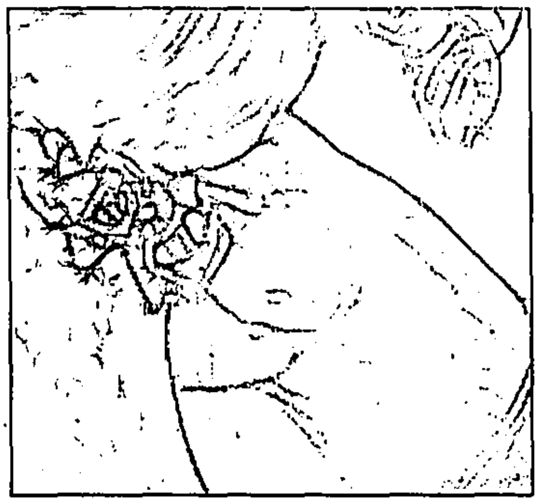
(a)

(b) (c)

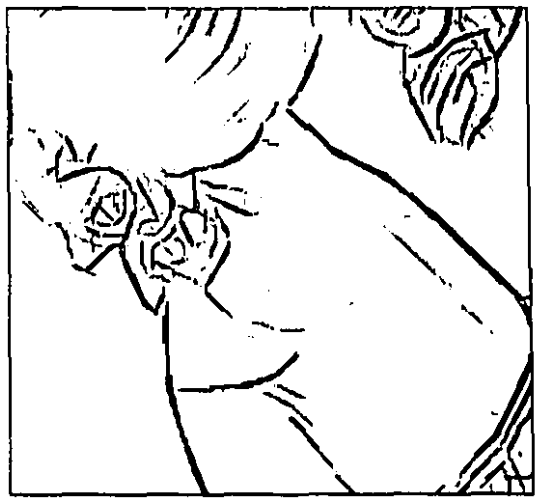
Figure 8.18: The statue image from Chap. 6 (a), and the results of applying initial edge operators (b), followed by 5 iterations of geometric relaxation (c). Since little detail is visible, note only the elimination of much background noise and the enhancement of the major bounding contours.



(a)

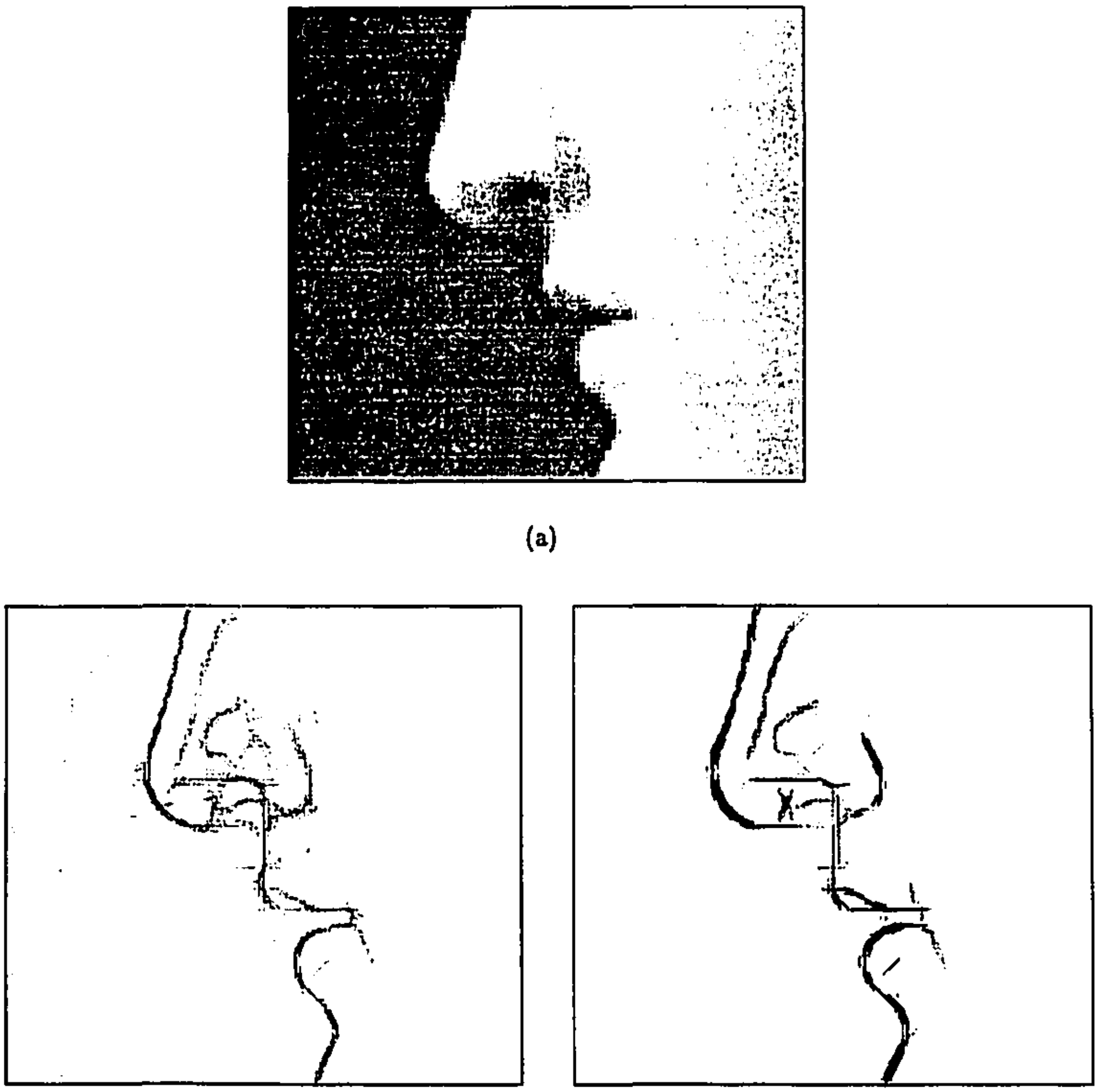


(b)



(c)

Figure 8.19: A detail of the statue image showing the area around the hand-neck occlusion (a), initial operator responses (b), and the results after 5 iterations of relaxation of the edge responses (c). Note the clear T-junction and corner where the hand is occluded by the neck and hair. Note also the ability of the relaxation to extract structure from very complicated regions like the fall of hair and the tuft at the nape of the neck.



(a) (b) (c)

Figure 8.20: A detail of the statue image showing an area from the lower left (a), initial operator responses (b), and the results after 5 iterations (c). Note the evolution of the local curvatures and the accuracy of the resulting sketch. In regions of such high curvature a curve smoothing system which minimized total curvature would likely displace the features significantly.

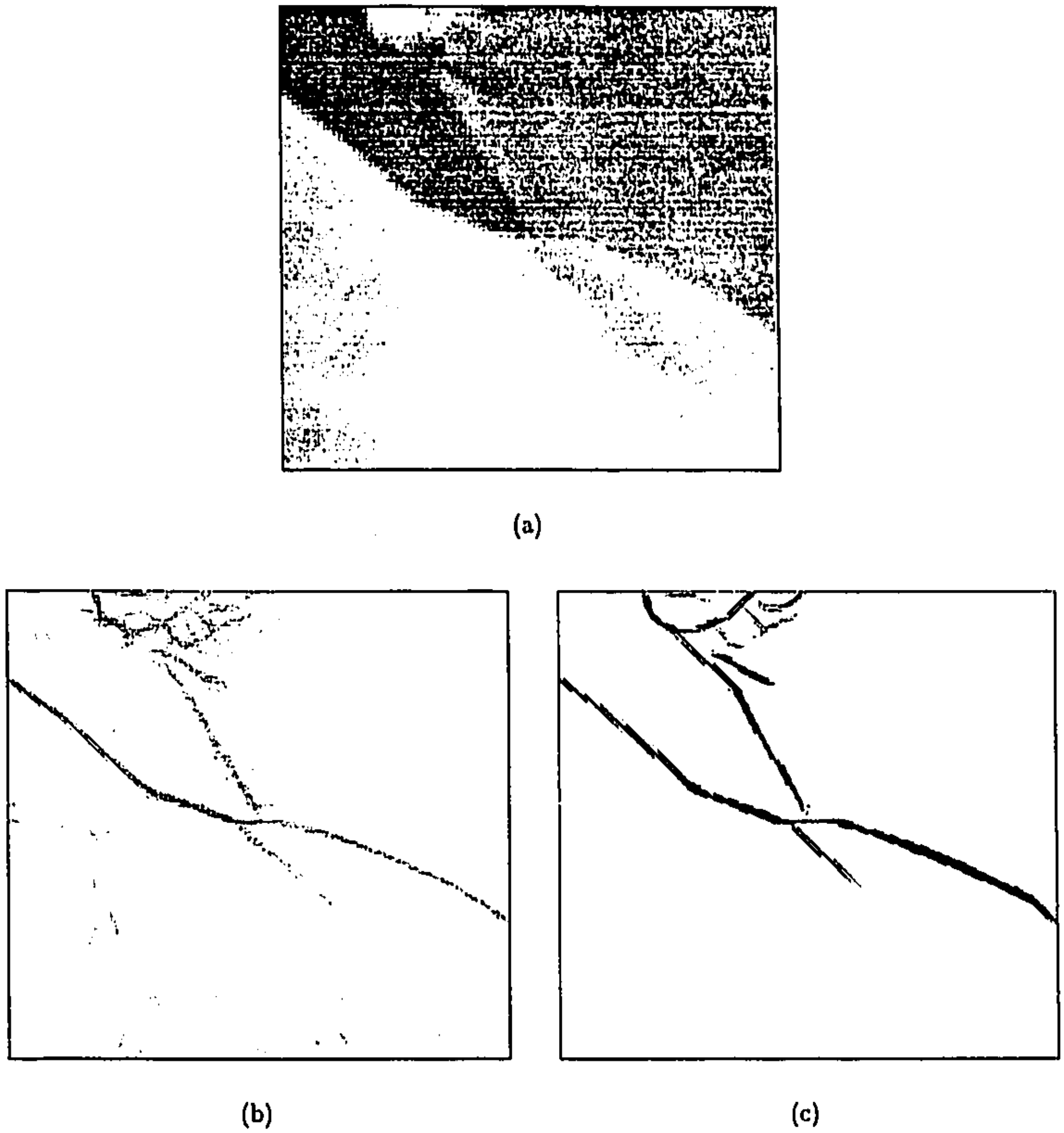


Figure 8.21: A detail of the statue image showing an area from the lower right (a), initial operator responses (b), and the results after 5 iterations (c). Note that within a two-pixel neighbourhood there is both a T-junction and apparent bifurcation. There seems to be little or no interference between them.

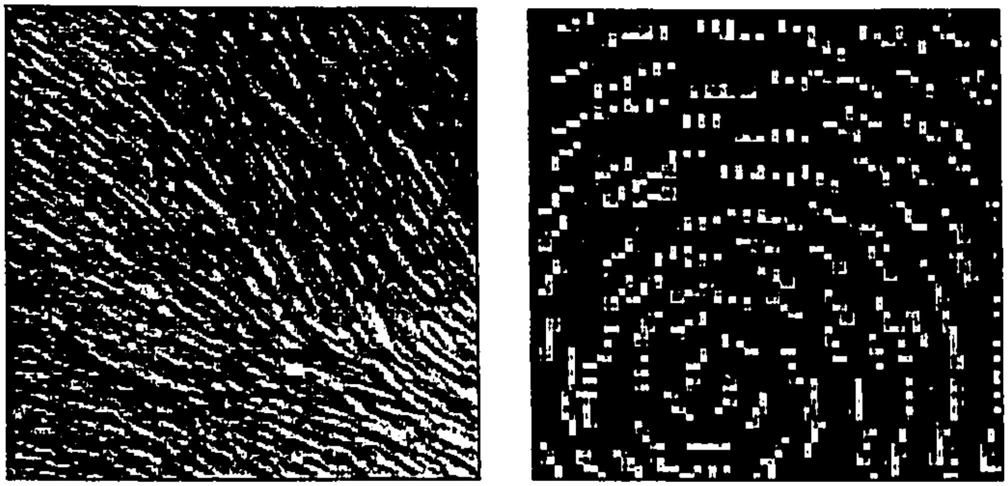
Chapter 9 Texture Flow

Texture flow is a term we use to describe a certain class of oriented, static textures. These textures consist of locally parallel, oriented elements which form a direction field in the image. Familiar examples of such textures are random dot moiré patterns (or Glass patterns [Gla73]) and hair patterns.

The perception of such textures has been extensively investigated, in both psychophysical [Gla73, GS76, LZ87, ZIH90] and computational realms [Ste78, Zuc84, KW87, RS91]. We will rely on two observations from the psychophysics to focus the development of the computations below. The first of these is that sparse orientation information can give rise to a dense texture flow percept, with implicit orientation perceived everywhere inside the field. This suggests that a great deal of interpolation is being performed. Because some of the effects observed (e.g. with moving, overlapping fields [ZIH90]) seem to indicate that very low-level features are implicitly constructed (i.e. as illusory features) within the interpolated areas—we suggest that this interpolation takes place early in the processing stream.

The second observation which we regard as significant has to do with the perception of discontinuities in these fields. In psychophysical experiments it was shown that, as with curves, the ability to reliably locate discontinuities in texture flow fields depends on the availability of *local curvature* information [LZ87]. Thus as with image curves, we conclude from this that curvature information is explicitly managed in the inference of texture flow fields. We will show below how local curvature can be defined in such a field.

