A Theory of Multi-Scale, Curvature and Torsion Based Shape Representation for Planar and Space Curves⁺

Farzin Mokhtarian

Technical Report 90-30 October 1990

Department of Computer Science University of British Columbia Vancouver, B.C. Canada V6T 1W5

+ A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.



Abstract

This thesis presents a theory of multi-scale, curvature and torsion based shape representation for planar and space curves. The theory presented has been developed to satisfy various criteria considered useful for evaluating shape representation methods in computer vision. The criteria are: *invariance*, *uniqueness*, *stability*, *efficiency*, *ease of implementation* and *computation of shape properties*. The regular representation for planar curves is referred to as the curvature scale space image and the regular representation for space curves is referred to as the torsion scale space image. Two variants of the regular representations, referred to as the renormalized and resampled curvature and torsion scale space images, have also been proposed. A number of experiments have been carried out on the representations which show that they are very stable under severe noise conditions and very useful for tasks which call for recognition of a noisy curve of arbitrary shape at an arbitrary scale or orientation.

Planar or space curves are described at varying levels of detail by convolving their parametric representations with Gaussian functions of varying standard deviations. The curvature or torsion of each such curve is then computed using mathematical equations which express curvature and torsion in terms of the convolutions of derivatives of Gaussian functions and parametric representations of the input curves. Curvature or torsion zero-crossing points of those curves are then located and combined to form one of the representations mentioned above.

The process of describing a curve at increasing levels of abstraction is referred to as the *evolution* or *arc length evolution* of that curve. This thesis contains a number of theorems about evolution and arc length evolution of planar and space curves along with their proofs. Some of these theorems demonstrate that evolution and arc length evolution do not change the physical interpretation of curves as object boundaries and others are in fact statements on the global properties of planar and space curves during evolution and arc length evolution and their representations. Other theoretical results shed light on the local behavior of planar and space curves just before and just after the formation of a cusp point during evolution and arc length evolution. Together these results provide a sound theoretical foundation for the representation methods proposed in this thesis.



Table of Contents

Ab	stract	ii	
List of Figures			
List of Tables			
Ac	knowledgments	viii	
1.	Introduction	1	
	1.1. Criteria for shape representation	2	
	1.2. Contents of thesis	3	
2.	Survey of literature	5	
3.	A theory of multi-scale shape representation for planar		
	curves	11	
	3.1. The curvature scale space image	13	
	3.2. The renormalized curvature scale space image	17	
	3.3. The resampled curvature scale space image	18	
	3.4. Evolution and arc length evolution properties of planar curves	20	
4.	A theory of multi-scale shape representation for space		
	curves	23	
	4.1. The curvature and torsion scale space images	24	
	4.2. The renormalized curvature and torsion scale space images	30	
	4.3. The resampled curvature and torsion scale space images	32	
	4.4. Evolution and arc length evolution properties of space curves	34	
5.	Algorithms and implementation	36	
	5.1. The convolution masks	36	
	5.2. The sampling algorithm	38	
	5.3. Algorithms for representations of planar curves	39	
	5.4. Algorithms for representations of space curves	41	
	5.5. Efficiency issues	44	
	5.6. Matching scale space images	46	
6.	Experimental results and evaluation	48	
	6.1. Planar curves	48	
	6.2. Space curves	80	
	6.3. Evaluation of the representations	100	
7.	Summary, conclusions and future directions	109	
	7.1. Summary	109	

7.2.	Conclusions		. 110
7.3.	Directions for fut	ure research	. 111
Refere	ces		. 114
Appen	lix A. Proofs o	of results on planar curves	. 120
Appen	lix B. Proofs o	f results on space curves	. 140

List of Figures

Figure	4.1.1	The Frenet trihedron for a space curve	25	
Figure	6.1.1	Coastline of Africa		
Figure	6.1.2	Africa during evolution		
Figure	6.1.3	The curvature scale space image of Africa	51	
Figure	6.1.4	Koch's snowflake curve	53	
Figure	6.1.5	Snowflake curve during evolution	54	
Figure	6.1.6	The curvature scale space image of the snowflake	55	
Figure	6.1.7	Design from a Persian carpet	56	
Figure	6.1.8	Carpet design during evolution	57	
Figure	6.1.9	The curvature scale space image of the carpet design	58	
Figure	6.1.10	Coastline of Africa with uniform, random noise	59	
Figure	6.1.11	The CSS image of Africa with uniform noise superim-		
po	sed on	the CSS image of Africa	60	
Figure	6.1.12	Coastline of Africa with severe, uniform noise	61	
Figure	6.1.13	The CSS image of Africa with severe noise superimposed		
on	the CS	SS image of Africa	62	
Figure	6.1.14	Coastline of Africa with non-uniform, random noise	63	
Figure	6.1.15	The curvature scale space image of Africa with non-		
un	iform 1	loise	64	
Figure	6.1.16	The renormalized curvature scale space image of Africa		
•••	•••••		66	
Figure.	6.1.17	The renormalized curvature scale space image of Africa		
W1	th non-	uniform noise	67	
Figure	6.1.18	The renormalized CSS image of Africa with non-uniform	00	
no	use sup	erimposed on the renormalized USS image of Africa	68	
Figure	6.1.19	The resampled curvature scale space image of Africa	69	
Figure	6.1.20	The resampled curvature scale space image of Africa	422-12-1	
with non-uniform noise				
Figure	6.1.21	The resampled CSS image of Africa with non-uniform		
nc	oise sup	erimposed on the resampled CSS image of Africa	71	
Figure	6.1.22	A self-crossing curve during evolution	74	
Figure	6.1.23	The CSS image of the curve of figure 6.1.22	75	

