Trace Inference,
Curvature Consistency,
and
Curve Detection

Pierre Parent

B. Sc. Economiques. (Université de Montréal). 1975

Department of Electrical Engineering

McGill University

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Engineering

September 1986

Abstract

This thesis describes a novel approach to curve inference in digital images based on curvature information. It divides the inference procedure into two stages: a trace inference stage, and a curve synthesis stage. Only the trace inference stage is addressed in this thesis, the curve synthesis stage being the second part of a two-part research effort. It is shown that recovery of the trace of a curve requires that the tangent and curvature information be recovered at the same time. These make it possible to specify powerful constraints between estimated tangents to a curve (in terms of a neighbourhood relationship called co-circularity), and between curvature estimates (in terms of a curvature consistency relation). Because all curve information is quantized, special care must be taken to obtain accurate estimates of trace points, tangents and curvatures. This issue is addressed by the introduction of a smoothness constraint and a maximum curvature constraint. The procedure is applied to two types of images: *(i)* artificial images designed to evaluate curvature sensitivity and noise sensitivity, and *(ii)* natural images.

Résumé

Cette thèse décrit une nouvelle approche basée sur la courbure au problème de l'inférence de courbes dans les images digitales. Elle scinde la procédure d'inférence en deux étapes: une étape d'inférence de trace, et une étape de synthèse de courbe. Seule l'étape de l'inférence de trace sera traitée dans cette thèse, l'étape de synthèse de courbe étant la seconde partie d'un travail de recherche en deux volets. Il est démontré que la récupération de la trace d'une courbe exige que la tangente et la courbure soient récupérées en même temps. Ces informations permettent de spécifier des contraintes puissantes entre les tangentes estimées à une courbe (en termes d'une relation de voisinage appelée co-circularité), et entre les courbures estimées (en termes d'une relation de cohérence de courbure). Puisque toutes les informations sur les courbes sont discrètes, l'estimation précise des points de la trace, des tangentes et des courbures doit se faire soigneusement. Ce problème est résolu par l'introduction d'une contrainte de différentiabilité continue et d'une contrainte de courbure maximale. La procédure est appliquée à deux types d'images digitales: *(i)* des images artificielles conçues pour évaluer la sensibilité à la courbure et au bruit, et *(ii)* des images naturelles.

Acknowledgements

My sincerest love and gratitude go to Dina El Masri, my life companion, who has shared in the many sacrifices necessary to the accomplishment of this research, who has provided me with ample moral support and encouragement during times of difficulty and depression, and who is now eager, deservedly so, to share in the profits brought upon by the Masters' degree. High also in my esteem are those who helped put some shape to my ideas during many stimulating discussions: Kamal Gupta, Peter Sander, Norah Link, Frank Ferrie, Yvan Leclerc, and Benjie Kimia (in order of office assignment). Some of the typesetting of this thesis (the references and the bibliography) would not have been so easy were it not for Wade Hong, whom I also thank for his presence during some lunch outings, which made the meals tax-less, if not priceless. Many special thanks go to Michael "der Mouse" Parker, CVaRL's very own wizard, whose computer literacy will always amaze me, who seemed to always have the answer to my problem, and to whom I still owe a dinner at La Petite Halle (one of these days, Mike!).

Serious work is not possible without a decent amount of R and R. Thus, acknowledgements would be incomplete without mentioning those who made student life all the more enjoyable, mainly through sports and games: Paul Freedman, Pierre Sicard, Iskender Paylan, Abdol-Reza Mansouri, and Faycal Kahloun; or through pleasant afternoon breaks: Maria Mastromonaco, who liked to *fare quattro chiacchiere*, and Mindle Levitt, whose persistence in learning TeX I admire.

I sincerely thank my advisor, Steve Zucker, who, it goes without saying, was so indispensable to the completion of this work. In particular, he managed to drill some rigour into my scientific writing style, and I am grateful for this.

And finally, thank God, it's done!

Table of Contents

List of Figures

Chapter 1 Introduction

Curves arise from the projection of various kinds of structure in the visual world, such as occluding contours of objects, curvature extrema in surfaces, and discontinuities in surface coverings and lighting. But curves are not directly observable in images; rather, curves are abstract entities (mappings) and images consist only of intensities. All that is observable in images is information about the trace of curves, or information about the set of image locations through which the (projected) curve passes. The curve must then be inferred from this information. In this thesis we formulate such an inference process in terms of traces, tangents, and curvatures, and develop consistency relationships between them.

The formation of images of curves is a forward problem, and is well-posed. The inverse problem—the inference of curves from images—is not, however, since information is lost during the imaging process. Additional constraints must be found, and we seek them through an analysis of the discrete nature of the problem.. We show, in particular, how discretized versions of standard notions from differential geometry lead naturally to smoothness constraints, and how quantization leads to minimization as a method for using these constraints.

Our approach differs from others in two fundamental ways. First, the standard approach to inferring curves assumes that the trace points are known. In spline interpolation, for example, a collection of points is given, and polynomial values are sought

between them [Pavlidis, 1982]. In our formulation, however, the trace points must also be determined. If images were purely binary, with dark points corresponding to trace points, and with adjacent trace points on the curve adjacent in the image, then trace inference would be straightforward. But images contain structure other than the raw traces, so that a preliminary problem—the inference (or separation) of the trace from other image structure—must be solved as well. We therefore separate the curve inference process into two distinct stages, the first in which local information (such as the trace) is determined, and the second in which the global curve is inferred.

Other attempts at curve inference lumped the problem of inferring the trace of the curve together with the problem of inferring the curve. However, this mixes local and global information together, and makes it difficult to take advantage of interactions between them. Martelli [Martelli, 1976], for example, minimized a functional of intensity differences along the curve with a global constraint on curvature; however, it was still necessary to specify the initial and final points, and the final result was dependent on properties of the noise. In general the trace of a curve is not the straightest sequence of pixels with the minimal intensity change along them. Pavlidis also examined the minimization of global functionals through a split-and-merge procedure [Pavlidis, 1982].

Our decomposition of the curve inference process into two stages corresponds naturally to their differential geometry. We show that reliable trace inference requires information about tangents and curvatures as well, so the goal of the first stage is to recover the trace together with tangent and curvature fields. Once these fields are given, since the tangent is the first derivative of the curve with respect to arc length, integrals through them can be readily found within the second stage. But there is still something of a chicken-and-egg problem, since the exact recovery of the trace requires information about the curve, and vice versa. Our solution to this problem is to first recover the trace, tangent, and curvatures only coarsely, so that discontinuities can be properly placed, and then, in the second stage, to recover them more exactly.

The second sense in which this approach differs from standard ones is the

manner in which we seek the constraints necessary to accomplish trace. tangent. and curvature inference. In fitting surfaces to disparity data. for example. it is now an accepted practice to assume a physical model. e.g. that the surfaces consist of thin plates and membranes [Terzopoulos, 1986]. Energy considerations then lead to elegant minimizations of second-order functionals. However. it is not at all clear that such physical assumptions should motivate the trace inference process. We begin with standard notions in differential geometry. and discretize them onto quantized grids. This leads to an analagous formulation. but suggests that we include one more derivative than is normally assumed. Rather than minimizing a functional through curvature [Terzopoulos, 1986]. we (implicitly) obtain a functional through *curvature variation*. This additional derivative appears necessary for localizing discontinuities. The minimization is accomplished using standard relaxation labelling techniques. and this formulation substantially outperforms earlier. more heuristic attempts [Zucker, Hummel and Rosenfeld. 1977: Zucker and Parent. 1984]. Of course. for reasons of numerical stability. one must be careful how these derivatives are estimated. and we present a novel approach to this as well. It is much more accurate than those based. for example. on the chain code [Davis and Freeman, 1977].

This thesis is the first part of a two-part research effort. In this thesis we develop the inference of the (discrete) trace. tangent. and curvature fields. Given these fields. in the second part of the effort [Kimia and Zucker. in preparation] it is shown how to find integral curves through them. i.e.. how to actually infer the global curves and their discontinuities. We begin. in this thesis. by motivating the constraints. and end with several real examples that illustrate the robustness of the approach.

Two different kinds of information are lost during the curve imaging process: (1) information about the third dimension, through projection; and (2) details about small-scale variations, because of sampling. The latter—quantization noise—introduces significant uncertainty in positional information and reduces the image to a finite set. Consideration of the details of the quantization, and how they affect the discretization of concepts from differential geometry, forms the backbone of our approach.

2.1 The discrete trace of a curve

The entire effect of the imaging process can be formalized as follows. Let the curve B be a mapping $y : I \rightarrow E^3$, from an interval I on the real line to Euclidian 3-space, such that

$$y(t) = (y_1(t), y_2(t), y_3(t)) \tag{2.1}$$

is a continuous function of t, a parameter running along the curve. y_1, y_2 and y_3 are the Euclidian coordinates of the *trace* of B, that is, the image of the mapping. Through a projection operator Π, B maps to a curve C in the plane

$$B \xrightarrow{\Pi} C, \tag{2.2}$$

where the curve C is a mapping $x : I \rightarrow E^2$, with

$$x(t) = (x_1(t), x_2(t)) \tag{2.3}$$
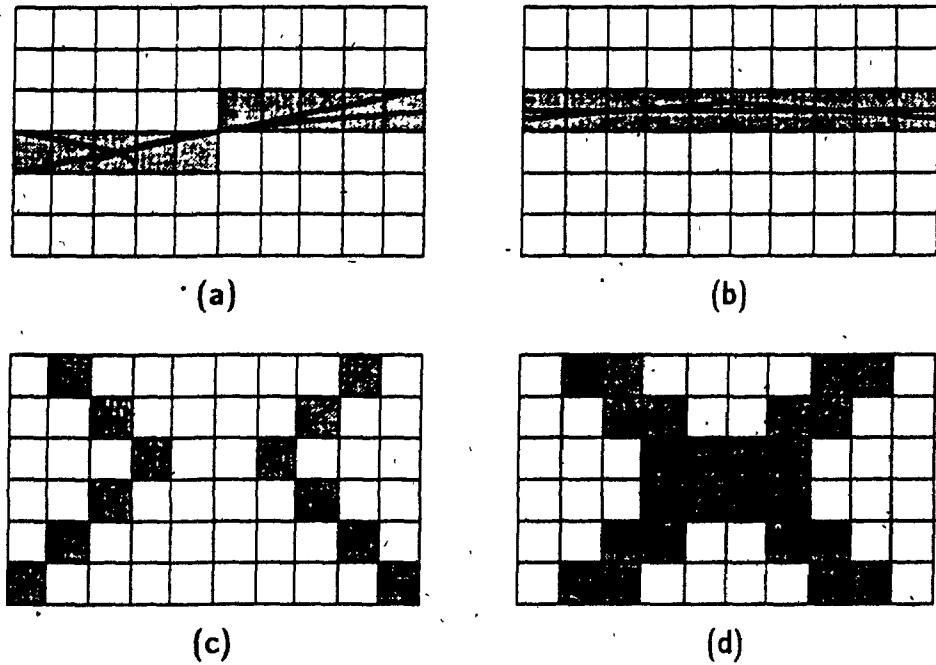
Figure 2.1   Various curves and corresponding discrete traces (shaded areas). (a) distinct planar curves may share a common discrete trace. (b) a small orientation change is undiscernible. (c) a corner and a bend of high curvature have identical traces. (d) two orientation changes in close proximity and a smooth bend have similar discrete traces.

being a continuous function of the parameter t. Finally, a sampling operator Σ takes the trace of C, which is the set $\{(x_1(t), x_2(t)) \mid t \in I\}$, into a discrete trace on a square sampling lattice with integer coordinates

$$\text{trace } C \xrightarrow{\Sigma} T. \tag{2.4}$$

T is a discrete trace, that is a set of points with integer coordinates. The sampling function is given by

$$\Sigma = \left\lfloor \mathbf{x}(t) + (\tfrac{1}{2}, \tfrac{1}{2}) \right\rfloor. \tag{2.5}$$

Observe that Σ is a many-to-one mapping, which maps all the points of the curve inside a unit square of the sampling grid to the center point of the square. Therefore, both the projection operator Π and the sampling operator Σ are non-invertible. Many distinct space curves—in fact an infinity of them, and some non-curves too—can give rise to identical projections. Likewise, distinct planar projections may have indistinguishable discrete traces, as depicted in Figure 2.1.

2.2 Smoothness assumptions permit trace inference

While the forward problems of obtaining a planar projection C from a space curve B, and obtaining a discrete trace T from the planar projection are well-posed, neither of the corresponding inverse problems are. Additional constraints are required to limit the family of solutions. In the case of the inverse projection problem, constraints about physical objects may come into play [Barrow and Tenenbaum 1981, Witkin 1981], while only general-purpose constraints, or constraints that must hold over large classes of images, are available to invert the sampling process. From this point on, we shall consider only the projected curve C. Since small-scale details are primarily what is lost by sampling, it is natural to impose a certain order of smoothness on the projected curve (except at discontinuities) giving rise to a given discrete trace, and nothing more.

The trace inference problem is further exacerbated by the fact that, in general, the trace of a curve is not directly observable in the image in which it is encoded. The trace itself must be inferred from the image intensities. We contend that the trace inference problem is closely linked to the sampling inversion problem, since the smoothness assumptions about the planar curve must influence the trace inference process. Thus it will be shown that it is not sufficient to infer only the trace of a curve, but that tangent and curvature fields must be inferred as well. Again, we formulate these discretely. The tangent and curvature fields embody the smoothness assumptions, and act as further constraints on the inverse sampling problem. Thus, the tangent and curvature estimates provide a local "model" of the curve in a neighbourhood around the putative trace point.

To illustrate the different orders of constraint, Figure 2.2 depicts a discrete trace to which continuous curves have been fitted. Given points in the discrete trace (a), a discrete orientation constraint (through the discrete tangent field) is added in (b), and then the combined orientation and curvature constraints (through the discrete tangent and curvature fields) are in (c). Observe that the curves in (c) and (b) satisfy the positional constraint in (a): that is, they all pass through the indicated positions. Similarly, the

6
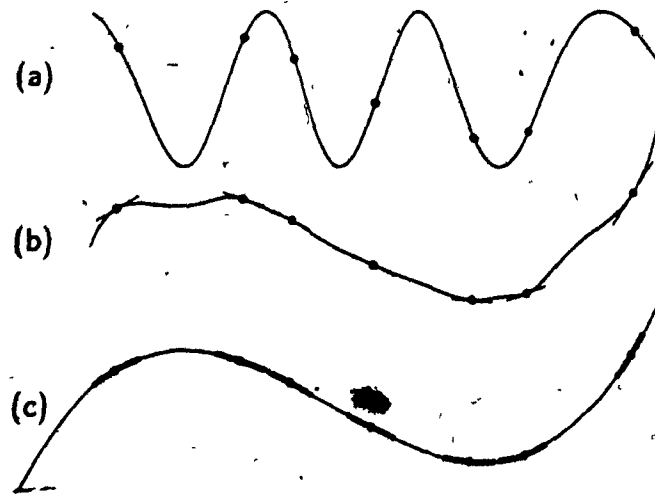
Figure 2.2 Three curves fitted to a given discrete trace: (a) positional constraint
only, one fit among a large family of curves with a broad spectrum of behaviour:
(b) position and orientation constraint. the family of curves is more constrained:
(c) position. orientation and curvature constraints combined with a smoothness
criterion. the family of solutions is reduced to a single curve Observe that the
curve in (c) satisfies the orientation (tangent) constraint and that both curves (c)
and (b) satisfy the positional constraint (a). But the curve in (a) does not satisfy
the tangent (b) or curvature (c) constraints

curves in (b) and (c) satisfy the tangent constraint in (b). but not (a). Finally. only

curve (c) satisfies a curvature constraint (depicted as short arcs of osculating circles in

(c)). and it is also the smoothest curve satisfying these combined constraints. Thus.

additional (smoothness) constraints limit the space of possible curves; what is required

for our problem is to provide sufficient constraints so that there is a unique curve which

satisfies them. The problem is then well-posed. We now start to concentrate on the

trace inference problem. and begin our search for constraints with a review of differential

geometry. More global smoothness constraints limiting the full space of possible curves

will. of course. influence the analysis. These will also be discussed in appropriate places.

2.3 Overview of differential geometry

.It is useful to review a few elementary notions of differential geometry [do

Carmo. 1976] to establish the context in which the smoothness constraints will be for-

mulated. The review will be centered on curves in the plane. although generalizations to

higher dimensional curves exist.