The description of texture flow fields will therefore involve the computation of dense descriptions interpolated from potentially sparse initial orientation estimates. These estimates will be provided by L/L operators tuned for both orientation and local curvature. Furthermore, the interpolation process will allow multiple, transparent flows to coexist in a region and will stabilize the discontinuities and boundaries of the



(a) (b)

Figure 9.1: Two examples of texture flow: (a) a hair pattern taken from [Bro66], and (b) an artificial Glass pattern. These are both perceived as dense, locally parallel fields of oriented texture.

flow fields themselves. As we have shown, these goals can be achieved by implementing the interpolation as a relaxation labelling process designed to extract thick traces of smooth texture flows. We will show that the relaxation will interpolate a dense field from sparse inputs without arbitrarily smoothing over discontinuities. This chapter will cover the definition and implementation of this system.

Before continuing, it is important to point out a significant analogy. It has been suggested that the similarities between texture flow and optical flow may be significant [RS91]. In fact, a smooth texture flow may be modelled as the direction map of a smooth velocity field. Beyond this there are basic similarities in the perception of these phenomena which we feel expose similar styles of processing:

- Both flows give rise to dense percepts from sparse data;
- Both are stable with respect to transparency; and
- Both are stable at boundaries.

The last two of these are issues which are simply not considered by current theories of processing either texture flow [KWS7, RS91] or optical flow [HS81, Hee87]. A system which can produce accurate descriptions of piecewise smooth texture fields, such as the one described below, will require little modification to also work for optic flow.

9.1 Representation

We derive our model of texture flow $\mathcal{F}$ from a two-dimensional motion field by associating the direction of flow in the motion field with a static orientation in the texture.

The simplest such motion field is clearly just constant parallel flow. However, acceleration is a significant parameter in understanding motion flow (it is associated with rotation and looming), and as we shall see is related to the *curvature* of the static flow field. Thus we augment our flow model with curl and divergence terms, generating a local non-deforming field with constant curl and divergence. Such a field $F(x, y)$ on the plane is defined by the equation

$$F(x,y) = \begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} 1 - \kappa_u y + \kappa_v x \\ \kappa_u x + \kappa_v y \end{pmatrix}.$$

Differentiation will verify that the divergence of this field is $2\kappa_u$ and the curl is $2\kappa_v$. Orientation maps of this field for a number of values of κ_u and κ_v is shown in Fig. 9.2.

The choice of the symbols κ_u and κ_v is perhaps puzzling since κ is usually associated with curvature. We justify this choice by examining the field at the origin. Consider the local direction of the field

$$\theta(x,y) = \arg(f_x, f_y).$$

If we take u as the unit vector parallel to the field at the origin and v as the unit normal then

$$\kappa_u = \lim_{\epsilon \to 0} \frac{\theta(\epsilon u)}{\epsilon},$$

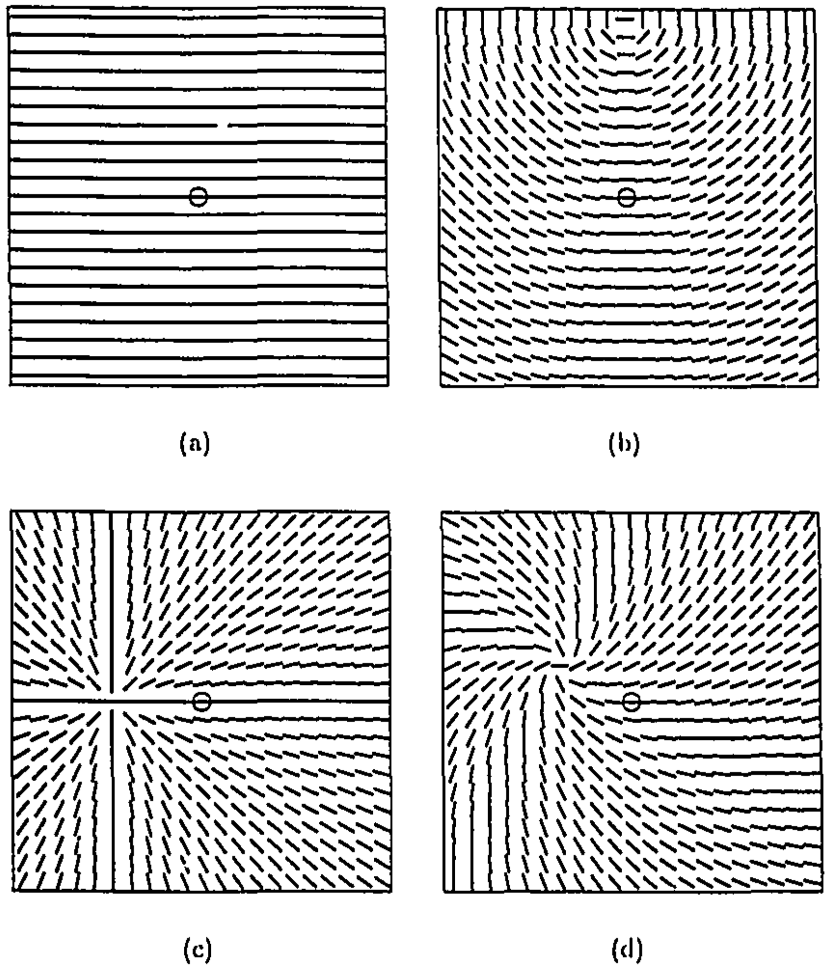


Figure 9.2: Four direction fields centered around the origin (circled). The fields are generated with $\theta = 0$ at the origin and (a) $\kappa_u = 0.0$ and $\kappa_v = 0.0$, (b) $\kappa_u = 0.1$ and $\kappa_v = 0.0$, (c) $\kappa_u = 0.0$ and $\kappa_v = 0.2$, and (d) $\kappa_u = 0.1$ and $\kappa_v = 0.2$. All fields are direction fields for velocity fields with constant curl and divergence.

$$\kappa_v = \lim_{\epsilon \to 0} \frac{\theta(\epsilon v)}{\epsilon}.$$

Thus κ_u is a measure of change of orientation tangent to the field and κ_v is the change of orientation normal to the field. Therefore we refer to κ_u as the *tangent curvature* and κ_v as the *normal curvature*.

We now have all of the building blocks needed to represent this texture flow field. Since this is a static image, our base space is again the image plane X. The fibre is now three-dimensional including direction and two curvatures $F = \Theta \times K_u \times K_v$. Thus a point in the total space $E(\mathcal{F})$ is $e_i = (x_i, y_i, \theta_i, \kappa_{ui}, \kappa_{vi})$. Sampling this fibre regularly can be done with the same samplings used in Chap. 8. For n discrete orientations, we sample at the points $\theta_i = i\pi/n$ for $i \in \{0, \ldots, n-1\}$. Both curvatures are sampled regularly over the range $K_u = K_v = [-\kappa_{\max}, \kappa_{\max}]$. With m curvatures we sample at $\kappa_j = \kappa_{\max}(-1 + 2j/(m-1))$ for $j \in \{0, \ldots, m-1\}$. The L_2 metrics for the components of this total space are then

$$\begin{aligned}
d(\theta_i, \theta_j) &= \frac{|\theta_i - \theta_j \operatorname{mod}_c \pi}{\pi/n} \\
d(\kappa_{ui}, \kappa_{uj}) &= \frac{|\kappa_{ui} - \kappa_{uj}|}{2\kappa_{\max}/(m_u - 1)} \\
d(\kappa_{vi}, \kappa_{vj}) &= \frac{|\kappa_{vi} - \kappa_{vj}|}{2\kappa_{\max}/(m_v - 1)}
\end{aligned}$$

and $d(e_i, e_j)$ is the L_2 combination of these metrics. The resulting Voronoi cells are 5-dimensional cubes.

The model derived from these constraints is straightforward. The directions derived from the motion field form the basic structure of the texture flow cross-sections F_{e_i} which form the model $\mathcal{F} = \{F_{e_i}\}$. The parameterization of F over the base space (x, y) gives a natural parameterization of the model. As with image curves we initially define the model around the origin.

$$F(x,y) = \begin{pmatrix} x \\ y \\ \arg(f_x, f_y) \\ \kappa_u(x,y) \\ \kappa_v(x,y) \end{pmatrix},$$

where

$$\begin{aligned} \kappa_u(x,y) &= (f_x\kappa_u + f_y\kappa_v + (f_y x - f_x y)(\kappa_u{}^2 + \kappa_v{}^2))/(f_x^2 + f_y^2)^{3/2} \\ \kappa_v(x,y) &= (f_x\kappa_v - f_y\kappa_u + (f_x x + f_y y)(\kappa_u{}^2 + \kappa_v{}^2))/(f_x^2 + f_y^2)^{3/2} \end{aligned}$$

The κ_u and κ_v components of this model are derived by reparameterizing the field around (x, y). This can be done easily by noting that the field $f(x,y)$ is singular at

$$\left(\frac{-\kappa_v}{\kappa_u{}^2 + \kappa_v{}^2}, \frac{\kappa_u}{\kappa_u{}^2 + \kappa_v{}^2}\right).$$

By maintaining the location of this singularity and calculating κ_u and κ_v with respect to the direction $\theta(x,y)$ we obtain the curvature values above.

Finally, we define the extension of the model cross-section over the base space by reparameterizing the static field at each point in the base space. As with image curves this is simply the translation and rotation of the fields defined around the origin.

$$F_{x'y'\theta\kappa_u\kappa_v}(x,y) = \mathrm{T}_{x',y'}\,\mathrm{R}_\theta \begin{pmatrix} x \\ y \\ \arg(f_x, f_y) \\ \kappa_u(x,y) \\ \kappa_v(x,y) \end{pmatrix}.$$

Note that κ_u and κ_v are invariant under rotation, since they are defined with respect to the local orientation. Since the transport of the field is actually defined by a reparameterization,

Corollary 9.1 *The model $\mathcal{F}$ is minimal.*

9.2 Initial Estimates

The initial measurement of confidence is so similar to that used for image curves that we need only slightly modify these estimates for texture flow. As we noted above, the tangent curvature is a measure of change of orientation parallel to the field. Because of this, there is a natural mapping from curved line or edge operators to operators for texture flows with $\kappa_v = 0$. There is a difficulty, however, in deriving operators specific to non-zero normal curvatures. Before we explain how this might be resolved, we modify the image curve operators slightly.

Psychophysical evidence suggests that locally correlated dots in Glass patterns must be of similar contrast in order to create a flow-like percept [GS76]. As one way of interpreting this, we suggest that the features underlying texture flow are contrast-sign specific, but that the responses from these building blocks combine across contrast and kind. That is, only oriented features with locally consistent contrast (i.e. bright or dark lines) will contribute to the texture flow field, but the responses for both signs of contrast will contribute equally to the *same* texture flow field. This is an argument for a complex cell [HW62] or "edge energy" building block [MB88], which responds equally to either contrast sign, but which is insensitive to neutral contrast inputs (e.g. a dot pair consisting of one white and on black dot). Within the context of L/L operators, this leads to an obvious extension of the image curve operators.

Operator 9.1 Select two L/L line operators Ψ_P and Ψ_N for bright and dark lines respectively, and two edge operators Ψ_E and Ψ'_E for opposite contrast edges. If they are all of the same size, and tuned for the same orientation and curvature then a L/L curved texture flow operator tuned for the given orientation and tangent curvature is given by

$$\Psi_F = \Psi_P \mathbin{\bar{\vee}} \Psi_N \mathbin{\bar{\vee}} \Psi_E \mathbin{\bar{\vee}} \Psi'_E.$$

Now we come back to the question of how to develop to obtain initial estimates of the normal curvature κ_v. One approach would be to augment these local operators with laterally displaced off-parallel components, but it is not clear how effective this would be. In general it should not be a requirement for the perception of a divergent flow that there be matched pairs of off-parallel line segments (see Fig. b).

Instead we have chosen an alternative strategy. We use the relaxation network itself to develop the estimates of normal curvature. This is possible only because κ_v is a directional derivative of θ. If the initial direction estimates are accurate, then κ_v is constrained everywhere by the local variation in this direction field. We thus initialize all estimates of p_i for a particular position, orientation and tangent curvature with the tuned initial estimates from Op. 9.1. All normal curvatures at this point are initialized with the same value. The relaxation then uses this starting position without the lateral components which restrict positive supports to local maxima in normal curvature. In this way, a local estimate of the normal curvature is actually derived from the texture flow support field. This is similar to the approach used by Parent [PZ85] to estimate local curvature without curvature-tuned local operators.

9.3 Relaxation

With image curves, had the initial operators performed perfectly (i.e. all and only those operators on thick traces respond positively) we could have avoided the relaxation step entirely. In that case, it was largely the realization that no simple local image operator can simultaneously resolve all of the competing demands that the thick trace representation requires and also be *completely insensitive to noise.* For texture flow however, relaxation is not an option, even with perfect inputs. Remember that texture flows generate a dense percept from potentially sparse data (e.g. Glass patterns). Since the initial operators can only extract information from the image data directly, the inference of dense structure must be left up to some interpolation process, in this case relaxation labelling. Moreover, as we showed above, the initial estimates do not even provide a complete description of the model parameters. Thus in order to create a dense description of the flow which includes normal curvature we must incorporate a relaxation stage.