Figure	6.1.24	A convex, self-crossing curve during evolution	76
Figure	6.1.25	The CSS image of the curve of figure 6.1.24	77
Figure	6.1.26	A simple curve during (regular) evolution	78
Figure	6.1.27	Curve of figure 6.1.26 during arc length evolution	79
Figure	6.2.1	Two views of a space curve depicting a fork	81
Figure	6.2.2	The fork during evolution	82
Figure	6.2.3	The torsion scale space image of the fork	83
Figure	6.2.4	Two views of a space curve depicting a bottle-opener	84
Figure	6.2.5	The bottle-opener during evolution	85
Figure	6.2.6	The torsion scale space image of the bottle-opener	86
Figure	6.2.7	A space curve depicting an armchair	87
Figure	6.2.8	The armchair during evolution	88
Figure	6.2.9	The torsion scale space image of the armchair	89
Figure	6.2.10	The armchair with random noise	90
Figure	6.2.11	The torsion scale space image of the armchair with noise	
			91
Figure	6.2.12	The renormalized torsion scale space image of armchair	
wi	th nois	e	92
Figure	6.2.13	The TSS image of armchair with noise superimposed on	
th	e TSS i	image of armchair	93
Figure	6.2.14	The armchair with severe random noise	94
Figure	6.2.15	The torsion scale space image of armchair with severe	1327564
nc	oise		95
Figure	6.2.16	The resampled torsion scale space image of the armchair	
		·····	96
Figure	6.2.17	The resampled torsion scale space image of armchair	07
Firmer	6 0 10	The recompled TSS increases from their with second points	91
rigure	0.2.10	osed on the resampled TSS image of armchair	99
Figure	631	A planar curve made up of straight line segments	102
Figure	632	The curve of figure 6.3.1 with added random noise	102
Figure	633	The curve of figure of the curve of figure	100
6.3.1			
Figure	6.3.4	The curvature scale space image of the curve of figure	
6.	3.2		105
Figure 6.3.5 The CSS image shown in figure 6.3.3 superimposed on the			
C	SS imag	ge shown in figure 6.3.4	106

List of Tables

Table	6.1.1	Comparison of regular, renormalized and resampled curva-	
tı	ire sca	ale space images	72
Table	A.1	Moment-pairs in equations (A.2) and (A.3)	128
Table	B.1	Moment-triples with $M_0^{\prime\prime}$	150
Table	B.2	Moment-triples with $M_1^{\prime\prime}$	150
Table	B.3	Moment-triples with M_2^{γ}	151
Table	B.4	Moment-triples with $M_3^{7/}$	151
Table	B.5	Moment-triples with $M_4^{\prime\prime}$	151
Table	B.6	Moment-triples with $M_5^{1/}$	151
Table	B.7	Moment-triples with $M_6^{1/}$	151

Acknowledgments

I am specially grateful to Alan Mackworth, my thesis supervisor, who contributed significantly to the development of this thesis. That contribution ranged from specific research results to general guidelines and ideas about how to proceed from a specific point. His enthusiasm, approachability and understanding helped produce work that I feel quite proud of.

I am also grateful to David Kirkpatrick, Peter Lawrence, Jim Little and David Lowe, members of my supervisory committee, and to Bob Woodham for discussions and for offering criticisms of earlier drafts of my thesis which helped improve both its quality and the presentation of its results. Tom Hurd of the Math department helped me gain further insight into the mathematical aspects of the problems I encountered. Alex Kean, Jane Mulligan, Marc Romanycia, and Ying Zhang listened to various aspects of my work at our group meetings and provided helpful comments.

I have no doubt that without the strong and continued support from my parents, I would not have been able to complete this work as it stands. Their love outlasted the dark clouds. I owe much to them and so does this thesis.

Roxana has only recently entered my life but has already added considerably to it in special ways. She too, has left her mark on this work.