Let I be an interval in one-dimensional Euclidian space E^1. A curve C is defined as a continuous mapping $\mathbf{x} : I \to E^2$ from the interval to the plane, where

$$\mathbf{x}(t) = (x_1(t), x_2(t)), \tag{2.6}$$

where $t \in I$ is a parameter running along the curve, and x_1, x_2 are continuous functions of t. The curve is said to have order of continuity k, denoted C^k, if all derivatives up to and including the kth derivative of x_1 and x_2 are continuous. Taking the first derivative with respect to t everywhere along C, we obtain the tangents

$$\mathbf{x}'(t) = (x_1'(t), x_2'(t)), \tag{2.7}$$

where the vectors $\mathbf{x}'(t)$ have bases at $(x_1(t), x_2(t))$. Their magnitudes can be interpreted as the velocity of a particle following the curve. A curve may be reparameterized in terms of its arc length s, equivalent to a particle traveling at constant unit velocity along the curve. In this case, the tangent vectors are unit length vectors.

$$\mathbf{x}'(s) = (x_1'(s), x_2'(s)), \tag{2.8}$$

where $s = f(t)$ is a reparameterization of the curve, and $\|\mathbf{x}'\| \equiv 1$.

The interesting aspect of the tangent is its orientation. The geometric interpretation of the tangent to a curve is depicted in Figure 2.3 (a). Letting P be a point on a curve, and A a neighbouring point, the tangent T at P is the limit of the line AP as A approaches P along the curve. The tangent yields the orientation of a curve at a point.

Taking the second derivative with respect to s everywhere along C, we obtain

$$\mathbf{x}''(s) = (x_1''(s), x_2''(s)), \tag{2.9}$$

where the vector $\mathbf{x}''(s)$ is normal to the vector $\mathbf{x}'(s)$, and the magnitude of $\mathbf{x}''(s)$ is called the curvature of C. Curvature is a measure of the rate of change of orientation per unit arc length. The geometric interpretation for the curvature is depicted in Figure 2.3 (b). Let P be a point on a curve, T the tangent at that point, and A a neighbouring point on the
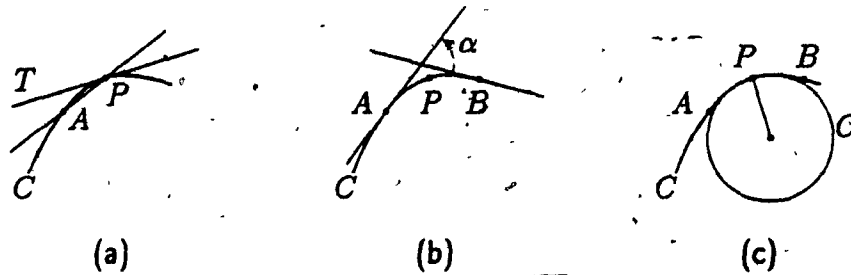
Figure 2.3 (a) Tangent T is the limit of segment $\overline{PA}$ as A approaches P along C. (b) The curvature κ of C at P is the limit of the ratio $\alpha/|\widehat{AB}|$ as A and B approach P independently along C. (c) The osculating circle O at P is the limit of the circle that passes through A. P and B as A and B approach P independently along C The osculating circle in (c) is a substantially better local approximation than the tangent in (a).

curve. Let α denote the angle between the line AP and T. The curvature κ at P is the limit of the ratio $\alpha/|AB|$ as A approaches P along the curve. Related to this interpretation of curvature is the osculating circle. Referring to Figure 2.3 (c). let A. P and B be three neighbouring points on a curve. and let O be a circle through these points. As A and B independently approach P along the curve, the circle O converges towards a limit. whose radius is precisely the inverse of the curvature κ at P.

2.4 Derivatives through curvature consistency

Although third and higher order derivatives can be defined for curves. practical considerations dictate that the process must stop somewhere. Our position is that the trace, tangent and curvature fields provide the local basis for inferring global curves. and are necessary for placing discontinuities. corners. or breakpoints. Qualitatively for people, interesting events along curves consist only of abrupt changes of orientation and curvature. local maxima of curvature. and inflection points (i.e. zero crossings of curvature) [Attneave. 1959; Hoffman and Richards. 1986; Koenderink and van Doorn. 1982; Fischler. 1983]. These are the places that a human observer is most likely to choose to segment long curves into shorter ones. Higher order discontinuities such as discontinuities in curvature variation do not seem to matter. Between the selected corner points. curves simply appear

to be smooth.

Although this argument for limiting the number of derivatives considered is informal and based on human perception, any machine vision system will have to confront this issue as well. While any number of derivatives can be defined, only a finite (and, in fact, small) number of them can actually be computed. Evolution, presumably, has settled upon an optimal number.

In terms of differential geometry, then, the visual system would appear to perceive curves as being piecewise-C^∞, with segmentation occuring at discontinuities in the first and second derivatives of the curves. In the sequel, we shall show that curvature consistency—a necessary relationship between discrete estimates of curvature along a smooth curve—amounts to a bound on the third derivative, i.e. on the curvature variation. Higher order discontinuities are smoothed over and ignored.

There is a numerical reason for limiting the analysis to curvature variation as well. Quantization can be modeled as the addition of "noise" [Papoulis, 1965], and each derivative numerically amplifies this noise. It is well known that the numerical stability of computing higher-order derivatives is poor. Although we shall come back to this point later in this thesis, for now suffice it to note that some number of derivatives are necessary to control smoothness and to signal discontinuities, and we shall take that number to be 3. Furthermore, real care must be used in computing them.

2.5   Tangent and curvature fields

So far, we have thought about curves intrinsically, i.e., as given functions $x(s)$ of an arc-length parameter s. However, since this is primarily the object we are after, and not given, it is necessary to formulate some of our algorithms extrinsically in terms of the Euclidian space in which the curves are embedded. Consider a retinotopic restriction of the plane E^2 to finite extent $D \subset E^2$. We shall be interested in 2 vector *fields* on D, one which is a mapping that associates a tangent vector $x'(s)$ to each point in D, and the

other which associates a curvature $x''(s)$. We refer to these fields as tangent and curvature fields, respectively. The fields form the basis of our representation for curves. Thus, those points in the trace of a curve will have unit tangents with associated curvatures mapped to them, while others will have null tangents and curvatures.

2.6 Discrete representation of trace, tangent, and curvature

Since our problem begins in the discrete domain, we choose a representation for curves based on their traces, with associated tangent fields and curvature fields. These latter fields are represented discretely as well, to reflect the fact that curve inference consists of a two-stage process. In this first step, the goal is to estimate the trace, tangent, and curvature fields finely enough so that discontinuities can be placed, but coarsely enough so that overwhelmingly restrictive assumptions are not made. The compromise solution, then, is with discrete tangent and curvature fields, which serve as inequality bounds on which the next, global stage can be based.

The discrete trace of a curve consists of a set of points in the discrete plane. The discrete tangent field is formed by finding, for each point of the discrete trace, the quantized orientation of the curve as it runs through that point. It thus consists of a set of unit tangents to curves, characterized by their position and orientation. Hence, assuming m discrete orientations, and letting θ_λ denote the discrete orientation of the tangent at a particular position, $\lambda = 1, \ldots, m$, the actual orientation θ^* lies in the interval

$$\theta_\lambda - \frac{\pi}{2m} \le \theta^* < \theta_\lambda + \frac{\pi}{2m}. \tag{2.10}$$

The discrete curvature field is similarly formed by associating to each unit tangent the quantized curvature at that point.

It is desirable to have a uniform representation for all points in the discrete plane, capable, among other things, of distinguishing between trace points and non-trace points. If, for example, the orientation of a curve is allowed to have one of 8 values, i.e.

the orientation is quantized to multiples of $\epsilon = \pi/8$, each point of the discrete plane could be associated with a vector of 8 elements, each one a predicate *true* or *false* according to whether or not a curve passes through that point with approximately that (quantized) orientation. Alternately, the *true* or *false* predicates could be replaced by real numbers in the interval $[0,1]$, where the extremes assert presence or absence of a curve with absolute certainty, while intermediate values represent less certain assertions. Thus, for curve points, there is at least one element of the certainty vector with a value near 1, while for non-curve points, all elements are near 0. Some points may have more than one near-1 value, e.g. curve crossings, and orientation discontinuities. The following notation is used for the certainty of tangent λ at position (x_i, y_i):

$$p_i(\lambda) \quad \text{for } i = 1, \ldots, n \quad \lambda = 1, \ldots, m. \tag{2.11}$$

assuming an image with n pixels and m discrete orientations of tangents at each pixel.

With each orientation vector element $p_i(\lambda)$ is associated a single discrete measure of curvature, $\kappa_i(\lambda)$. It becomes part of the discrete curvature field when the corresponding tangent is part of the discrete tangent field.

2.7 Curve inference as a two-stage procedure

Sufficient background material has now been developed to specify the two stages involved in inferring a curve.

Stage 1: *Trace Inference and Orientation Selection*

Taking an image as input, infer the discrete trace, tangent, and curvature fields subject to quantization and maximal curvature constraints.

Stage 2: *Curve Synthesis*

Taking the discrete trace, tangent, and curvature fields as input, locate discontinuities and find integral curves running through them, subject to discontinuity and smoothness constraints.

Stage 2, curve synthesis, will be treated in subsequent work [Kimia and Zucker, in preparation]. We now concentrate on Stage 1.

Chapter 3   Trace inference and orientation selection

The goal of the first stage of our curve inference process is the recovery of local information. Clearly this must include the recovery of trace points. If the curve were known, then trace points could be separated from other image structure simply by calculating them. But the curve is not known, so we are forced to estimate the structure of the curve in the neighborhood of each putative trace point. As we shall show, coarse estimates of the tangent and curvatures provide sufficent local information about the curve. These estimates provide a partial local model for the curve sufficient to gather evidence about individual trace points from their neighbours.

Two terms used above—local and coarse—warrant further expansion, because they are related in a fundamental way. Observe that, when searching for a book in the library, one first searches through broad categories before finely scanning the exact titles. Analogously, curve recovery is facilitated by first obtaining a rough—or coarse—estimate of its structure to guide subsequent analyses. The need for a local analysis follows for similar reasons, since few (if any) assumptions can be made *a priori* about the global structure of the curve. Moreover, given the presence of noise from both sensors and quantization, such a coarse, local analysis becomes necessary: imagine trying to exactly estimate the tangent of a contour from an image to three decimal places without strict *a priori* assumptions, such as the straightness of the sides of a block [Binford, 1981]. Seeking higher-order approximations suffers the same problems as well.

Similar arguments could be made in detail about the tangents and curvatures.

If the curve were known, then these could be computed exactly. But since it is not, then they must be estimated as well. There is something of a hierarchy of information here, with the (estimated) tangents supplying constraints on the positions of nearby (estimated) trace points, with the (estimated) curvatures supplying constraints on the (estimated) tangents and their (estimated) locations; and, finally, with curvature consistency relationships supplying constraints on (estimated) curvatures. We now develop these constraints in detail, based on quantizations of the differential geometry already described. In the end we will have obtained an inference procedure for estimating (quantized) trace, tangent, and curvature fields such that a particular functional with terms through curvature variation is minimized.

3.1   The two steps of Stage 1

The Stage 1 inference procedure consists of two distinct steps, a measurement step and an interpretation step. The functional minimization and tangent field inference are accomplished in the second step.

Step 1: *Measurement*

Convolution of linear operators against the image to obtain initial tangent certainty estimates at each position and for each (quantized) orientation.

Step 2: *Interpretation*

Selection of a subset of the tangents signaled in Step 1 according to the functional minimization procedure to be defined.

Classically, of course, the linear operators amount to "line detectors" [Rosenfeld and Kak, 1976], although our selection procedure is much more complicated than simply taking the "strongest" convolutions. Rather, it amounts to selecting those "strong" convolutions that are strongly supported by—or consistent with—the other convolutions in

Figure 3.1 The initial convolutions are performed with this "line detector" operator which is a difference of three Gaussians in the x direction, multiplied by a single Gaussian in the y direction

their neighbourhood according to the estimated local curve model. We now discuss the two steps in turn.

3.1.1 Step 1: Initial tangent estimates

The requirement for the first step is a set of operators that estimate the presence of a tangent at each position in the image. Since discretely the tangent can be viewed as a short, straight line segment, templates tuned to this structure are the obvious candidates. Such templates amount, of course, to so-called "line detectors" [Rosenfeld and Kak, 1976; Zucker, 1982], and we use the following one (see Fig. 3.1):

$$\mathcal{G}(x,y) = LSF(x) \cdot \exp(-y^2/\sigma_y^2) \tag{3.1.a}$$

with

$$LSF(x) = \exp(-x^2/\sigma_1^2) - B\exp(-x^2/\sigma_2^2) + C\exp(-x^2/\sigma_3^2) \tag{3.1.b}$$

where B and C are the normalized weights of the second and third Gaussians.

The classical rationale for choosing such operators is clear; they are template representations of short, straight line segments. The Gaussian kernels have the attractive property that they smooth over intensity variations along the tangent direction, but sharpen them in the orthogonal direction [Zucker and Hummel, 1986]. Such operators resemble the receptive fields of so-called simple-cells in primate visual cortex [Hubel and Wiesel, 1977], and hence are also attractive from a biological point of view.

3.1.2 Step 2: Interpreting the initial tangent estimates

Classical treatments of curve detection also involve two steps, the first of which is very similar to the one just described. But the second-step—interpretation of the operator convolutions—is usually much simpler than the scheme that we shall be describing. Since the operator templates match high-contrast straight lines so well, it is often assumed that simply selecting the strongest convolutions is sufficent for obtaining a local representation of the contour (what we are calling a tangent field). But this is not the case for any pattern other than widely spaced, straight lines. Curvature, corners, and nearby contours all affect the convolutions, and all are sufficent to invalidate the maximum convolution selection strategy [Zucker, 1982: 1985]. A richer model for curves is clearly needed.

From the differential geometry reviewed in Chapter 2, it is clear that our model must at least include curvature. Recall that, in the neighborhood of each point, the osculating circle is a substantially better approximation to the curve than the tangent (Fig. 2.3). We shall later argue that curvature is also a high-enough approximation to separate closely spaced curves, so that incorrect convolutions that cover distinct curves can be properly interpreted. Therefore we shall focus on curvature, and shall begin to derive an estimation procedure based on (a quantized version of) it. Our goal, briefly stated, is to minimize the curvature variation at each point by maximizing circularity.

Orientation, curvature,

smoothness, and position constraints

In this chapter, we shall first establish the neighbourhood circularity measure in terms of a pairwise relation between (estimated) tangent elements, called *co-circularity*, which determines an orientation constraint. Introducing the maximum curvature constraint that arises from grid quantization is then straightforward. Secondly, it will be shown that the orientation constraint is not sufficient and that interaction between neighbours should be mediated by a *curvature consistency* constraint. This constraint can be applied provided local estimates of curvature are available. It will be shown that these estimates can be obtained by partitioning neighbourhoods into regions called curvature classes, and that tangent estimates can be obtained by propagating support through these regions. Consistency of curvature is achieved by comparison of the curvature classes. Thirdly, the smoothness constraint is examined in relation to the maximum curvature constraint and the neighbourhood size.

Finally, a level of constraint is required to achieve the high localization accuracy which is characteristic of curvilinear patterns in images. This positional constraint is a form of lateral maximum selection whereby a pixel-wide region is determined to be part of the curve. It is required because support for tangent elements may occur over a relatively wide area near curves, whereas the discrete trace of a curve is composed only of the set of pixels through which the curve actually passes. This level of constraint thus improves the accuracy of the trace inference procedure.

4.1 Co-circularity

The standard approach to estimating curvature is to fit a polynomial to a collection of points, and then to differentiate the polynomial twice [Pavlidis. 1982]. This, however, amplifies noise. and hence is unusable in our quantized context. We shall present a different scheme. in which the information contained in tangent estimates provides the basis for curvature estimates through the co-circularity relationship.

4.1.1 Definition

The relation of co-circularity applies to distinct tangents to a circle (see Figure 4.1(a)). A property of this spatial configuration is that the tangents form angles of equal magnitude. but of opposite sign. with the line joining the points of tangency (see Figure 4.1(b)). Thus. abstraction can be made of the circle. and the symmetry of the configuration can be retained as the characteristic of co-circularity. Note that when the radius of the circle becomes infinite. we obtain a special case of co-circularity; co-linearity. Co-circularity is therefore a function of both the orientations and the positions of the tangents.

Definition. *Two unit tangents λ and λ' are co-circular, denoted*

$$\lambda \asymp \lambda'$$

iff there exists a circle to which they are both tangent.

Co-circularity is a kind of symmetry relation between tangents. and bears some relationship to the way in which [Brady and Asada. 1984] define a local symmetry. But our discretization and use of the notion differs substantially from theirs.