The design of the support network evolves directly from the equations in §7.3. The model $\mathcal{F}$ developed above has a simple tubular map. Since each model cross-section covers the entire base space, in order to calculate e_i^* for a given e_j we need only project onto F_{e_j} on the fibre over e_i. Assume for the moment that $e_j = (0,0,0,\kappa_u,\kappa_v)$ and $e_i = (x_i y_i, \theta_i, \kappa_{ui}, \kappa_{vi})$. Computing F_{e_j} on the fibre over e_i (i.e. at $(x,y) = (x_i, y_i)$) we thus have

$$e_i^* = \begin{pmatrix} x \\ y \\ \arg(f_x, f_y) \\ \kappa_u(x,y) \\ \kappa_v(x,y) \end{pmatrix}.$$

As with image curves, since only relative position $(x_j - x_i, y_j - y_i)$ is significant, we can transform this into general position by translation and rotation. Thus for $e_i = (0,0,\theta_i,\kappa_{ui},\kappa_{uv})$ and $e_j = (x_j, y_j, \theta_j, \kappa_u, \kappa_v)$ we have

$$\begin{aligned} (x,y) &= \mathrm{R}_{-\theta_j}(-x_j, -y_j) \\ e_i^* &= \begin{pmatrix} 0 \\ 0 \\ \arg(f_x, f_y) - \theta_j \\ \kappa_u(x,y) \\ \kappa_v(x,y) \end{pmatrix}. \end{aligned}$$

Thus gives the incompatibility and transport distances directly:

$$\begin{aligned} d^c_{\mathcal{F}}(e_i, e_j) &= d(e_i^* - e_i), \\ d^t_{\mathcal{F}}(e_i, e_j) &= |(x_j, y_j)|. \end{aligned}$$

We now have the necessary building blocks to define the local geometric support for a texture flow label. Since the tubular map restricts the projection to an individual

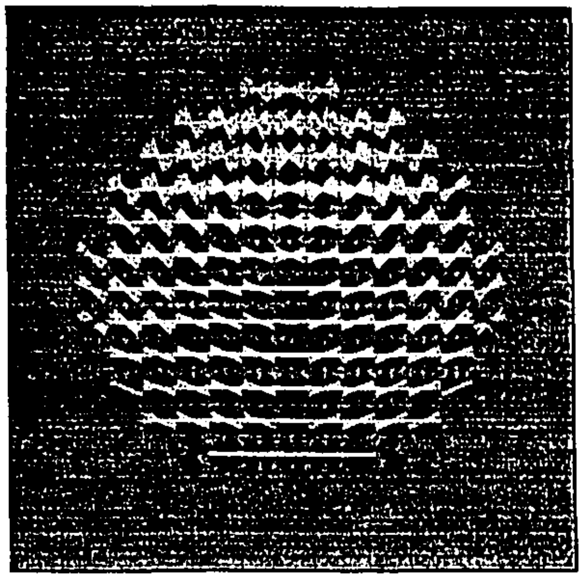


Figure 9.3: The linear reduction of the texture flow field supporting a label with $\theta_i = 0, \kappa_u = -0.1$, and $\kappa_v = 0.0$.

fibre, any basis for the fibre is a basis for the tubular neighbourhood of a point. Thus the usual orthonormal basis for the fibre can be used, namely $\Gamma = \{\gamma_1, \gamma_2, \gamma_3\}$ where

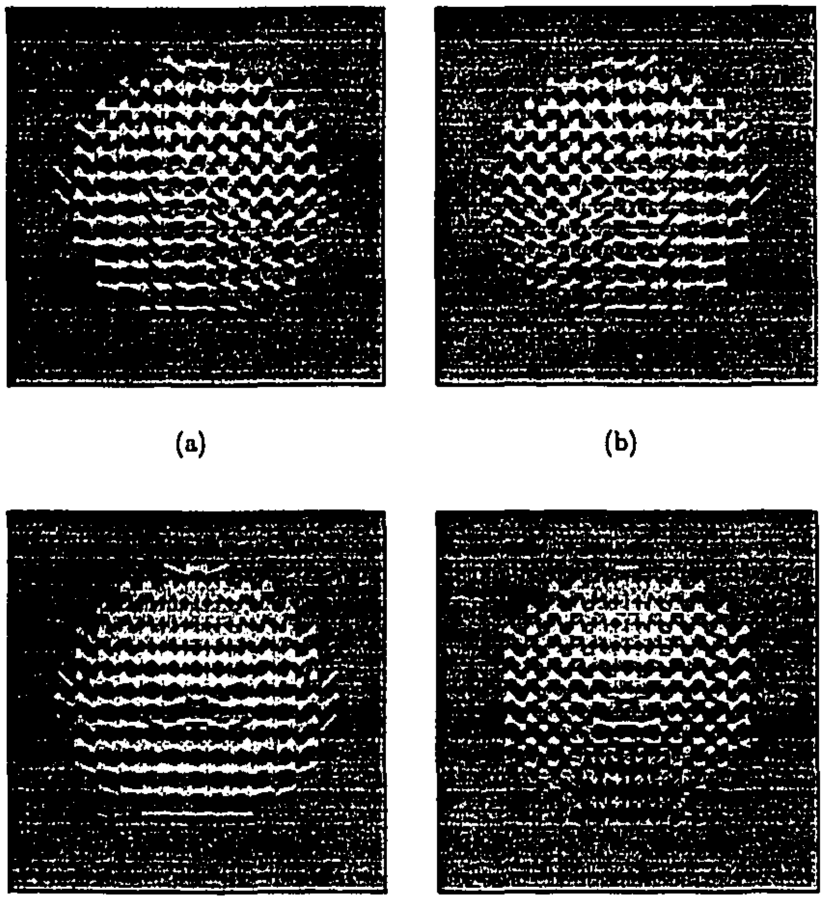
$$\begin{aligned}\gamma_1 &= (0,0,1,0,0),\\ \gamma_2 &= (0,0,0,1,0),\\ \gamma_3 &= (0,0,0,0,1).\end{aligned}$$

However, since κ_v is a construct of the network, we do not create a pair of lateral inhibition components in the γ_3 direction. Thus we have four lateral components making up the support network around e_i (shown in Fig. 9.4)

$$\begin{aligned}r^1_{ij} &= G^j_{\theta_i}(e_i - \rho\gamma_1), & r^2_{ij} &= -G^j_{\theta_i}(e_i + \rho\gamma_1),\\ r^3_{ij} &= G^j_{\kappa_{ui}}(e_i - \rho\gamma_2), & r^4_{ij} &= -G^j_{\kappa_{ui}}(e_i + \rho\gamma_2),\end{aligned}$$

where

$$G^j(e_i) = \mathrm{G}_{\sigma_t}(d^t_{\mathcal{F}}(e_i, e_j))\, \mathrm{G}_{\sigma_c}(d^c_{\mathcal{F}}(e_i, e_j)).$$



(a) (b)

(c) (d)

Figure 9.4: Shown are the four lateral inhibition component networks of the full support network for a texture flow label with $\theta_i = 0.0, \kappa_u = -0.1$, and $\kappa_v = 0.0$. The interactions shown are (a) r^1_{ij}, (b) r^2_{ij}, (c) r^3_{ij}, (d) r^4_{ij}. Note that (a) and (b) together select local maxima in orientation, (c) and (d) in tangent curvature.

The decomposition into regions is similarly straightforward. The real constraint on this is that we wish discount the effect of isolated curves. When a single curve is not supported by other laterally displaced parallel curves, it should not by itself cause a significant texture flow percept. One way to achieve this is to ensure that the initial responses caused by a single curve are segregated into just two of the quadrants around a point. Since we are free to choose *any* orthonormal basis for the base space X, we choose the one for which ζ_1 is offset by $\pi/4$ from θ_i. Thus if we set $\theta_i' = \theta_i + \pi/4$ the basis is given by

$$\begin{aligned}\zeta_1 &= (\cos\theta_i', \sin\theta_i', 0, 0, 0),\\ \zeta_2 &= (-\sin\theta_i', \cos\theta_i', 0, 0, 0).\end{aligned}$$

This produces four regions (see Fig. 9.5):

$$\begin{aligned}q_1(e_i, e_j) &= \sigma(\zeta_1 \cdot (e_j - e_i))\,\sigma(\zeta_2 \cdot (e_j - e_i)),\\ q_2(e_i, e_j) &= \sigma(\zeta_1 \cdot (e_i - e_j))\,\sigma(\zeta_2 \cdot (e_j - e_i)),\\ q_3(e_i, e_j) &= \sigma(\zeta_1 \cdot (e_j - e_i))\,\sigma(\zeta_2 \cdot (e_i - e_j)),\\ q_4(e_i, e_j) &= \sigma(\zeta_1 \cdot (e_i - e_j))\,\sigma(\zeta_2 \cdot (e_i - e_j)).\end{aligned}$$

9.4 Results

For the most part, the parameters controlling the texture flow relaxation are identical to those for image curves. We reuse the image operators from Table 8.1 to build the initial operators above. For this reason we have discretized κ_u into five classes with values $\{-0.2, -0.1, 0.0, 0.1, 0.2\}$, and θ into 8 classes with the difference between adjacent directions was 22.5°. The lateral components of the support were implemented with $\sigma_c = \epsilon = \sqrt{2}/2$, normalized to the distance metric. Since κ_v is a construct of the relaxation support, we tested discretizations with only one class

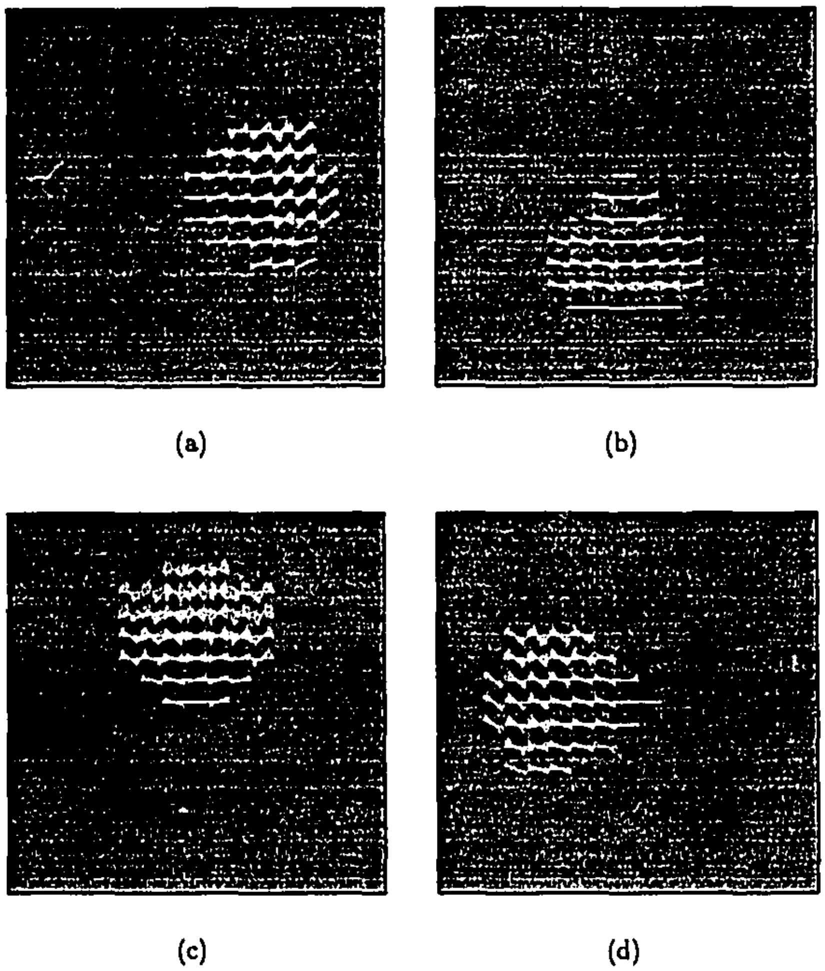


Figure 9.5: Decomposition into regions of the support network for a texture flow label with $\theta_i = 0$, $\kappa_{u_i} = -0.1$, and $\kappa_{v_i} = 0.0$. Shown are the linear reductions of the networks for four quadrants around the origin.

$\kappa_v = 0$ and with three $\kappa_v \in \{-0.1, 0.0, 0.1\}$. Since the κ_v values are not localized with a pair of lateral inhibition components, we use a smaller $\sigma_c = 0.5$ in the κ_v direction. As we explained above, the accuracy of the orientation estimates ensures that the cross-section remains localized in this direction.

Since the support fields are circular, the support is divided into four boundary-stability components as described above. The radius of this neighbourhood was established by setting $\sigma_t = 2.5$. The compatibilities were normalized exactly as in Chap. 8 and only those within 5% of maximum were used. Finally, the initial measurements used the ρ-approximate L/L operators with $\rho = 16$, and the relaxation used the adaptive ρ-approximates with $\rho' = 4$.

Note that except for the special treatment of κ_v the parameters used in this case are identical to those used for image curves. We take this as empirical verification of the claim in §7.3 that the choice of values for most of the "free parameters" in the relaxation is independent of the particular models used.

Again, we leave most of the specific comments on the results to the figure captions, and focus on general observations:

- The interpolation performed by the relaxation is very fast, producing dense descriptions from initially sparse ones after only one or two iterations (Fig. 9.7). Note also (Fig. 9.13) that the interpolation takes place even through non-empty regions in the case of transparent overlapping fields.