Chapter 1: Introduction

Computer Vision is an area within the field of Artificial Intelligence which aims to develop techniques for analysis and interpretation of images. Those images may depict two- or three-dimensional real or synthetic objects. An important area of research within Computer Vision is the problem of shape representation. This thesis considers the problem of representing the shapes of twodimensional or planar and three-dimensional or space curves. A planar or a space curve is a set of points whose position vectors are the values of a continuous vector-valued function [Goetz 1970]. A planar curve can be represented by the parametric vector equation

$$\mathbf{r}(u) = (x(u), y(u))$$

and a space curve can be represented by

$$\mathbf{r}(u) = (x(u), y(u), z(u))$$

where coordinate functions x(u), y(u) and z(u) are components of $\mathbf{r}(u)$ and $u \in [a,b]$ is the parametrization variable. Note that a planar or space curve as defined above is connected and has no branches.

A note on the title of this thesis is appropriate. The title speaks of a "theory of multi-scale, curvature and torsion based shape representation for planar and space curves" rather than a "multi-scale, curvature and torsion based shape representation technique for planar and space curves." The word *theory* was chosen to indicate the approach used by the author in proposing a shape representation technique in this thesis. That approach is to propose and motivate a number of criteria that a general purpose shape representation technique should satisfy. Then shape representations were developed and evaluated according to those criteria. The evaluation shows that the representations satisfy nearly all of the proposed criteria. The properties of the representations were also studied formally and in detail in order to understand their possible behaviours. The result is

a number of theorems which clearly describe properties of the representations under conditions specified by the theorems. It is believed that the approach taken justifies the use of the term *theory*.

1.1. Criteria for shape representation

In the following, when two planar or space curves are described as having the *same shape*, there exists a transformation consisting of uniform scaling, rotation and translation which will cause one of those curves to overlap the other.

We believe that it is important for a general-purpose shape representation to satisfy the following criteria:

- Invariance: if two curves have the same shape, they should also have the same representation.
- Uniqueness: if two curves do not have the same shape, they should have different representations.
- Stability: if two curves have a small shape difference, their representations should also have a small difference and if two representations have a small difference, the curves they represent should also have a small shape difference.

The importance of the invariance criterion is that it guarantees that all curves with the same shape will have the same representation. It will therefore be possible to conclude that two curves have different shapes by observing that they have different representations. Without the invariance criterion, two curves with the same shape may have different representations.

The uniqueness criterion is important since it guarantees that two curves with different shapes will have different representations. It will therefore be possible to conclude that two curves have the same shape by observing that they have the same representation. Without the uniqueness criterion, two curves with different shapes may have the same representation.

The significance of the stability criterion is that it guarantees that a small change in the shape of a curve will not cause a large change in its representation and a small difference between two representations does not indicate a large shape difference between the curves they represent. As a result, when two representations are close, the curves they represent are close in shape and when two representations are not close, the curves they represent are not close in shape. When this criterion is satisfied, the representation can be considered to be stable with respect to noise. One way to measure the shape difference between two planar or space curves is the *Hausdorf distance* [Hong & Tan 1988]. A useful shape representation in computational vision should make accurate and reliable recognition of a shape possible. Therefore it is useful for a shape representation to satisfy a number of additional properties in order to become suitable for shape recognition tasks in computer vision. The following is a list of such criteria. Note that similar criteria have been proposed in [Nishihara 1981], [Mokhtarian & Mackworth 1986] and [Haralick *et al.* 1989].

- Efficiency: The representation should be efficient to compute and store. This is important since it may be necessary for an object recognition system to perform real-time recognition. By *efficient* we mean that the computational complexity should be a low-order polynomial in time and space (and in the number of processors if a parallel computing architecture is used) as a function of the size of the input curve.
- Ease of implementation: If two or more competing representations exist, it is advantageous to choose one of those representations such that the implementation of the computer program which computes that representation requires the least time spent on programming and debugging.
- Computation of shape properties: It may be useful to be able to determine properties of the shape of a curve using its representation. For example, if a curve has a symmetric shape, it may be desirable to be able to determine that fact from its representation (the *symmetry* criterion). Furthermore, if the shape of a whole curve or part of a curve is the same as the shape of part of another curve, it may be useful to be able to determine that relationship using their representations (the *part/whole* criterion).

1.2. Contents of thesis

The following is a description of the contents of the remaining chapters of this thesis:

Chapter 2 reviews the computer vision literature in the area of shape representation which is relevant to planar and space curves. A comprehensive survey of techniques for representation of shapes of planar curves is given and ways to generalize each technique to apply to space curves are described. Each technique is then evaluated according to the criteria proposed in this chapter.

Chapters 3 and 4 propose a novel theory of multi-scale shape representation for planar and space curves. Chapter 3 introduces a multi-scale, curvature based theory of shape representation for planar curves. Three variations of the representation have been developed, each most suitable for particular applications. A number of theorems provide a sound theoretical foundation for the representation. Chapter 4 introduces a multi-scale, torsion based theory of shape representation for space curves. Again three variations of the representation are described. A number of parallel theorems provide a similarly sound theoretical foundation for the representation.