4.1.2 Co-circularity in the discrete case

When position and orientation are quantized. tangent pairs are seldom exactly co-circular. The small perturbations introduced by quantization must somehow be taken

(a)   (b)
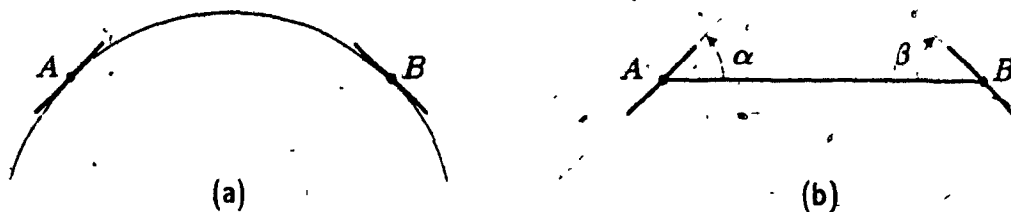
Figure 4.1 In (a). unit tangents A and B are both tangent to the same circle. therefore A is co-circular to B (denoted $A \asymp B$) This condition is geometrically equivalent to that depicted in (b). where $\alpha = -\beta$.
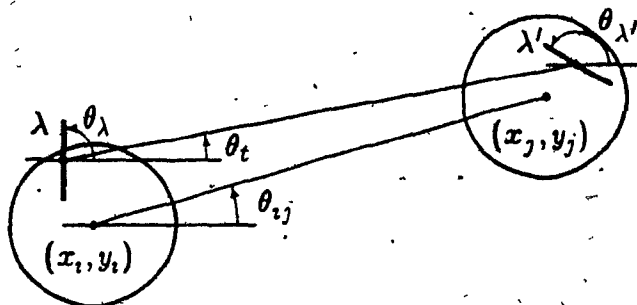
Figure 4.2 Two unit tangents λ and λ' with respective orientations θ_λ. $\theta_{\lambda'}$ whose positions are restricted to the circle of radius $1/2$ centered at the pixel positions (x_i, y_i). (x_j, y_j). the line with orientation θ_{ij} joining the centers of the pixels; the line with orientation θ_t joining the tangents.

into account. To do this. we begin by allowing the position of the tangents to be anywhere within the circle of radius $1/2$ pixel centered at the pixel. Likewise. we let their orientation vary within a neighbourhood of size ϵ. for orientations quantized to multiples of ϵ. The tangent pair is thus co-circular if there exists at least one assignment of the position and orientation variables for which co-circularity as defined above is true.

Let (x_i, y_i) and (x_j, y_j) be the coordinates of nodes i and j. and let (x, y) be an arbitrary point within the circle of radius $1/2$ centered at (x_i, y_i). and (x', y') a point in a circular neighbourhood of (x_j, y_j); let λ and λ' be unit tangents at these locations and θ_λ and $\theta_{\lambda'}$ be their respective orientations; let θ be an orientation in an ϵ-neighbourhood of θ_λ. and θ' an orientation in an ϵ-neighbourhood of $\theta_{\lambda'}$ (see Figure 4.2). The orientation of the line joining the centers of the pixels is given by

$$\theta_{ij} = \arctan(\Delta y / \Delta x) \tag{4.1}$$

where $\Delta x = x_j - x_i$ and $\Delta y = y_j - y_i$.
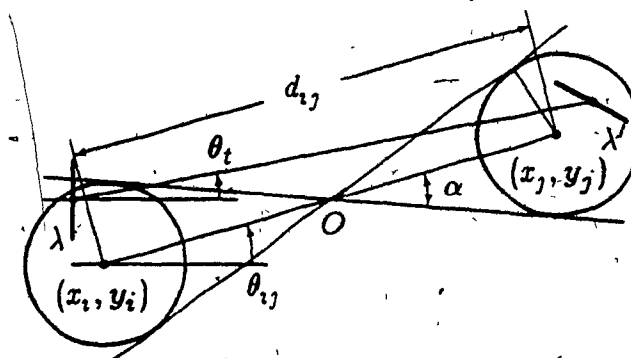
Figure 4.3 The interval for the orientation θ_t of the line joining two unit tangents λ and λ' located in circles of radius $1/2$ centered at (x_i, y_i) and (x_j, y_j) is $(\theta_{ij} - \alpha, \theta_{ij} + \alpha)$, where θ_{ij} is the orientation of the line joining the centers of the circles, and α depends on the distance d_{ij} separating them. The sine of α is $1/2$ divided by the distance from O to (x_i, y_i), which is $d_{ij}/2$, hence $\alpha = \arcsin(1/d_{ij})$.

We wish to determine the minimum and maximum values that may be taken by θ_t, the orientation of the line joining the tangents as (x, y) and (x', y') are allowed to vary within their circular neighbourhoods. As in Figure 4.3, the extrema coincide with the two intersecting tangents common to the circular neighbourhoods. In the case of circles of equal radii, it can be shown that the angle between the common tangent and the line joining their centers is given by

$$\alpha = \arcsin(1/d_{ij}) \qquad (4.2)$$

Let the function $\Gamma(\beta, \gamma)$ designate the interior angle between a pair of lines with orientations β and γ, as in Figure 4.4. Let the sign of this function be the same as the direction in which the first line must rotate in order to close the interior angle and coincide with the second, that is, positive for counterclockwise and negative for clockwise rotations. The formal definition of this function, assuming that $0 \leq \beta \leq \pi$ and $0 \leq \gamma \leq \pi$, is the following

$$\Gamma(\beta, \gamma) = \begin{cases} \gamma - \beta, & \text{if } |\gamma - \beta| \leq \frac{\pi}{2}; \\ \gamma - \beta - \pi, & \text{if } \frac{\pi}{2} < \gamma - \beta \leq \pi; \\ \gamma - \beta + \pi, & \text{if } -\pi \leq \gamma - \beta < -\frac{\pi}{2}. \end{cases} \qquad (4.3)$$

Turning back to Figure 4.2 and recalling the geometrical definition of co-circularity, we find that the tangents at (x_i, y_i) and (x_j, y_j) are co-circular if the interior angle
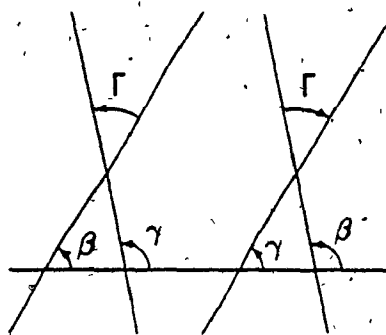
Figure 4.4 The interior angle function $\Gamma(\beta,\gamma)$ is the interior angle between a pair of lines with orientations β and γ. The sign of this function is determined by the direction in which the first line must rotate in order to close the interior angle.

between the first tangent and the line joining the tangents is the same as that between the latter line and the second tangent. Formally,

$$\lambda \asymp \lambda' \quad \text{iff} \quad \Gamma(\theta,\theta_t) = \Gamma(\theta_t,\theta'), \tag{4.4}$$

for some $\theta_t \in (\theta_{ij} - \alpha, \theta_{ij} + \alpha)$, $\theta \in (\theta_\lambda - \frac{\epsilon}{2}, \theta_\lambda + \frac{\epsilon}{2})$, $\theta' \in (\theta_{\lambda'} - \frac{\epsilon}{2}, \theta_{\lambda'} + \frac{\epsilon}{2})$. This condition is clearly equivalent to

$$\lambda \asymp \lambda' \quad \text{iff} \quad |\Gamma(\theta_\lambda,\theta_{ij}) - \Gamma(\theta_{ij},\theta_{\lambda'})| < \epsilon + 2\alpha. \tag{4.5}$$

Condition (4.5) is a discrete co-circularity condition: it is either true or false. A continuous version of this condition can be implemented by measuring the closeness to co-circularity, and we refer to this measure as a co-circularity coefficient. Departure from exact co-circularity occurs by rotation of one of the tangents, which suggests that a function of the difference in orientation between a tangent and the co-circular tangent in the same position could be used as a measure of co-circularity. The co-circularity coefficient then consists of a real number between 0 and 1, where 0 means not co-circular, 1 means co-circular, and values close to 1 are interpreted as nearly co-circular.

Generally, the coefficient is 1 for a certain range of orientations of tangents at the neighbouring node, because of built-in quantization noise tolerance in (4.5). Denote this range by $[\theta_{min}, \theta_{max}]$. We let the magnitude of the coefficient fall off monotonically outside this range (see Figure 4.5).
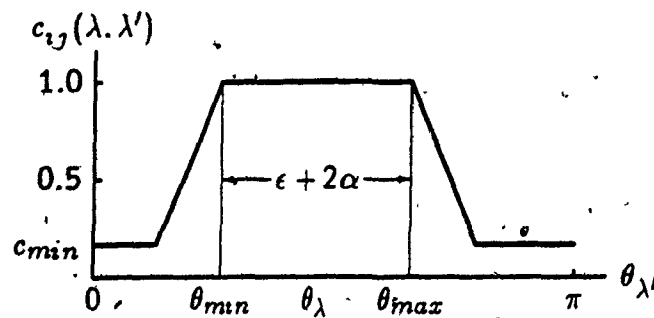
Figure 4.5 Co-circularity coefficient $c_{ij}(\lambda, \lambda')$ as a function of orientation $\theta_{\lambda'}$ of neighbouring tangent. The length of the interval $[\theta_{min}, \theta_{max}]$ is $\epsilon + 2\alpha$ (see Condition 4.5).

Thus the co-circularity coefficients $c_{ij}(\lambda, \lambda')$ are given by

$$c_{ij}(\lambda, \lambda') = \begin{cases} 1, & \text{if } |\Gamma(\theta_\lambda, \theta_{ij}) - \Gamma(\theta_{ij}, \theta_{\lambda'})| < \epsilon + 2\alpha: \\ \max(1 - \eta|\theta_{\lambda'} - \theta_m|, c_{min}), & \text{otherwise.} \end{cases}$$

(4.6)

where, θ_m is the extreme of the range $[\theta_{min}, \theta_{max}]$ closest to $\theta_{\lambda'}$. and η is the absolute value of the slope of the drop-off region. assuming a linear decrease.

Because of the grid and tangent quantization. it is necessary to consider tangents distributed in a neighbourhood around each image point. If only the 3 by 3 immediate neighbours were considered. then the angular quantization of θ_{ij} would be much too severe. The co-circularity coefficient so defined can be measured for all neighbouring tangents in a neighbourhood of a given size. and the set of these measures forms the neighbourhood support set.

4.1.3 Maximum curvature constraint

In order to introduce the maximum curvature constraint (recall Fig. 2.1(c)). a measure of the radius of curvature implied by a pair of tangents is required. Letting $\hat{\rho}_{ij}(\lambda)$ denote the implied radius, we use

$$\hat{\rho}_{ij}(\lambda) = \frac{d_{ij}}{2\sin|\Gamma(\theta_\lambda, \theta_{ij})|}$$

(4.7)

as its measure. The maximum curvature constraint implies that

$$c_{ij}(\lambda, \lambda') = 0 \quad \text{whenever} \quad \hat{\rho}_{ij}(\lambda) < \rho_{min}.$$

(4.8)

23

Figure 4.6(a) shows the set of neighbours co-circular to a tangent with a vertical orientation for a neighbourhood diameter of 15 with the maximum curvature constraint applied.

4.2 Co-circularity support

We are now in a position to estimate how well a particular (estimated) tangent is supported by other (estimated) tangents in its neighbourhood. Recall that the first measurement stage consisted in convolutions against "line detectors" (Section 3.1). Letting θ_{λ_i} denote the orientation of the operator at node i with coordinates (x_i, y_i), for $\lambda = 1, \ldots, m$, the normalized convolutions $\{p_i(\lambda), i = 1, \ldots, n, \lambda = 1, \ldots, m\}, 0 \leq p_i(\lambda) \leq 1$, provide an initial estimate of the confidence in tangent λ at node i. Note, for a long straight line of orientation θ_{λ^*} passing through i, that $p_i(\lambda^*)$ will be maximal at that position, and that $p_i(\lambda)$ will drop off from $p_i(\lambda^*)$ according to the *orientation tuning curve* for the operator.

But when the curve does not consist of long straight lines, $p_i(\lambda)$ can follow a more complex distribution at each node i. Therefore, the circularity measure must be evaluated over a local neighbourhood around i and must depend on the entire distribution of possible tangents at each point. We take a linear weighted sum to indicate the co-circularity support for a unit tangent λ at position i:

$$s_i(\lambda) = \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}(\lambda, \lambda') p_j(\lambda') \tag{4.9}$$

where $r_{ij}(\lambda, \lambda') = c_{ij}(\lambda, \lambda')$, the co-circularity coefficient. Clearly those tangents lying along a curve should have maximal support. More remains to be done before this is guaranteed, however, because linear averaging schemes such as this have the potential to smooth across different but nearby (within the given neighbourhood) curves.

4.3 Curvature classes

Consider a small neighbourhood of an image containing many curves. Within this neighbourhood, many tangent pairs are mutually co-circular, with some co-circular

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

(e)   (f)   (g)   (h)
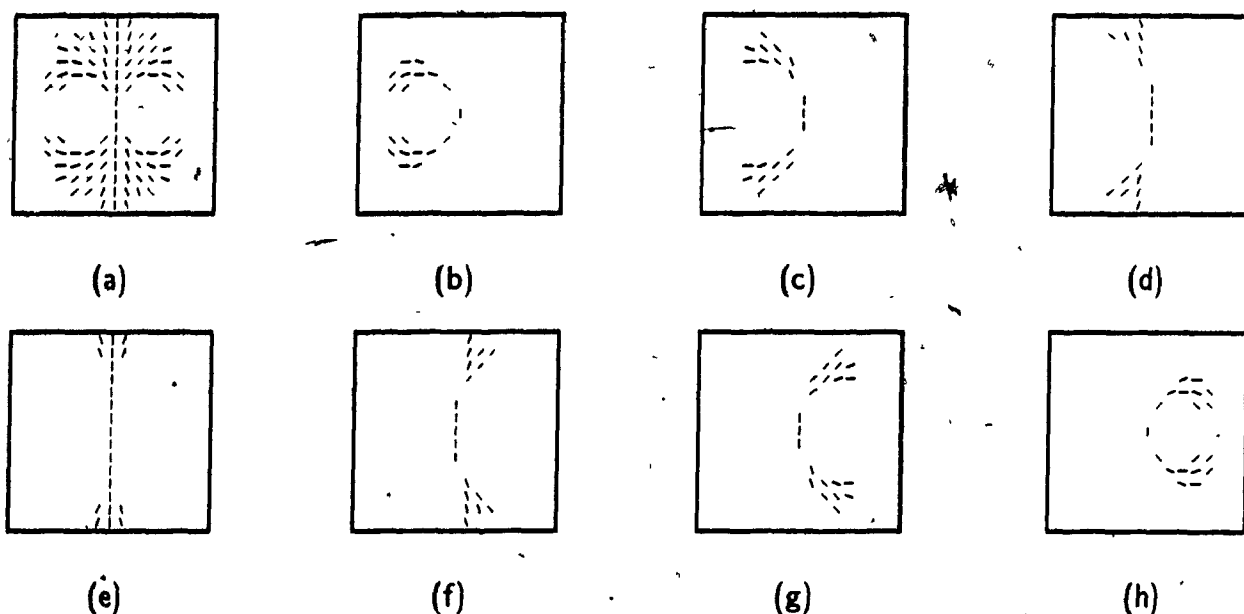
Figure 4.6 A diameter 15 neighbourhood (a), partitioned into 7 curvature classes for the vertical orientation (b) — (h). (a) bears some resemblance to the consistency operators for curve enhancement developed in [Hedlund, Granlund and Knutsson, 1982]

pairs belonging in fact to distinct curves. More specifically, given three tangents A, B and C in such a neighbourhood and given that $A \asymp B$ and $A \asymp C$, it does not follow necessarily that $B \asymp C$. In particular, the interpretation of A remains ambiguous when $B \not\asymp C$: does A belong to the curve going through A and B, or to the one through A and C? One way to decide the situation is to partition the neighbourhood support set about A into sufficiently narrow curvature classes $K_k(A)$, $k = 1, \ldots, K$ as in Figure 4.6(b)-(h).

Each curvature class consists of all the osculating circles whose radius is between the limits for that class or, equivalently, whose curvature is within certain limits. Thus, if $A \asymp B$, $A \asymp C$, and B, C belong to the same curvature class $K_k(A)$ with respect to A, it may be concluded that $B \asymp C$.