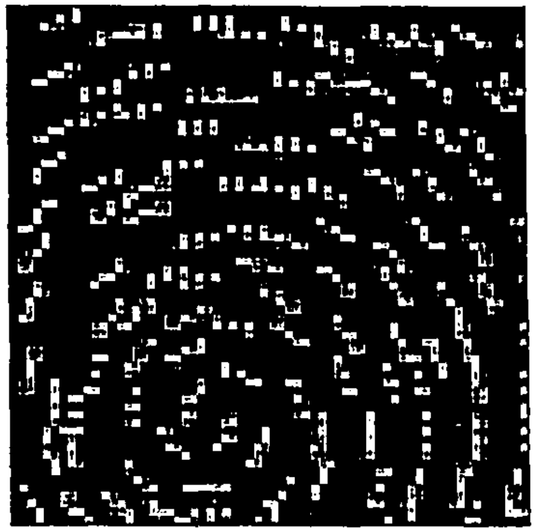
- The interpolation appears to be stable around singularities of the field such as centers of rotation or expansion.

- Overlapping fields are clearly indicated by the presence of multiple, disjoint labels coexisting at the same point in the image (Fig. 9.13).

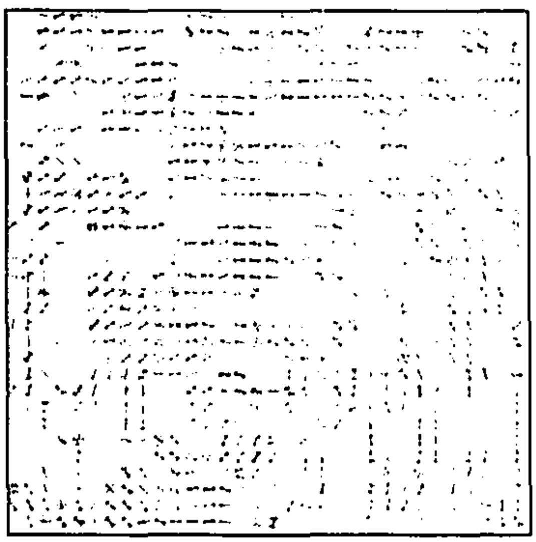
- The boundaries of the regions appear to be stable and appear to "infer" a smooth boundary around the field. One effect of the "surround" requirement is that these boundaries are slightly concave when they are interpolated through blank regions.

- Discontinuities in the flow fields are signalled by *coincident boundaries* (Fig. 9.11). The fields themselves need not actually overlap, but if the boundaries coincide within a few pixels over a long distance, then they should be interpreted as a texture discontinuity or occlusion and not as independent transparent flows.

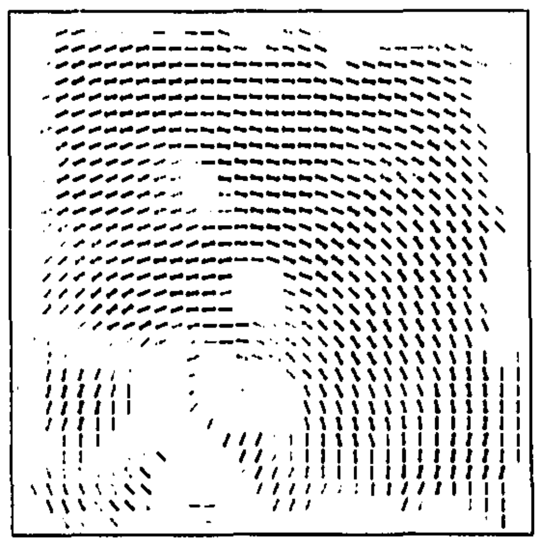
- Flaws in the flow fields are clearly represented by either a hole or a discontinuous patch. This may be essential information in situations where the flaws are the points of interest (e.g. for locating knots in wood).



(a)



(b)



(c)

Figure 9.6: A circular Glass pattern (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). Note that the flow is interpolated densely everywhere even though neither the image nor the initial operator responses are dense.

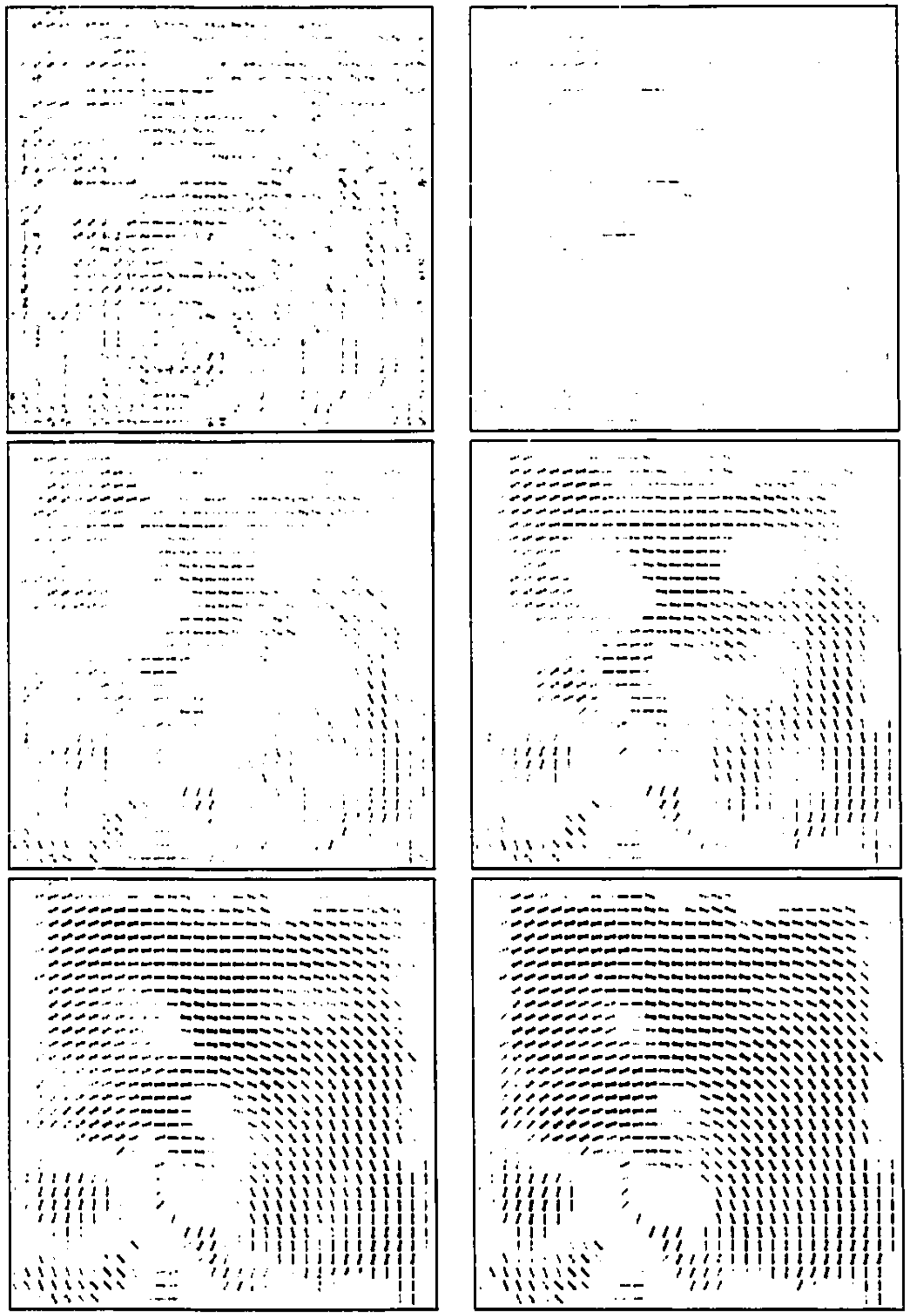
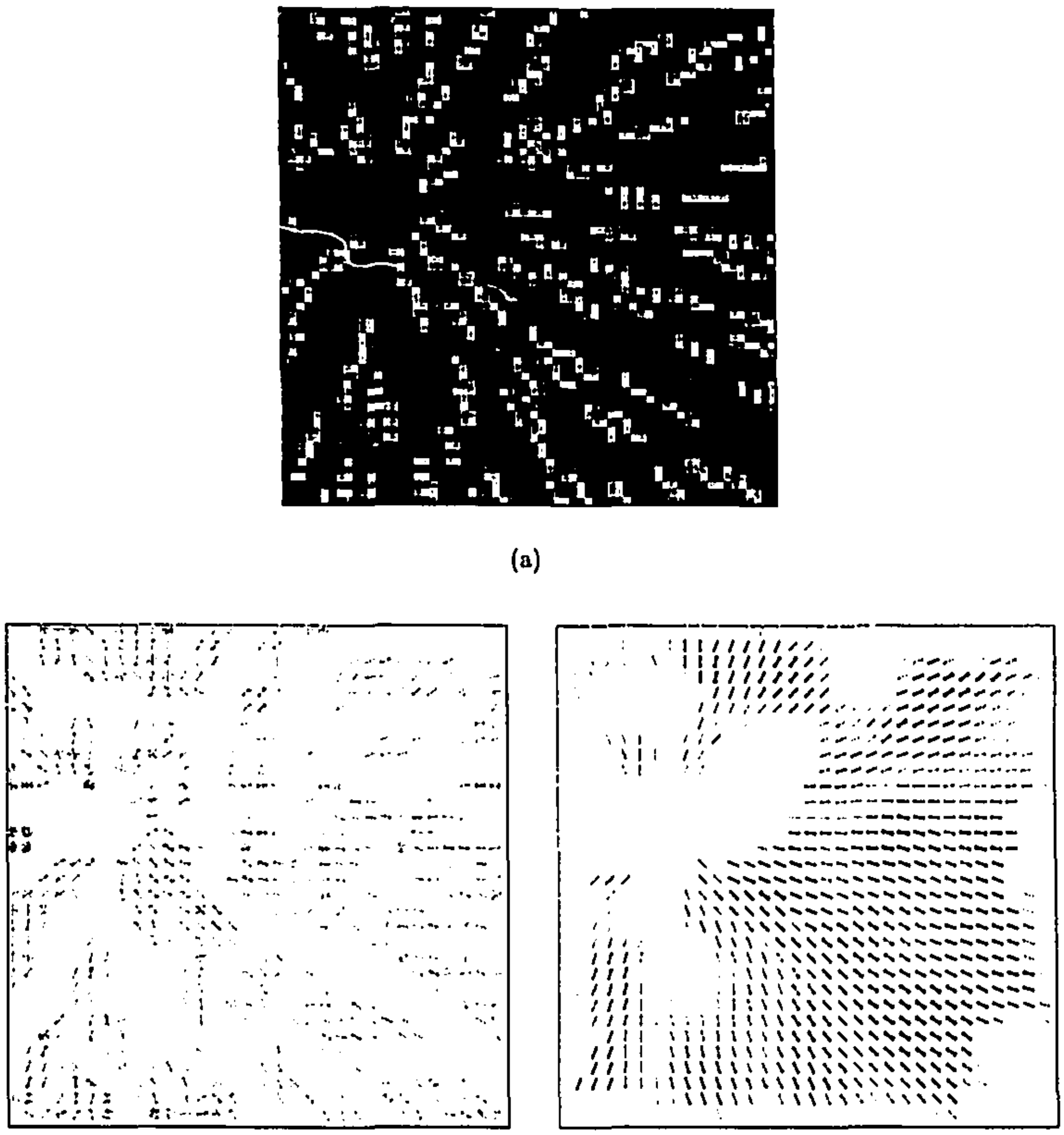


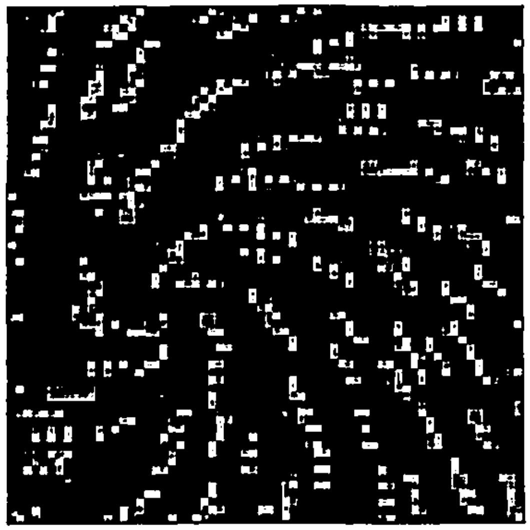
Figure 9.7: The state of the relaxation of the texture flow for the circular Glass pattern after initial operators (top left), and through five iterations (across and down). Note that the interpolation takes place on the first two iterations after which orientations and curvatures are refined.



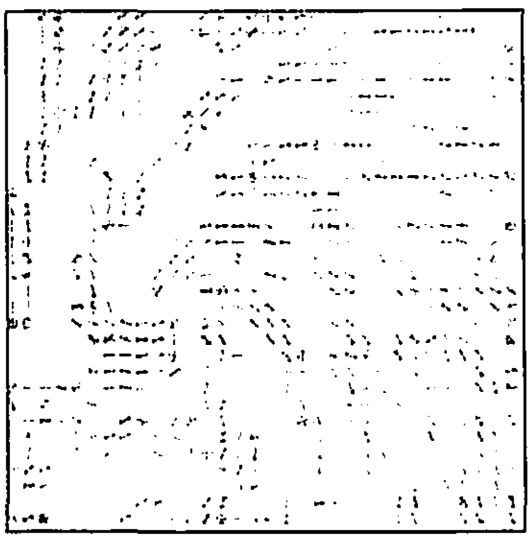
(a)

(b) (c)

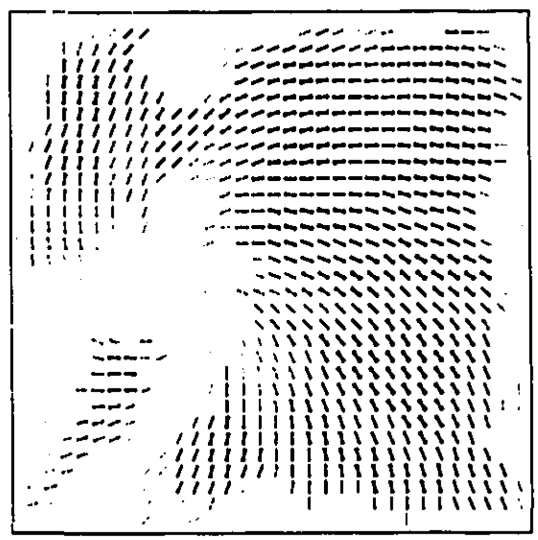
Figure 9.8: A divergent Glass pattern (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). This relaxation is performed with normal curvature $\kappa_{v} = 0.0$. Note that it is only very near the singularity that we see an effect of the assumption that κ_{v} does not vary from zero.