Two benefits accrue from the use of partitioning. First, it imposes n-*wise* consistency of tangents within curvature classes at a low cost in complexity. The tangent support function can be modified to take advantage of the partition, by measuring support independently by curvature class, and by selecting the highest as the final tangent support.
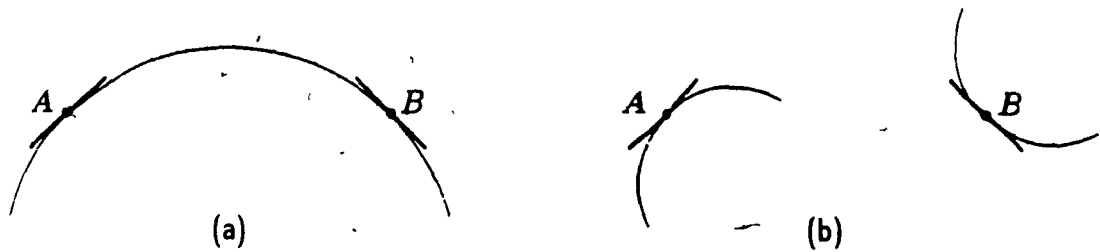
Figure 4.7 In (a). $A \asymp B$ and A and B are tangent to the same curve. In (b) however. the spatial configuration of A and B is the same as in (a). but they are tangent to distinct curves.

The support of tangent λ at node i is thus given by

$$s_i(\lambda) = \max_{k=1,K} \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}^k (\lambda, \lambda') \, p_j(\lambda'), \tag{4.10}$$

where the coefficient $r_{ij}^k(\lambda, \lambda')$ is the product of the co-circularity coefficient $c_{ij}(\lambda, \lambda')$ and a partitioning function

$$K_{ij}^k(\lambda, \lambda') = \begin{cases} 1, & \text{if } \rho_{min}^k \le \hat{\rho}_{ij}(\lambda) \le \rho_{max}^k; \\ 0, & \text{otherwise}; \end{cases} \tag{4.11}$$

for given, curvature class limits ρ_{min}^k and ρ_{max}^k.

The second benefit is that a discrete estimate of curvature $\kappa_i(\lambda)$ is obtained by correspondence with the curvature class that maximizes the support function. With this estimate, it is be possible to introduce a further constraint on the selection of neighbouring tangents for local support. This constraint is examined in the next Section. First note, however, that these discrete curvature estimates have been obtained without the numerical problems inherent in standard, spline-fitting techniques.

4.4 Consistency of curvatures

One kind of ambiguity persists even after partitioning into curvature classes. Consider for example Figures 4.7(a) and (b).

In Figure 4.7(a). tangents A and B are co-circular and they are tangent to the same curve. In Figure 4.7(b) however. tangents A and B occur in the same spatial

(a)   (b)
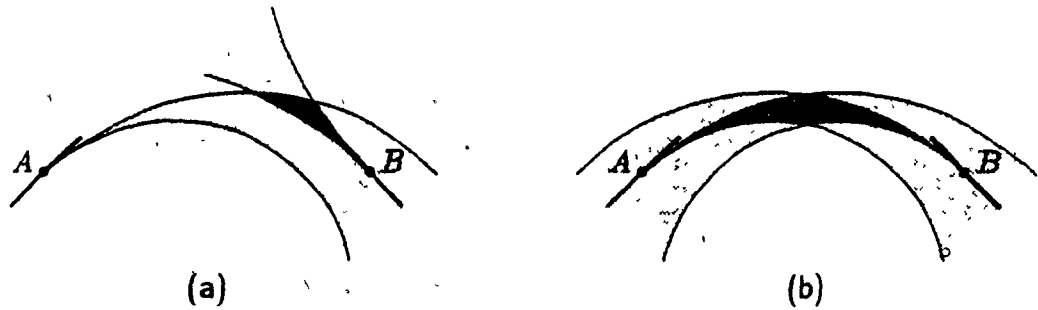
Figure 4.8 In (a), $A \asymp B$ but the estimated curvature at B is inconsistent with the interpretation of a curve through A and B. In (b) however, mutual support is possible because the curvatures are consistent

configuration as in (a), yet they are tangent to *distinct* curves. Should the tangents A and B mutually support each other in (b)? If not, how can configurations (a) and (b) be told apart?

The solution to this problem requires comparison of local curvature class estimates. The orientation of a tangent and the local curvature class estimate together determine a region about the position of the tangent where a curve is most likely to exist. In Figure 4.8, $B \asymp A$, therefore B belongs to at least one of the curvature classes of A. But in Figure 4.8(a), the local curvature class estimate at B does not include A as a member. The curvatures are said to be inconsistent and this condition precludes mutual support of A and B. Figure 4.8(b) depicts a situation where curvatures are consistent with the interpretation that a curve passes through A and B. Note that in this case we have both $A \in K_k(B)$ and $B \in K_{k'}(A)$ for some k and k' (not necessarily equal).

Letting $C_{ij}^{kk'}(\lambda, \lambda')$ denote the predicate variable for curvature consistency, i.e.

$$C_{ij}^{kk'}(\lambda, \lambda') = \begin{cases} 1, & \text{if curvature class } k \text{ of } \lambda \text{ at } i \text{ is consistent} \\ & \text{with estimated curvature class } k' \text{ of } \lambda' \text{ at } j; \\ 0, & \text{otherwise;} \end{cases} \quad (4.12)$$

we obtain a new coefficient

$$r_{ij}^{kk'}(\lambda, \lambda') = c_{ij}(\lambda, \lambda') \, K_{ij}^k(\lambda, \lambda') \, C_{ij}^{kk'}(\lambda, \lambda'), \quad (4.13)$$

which is a function of curvature as well as orientation and position.

27

4.5   Smoothness constraint

So far. it has been assumed that the tangent and curvature fields embodied the smoothness constraint without explicitating its actual mechanism. In this Section. the smoothness constraint will be discussed in relation to the circularity measure. which is a function of the maximum curvature constraint and the neighbourhood size.

The smoothness constraint takes on two distinct but related forms. First and foremost. it is mediated through the maximum curvature constraint. whose role is to filter out quantization noise introduced by discretization of the image. The upper bound on maximum curvature is dictated directly by the resolution of the discrete image. As depicted in Figure 2.1 (c). very high curvature is indistinguishable from a corner. so the limit is at the point where the maximum curvature leads to a trace different from a corresponding corner. Experiments with the trace inference procedure suggest that the minimum radius of curvature should be in the range of 2.5 to 3.0 pixels.

The second form of smoothness constraint depends on the size of the neighbourhood used to collect support for unit tangents along a curve. It is implicitly assumed that the osculating circle is a good approximation of the neighbourhood. that is. the curvature variation is small within a given neighbourhood size. Thus. the greater the neighbourhood is. the less sensitive the circularity measure will be to small details in the curve. and the smoother the inferred trace will be. For example. a regular polygon would be perceived as a circle using a sufficiently large neighbourhood. Chapter 6 describes an experiment showing the effect of varying neighbourhood size on inferred trace and tangent and curvature fields.

Neighbourhood size and maximum curvature are not totally independent. In general. the minimum radius of curvature scales with neighbourhood size. just as the radius of curvature limits of the curvature classes (assuming a fixed number of curvature classes). As will be seen in the next chapter. the extent of the neighbourhood is defined in such a way that all circles tangent to the center of the neighbourhood have equal arc length within the neighbourhood. This is one solution to avoiding the problem of preference for

28

certain curvature classes over others. Thus, the minimum radius of curvature must scale with neighbourhood size. Another approach to the curvature preference problem would be to adjust the co-circularity coefficients to offset the effects of variable intra-neighbourhood arc lengths. Then, a large neighbourhood size could be chosen while maintaining a low minimum radius of curvature, perhaps at the cost of increasing the number of curvature classes to cover the extended range more effectively. This solution was not developed due to the high complexity cost of increasing the number of curvature classes.

4.6 Lateral maxima

In this Section, we address the problem of inferring the trace of a curve from the certainties associated with the tangents during our procedure. The objective is to extract a pixel-wide region about a curve. The method is based on comparisons between certainties in a small neighbourhood, with selection of the tangents with highest certainty, i.e. the lateral maxima.

4.6.1 Techniques for extracting lateral maxima

The rationale for lateral maximum selection is the observation that the support function exhibits a characteristic maximum at the precise location of a curve. The magnitude of the support function decreases gradually on either side of this location.

Lateral maximum selection is, in principle, a simple technique, but its implementation on a discrete grid requires careful consideration. The most obvious problem is that, on an orthogonal grid, only tangents with orientation parallel to a grid axis have lateral neighbours with which to compare. For other orientations, interpolated values for lateral neighbours must be used, rather than the neighbour closest to the ideal position. A straightforward linear interpolation based on a plane fitted to the three nearest neighbours is quite sufficient. A tangent λ at (x, y) is therefore a lateral maximum if its certainty is maximal among all tangents in the 3 by 3 neighbourhood in position-orientation space.

Letting $m_i(\lambda)$ denote the predicate variable for a lateral maximum for orientation λ at node i. we have

$$m_i(\lambda) = \begin{cases} 1, & \text{iff } p_i(\lambda) > p_l(\lambda'). \ \forall\lambda' \in \{\lambda - 1, \lambda, \lambda + 1\}. \text{ and} \\ & \quad p_i(\lambda) > p_r(\lambda'). \ \forall\lambda' \in \{\lambda - 1, \lambda, \lambda + 1\}. \text{ and} \\ & \quad p_i(\lambda) > p_i(\lambda'). \ \forall\lambda' \in \{\lambda - 1, \lambda + 1\}: \\ 0, & \text{, otherwise:} \end{cases} \tag{4.14}$$

where $p_l(\lambda')$ and $p_r(\lambda')$ denote the certainties of the left and right. or interpolated left and right certainties for tangent λ'.

As an alternative to interpolated certainties. one could also perform a strict comparison against a set of neighbouring certainties determined by the orientation of the certainty under test. Comparison sets can be defined in such a way that the selection process is stable. More formally.

$$m_i(\lambda) = \begin{cases} 1, & \text{iff } p_i(\lambda) > p_j(\lambda'). \ \forall\lambda' \in \{\lambda - 1, \lambda, \lambda + 1\}. \ \forall j \in N_\lambda^k: \\ 0, & \text{otherwise:} \end{cases} \tag{4.15}$$

where the neighbour set N_λ^k is a predefined set depending on the local estimated orientation λ and estimated curvature k.

The lateral maximum property of neighbouring tangents can be used as an additional constraint in the support function. Since only those tangents in the neighbourhood that are lateral maxima are compatible with a curve interpretation (observe, that the initial convolutions drop off with lateral displacement from the curve). the support function should be correspondingly restricted. Thus, we obtain a new expression for the support of a tangent:

$$s_i(\lambda) = \max_{k=1,K} \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}^{kk'}(\lambda, \lambda') \ p_j(\lambda') \ m_j(\lambda'). \tag{4.16}$$

The net effect of this constraint is to further narrow the region near curves where tangents receive high support.

Since the expression (4.16) for the support of a tangent should be maximized at each position, a natural choice for a global functional is the average local support

$$A(p) = \sum_{i=1}^{n} s_i(\lambda) p_i(\lambda) \qquad (5.1)$$

Qualitatively, the $p_i(\lambda)$ indicate which tangents and positions are chosen, and the $s_i(\lambda)$ indicate how mutually consistent they are through the quantized geometrical constructs just developed. That is, the $s_i(\lambda)$ codes the local model for the curve in the neighbourhood of tangent λ at node i. In particular, as the positional quantization $\Delta i \to 0$, orientation quantization $\Delta \lambda \to 0$, and the curvature classes approach the actual curvature, $s_i(\lambda) \to 1$ for all tangents along the (smooth) curve, and $s_i(\lambda) \to 0$ elsewhere.

Relaxation labelling is a procedure for maximizing expressions such as (5.1) [Hummel and Zucker, 1983], and in the following Section, we review the relaxation labelling procedure and tailor it to this application.

5.1 Overview of relaxation labelling

Relaxation labelling is an iterative procedure applied over a network of nodes. Associated with each node is a set of labels, and associated with each label is a measure of confidence, or certainty. Let there be n nodes and, for the sake of simplicity, a unique set of m labels at each node. Further, let $p_i(\lambda)$ denote the confidence of label λ at node

i. The values of the $p_i(\lambda)$ are restricted to the interval $[0, 1]$. and are subject to the added constraint that at each node *i*,

$$\sum_{\lambda=1}^{m} p_i(\lambda) = 1 \quad \text{for } i = 1, \ldots, n. \tag{5.2}$$

In vector notation. this constraint can be written as $\bar{p}_i \cdot \mathbf{1} = 1$. where $\mathbf{1}$ is the m-dimensional vector of 1's $(1, 1, \ldots, 1)$. The degree of compatibility between a label and its neighbourhood can be measured by what is known as the label's *support*. which is a function of other label certainties in the neighbourhood and their compatibility (pair-wise) with the label being supported. The constraints between labels are represented by a matrix of compatibilities. $r_{ij}(\lambda, \lambda')$. which serve to embody the problem-dependent knowledge. In this notation. $r_{ij}(\lambda, \lambda')$ denotes the compatibility between label λ' associated with node *j* and label λ associated with node *i*. We use the following expression for the support $s_i(\lambda)$ of label λ at node *i*:

$$s_i(\lambda) = \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}(\lambda, \lambda') p_j(\lambda') \tag{5.3}$$

Relaxation labelling is the process of achieving consistency. Hummel and Zucker [Hummel and Zucker. 1983] defined consistency in variational terms: they also proved that. for symmetric compatibility coefficients. consistent states of the relaxation network. maximized functionals of the form $A(p)$ in (5.1). Such maxima are achieved iteratively: beginning with an initial labelling $\{p_i^0(\lambda)\}$. the iteration

$$p_i^{k+1}(\lambda) = f(p_i^k(\lambda) : s_i^k(\lambda)) \tag{5.4}$$

continues until convergence. Hummel and Zucker [Hummel and Zucker. 1983] develop a general scheme for the iteration (5.4) utilizing the Mohammed [Mohammed, Hummel and Zucker. 1983] projection operator. However. the efficiency of this scheme can be improved substantially for this application: see [Parent and Zucker. 1985] and Appendix A.

5.2 The relaxation graph

The original approach [Zucker, Hummel and Rosenfeld. 1977] to representing curves in a relaxation graph used m orientations of tangents to a curve at each node. plus

32

the *no-line* label, with certainties subject to condition

$$\sum_{\lambda=1}^{m+1} p_i(\lambda) = 1, \quad \text{for } i = 1, \dots, n. \tag{5.5}$$

According to this convention, a labelling is consistent and certain only if at most one orientation is chosen by the relaxation process. However, curve intersections and corners require that multiple orientations be chosen at certain nodes.

Our solution is to have not one but many superimposed relaxation graphs, one for each label $\lambda = 1, \dots, m$. We shall refer to this collection as a relaxation *hyper-graph*. Each node of graph λ requires 2 labels: λ and *no-line*. Interactions are permitted between the graphs through the compatibility coefficients.

Note that the certainty vector at each node of each graph consists of two components with unit sum. Clearly, nothing is gained by representing both components of the vector explicitly, one of the components being simply the complement of the other. Similar savings can be achieved in the compatibility matrix by eliminating the *no-line* label completely.

Consider the contribution of node j to the support of the two labels (0 and 1) at node i. Letting $r_{11} = r_{ij}(1,1)$, $r_{10} = r_{ij}(1,0)$, etc., and $p_j = p_j(1)$, we obtain, from (5.3),

$$s_i(1) = r_{10}(1 - p_j) + r_{11}p_j \tag{5.6}$$

$$s_i(0) = r_{00}(1 - p_j) + r_{01}p_j. \tag{5.7}$$

Assuming that $r_{00} = r_{11}$, $r_{01} = r_{10} = -r_{11}$, the above are reduced to

$$s_i(1) = 2 \cdot r_{11}\left(p_j - \frac{1}{2}\right) \tag{5.8}$$

$$s_i(0) = -s_i(1), \tag{5.9}$$

where both expressions are void of terms related to label 0. Hence, for this special case, (and many similar ones where all compatibility coefficients are functions of a single one),

it is possible to reduce the complexity of the relaxation graph and the compatibility matrix. The assumptions about the compatibility coefficients are not unreasonable when label 0 is interpreted as *no-line*. This result shows that if certain labels are left out and only the compatibility coefficients between the remaining labels are specified, the system is equivalent to another one where all the labels and compatibility coefficients are specified. Thus, truncation of the graph and compatibility matrix does not violate relaxation labelling theory. The result is extensible to systems of more than two labels.

5.3 Complete relaxation model

To summarize, a sketch of the complete relaxation model as it will be used in the trace inference process is as follows:

Hyper-graph: m graphs of n nodes, one for each orientation. The labels represent tangents with orientations quantized to multiples of $\epsilon_\varphi = \frac{\pi}{m}$. The "no-line" label is not explicitly represented. The m labels at node i have *independent* certainties $p_i(\lambda) \in [0,1]$.

Support function: the support $s_i(\lambda)$ of label λ at node i is a normalized function of the sum of the products of neighbouring label certainties by appropriately chosen compatibility coefficients, as in Eq. (4.16).

Update formula: each $p_i(\lambda)$ is updated as though in a two-label graph, with the other label interpreted as "no-line" and having complementary support, i.e. $-s_i(\lambda)$. $p_i(\lambda)$ is updated in such a way as to converge *asymptotically* to 1 when excited, and to 0 when inhibited (see Appendix A).
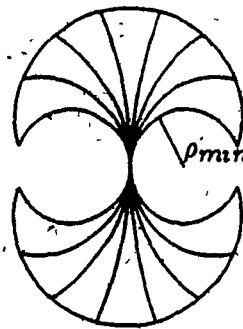
Figure 5.1 An equi-length neighbourhood for the vertical orientation with 7 curvature classes. All arcs of circle, vertically tangent to the center of the neighbourhood have the same length. ρ_{min} $(= 1/\kappa_{max})$ is the minimum radius of curvature. which limits the extent of the neighbourhood according to the maximum curvature constraint.

5.4 Normalization details

Two details remain, but they are central to successful implementations. First, the support functions must be normalized to account for arc-length effects, and secondly, they have to be mapped into a common interval so that values are comparable across positions. We discuss each of these normalizations in turn.

5.4.1 Extent of neighbourhood support

The unit tangent support function has so far been described in terms of co-circularity coefficients, but nothing has been said about the shape or extent of the neighbourhood over which tangents interact. We could assume a circular neighbourhood shape, determined by the neighbourhood diameter and the maximum curvature constraint. However, this circular shape is not ideal because arcs of different curvature have distinct intra-neighbourhood lengths, implying a bias of the support function towards longer arcs.

In order to make the support of a tangent to a curve independent of the particular curvature, the extent of the neighbourhood is adjusted so that arcs of circles with all curvatures have the same intra-neighbourhood length.

Letting R denote the maximum arc length distance of interaction between unit

35

tangents. the curve describing the boundary of the equi-length region is given by

$$\rho = R|\sin c(\frac{\pi}{2} - |\theta|); \quad \text{for} \quad -\pi \le \theta \le \pi \tag{5.10}$$

in polar coordinate form. Figure 5.1 depicts such a neighbourhood. with the minimum radius of curvature $r_{min} = 1/\kappa_{max}$ constraint applied. The neighbourhood of node i is thus composed of all unit tangents whose estimated arc length distance from (x_i, y_i) $\hat{D}_{ij}(\lambda) \le R$. The estimated arc length distance can be obtained from the estimated radius of curvature $\hat{\rho}_{ij}(\lambda)$ with

$$\hat{D}_{ij}(\lambda) = 2\hat{\rho}_{ij}(\lambda)|\Gamma(\theta_\lambda, \theta_{ij})|. \tag{5.11}$$

Letting $E_{ij}(\lambda)$ denote the neighbourhood extent predicate. we have

$$E_{ij}(\lambda) = \begin{cases} 1, & \text{if } \hat{D}_{ij}(\lambda) \le R: \\ 0, & \text{otherwise.} \end{cases} \tag{5.12}$$

5.4.2 Intra-pixel length correction

Next. we consider the intra-pixel length correction to the compatibility coefficients. This is necessary in order to insure that the process is isotropic. A straight line of a given length intersects a different number of pixels of a digital grid depending on its orientation. For example. diagonal lines intersect fewer pixels than lines of the same length at any other orientation. while the lines parallel to a grid axis intersect the most. Unless a compensation is introduced. the process will therefore "prefer" orthogonal orientations to diagonal ones. A simple normalization coefficient is given by

$$l(\lambda') = \frac{\sqrt{2}}{2\cos\omega_{\lambda'}}, \tag{5.13}$$

where $\omega_{\lambda'}$ is the angular difference between $\theta_{\lambda'}$ and the nearest grid axis.

With these two compensations. we obtain the final expression for the compatibility coefficients by multiplying the expression of Equation (4.13) by $E_{ij}(\lambda)$ and $l(\lambda')$. giving

$$r_{ij}^{kk'}(\lambda, \lambda') = c_{ij}(\lambda, \lambda')\, E_{ij}(\lambda)\, K_{ij}^k(\lambda, \lambda')\, C_{ij}^{kk'}(\lambda, \lambda')\, l(\lambda'), \tag{5.14}$$

where $c_{ij}(\lambda, \lambda')$ is the co-circularity coefficient. $K_{ij}^k(\lambda, \lambda')$ is the curvature class partitioning function. and $C_{ij}^{kk'}(\lambda, \lambda')$ is the curvature consistency predicate.

5.4.3   Normalization of the support function

The support function obtained previously is

$$s_i(\lambda) = \max_{k=1,K} \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}^{kk'}(\lambda, \lambda') \, p_j(\lambda') \, m_j(\lambda').$$ (5.15)

but its range must be normalized before use.

For a given neighbourhood radius, one can compute the integral of the local compatibility coefficients given that a single curve (assume a straight line) traverses the entire neighbourhood in the proper orientation. Denote this integral by s_{max}. This sum determines the maximum support that a label can obtain from its neighbourhood, and the sum varies according to the radius of the neighbourhood. This is the maximum support that can be achieved.

The minimum acceptable support for a label depends both on geometry and noise. First, the process should be stable near the end of a genuine (non-noise) curve; that is, the curve should neither grow nor shrink during relaxation. Second, a threshold can be established according to the response of the initial operator convolutions in the presence of pure noise. Now, consider the unit tangent at the end of a low contrast line. The operator response, which is sensitive to contrast, will be rather low at that point, while the support for the tangent would be approximately $s_{max}/2$ (the curve traverses only half the neighbourhood), times the average certainty of the tangents on the curve. Assuming a minimum contrast criterion, p_{min}, we fix the minimum acceptable support of a label as

$$s_{min} = \frac{p_{min} \, s_{max}}{2}$$ (5.16)

With this choice of s_{min}, relative stability at end-lines is insured for a wide range of initial contrasts. If, on the other hand, the initial contrast of a curve is below the minimum, then that curve will gradually shorten until it disappears.

The label support is normalized by mapping the interval $[s_{min}, s_{max}]$ linearly into the interval $[0,1]$. The required normalized support, $S_i(\lambda)$, is therefore

$$S_i(\lambda) = \frac{s_i(\lambda) - s_{min}}{s_{max} - s_{min}}. \qquad (5.17)$$

It can readily be seen that this expression is equal to 1 when the raw support $s_i(\lambda)$ equals the maximum support, s_{max}, while the numerator vanishes if the raw support equals the minimum acceptable support, s_{min}. Of course, anything less for the raw support leads to a negative normalized support. The normalization of the support occurs before the projection of the support vector at a node onto the positive quadrant; see Appendix A.

This completes the discussion of the technical issues related to the implementation of a discrete trace inference process within a relaxation labelling computational framework. We turn our attention now to the results of the trace inference process over various kinds of images.

In discussing the experiments performed with the trace inference process described in the preceding Chapters, we will go from simple to complex. Thus, in the first part of this Chapter, we discuss experiments based on artificial images designed to evaluate specific features of the process, such as sensitivity to curvature, robustness in the presence of noise, and the effect of neighbourhood size on the smoothness of the inferred trace. At the same time, the trace inference process is compared to other procedures to extract curves from images, in the presence of large amounts of noise. Further experiments based on real images, such as satellite and bio-medical imagery, and fingerprints, are discussed in the second part of this Chapter.

6.1 Artificial images

6.1.1 Sensitivity to curvature

The first experiment is designed to evaluate sensitivity to curvature. Referring to Figure 6.1, the image is composed of 4 concentric circles whose radii were chosen to match individual curvature classes. The following parameters are used for the initial operators (see Eq. (3.1) for the parameterized form of the operator):

$$\sigma_1 = 1.14 \quad \sigma_2 = 1.8 \quad \sigma_3 = 2.28 \quad \sigma_y = 3.6$$

$$B = 1.266 \quad C = 0.5 \tag{6.1}$$

	nearly straight
⊢⊷	low curvature
⊷	medium curvature
⊶	high curvature
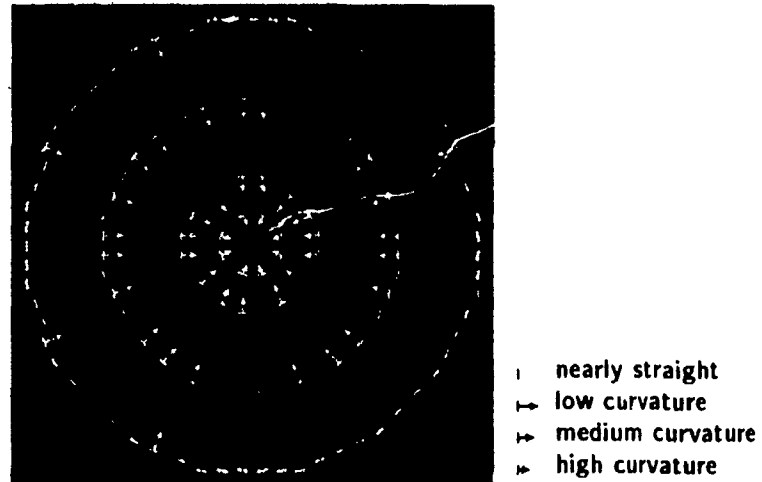
Figure 6.1 Trace inference process on concentric circles after 2 iterations. The resulting tangent field (short segments) and curvature field (arrows pointing to center of curvature) are superimposed on the image (filled pixels) The assigned tangent and curvature fields are perfect everywhere except at a few locations were quantization affects the local structure of the curve more severely These errors would disappear using a larger neighbourhood size

A neighbourhood diameter of 15 pixels is used. with 7 curvature classes determined by the following radius limits (in pixels):

$$\text{radius limits:}\quad 2.7, 4.2, 7.2, 21.0 \tag{6.2}$$

The·circles have the following radii:

$$\text{radii of circles:}\quad 3.5, 6.8, 15.0, 23.0 \tag{6.3}$$

The result displayed in Figure 6.1 is after 2 iterations. with step size 1. and using a supporting threshold $s_{min} = 0.5$. In these displays, tangents are indicated by short line segments, and curvatures by vectors pointing toward the center of the osculating circle. The magnitude of the curvature vector is proportional to the radius of curvature.

6.1.2   Sensitivity to noise

The next experiment is designed to evaluate the effect of noise on the trace inference process. The image consists of a single hand-drawn curve on a uniform background. featuring a sharp orientation discontinuity and varying curvature. The experiment

40

is performed independently for two neighbourhood sizes, 15 and 25 pixels respectively, and for each size there are 5 noise levels: $S/N = \infty$, 1.8, 0.9, 0.6, and 0.45. The noise added to the image has uniform distribution $U(0, A)$, where A is the peak-to-peak amplitude of the noise. Given that the curve has constant intensity I_c over a background of constant intensity I_b, the S/N ratio is obtained by

$$S/N = \frac{|I_c - I_b|}{A} \qquad (6.4)$$

Both experiments use 7 curvature classes, and the radius limits for each size are as follows:

$$\text{size 15 radius limits:} \quad 2.7, 4.2, \ 7.2, 21.0$$
$$\text{size 25 radius limits:} \quad 4.5, 7.0, 12.0, 35.0 \qquad (6.5)$$

The initial convolutions are with operators whose size is adjusted to the respective neighbourhood sizes. For the smaller neighbourhood size, the parameters are those in (6.1), and for the larger one, the parameters are as follows:

$$\sigma_1 = 1.9 \quad \sigma_2 = 3.0 \quad \sigma_3 = 3.8 \quad \sigma_y = 6.0$$
$$B = 1.266 \quad C = 0.5 \qquad (6.6)$$

The results displayed in Figure 6.2 are after 2 iterations, using a supporting threshold $s_{min} = 0.5$. The performance of the trace inference process in the presence of noise is very satisfactory, especially for the larger neighbourhood size.

To emphasize these results, a comparison between various other methods for selecting curves is displayed in Figure 6.3. These are based on the same curve as in Figure 6.2, at $S/N = 0.45$, the noisiest case.

In (a), an optimal intensity threshold is chosen, interactively, so that most of the curve points are selected, but this results in too many non-curve points being selected at the same time. Significant additional processing would be required to remove these noise points.

Neighbourhood size 15   Neighbourhood size 25
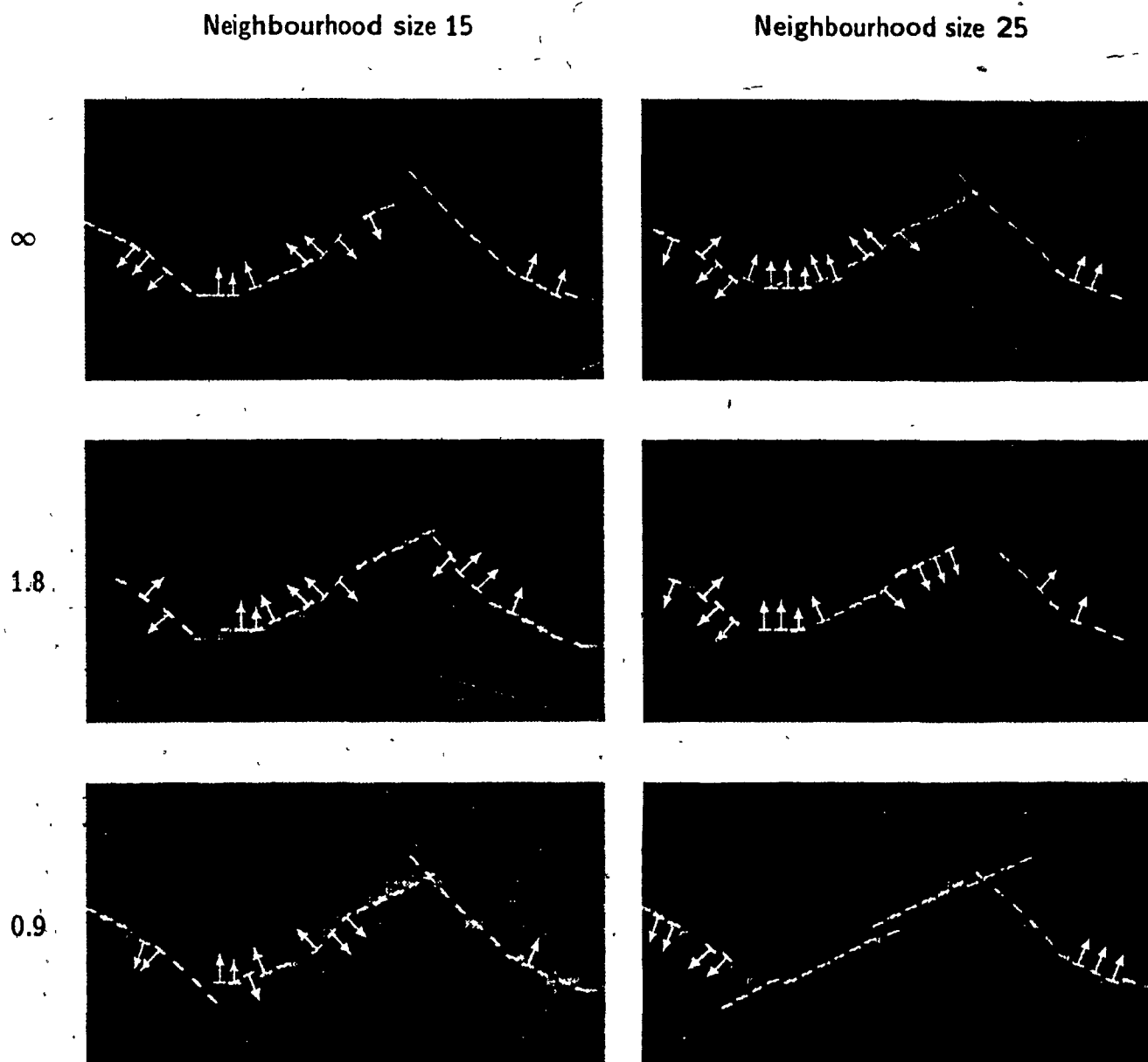
Figure 6.2 Trace inference on an image consisting of a single curve. using two neighbourhood sizes. and with the addition of various amounts of noise; to obtain S/N ratios of ∞, 1.8, 0.9 on this page and ratios 0.6 and 0.45 on the following page. After two iterations. only those tangents with certainties above 0.5 are displayed. The smaller neighbourhood size results in fairly stable inference down to $S/N = 0.9$, while the larger neighbourhood size remains quite stable for S/N down to 0.45, where the curve is nearly imperceptible at close range.