(a)

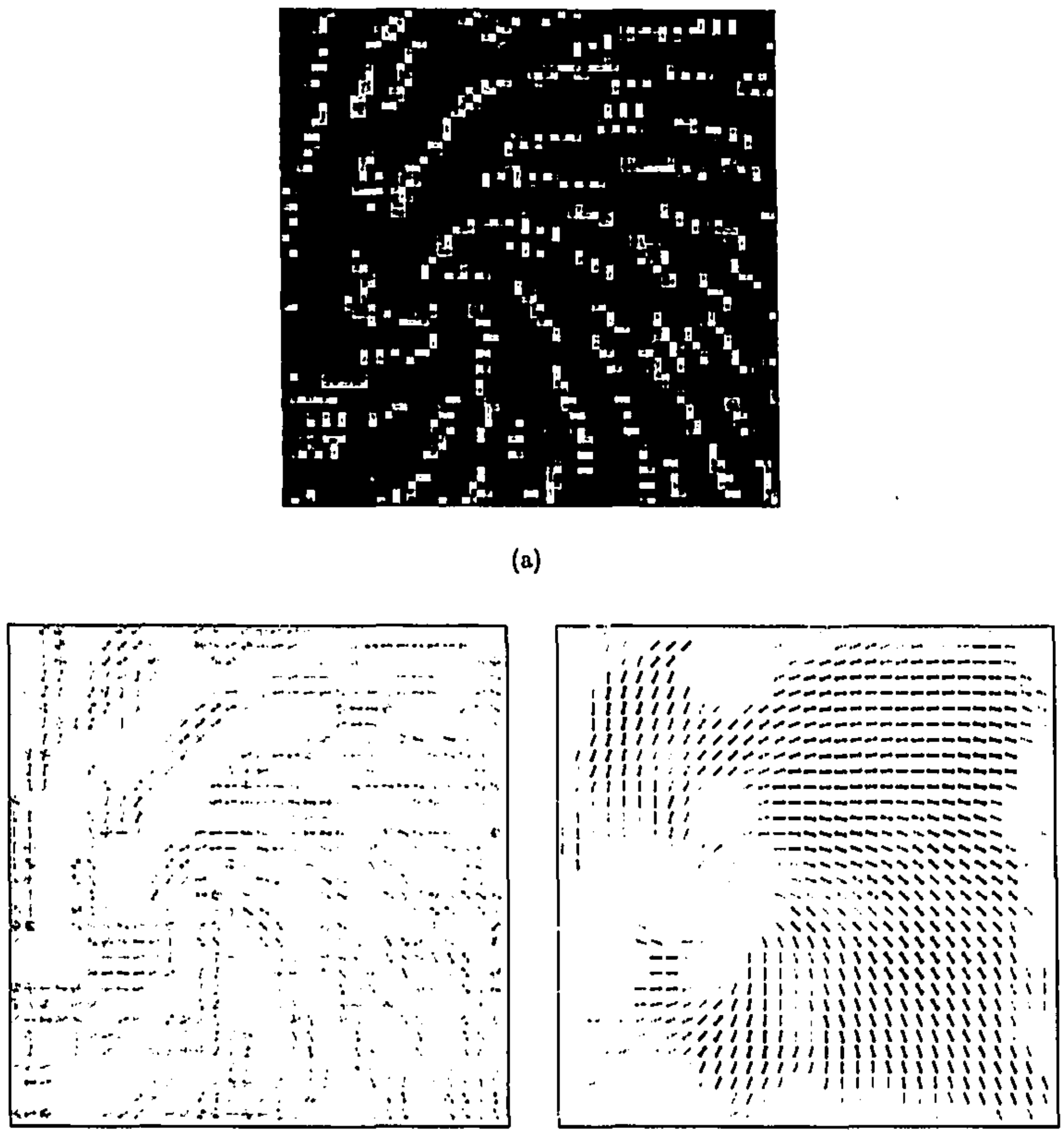


(b)



(c)

Figure 9.9: A Glass pattern with both rotation and divergence (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). This relaxation is performed with normal curvature $\kappa_{\upsilon} = 0.0$. There are gaps in the field near the singularity, where the normal curvature differs significantly from zero.

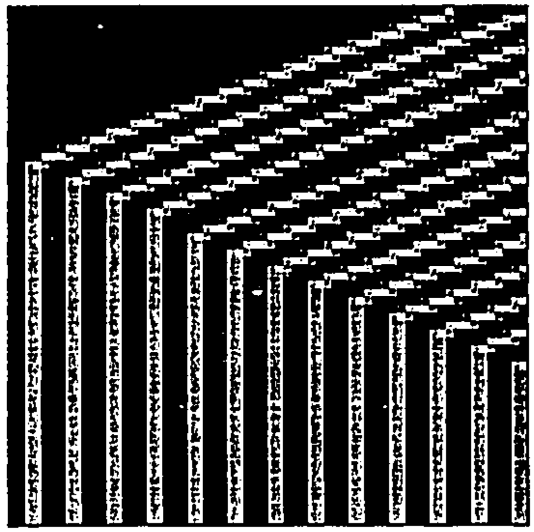


(a)

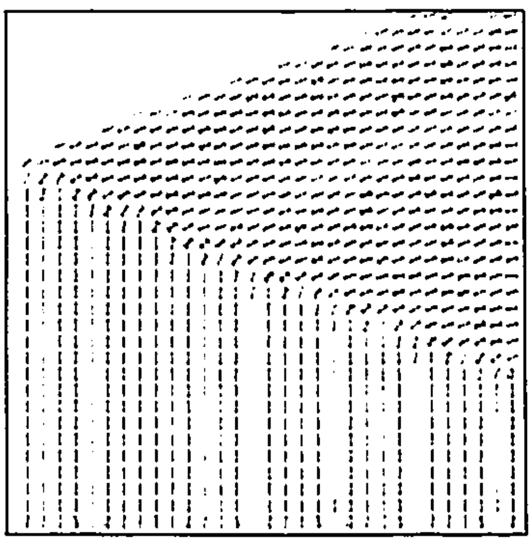
(b)

(c)

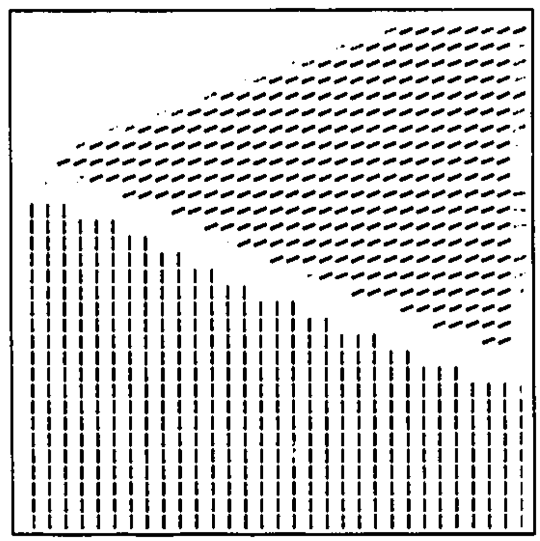
Figure 9.10: A Glass pattern with both rotation and divergence (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). This relaxation is performed from the same starting point as Fig. 9.13 except with three normal curvature classes $\kappa_v \in \{-0.1, 0.0, 0.1\}$. There are no longer any gaps in the field near the singularity, where the normal curvature differs significantly from zero.



(a)



(b)



(c)

Figure 9.11: A pattern of parallel lines with a local orientation discontinuity (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). The flow field extracted shows two distinct flow fields with a common boundary (to within 2 pixels). Common boundaries signal a discontinuity in the field, either because of a flow discontinuity or an occlusion.