In (b). a threshold is applied to the lateral maxima based on an initial operator size of .25. as for the large neighbourhood of Figure 6.2. This result is in fact the 0th iteration of the trace inference process. and hence resembles a process of selecting the maximal response at each position. Again. to obtain all curve points. a low threshold must

Neighbourhood size 15 Neighbourhood size 25

0.6

0,45
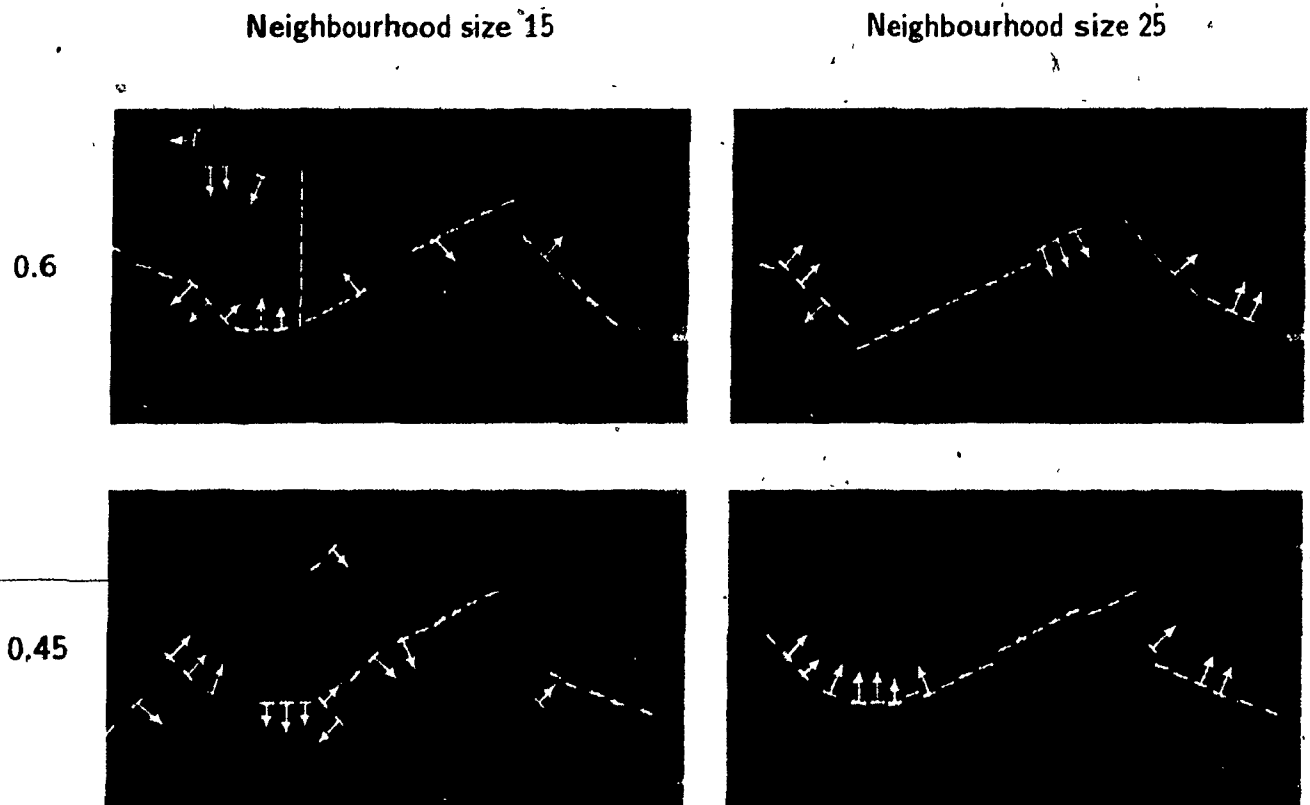
Figure 6.2 (continued) Noise sensitivity experiment for $S/N = 0.6$ and 0.45.

be chosen, resulting in too many non-curve points, and significant post-processing would be required.

In (c), the result displayed comes from an earlier attempt to formulate an inference process [Zucker and Parent, 1984] in which consistency is achieved through comparison of expected versus observed operator responses. This method is corner-sensitive, and slightly curvature-sensitive, and thus represents an improvement over an earlier method using only tangent information [Zucker, Hummel and Rosenfeld, 1977], the result of which is not displayed here. The effective neighbourhood diameter for this experiment is 3. Thus, this procedure has only slight curvature sensitivity, and does not have a built-in noise removal capability for short low-contrast segments as in the trace inference process. It is no surprise then, that this method also degrades rapidly in the presence of noise.

Finally, in (d), the result for the largest neighbourhood at $S/N = 0.45$ from Figure 6.2 is reproduced. Comparison with the rest of the figures clearly indicates the

(a)

(b)

(c)

(d)
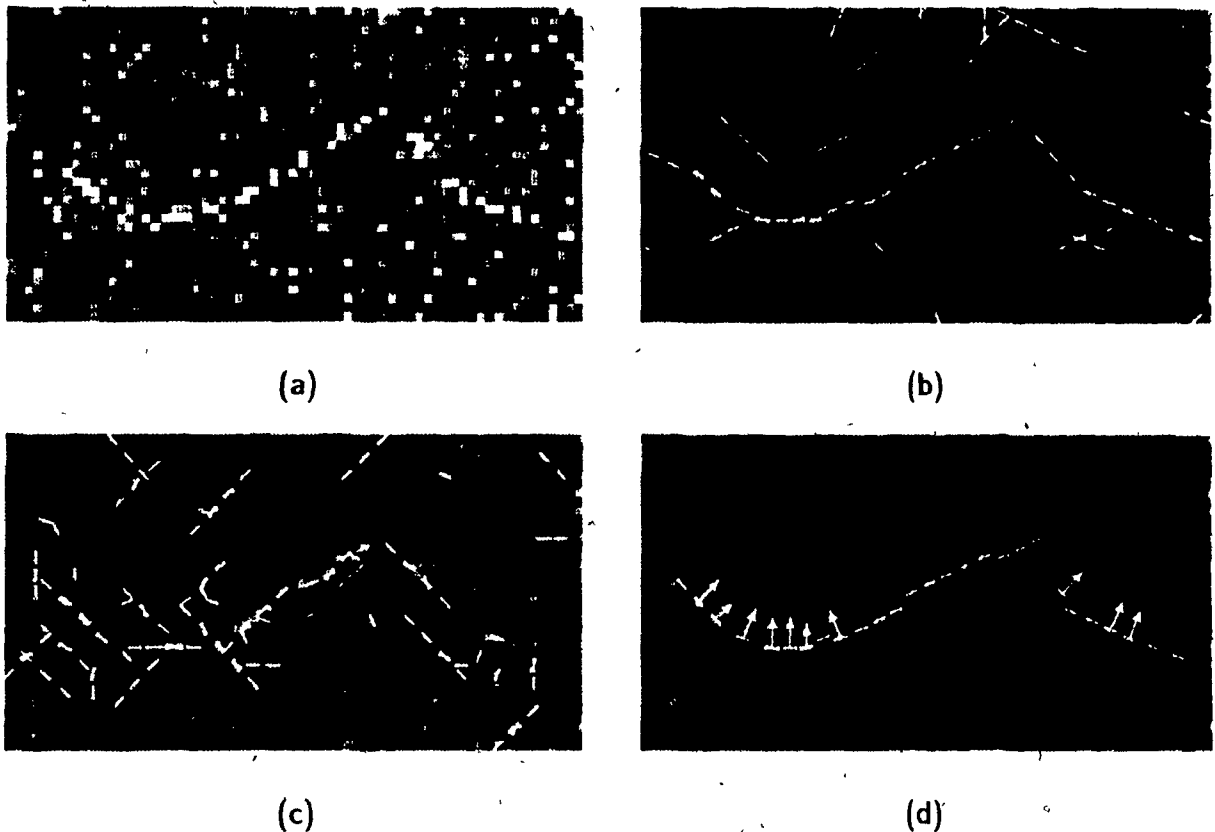
Figure 6.3 A comparison of different methods for detecting curves (a) intensities thresholded to include most of the curve points, resulting in many non-curve points being selected. (b) thresholded lateral maxima, equivalent to the 0th iteration of the trace inference process; again, only a low threshold allows all curve points to be selected, at the cost of including some non-curve points; (c) previous Parent and Zucker method, based on comparison of expected versus observed operator responses, this method is not curvature-based, and depends only on the immediate 3 by 3 neighbourhood for support, hence the noise sensitivity. (d) trace inference process, same as previous Figure for neighbourhood size 25 and $S/N = 0.45$. It clearly illustrates the advantage of using curvature information

importance of curvature information.

6.1.3 Neighbourhood size and smoothness

The experiment designed to evaluate the effect of neighbourhood size on smoothness is performed on a single curve, using three neighbourhood sizes. The curve is essentially a straight line, with a gaussian-like bump in its center being the region of interest for this experiment. Three neighbourhood diameters were used:

size 15 radius limits: 2.7, 4.2, 7.2, 21.0

$$\text{size 25 radius limits:} \quad 4.5, \quad 7.0, 12.0, 35.0$$

$$\text{size 45 radius limits:} \quad 8.1, 12.6, 21.6, 63.0 \tag{6.7}$$

and each neighbourhood size was matched with initial convolution operators of corresponding size:

Size 15:
$$\sigma_1 = 1.14 \quad \sigma_2 = 1.8 \quad \sigma_3 = 2.28 \quad \sigma_y = 3.6$$
$$B = 1.266 \quad C = 0.5$$

Size 25:
$$\sigma_1 = 1.9 \quad \sigma_2 = 3.0 \quad \sigma_3 = 3.8 \quad \sigma_y = 6.0$$
$$B = 1.266 \quad C = 0.5$$

Size 45:
$$\sigma_1 = 3.8 \quad \sigma_2 = 6.0 \quad \sigma_3 = 7.6 \quad \sigma_y = 12.0$$
$$B = 1.266 \quad C = 0.5 \tag{6.8}$$

The results are displayed in Figure 6.4. The progression of neighbourhood size is from the smallest in (a) to the largest in (c). While the bump is fully represented in (a), with position, orientation and curvature, there is only a slight variation in position in (c), and neither orientation nor curvature variation. Thus, the inferred trace is smoother when using a larger neighbourhood.

Note that this experiment is possible only because the macro-structure of the curve used in this experiment is that of a straight line, hence that its curvature is within certain limits. It is always possible to utilize large neighbourhoods when the macro-structure of the curves in an image have curvatures within the maximum curvature constraint implied by the neighbourhood size (as discussed in Section 4.5), and it may be interesting to do so, as in multiple-level processing. However, if the image consists of curves with long high-curvature segments, the effect of utilizing a larger neighbourhood, or of imposing a smaller maximum curvature constraint, would be to filter out the higher curvature components of the curves. This happens because the tangent support along these segments remains small for any given curvature class, because the curves will tend to traverse many curvature classes within such a large neighbourhood.

(a)

(b)

(c)
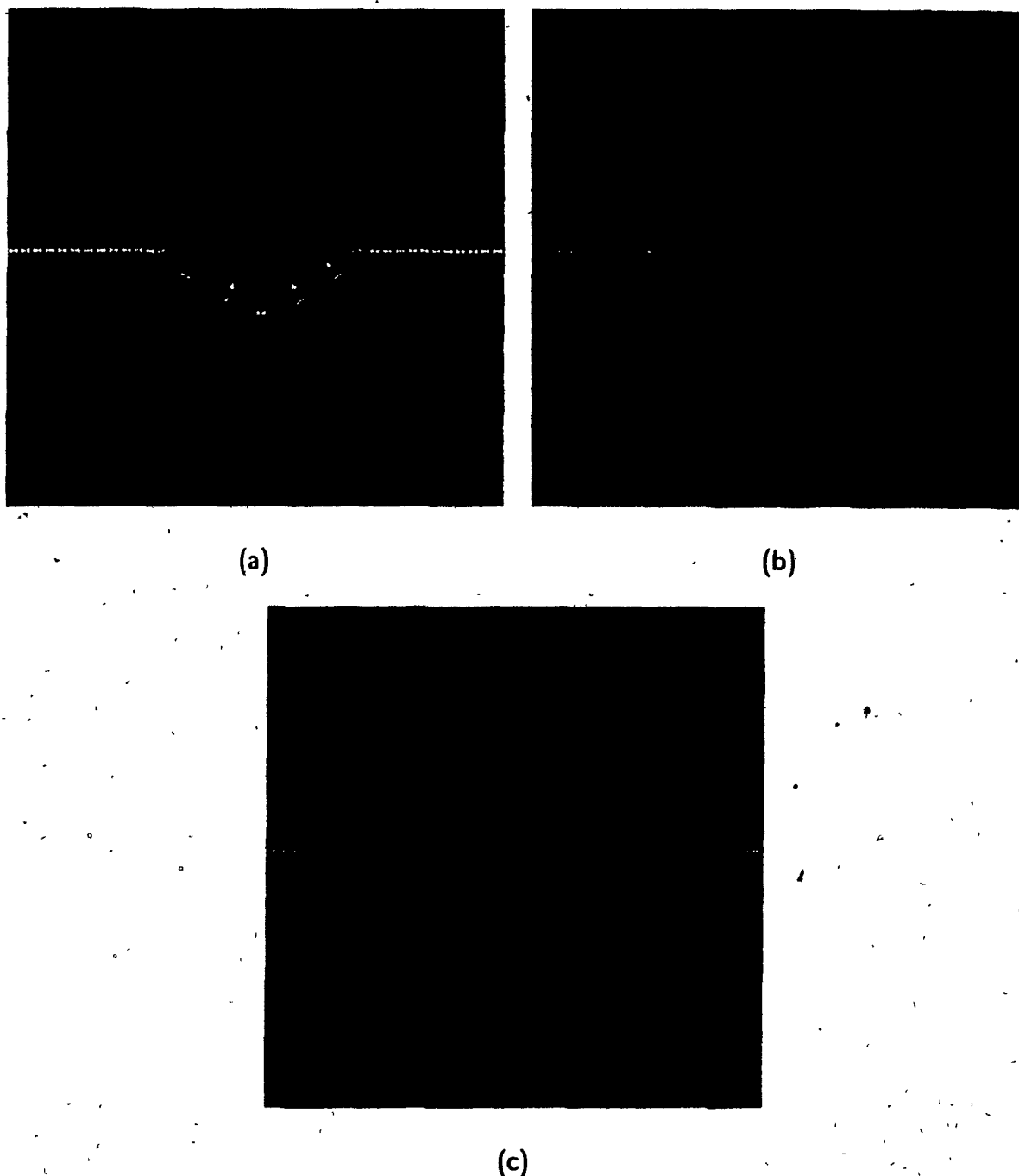
Figure 6.4 Effect of varying neighbourhood size on the smoothness of the inferred trace of a curve. The image is a single curve, essentially straight but with a gaussian-like bump. Three neighbourhood sizes are used, going from the smallest in (a) to the largest in (c). In (a), the diameter of the neighbourhood is 15, and the bump is well represented in the result, its curvature being within the maximum curvature constraint. In (b), the diameter is 25, and already the bump is smoother than in (a). Finally, in (c), where the diameter of the neighbourhood is 45, the curvature field inferred is straight everywhere, and there is only a slight lateral displacement of the trace where the bump is.

(a)   (b)
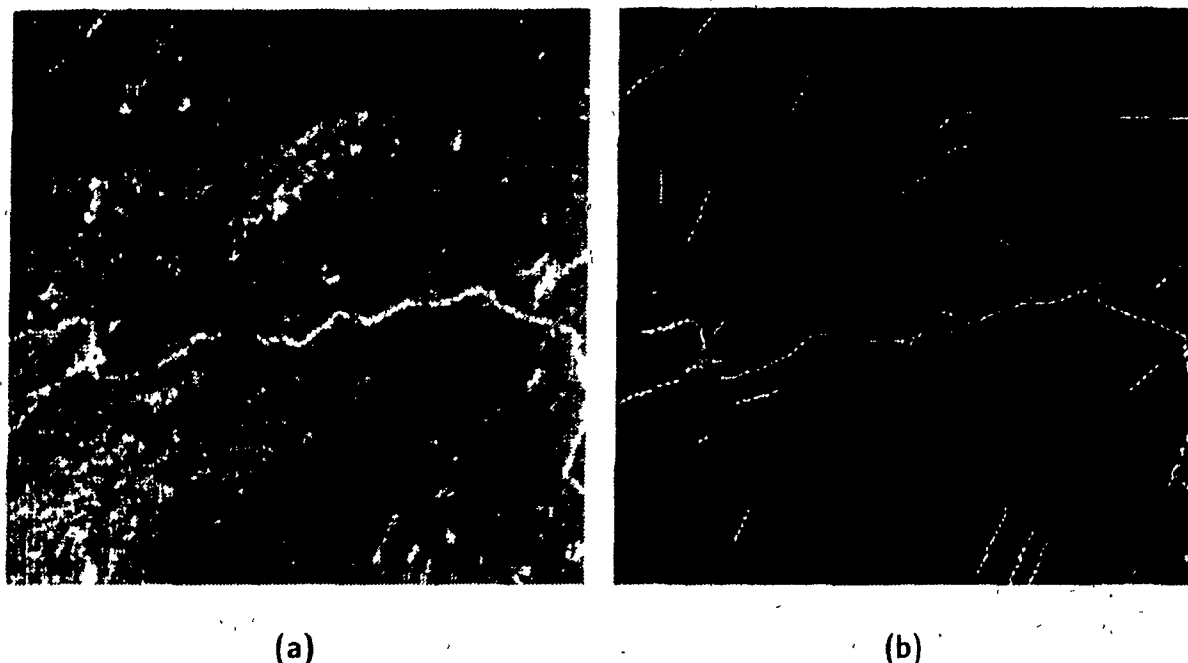
Figure 6.5   (a) a satellite image of a forest with logging roads, and (b) the result of
2 iterations of the trace inference process, using a neighbourhood size of 25, a step
size of 1.0, and displayed at a confidence threshold of 0.6 - The curvature vectors ar
omitted for the sake of clarity

6.2   Natural images

The preceding experiments on artificial images showed that the curvature in-
formation helped the trace inference process to find curves in controlled situations. The
main interest of the procedure, however, lies in its application to finding curves in real im-
ages, such as satellite images, bio-medical images, and fingerprints. In this Section, three
experiments based on such images are described.

6.2.1   Satellite image: logging roads in forest

The first natural image experiment is based on a satellite image of forest terrain
in which logging roads are visible as light, elongated streaks, on a slightly darker back-
ground. The experiment is performed using a size 25 neighbourhood, with parameters for
the initial convolution and radius limits as defined in Eqs. (6.7) and (6.8) for Size 25. The
inference process is run for two iterations, with a step size of 1.0.

(a)   (b)
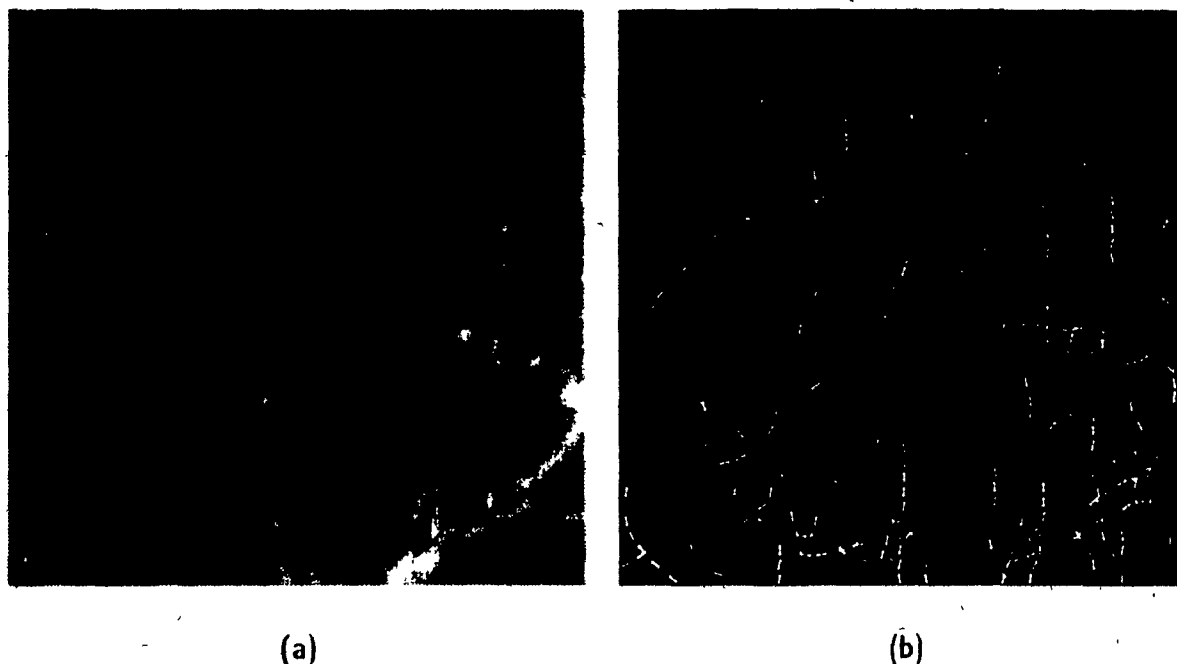
Figure 6.6   (a) an angiogram. or radiograph of blood vessels in the brain. and. (b) the result after 2 iterations of the trace inference process. using a neighbourhood size of 15. a step size of 1.0. and displayed at a confidence threshold of 0.5.

The result displayed in Figure 6.5 is thresholded at a confidence level of 0.6.

6.2.2   Angiogram

The second experiment takes a bio-medical image as its input. an angiogram. or a radiograph of blood vessels in the brain. It is a good example of an image with many curves; all of which have varying curvature. with many curve crossings at various angles.

The image is first convolved with operator parameters as given in Eq. (6.1). The neighbourhood size is 15. the number of iterations is 2. with a step size of 1.

All tangents with a certainty of at least 0.5 are displayed in Figure 6.6. however the curvature vectors are omitted for the sake of clarity.

6.2.3   Fingerprint

The final experiment involves a fingerprint image. The neighbourhood size which

(a)   (b)
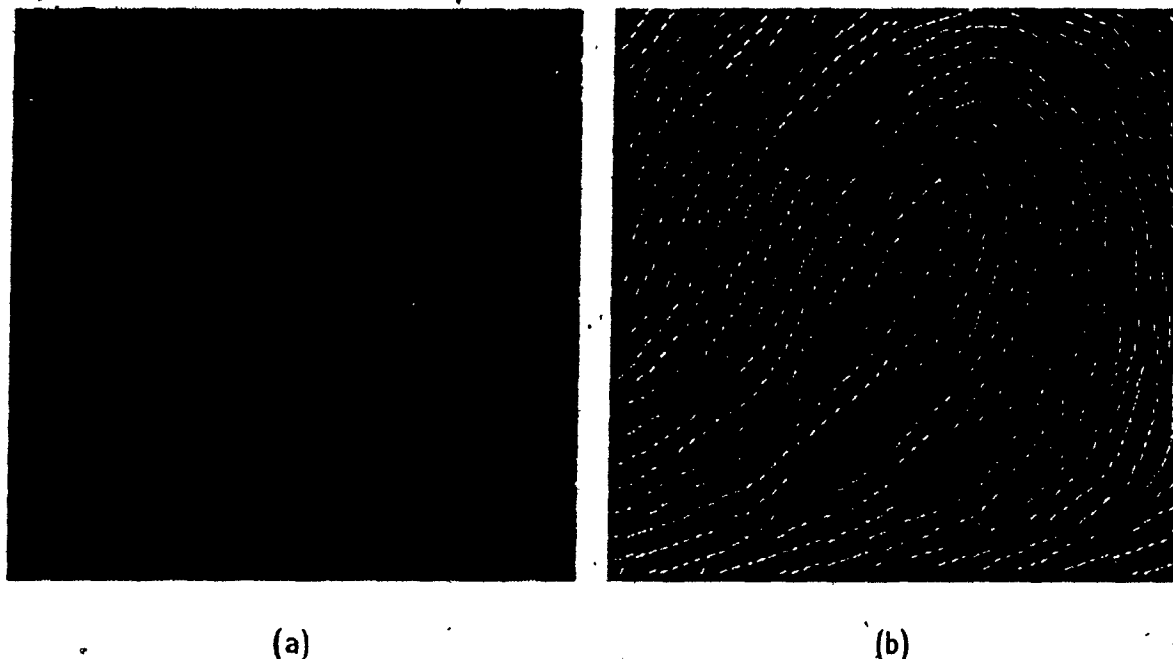
Figure 6.7   (a) the image of a fingerprint. and (b) the result after 2 iterations of the
trace inference process. using a neighbourhood size of 11. a step size of 1 0. and
displayed at a confidence threshold of only 0 3. due to the extremely low contrast
of the original image.

was chosen. 11 pixels in diameter. is smaller than in the other experiments. However, the
initial convolutions were performed with the same operator as for the angiogram experiment.
i.e. with parameters as in Eq. (6.1).

Again, 2 iterations of the process were used. The display threshold for the
tangents is set to 0.3, a low value because the image itself had low contrast.

Information about the exact structure of curves is lost when they are projected into quantized images. Hence curve detection is an inferential process, utilizing both image information and other constraints. We formulate the curve inference process as two distinct stages, in which local information is first recovered so that it can guide the global stage. In this thesis we concentrated on the recovery of local information, and demonstrated that it could be accomplished both in theory and in practice.

Images contain information not directly about curves, but rather about their traces, or the set of quantized image positions through which the curve passes. But these traces are not uniquely specified; they must be separated from other image structure. Therefore the first stage of curve inference must include trace inference, and the development of an approach to this occupied most of this thesis.

Inherent in trace inference is a chicken-and-egg problem. If the functional form of the curve were known, then the trace could simply be calculated. But since it is not, most of the effort went into developing an estimation procedure sufficently powerful to provide a model for the curve in the neighborhood of each possible trace point. The model for curves included the tangent and curvature at each point, and it is this model that guided trace inference. The result was what we called a tangent field, or a representation of the trace, tangent, and curvatures suitably quantized.

Since it is the high-frequency micro-structure of curves that is lost through

quantization, it is natural to employ smoothness constraints while estimating them. We derived such an estimation procedure by examining how discretization and quantization affect their differential-geometric definitions. The result was a functional with terms through curvature variation which could be maximized to guide trace inference. It appears that functionals through curvature variation are necessary to properly separate the influences of nearby curves, yet are sufficient to place discontinuities.

The computation of curvature is notoriously sensitive to noise. To avoid these problems, we introduced an alternate method for coarsely estimating it, based on average values of tangent estimates within spatial neighbourhoods called curvature classes.

Computing the tangent field was the overall goal of the first stage of curve inference, and we demonstrated that curvature and curvature consistency (or limits on curvature variation) can be utilized advantageously. The information in the tangent field certainly provides a rich, stable foundation for global curve inference.

Bibliography

Attneave, F., *Applications of Information Theory to Psychology: a summary of basic concepts, methods, and results*, Holt, Rinehart and Winston, New York, 1959.

Barrow, H. G., and Tenenbaum, J. M., "Interpreting line drawings as three-dimensional surfaces," *Artificial Intelligence*, vol. 17, 1981, pp. 75-116.

Binford, T. O., "Inferring surfaces from images," *Artificial Intelligence*, vol. 17, 1981, pp. 205-244.

Brady, M., and Asada, H., "Smoothed local symmetries and their implementation," *Robotics Research*, vol. 3, 1984, pp. 37-61.

Davis, L. S., and Freeman., "Corner detection in chain codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1977.

Davis, L. S., and Rosenfeld, A., "Cooperating processes for low-level vision: a survey," *Artificial Intelligence*, vol. 17, 1981, pp. 245-263.

do Carmo, M. P., *Differential Geometry of Curves and Surfaces*, Prentice Hall, Englewood Cliffs, N.J., 1976.

Faugeras, O., and Berthod, M., "Scene labeling, an optimization approach," *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, 1979, pp. 318-326.

Fischler, M. A., and Bolles, R. C., "Perceptual organization and curve partitioning," *Proceedings of the 1983 Image Understanding Workshop*, 1983.

Hartshorne, R., *Algebraic Geometry*, Springer Verlag, 1972.

Hedlund, M., Granlund, G. H., and Knutsson, H., "A consistency operation for line and curve enhancement," *Proceedings of the Pattern Recognition and Image Processing conference*, 1982, pp. 93-96.

Hoffman. D. D,. and Richards, W. A., "Parts of recognition." *From Pixels to Predicates,* edited by A. Pentland. Ablex. Norwood. N.J.. 1986.

Hubel, D.. and Wiesel. T.. "Functional architecture of macaque monkey visual cortex." *Proceedings of the Royal Society (London),* vol. B198, 1977, pp. 1–59.

Hummel, R.. and Zucker, S. W.. "On the foundations of relaxation labeling processes." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 5, 1983, pp. 267–287.

Koenderink, J., and van Doorn, A.. "The shape of smooth objects and the way contours end." *Perception,* vol. 11, 1982, pp. 129–137.

Martelli, A., "An application of heuristic search methods to edge and contour detection." *Communications of the Association for Computing Machinery,* vol. 19, 1976, pp. 73–83.

Mohammed, J. L., Hummel. R.. and Zucker, S. W.. "A gradient projection algorithm for relaxation methods." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 5, 1983, pp. 330–332.

Montanari, U.. "On the optimal detection of curves in noisy pictures." *Communications of the Association for Computing Machinery,* vol. 14, 1971, pp. 335–345.

Papoulis, A.. *Probability, Random Variables, and Stochastic Processes,* McGraw-Hill, New York, 1965.

Parent, P.. and Zucker, S. W.. *Radial Projection: An Efficient Update Rule for Relaxation Labelling.* Computer Vision and Robotics Laboratory Technical Report TR-85-15R, McGill University, 1985.

Pavlidis. T.. *Algorithms for Graphics and Image Processing.* Computer Science Press, Rockville, Maryland, 1982.

Peleg, S., and Rosenfeld, A., "Determining compatibility coefficents for curve enhancement relaxation processes," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 8, 1978, pp. 548–555.

Rosenfeld, A., Hummel, R., and Zucker, S. W., "Scene labeling by relaxation operations," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 6, 1976, pp. 420–433.

Rosenfeld, A., and Kak, A., *Digital Picture Processing,* Academic Press, New York, 1976.

Terzopoulos, D., "Integrating visual information from multiple sources," *From Pixels to Predicates,* edited by A. Pentland, Ablex, Norwood, N.J., 1986.

Waltz, D., "Understanding line drawings of scenes with shadows," *The Psychology of Computer Vision,* edited by P. H. Winston, McGraw-Hill, 1975, pp. 19–91.

Witkin, A. P., "Recovering surface shape and orientation from texture," *Artificial Intelligence,* vol. 17, 1981, pp. 17–45.

Zucker, S. W., Hummel, R., and Rosenfeld, A., "An application of relaxation labeling to line and curve enhancement," *IEEE Transactions on Computers,* vol. 26, 1977, pp. 394–403, and pp. 922–929, for figures.

Zucker, S. W., *Early orientation selection and grouping: Evidence for type I and type II processes,* Computer Vision and Robotics Laboratory Technical Report TR–82–8R, McGill University, 1982.

Zucker, S. W., and Parent, P., "Multiple-size operators and optimal curve finding," *Multiresolution Image Processing and Analysis,* edited by A.Rosenfeld, Springer-Verlag, 1984.

Zucker, S. W., "Early orientation selection: Tangent fields and the dimensionality of their support," *Computer Vision, Graphics, and Image Processing,* 1985.

Zucker, S. W., and Hummel, R., "Receptive fields and the representation of visual information," *Human Neurobiology,* to be published, 1986.

A.   Radial Projection: An Efficient Update Rule for Relaxation Labelling

A.1   Notation and terminology

Let the relaxation graph consist of n nodes, with m labels at each node. For simplicity, we assume that each node has the same set of labels $\Lambda = \{\lambda \mid \lambda = 1, \ldots, m\}$. Let $p_i(\lambda)$ denote the certainty of label λ at node i, restricted to the interval $[0,1]$, and such that the sum of the certainties at any given node is

$$\sum_{\lambda=1}^{m} p_i(\lambda) = 1 \quad \text{for } i = 1, \ldots, n. \tag{A.1}$$

In vector notation, the certainty vector at node i is $\vec{p_i}$, and Eq. (A.1) can be expressed as $\vec{p_i} \cdot \mathbf{1} = 1$, where $\mathbf{1}$ denotes the m-dimensional vector of 1's $(1, 1, \ldots, 1)$.

The support of label λ at node i is denoted $s_i(\lambda)$, and the support vector at node i is denoted $\vec{s_i}$. The normalized support vector is obtained from $\vec{s_i}$ by scaling it by $1/(\vec{s_i} \cdot \mathbf{1})$: the sum of the components of the normalized support vector is 1, and it thus satisfies constraint (A.1).

Let a superscript on a vector denote the iteration number. Thus, $\vec{p_i}^k$ and $\vec{s_i}^k$ denote respectively the certainty vector and the support vector of node i at iteration k.