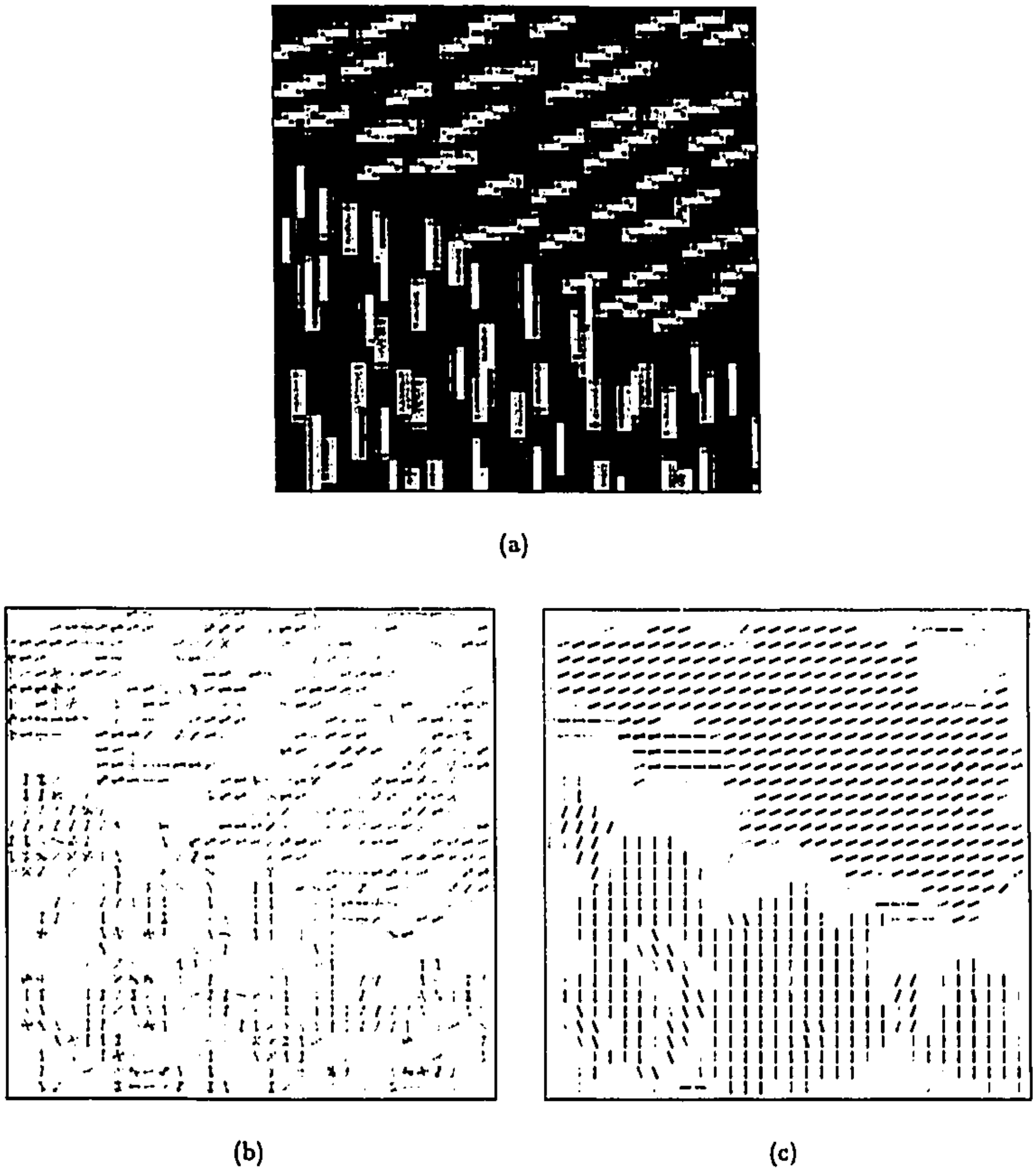
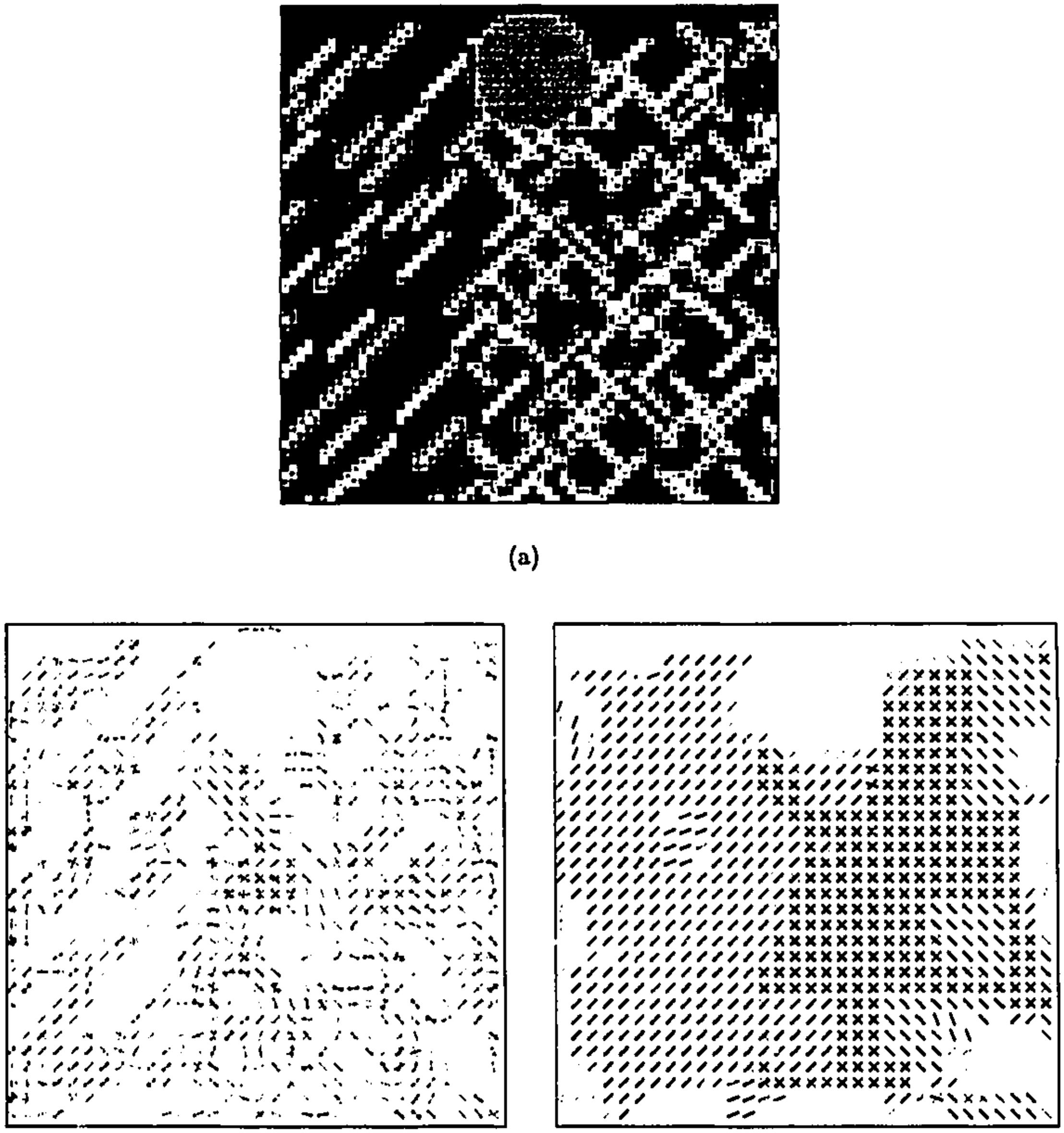
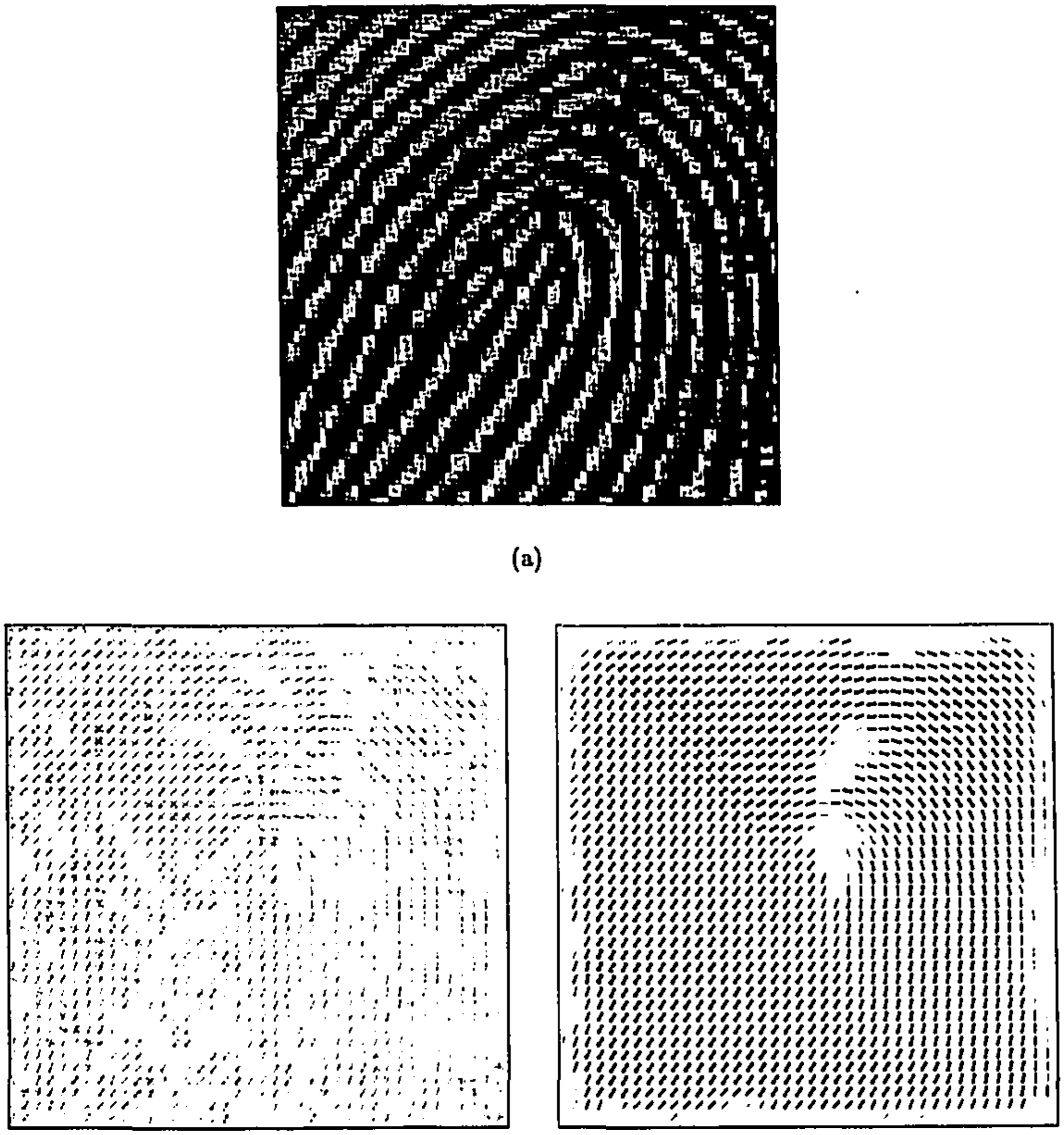


Figure 9.12: A Glass pattern with a local orientation discontinuity (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). Note that exactly as with the ideal parallel lines, the flow field extracted shows two distinct flow fields with a common boundary.



(a) (b) (c)

Figure 9.13: A complex texture flow pattern with overlapping textures and a hole (a), and the results of applying initial flow operators (b), followed by 10 iterations of geometric relaxation (c). The overlapping fields are represented by multiple unconnected labels being supported over the same position. Note that the presence of this overlap does not interfere with the ability to locate the boundary of the field overlayed on the right hand side of the image, even though there is no such discontinuity in the other field. Of course, neither field encroaches on the hole at the top.

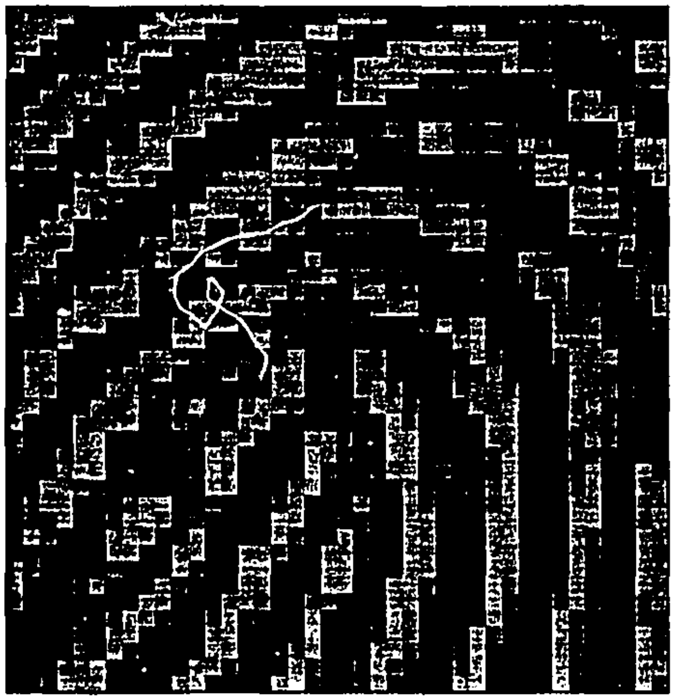


(a)

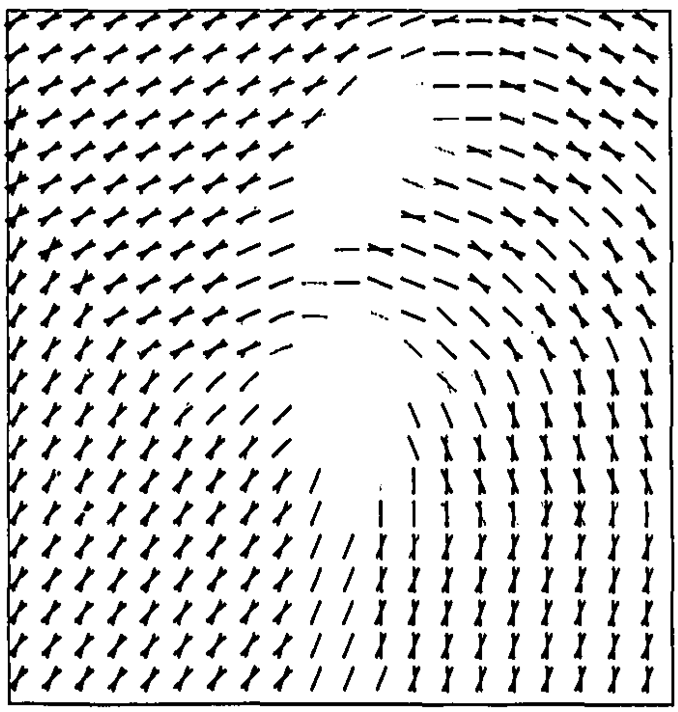
(b)

(c)

Figure 9.14: A whorl in the fingerprint image (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). Note that the flow is stable around the singularity, and near the "rift" directly above the singularity.

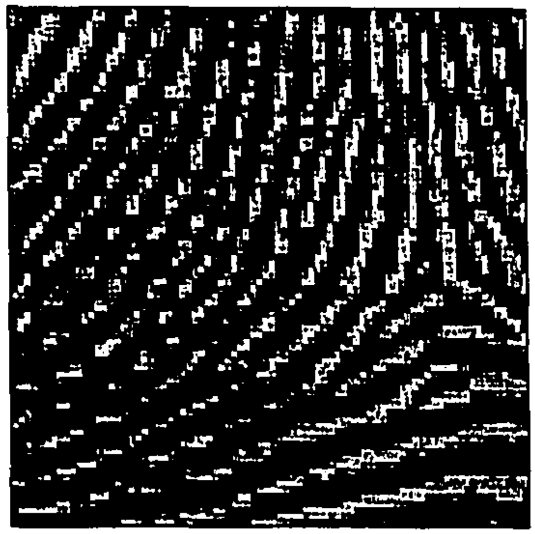


(a)

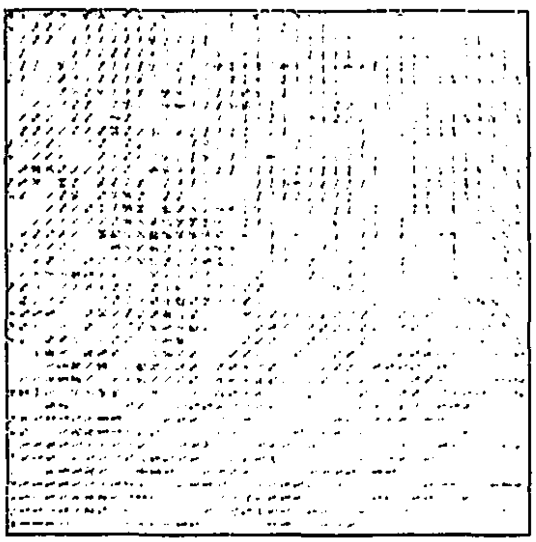


(b)

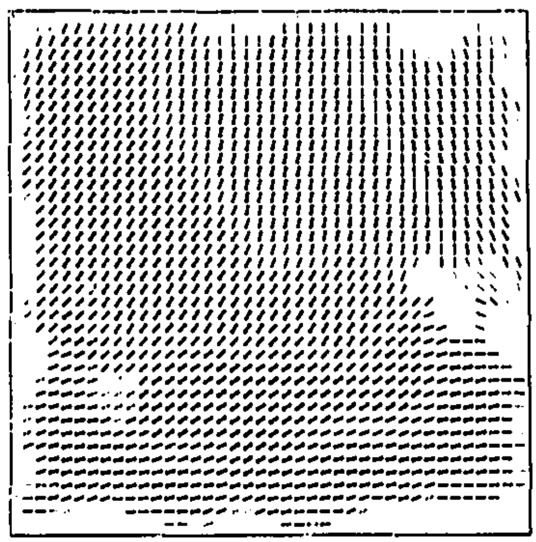
Figure 9.15: Blowup of the fingerprint (a) and the flow field (b) around the singularity. Shown are only the labels which received positive support at the fifth iteration. Notice the curvatures associated with these labels. Also note the "holes" at the singularity and rift.



(a)

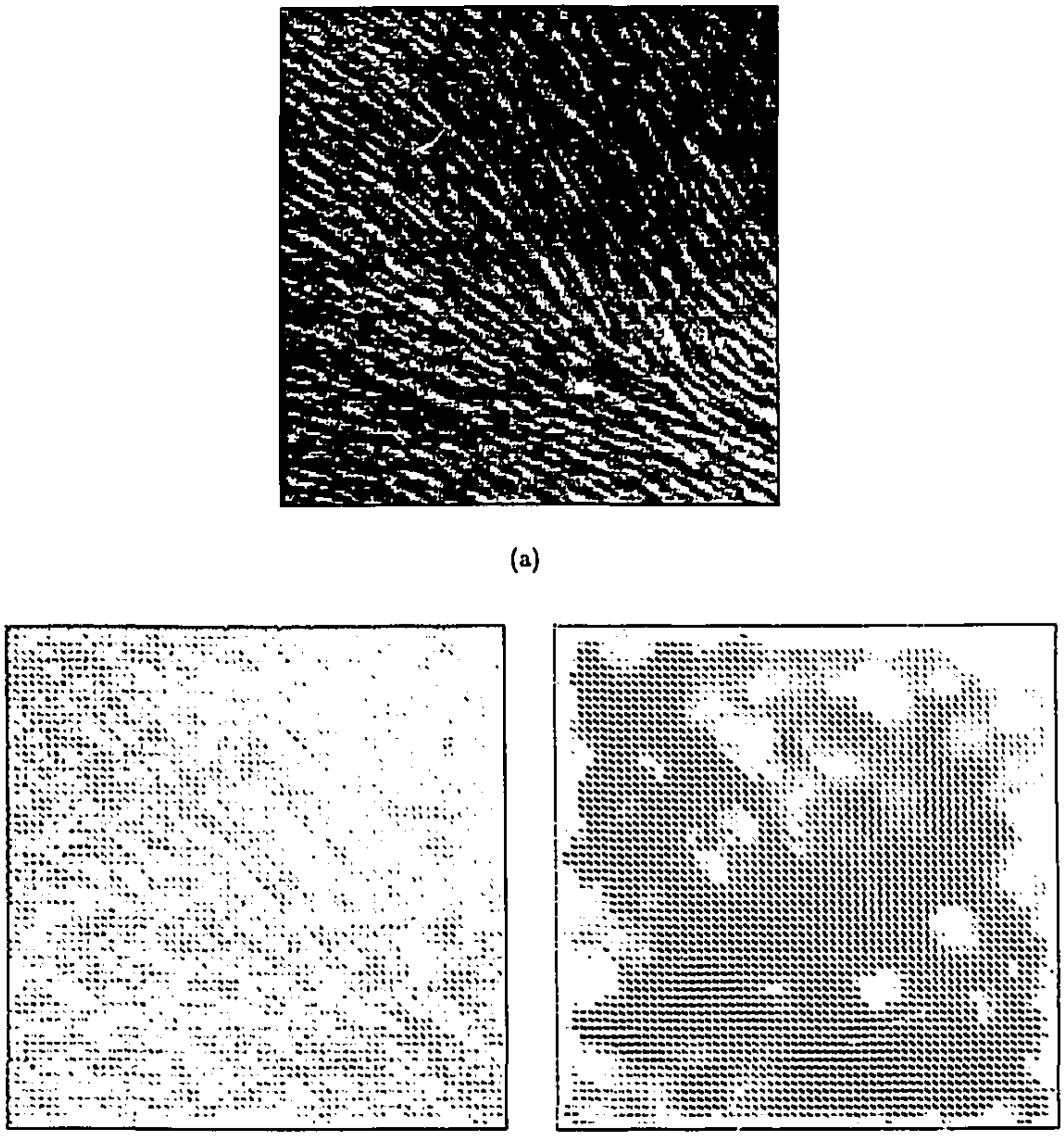


(b)



(c)

Figure 9.16: Another singularity in the fingerprint image (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). Note that the flow is stable around this very different type of singularity.



(a)

(b) (c)

Figure 9.17: A picture of fur taken from [Bro66], pp. 93 (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). The direction of flow varies smoothly over the image.

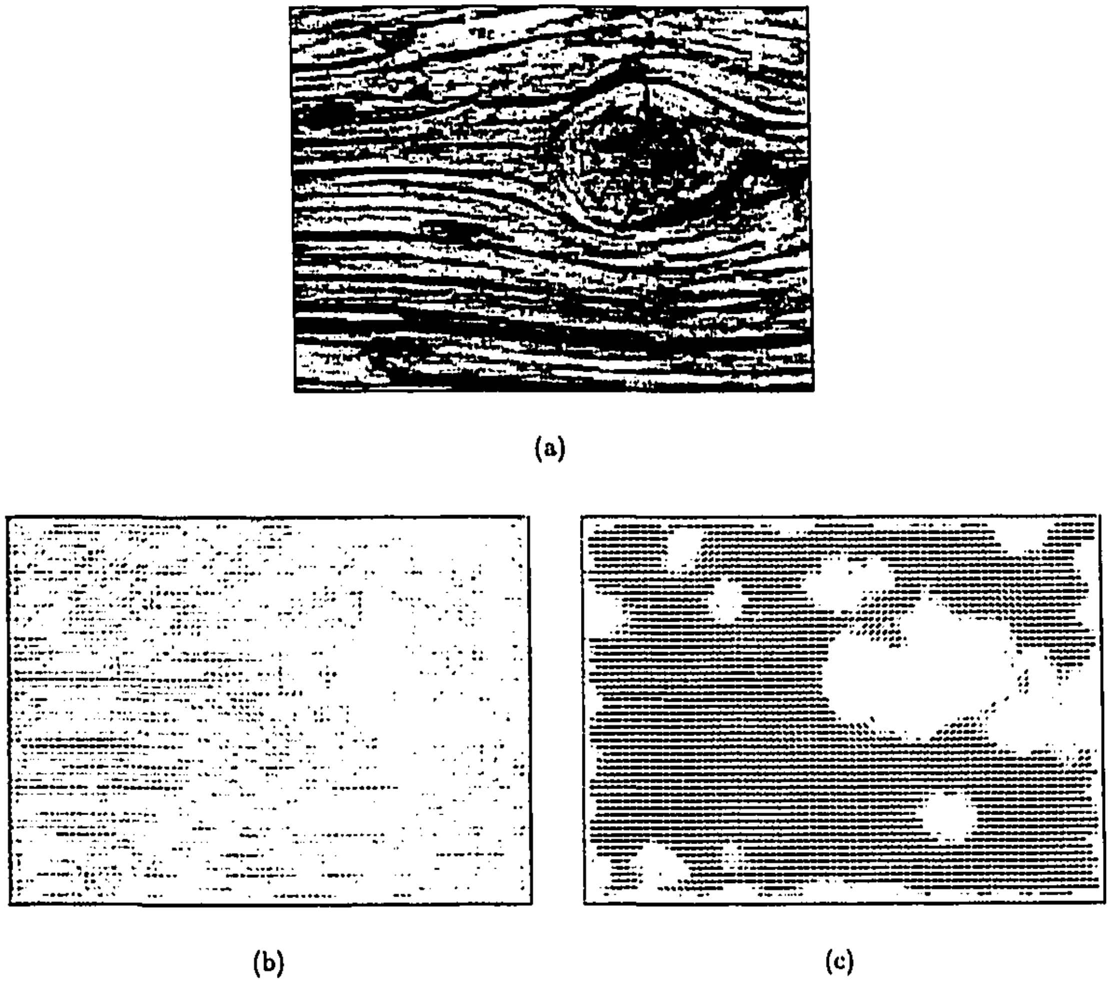


Figure 9.18: A picture of tree bark taken from [Bro66], pp. 72 (a), and the results of applying initial flow operators (b), followed by 5 iterations of geometric relaxation (c). Note the holes in the resulting flow field wherever there are knots or flaws in the wood surface.

Chapter 10 Conclusions

From a strict engineering point of view, this thesis presented a general-purpose system for analyzing curves and texture flow fields in early vision. In both cases, the results obtained from this system are clearly accurate and robust.

For curves, the system is able to categorize the three kinds of local contrast-defined image curves (bright and dark lines, and edges) accurately and without confusion. The resulting descriptions stabilize the end-points of these curves, make coarse and accurate measurements of both orientation and curvature, and implicitly represent the locations of both corners and junction points. Moreover, the topological properties of the thick trace representation ensure that there is some continuous, smooth model curve underlying each connected subset of points which survive the relaxation. The resulting descriptions should be an excellent starting point for higher-level vision systems. This is in contrast to the "industry-standard" alternatives now available which: mislocalize line endings, are unable to properly represent corners and junctions, impose continuity rather than revealing it, and depend on arbitrary thresholds thus limiting sensitivity to faint stimuli.

For texture flow we have developed a system which can accurately describe smooth texture flows in the presence of both discontinuities and transparency. It is able to interpolate dense, smooth flows into blank regions, but only does so when those regions are surrounded by consistent flow. This behaviour seems to mimic human perception when faced with Glass pattern stimuli. The system also localizes the boundaries of regions of smooth flow even in the presence of independent, transparent flows without such discontinuities. Altogether, the system presented is sufficient to capture most of the fundamental properties which characterize the perception of such flow-like textures.

Three new theoretical building blocks were developed in order to achieve these goals: the *thick trace* distributed representations, the *Logical/Linear algebra,* and the theory of *geometric compatibility* for relaxation labelling. In each case, the full generality of the results should be clear. It should thus be straightforward to apply the techniques we have developed to other problems in early vision.

10.1 Future Directions

In considering the directions open for future research, we focus first on extensions to the particular applications investigated, namely image curves and texture flows, and then on broader issues.

Global Descriptions. The most obvious focus for future work is the integration of these distributed representations into more global descriptions. Clearly a system which produces more global representations of curves or fields would be useful, especially one which made explicit some of the information which is implicit in the thick traces: the locations of end-points, corners, junctions and bifurcations. We have taken care throughout to clearly define the descriptions which our system produces—thick traces of cross-sections—and in Chap. 2 we described the properties of these traces, so the input to such a system is well-defined. Moreover, a global integration problem was solved implicitly during the design of the relaxation networks, using a technique reminiscent of the solution for curves proposed in [DZ90]. It is still unclear, however, how to efficiently integrate a connected trace of local descriptions into a unique global description of a curve or flow field.

Scale. Another unresolved issue is the role of scale. In both of the systems designed in this thesis, the scale of processing was fixed and restricted to the smallest possible size with respect to the image sampling. Yet a curve which appears straight locally may be obviously curved if viewed more globally, and the same can be said of flow fields. Our intuitions about the human visual system are often pointers to solutions to general vision problems, and in this case our ability to choose the "right" scale at which to describe a curve indicates that some sort of multiscale representation is being passed on to later stages of processing (after all we seem to be able to pick and choose the scale we pay attention to depending on circumstances). Yet this multi-scale representation does not appear to be simply a function of smoothing (as suggested by scale-space approaches [Wit86]), since fine low-contrast curves (which cannot be detected once the image is smoothed) can produce very large, salient regions of low perceived curvature. The same effect is seen in random dot Moiré patterns in which very small dots can be used to produce large, slowly curving fields. Clearly there is something more going on here. One possible suggestion is provided by theories of multi-grid relaxation (e.g. [Ter84]) which involve interactions between descriptions at different scales.

Optical Flow. The means by which the texture flow model was developed in Chap. 9 and of course the name "texture flow" deliberately evoke images of motion fields. We suggest that the extraction of optical flow may involve a natural extension of the texture flow system to spatio-temporal images. The path is fairly clear. Linear operators for detecting and describing local motion have been described (e.g. [Hee87]) and these would almost certainly benefit from the introduction of L/L non-linearities. Furthermore, just as the texture flow relaxation manages to resolve the competing goals inherent when smoothing in the presence of discontinuities, so could a similar approach to optical flow. The presence of transparent, overlapping fields is perhaps even more endemic in motion that with static textures. Finally, short range motion capture may be explained in terms of the "surround" processing of motion fields.

Implementation. Rather considering other applications of these ideas, another issue of pressing importance is the question of implementation. Even on modern SIMD machines such as the MasPar or Connection Machine the computational scale of the systems we have described can be overwhelming. Consider the combinatorics of image curves. For the initial operators (for edges), we have 80 local operators each of which is composed of 20 linear components. This means that 1600 convolutions of the image are performed just to extract initial estimates. For a 512×512 image this will produce an image bundle with 20,000,000 sampled points. To relax this system involves evaluating as many as several hundred local interactions per node. At least in scale, we have produced a system that has more in common with visual cortex than "neural networks".

Yet there is hope for implementing these systems efficiently. We note that all computations performed in the system can be decomposed into identical building blocks of the form

$$y_i = (\sum_j a_{ij} x_j) \triangle (\sum_j b_{ij} x_j)$$

where a_{ij} and b_{ij} are *fixed weights* and the x_j are small, bounded values communicated over some local network. So the entire systems consists of nothing more than fixed summing networks and point non-linearities. Moreover for both the initial operator responses and the relaxation iterations, *all updates may be performed simultaneously.* This suggests that some form of dedicated, highly interconnected VLSI system might be able to perform these computations in real-time.