A labelling at node i is said to be on the boundary of the labelling space if $p_i(\lambda) = 0$ for at least one value of λ. Saturation of node i occurs when $p_i(\lambda) = 1$ for some λ; this condition is alos referred to as an unambiguous labelling of node i. If $\forall \lambda, 0 < p_i(\lambda) < 1$, then the labelling is said to be ambiguous; a labelling is perfectly ambiguous if $\forall \lambda, p_i(\lambda) = 1/m$.

A.2   Consistent labellings

Definition. *A labelling is consistent if and only if the certainty vector matches the nor-*

malized support vector at every node, that is,

$$\vec{p}_i = \frac{\vec{s}_i}{\vec{s}_i \cdot 1} \quad \text{for } i = 1, \ldots, n. \tag{A.2}$$

To find a consistent labelling from an inconsistent one, one possibility would be to assign the normalized support vector directly to the certainty vector of the following iteration, with

$$\vec{p}_i^{k+1} = \vec{s}_i^k. \tag{A.3}$$

However, since the support vector $\vec{s}_i$ may vary considerably from one labelling assignment to another, it is more cautious to take a series of small steps towards the support vector instead. Let $\vec{v}_i^k$ denote the difference between the normalized support vector and the certainty vector at iteration k:

$$\vec{v}_i^k = \frac{\vec{s}_i^k}{\vec{s}_i^k \cdot 1} - p_i^k \tag{A.4}$$

A consistent labelling is obtained by taking, at each iteration k, a step in the same direction as $\vec{v}_i^k$. Note that vector $\vec{v}_i^k$ lies in a plane tangent to constraint (A.1) on the labelling assignment. It seems natural therefore to obtain an update vector by projecting the support vector $\vec{s}_i^k$ onto this tangent plane, which is indeed the standard solution to such problems [Faugeras and Berthod, 1979; Mohammed, Hummel and Zucker, 1983]. Care must be taken, however, to avoid projecting $\vec{s}_i^k$ in such a way that the resulting updated labelling no longer satisfies the constraint

$$0 \le p_i(\lambda) \le 1 \quad \text{for } i = 1, \ldots, n, \ \lambda = 1, \ldots, m. \tag{A.5}$$

The case of an initial assignment inside the boundary is trivial and requires only normal projection onto the tangent plane, with the proviso that the update vector may have to be scaled to avoid violating the constraint. On the other hand, when the initial assignment is on the boundary—when at least one of the $p_i(\lambda)$ is 0—scaling is not sufficient, and a more sophisticated projection scheme must be utilized to find a proper direction for the update vector [Mohammed, Hummel and Zucker, 1983]. To date, all proposed projection methods suffer from this basic drawback—the added complexity incurred when boundaries are encountered—which makes them computationally expensive.
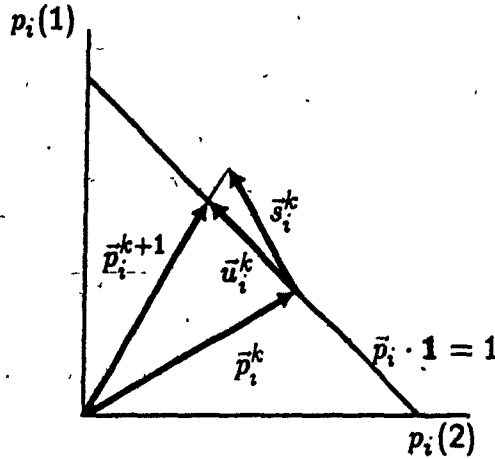
Figure A.1   For the case of a node with two labels: an updated labelling assignment is obtained by scaling $\vec{p}_i^k + \vec{s}_i^k$ to satisfy the constraint $\vec{p}_i \cdot 1 = 1$.

We propose a method that avoids this problem entirely by restricting support vectors to the positive quadrant, and by using *radial* projection instead of *normal* projection. Thus, if the initial labelling is strictly within (not on the boundary of) the positive quadrant, then all updated labellings will also be strictly within the positive quadrant, hence avoiding any complex computations at the boundary. This property holds although the labelling may in fact be converging towards some boundary or intersection of boundaries of the labelling space.

A.3   Radial projection update rule

The update rule that we propose does not use normal projection onto the tangent plane. Instead, it uses a radial projection with the origin as center. Thus, an updated labelling assignment is obtained simply with

$$\vec{p}_i^{k+1} = \frac{\vec{p}_i^k + \vec{s}_i^k}{(\vec{p}_i^k + \vec{s}_i^k) \cdot 1} = \frac{\vec{p}_i^k + \vec{s}_i^k}{1 + \vec{s}_i^k \cdot 1} \quad \text{for } i = 1, \ldots, n. \tag{A.6}$$

Referring to Figure A.1, it can be seen that (A.6) amounts merely to scaling the sum of the support and certainty vectors, $\vec{p}_i^k + \vec{s}_i^k$, to satisfy the assignment constraint (A.1).

What we require is that the update vector $\vec{u}_i^k \doteq \vec{p}_i^{k+1} - \vec{p}_i^k$ be parallel to the vector $\vec{v}_i^k$ of Equation (A.4). We have (dropping the subscript i momentarily)

$$\vec{u}_i^k = \frac{\vec{p}^k + \vec{s}^k}{1 + \vec{s}^k \cdot 1} - \vec{p}^k$$

$$= \frac{\vec{s}^k - (\vec{s}^k \cdot \mathbf{1})\vec{p}^k}{1 + \vec{s}^k \cdot \mathbf{1}}$$

$$= \frac{\vec{s}^k \cdot \mathbf{1}}{1 + \vec{s}^k \cdot \mathbf{1}} \times \frac{\vec{s}^k - (\vec{s}^k \cdot \mathbf{1})\vec{p}^k}{\vec{s}^k \cdot \mathbf{1}}$$

$$= \frac{\vec{s}^k \cdot \mathbf{1}}{1 + \vec{s}^k \cdot \mathbf{1}} \times \left(\frac{\vec{s}^k}{\vec{s}^k \cdot \mathbf{1}} - \vec{p}^k \right)$$

$$= \alpha_i^k \vec{v}_i^k \quad \text{for } i = 1, \dots, n. \tag{A.7}$$

Thus, $\vec{u}_i^k$ is a scalar multiple of $\vec{v}_i^k$, and consequently, these two vectors are parallel. $\vec{u}_i^k$ is therefore in the correct direction. The scalar α_i^k, under the conditions defined in the next Section, satisfies the condition $0 \leq \alpha_i^k < 1$.

A.3.1 Avoiding projection outside labelling space

The update rule (A.6) produces valid labelling assignments as long as the vector sum $\vec{p}_i^k + \vec{s}_i^k$ lies in the positive quadrant. This is always the case when the support vector itself points to that quadrant, i.e. when all of its components are non-negative. If such is not the case, however, the only way to insure correctness is to first project (see Figure A.2) the support vector onto the boundary of the positive quadrant with the formula

$$\tilde{s}_i = \min_{\lambda=1,m} s_i(\lambda) \tag{A.8}$$

$$s_i^*(\lambda) = \begin{cases} s_i(\lambda) - \tilde{s}_i & \text{if } \tilde{s}_i \leq 0; \\ s_i(\lambda) & \text{otherwise.} \end{cases} \tag{A.9}$$

In vector notation, $\vec{s}_i^{*k} = \vec{s}_i^k + |\tilde{s}_i|\mathbf{1}$, when $\tilde{s}_i < 0$. After this transformation, all components of the support vector are either positive or 0. It follows that labelling assignments updated using $\vec{s}_i^{*k}$ with (A.6) always lie in the valid labelling space.

It is interesting to note that the normal projections onto the tangent plane of $\vec{s}_i^k$ and $\vec{s}_i^{*k}$ coincide. Thus, as far as the standard projection methods are concerned, $\vec{s}_i^k$ and $\vec{s}_i^{*k}$ are strictly equivalent.
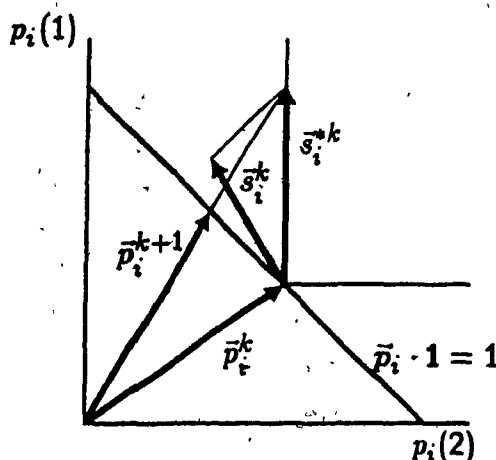
Figure A.2 When the vector sum $\vec{p}_i^k + \vec{s}_i^k$ lies outside the positive quadrant, the vector $\vec{s}_i^k$ is first projected onto the boundary of the positive quadrant $(\vec{s}_i^{*k})$. Thus, $\vec{p}_i^k + \vec{s}_i^{*k}$ is guaranteed to lie in the positive quadrant. $\vec{s}_i^{*k}$ is obtained from $\vec{s}_i^k$ by subtracting its most negative component from all its components.

A.3.2 Convergence properties and preservation of order information

None of the components of the updated labelling assignment can actually be 0, unless the initial assignment contained null components as well. This follows because the radial projection rule is such that successive labelling assignments converge asymptotically towards a limiting value. The rate of convergence is determined by the magnitude of the support vector $\vec{s}_i^k$.

This property is what distinguishes the radial projection method most from traditional projection methods. While the latter methods allow labellings to saturate at unambiguity, the radial update rule does not, provided the initial labelling is everywhere ambiguous. Moreover, in a hypothetical graph where many nodes in a small neighbourhood received essentially unambiguous support, traditional updating would eventually saturate all the nodes, regardless of the magnitude of the support vector at each node. In the event that the outcome of the relaxation process is meant to be further processed by techniques which rely on relative magnitudes of certainties in a small neighbourhood, saturation would be totally unwelcome.

Radial updating, on the other hand, produces a labelling structure that reflects the relative magnitudes of the support vectors throughout the relaxation network. Recall

from Section A.3 that the update vector $\vec{u}_i^k$ is equal to a constant times the difference vector $\vec{v}_i^k$. In the light of the saturation avoidance technique of the preceding Section, we are now in a position to determine the range of this constant. Substituting $\vec{s}_i^{*k}$ for $\vec{s}_i^k$ in Equation (A.7), we obtain

$$\alpha_i^k = \frac{\vec{s}_i^{*k} \cdot \mathbf{1}}{1 + \vec{s}_i^{*k} \cdot \mathbf{1}}. \tag{A.10}$$

But since all components of $\vec{s}_i^{*k}$ are non-negative, by definition, it follows that the quantity $\vec{s}_i^{*k} \cdot \mathbf{1}$ is also non-negative. Consequently, α_i^k is non-negative, strictly less than unity, and is a monotonic increasing function of the magnitude of $\vec{s}_i^{*k}$, over all $\vec{s}_i^{*k}$ in a constant direction. Thus, given the same initial certainties, two nodes receiving supports which differ *only in magnitude* will obtain distinct certainties, with the node receiving greater support converging fastest towards its limit.

A.3.3   Comparison with previous rules

The radial projection rule is identical in essence to the normal projection rule employed in [Faugeras and Berthod, 1979] when the labelling is made to converge towards purely consistent labellings. The inconsistency term that is minimized in that method is the norm $\|\vec{v}_i^k\|$. Given the same support vector $\vec{s}_i^k$, and provided that all components of the support vector are non-negative, both their method and the radial projection rule converge to the same result. The principal difference between the two methods lies in the handling of the case of projection outside the valid labelling space.

The radial projection rule can also be related to the framework described in [Hummel and Zucker, 1983]. In this method, Hummel and Zucker propose a global functional to maximize which is effectively the dot product $\vec{p}_i^k \cdot \vec{s}_i^k$. Note that the magnitude of $\vec{p}_i^k$ varies with position on the constraint $\vec{p}_i^k \cdot \mathbf{1} = 1$. In fact, the magnitude of $\vec{p}_i^k$ reaches a maximum in the corners of the constraint, i.e. for perfectly unambiguous labellings. Thus, for constant $\vec{s}_i^k$ over the entire labelling space, the functional will be maximized for the unambiguous labelling corresponding to the maximal component of $\vec{s}_i^k$. This explains the built-in bias towards unambiguous labellings of their method.
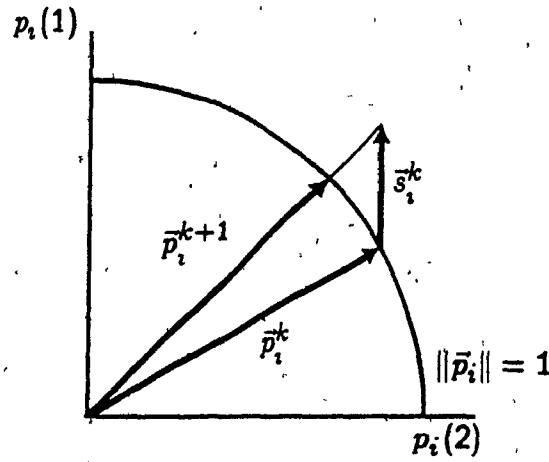
Figure A.3 If the constraint $\|\vec{p}_i\| = 1$ is substituted for the usual labelling con-
straint, the Hummel and Zucker framework converges to the same labelling as with
the radial projection rule, assuming that all components of $\vec{s}_i^k$ are non-negative.

Moreover, if more than one component of $\vec{s}_i^k$ is maximal, there is no unique
solution to the optimization problem as they have formulated it. In the extreme case where
all components of $\vec{s}_i^k$ are equal, $i = 1, \ldots, n$, the process terminates, regardless of the
initial labelling assignment, because the normal projection of the support vector onto the
tangent plane is the null vector.

Consider the following modification to the Hummel and Zucker framework: if
the constraint (A.1) on local labels is changed to

$$\sum_{\lambda=1}^{m} (p_i(\lambda))^2 = 1 \quad \text{for } i = 1, \ldots, n, \tag{A.11}$$

then their method is virtually identical to the radial projection rule. Referring to Figure A.3,
it can readily be seen that the constraint over which the global functional would then be
maximized is a hypersphere instead of a hyperplane. Because the magnitude of $\vec{p}_i^k$ is now
constant over the entire labelling space, the aforementioned bias no longer takes effect.

And since their method maximizes a linear functional, the process is guaranteed to have a
unique solution over this non-linear convex set. In particular, when all components of $\vec{s}_i^k$
are positive, the solution is the same as with the radial projection rule.

A.3.4 Summary of radial projection method

In summary, three steps are required to obtain updated labelling assignments

using the radial projection method. Given the initial measures of confidence, $p_i^0(\lambda)$, and the compatibility coefficients, $r_{ij}(\lambda, \lambda')$, the first step is accumulating the support evidence for each label:

$$s_i^k(\lambda) = \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}^k(\lambda, \lambda')\, p_j^k(\lambda'), \qquad (A.12.a)$$

The second step is a projection, if required, of the support vector at a node onto the boundary of the positive quadrant, to avoid labelling outside the valid assignment space:

$$\tilde{s}_i^k = \min_{\lambda=1,m} s_i^k(\lambda) \qquad (A.12.b)$$

$$s_i^{*k}(\lambda) = \begin{cases} s_i^k(\lambda) - \tilde{s}_i^k & \text{if } \tilde{s}_i^k \le 0; \\ s_i^k(\lambda) & \text{otherwise.} \end{cases} \qquad (A.12.c)$$

Finally, the third step is the radial projection method as such. It consists, for each node i, of the scaled vector sum

$$\vec{p}_i^{k+1} = \frac{\vec{p}_i^k + \vec{s}_i^{*k}}{1 + \vec{s}_i^{*k} \cdot \mathbf{1}}. \qquad (A.12.d)$$

This rule amounts to taking, from the current assignment, a step in the direction of the nearest consistent assignment. The magnitude of this step is governed by the quantity

$$\alpha_i^k = \frac{s_i^{*k} \cdot \mathbf{1}}{1 + s_i^{*k} \cdot \mathbf{1}}, \qquad (A.12.e)$$

where it is known that $0 \le \alpha_i^k < 1$. The rate of convergence α_i^k can be modified by scaling the support vector $\vec{s}_i^k$ uniformly throughout the network by any positive constant.

A.4   Conclusion

In this Appendix, we have shown that an efficient update mechanism for continuous relaxation labelling is possible. The radial projection rule is formally related to previous rules, but overcomes the complexity of update vector projection at the boundaries of the labelling space.

The reduction in complexity is achieved by first restricting support vectors to the positive quadrant, and then using radial projection onto the constraint instead of normal

projection. Among the features of the new update rule are smooth convergence towards a solution. and preservation of crucial order information which is sometimes required for other cooperative processes.