But what would this system look like? The initial L/L operators were implemented entirely in low-precision integer arithmetic (8-16 bits), and the relaxation labelling confidences were passed between iterations with similarly low-precision integers (8 bits). This suggests that useful systems can be developed with very low bit-rate communication paths. A not unreasonable path to pursue would thus be dedicated VLSI, perhaps even analog VLSI [Mea89]. With interconnections built in and weights either programmed or burned in, it would take no more than a few layers of hardware to go from an image to a fully connected thick trace describing curves or texture flow.

Biology. The final suggestion for future work is really just an exploration of an undercurrent of the entire project—the structure of biological vision systems. Anyone even vaguely familiar with the neurophysiology of early vision [HW62, Orb84] will have noticed echos of biology throughout this work. We have previously investigated some of the implications of the L/L operators for theories of simple cells in visual cortex [DIZ90]. There were very promising connections between the computational theory and the behaviour of individual neurons. We believe that this should be investigated further. Moreover, if the Logical/Linear non-linearities do indeed have correlates in the operations of neurons then the relaxation techniques may constitute a theory of at least some of the interactions between neurons in primary visual cortex.

References

[Arm83] M. A. Armstrong. *Basic Topology*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, N. Y., 1983.

[Aur91] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):347–405, September 1991.

[BA84] M. Brady and Asada. Smoothed local symmetries and their implementation. *Int. J. Robotics Research*, 3:36–61, 1984.

[Bie85] Irving Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics and Image Processing*, 32(1):29–73, October 1985.

[BM77] Garrett Birkhoff and Saunders MacLane. *A Survey of Modern Algebra (fourth edition)*. Macmillan Publishing Co., Inc., New York, N. Y., 1977.

[BPR83] J. Beck, K. Pradzny, and A. Rosenfeld. A theory of textural segmentation. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*, chapter 1, pages 1–38. Academic Press, New York, N.Y., 1983.

[Bro66] Phil Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, New York, N.Y., 1966.

[Bur89] Richard L. Burden. *Numerical Analysis*. PWS-Kent, Boston, Mass., fourth edition, 1989.

[BZ87] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, Massachusets, 1987.

[Can86] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 679–698, 1986.

[Con93] John Horton Conway. *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, New York, N.Y., 2nd edition, 1993.

[dC76] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, Englewood Cliffs, N.J., 1976.

[Der87] Rachid Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.

[DH73] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, 1973.

[DIZ90] Allan Dobbins, Lee Iverson, and Steven W. Zucker. A logical/linear model of cortical subunit interactions. In *ARVO Annual Meeting: Abstract Issue*, page 397, Sarasota, Florida, May 1990.

[Dob92] Allan Dobbins. *Difference Models of Visual Cortical Neurons.* PhD thesis, McGill University, Montréal, Canada, 1992.

[Doo84] J. L. Doob. *Classical Potential Theory and its Probabilistic Counterpart*, volume 262 of *Grundlehren der mathematischen Wissenschaften.* Springer-Verlag, New York, N.Y., 1984.

[DRA76] Larry Davis, Azriel Rosenfeld, and A. Agrawala. On models for line detection. *IEEE Transactions on Systems, Man and Cybernetics*, 6(2):127–133, 1976.

[DZ90] Chantal David and Steven W. Zucker. Potentials, valleys and dynamic global coverings. *IJCV*, 5(3):219–238, 1990.

[DZC87] Allan Dobbins, Steven W. Zucker, and Max Cynader. Endstopping in the visual cortex as a substrate for calculating curvature. *Nature*, 329:438–441, 1987.

[DZC88] Allan Dobbins, Steven W. Zucker, and Max Cynader. Endstopping and curvature. McGill Research Centre for Intelligent Machines Technical Report TR-CIM-88-3, McGill University, Montréal, 1988.

[FA91] W. Freeman and E. Adelson. The design and use of steerable filters for image analysis, enhancement and wavelet decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[Gla73] Leon Glass. Perception of random dot interference patterns. *Nature*, 246:360–362, 1973.

[GS76] Leon Glass and E. Switkes. Pattern recognition in humans: Correlations which cannot be perceived. *Perception*, 5:67–72, 1976.

[Har82] Robert M. Haralick. Digital step edges from zero-crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:58–68, 1982.

[HB70] A. Herskovitz and T. O. Binford. On boundary detection. MIT AI Memo 183, MIT AI Lab, Cambridge, Mass., 1970.

[Hee87] David J. Heeger. Optical flow from spatiotemporal filters. In *Proc. First International Conference on Computer Vision*, pages 181–190, London, England, June 1987.

[Heu71] M. H. Heuckel. An operator which locates edges in digital pictures. *J. Association for Computing Machinery*, 18:113–125, 1971.

[HL87] David H. Hubel and Margaret S. Livingstone. Segregation of form, color, and stereopsis in primate area 18. *J. Neuroscience*, 7(11):3378–3415, 1987.

[Hor77] Berthold K. P. Horn. Understanding image intensities. *Artificial Intelligence*, 8:201–231, 1977.

[HR85] D. D. Hoffman and Whitman A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1985.

[HS81] Berthold K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[Hus66] Dale Husemoller. *Fibre Bundles*. Graduate Texts in Mathematics. Springer-Verlag, New York, 1966.

[HW62] David H. Hubel and Torsten N. Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *J. Physiology (London)*, 160:106–156, 1962.

[HZ83] Robert A. Hummel and Steven W. Zucker. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:267–287, 1983.

[JP87] J. P. Jones and L. A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophys.*, 58(b):1233–1258, 1987.

[Kim91] Ben Kimia. *Toward a Computational Theory of Shape*. PhD thesis, McGill University, Montréal, Canada, 1991.

[Koe84] Jan J. Koenderink. The internal representation of solid shape and visual exploration. In *Sensory Experience, Adaptation, and Perception*, chapter 7, pages 123–142. Lawrence Erlbaum Associates, Inc., 1984.

[Koe88] Jan J. Koenderink. Operational significance of receptive field assemblies. *Biological Cybernetics*, 58:163–171, 1988.

[Kos93] Antoni A. Kosinski. *Differential Manifolds*, volume 138 of *Pure and Applied Mathematics*. Academic Press, Inc., Boston, Mass., 1993.

[KTZ92] Ben Kimia, A. Tannenbaum, and Steven W. Zucker. On the evolution of curves via a function of curvature, I: The classical case. *J. Math. Analysis and Applications*, 163(2):438–458, 1992.

[KvD76] Jan J. Koenderink and Andrea J. van Doorn. The singularities of the visual mapping. *Biological Cybernetics*, 24:51–59, 1976.

[KvD82] Jan J. Koenderink and A. J. van Doorn. The shape of smooth objects and the way contours end. *Perception*, 11:129–137, 1982.

[KvD87] Jan J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*. 55:367–355, 1987.

[KvD90] Jan J. Koenderink and Andrea J. van Doorn. Receptive field families. *Biological Cybernetics*, 63:291–2977, 1990.

[KW87] M. Kass and A. Witkin. Analyzing oriented patterns. *Computer Vision, Graphics and Image Processing*, 37:362–385, 1987.

[Lan85] Serge Lang. *Differential Manifolds*. Springer-Verlag, New York, N.Y., 1985.

[Lec85] Yvan Leclerc. Capturing the local structure of image discontinuities in two dimensions. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 34–38, San Francisco, Ca., 1985.

[Lee90] D. Lee. Coping with discontinuities in computer vision: Their detection, classification, and measurement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:321–344, 1990.

[Ley88] Michael Leyton. A process grammar for shape. *Artificial Intelligence*, 34:213–247, 1988.

[LZ84] Yvan Leclerc and Steven W. Zucker. The local structure of image discontinuities in one dimension. Computer Vision and Robotics Lab Technical Report 83-19R, McGill University, Montréal, 1984.

[LZ87] Norah Link and Steven W. Zucker. Sensitivity to corners in flow patterns. *Spatial Vision*, 2:233–244, 1987.

[LZ88] Norah Link and Steven W. Zucker. Corner detection in curvilinear dot grouping. *Biological Cybernetics*, 59:247–256, 1988.

[Mar82] David Marr. *Vision*. W.H. Freeman, San Francisco, 1982.

[MB88] M. Morrone and D. Burr. Feature detection in human vision: a phase dependent energy model. *Proc. Royal Society*, 235:221–245, 1988.

[Mea89] Carver Mead. *Analog VLSI and Neural Systems*. Menlo Park, Ca. Addison-Wesley, 1989.

[MH80] David Marr and Ellen Hildreth. Theory of edge detection. *Proc. Royal Society*, 207:187–217, 1980.

[MP90] Jitendra Malik and Pietro Perona. Preattentive texture discrimination with early vision mechanisms. *J. Optical Society of America A.*, 7:923–931, 1990.

[MTT78] J. Anthony Movshon, I. D. Thompson, and D. J. Tolhurst. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *J. Physiology (London)*, 283:53–77, 1978.

[MZ92] Doug Miller and Steven W. Zucker. Efficient simplex-like methods for equilibria of nonsymmetric analog networks. *Neural Computation*, 4(2):167–190, 1992.

[Orb84] Guy Orban. *Neuronal Operations in the Visual Cortex*. Springer-Verlag, Berlin, 1984.

[Per92] Pietro Perona. Steerable-scalable kernels for edge detection and junction analysis. In *Second European Conference on Computer Vision*, pages 3–18, Santa Margherita Ligure, Italy, May 1992.

[PM91] Pietro Perona and Jitendra Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *Third International Conference on Computer Vision*, pages 52–57, Osaka, 1991. IEEE Computer Society.

[PTK85] Tomaso Pggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[PW84] Murray H. Protter and Hans F. Weinberger. *Maximum Principles in Differential Equations*. Springer-Verlag, New York, N. Y., 1984.

[PZ85] Pierre Parent and Steven W. Zucker. Trace inference, curvature consistency, and curve detection. McGill Research Centre for Intelligent Machines Technical Report TR-CIM-86-3, McGill University, Montréal, 1985.

[Raj68] Des Raj. *Sampling Theory*. McGraw-Hill, New York, N.Y., 1968.

[Rat65] Floyd Ratliff. *Mach Bands: Quantitative Studies on Neural Networks in the Retina*. Holden-Day, San Francisco, Ca., 1965.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In David E. Rumelhart, James L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, chapter 8, pages 318–362. The MIT Press, Cambridge, Mass., 1986.

[RS91] A. Ravishankar Rao and Brian G. Schunk. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 52(2):157–185, 1991.

[Sik60] Roman Sikorski. *Boolean Algebras*. Springer-Verlag, New York, N. Y., 1960.

[SL85] Robert Shapley and Peter Lennie. Spatial frequency analysis in the visual system. *Annual Review of Neuroscience*, 8:547–583, 1985.

[SM84] Robert A. Schumer and J. Anthony Movshon. Length summation in simple cells of cat striate cortex. *Vision Research*, 24(6):565–571, 1984.

[Spi65] Michael Spivak. *Calculus on Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus.* Mathematics monographs. W.A. Benjamin, New York, N.Y., 1965.

[Spi79] Michael Spivak. *Differential Geometry*, volume 1. Publish or Perish, Inc., Berkeley, Calif., 1979.

[Ste78] K. Stevens. Computation of locally parallel structure. *Biological Cybernetics*, 29:19–26, 1978.

[Ter84] Demitri Terzopoulos. *Multilevel Computational Processes for Visible Surface Reconstruction.* PhD thesis, MIT, Cambridge, Mass., 1984.

[Ull79] Shimon Ullman. *The Interpretation of Structure from Motion.* MIT Press, Cambridge, Mass., 1979.

[Voo87] H. Voorhees. Finding texture boundaries in images. Technical Report 968, M.I.T. A.I. Lab, Cambridge, Mass., 1987.

[Wal75] D. Waltz. Understanding line drawings of scenes with shadows. In Patrick H. Winston, editor, *The Psychology of Computer Vision.* McGraw-Hill, 1975.

[Wit86] A. Witkin. Scale space filtering. In A. Pentland, editor, *From Pixels to Predicates.* Ablex, Norwood, N.J., 1986.

[WM83] R. J. Watt and Michael J. Morgan. Mechanisms responsible for the assessment of visual location: theory and evidence. *Vision Research*, 23:97–109, 1983.

[You85] R. A. Young. The gaussian derivative theory of spatial vision: Analysis of cortical receptive field line-weighting profiles. Tech. Rep. GMR-4920, Gen. Motors Res., 1985.

[ZDDI88] Steven W. Zucker, Chantal David, Allan Dobbins, and Lee Iverson. The organization of curve detection: Coarse tangent fields and fine spline coverings. In *Second International Conference on Computer Vision*, pages 568–577, Tampa, Florida, December 1988. IEEE Computer Society.

[ZDI89] Steven W. Zucker, Allan Dobbins, and Lee A. Iverson. Two stages of curve detection suggest two styles of visual computation. *Neural Computation*, 1, 1989.

[ZIH90] Steven W. Zucker, Lee Iverson, and Robert Hummel. Coherent compound motion: Corners and non-rigid configurations. *Neural Computation*, 2:44–57, 1990.

[ZRD75] Steven W. Zucker, Azriel Rosenfeld, and Larry S. Davis. Picture segmentation by texture discrimination. *IEEE Trans. on Computers*, C-24:1228–1233, 1975.

[Zuc84] Steven W. Zucker. Type I and type II processes in early orientation selection. In T. Caelli and P. Dodwell, editors, *Figural Synthesis*. Lawrence Erlbaum Assoc., Hillsdale, N.J., 1984.

[Zuc86] Steven W. Zucker. The computational connection in vision: Early orientation selection. *Behavior Research Methods, Instruments and Computers*, 18:608–617, 1986.

Index