Chapter 5 discusses a number of issues which arise when implementing the curvature and torsion scale space representations of planar and space curves and gives algorithms to compute various curvature and torsion scale space representations proposed in chapters 3 and 4. It also presents a complexity analysis of each of those algorithms.

Chapter 6 provides a number of examples of curvature scale space representations of planar curves and torsion scale space representations of space curves. It also provides examples of curves distorted by uniform and non-uniform noise and its effect on the representations of those curves. This chapter also includes a discussion of the significance of the theorems of chapters 3 and 4 and an evaluation of the curvature and torsion scale space representations according to the criteria set forth in chapter 1.

Chapter 7 presents a summary and the conclusions of this thesis. It also proposes avenues of further research on multi-scale geometry-based shape representation techniques.

Appendix A contains the proofs of the theorems of chapter 3 and appendix B contains the proofs of the theorems of chapter 4.

Chapter 2:

Survey of Literature

This chapter contains a comprehensive survey of representation techniques for the shapes of planar and space curves. Each technique is described and a possible way to generalize it to apply to space curves is explained. The reason is that almost all three-dimensional shape representation techniques proposed in computer vision have been concerned with surfaces but not curves. An evaluation of each technique according to the criteria proposed in chapter 1 is also presented. By default, each technique satisfies all criteria of chapter 1 except the ones pointed out following the description of that technique.

It should be noted that any of the techniques listed below may be quite suitable for special-purpose shape representation and recognition tasks. Therefore failure to satisfy one or more of the criteria listed in chapter 1 does not imply that a particular shape representation should not be used under any circumstances. The criteria listed in chapter 1 have been proposed for a generalpurpose shape representation technique independent of any particular applications. Furthermore, note that some of the representations were not originally proposed for shape matching applications and therefore invariance, stability and other issues were not a major concern. It may well be possible to deal with some of these issues through, for example, the use of normalization techniques.

a. Hough transform: Has been used to detect lines [Hough 1962], circles [Duda & Hart 1972] and arbitrary shapes [Ballard 1981]. Edge elements in the image vote for the parameters of the objects of which they are parts. The votes are accumulated in a parameter space. The peaks of the parameter space then indicate the parameters of the objects searched for. In theory, the Hough transform can be generalized to space curves by increasing the dimensions of the parameter space. However, the parameters which define an object change when it undergoes rotation, uniform scaling or translation therefore the invariance criterion is not satisfied.

- b. Chain encoding [Freeman 1974, McKee & Aggarwal 1977, Gallus & Neurath 1970]: The curve is approximated using connected line segments which lie on a grid. This method can be generalized to space curves by using a threedimensional instead of a two-dimensional grid. Note that this technique was originally used to encode contour data but was later used for recognition as well. This method does not satisfy the invariance criterion since the approximating chain code rotates, scales and moves as the original curve rotates, scales and moves. The uniqueness criterion is not fully satisfied either since the mapping of a curve to its chain code involves loss of information. The degree of information loss depends on the resolution of the approximating grid.
- c. Medial axis transform [Blum 1973, Marr 1977]: The medial axis transform computes the skeleton of a two-dimensional object by a thinning algorithm that preserves connectivity of regions. This method can be generalized to space curves by using a three-dimensional thinning algorithm. Let S be the set of boundary points of a region. For each point P in that region, find its closest neighbors on the region boundary. If more than one boundary point is at a minimum distance from P, then P is on the skeleton of the region. This method does not satisfy the invariance criterion since the skeleton of an object rotates, scales and moves as the original curve rotates, scales and moves. It also fails to satisfy the stability criterion since a small change in the shape of the object could result in a significant change in its representation.
- d. Quad trees [Samet 1980, Schneier 1979]: A spatial occupancy array is first computed for a region on a raster device by assigning the value 1 to any array element which is in the region and value 0 otherwise. The quad tree can then be thought of as an intermediate pyramid representation of the spatial occupancy array. Each pixel in images above the lowest level has one of three values: black, white or gray. A pixel in an upper level is black or white if all its corresponding pixels in the next lower level are black or white respectively. If some of the lower level pixels are black and others are white, the corresponding pixel in the higher level is gray. This method is area-based and can not be applied to space curves. Note that Quad trees were proposed mainly for use in computer graphics and image processing. Quad trees do not satisfy the invariance criterion since shape-preserving transformations of the object change the structure of the representation. They also do not satisfy the stability criterion since making a small change to the spatial occupancy array can cause a dramatic change in the resulting quad tree.
- e. Shape factors and quantitative measurements [Danielsson 1978, Brown 1979]: The shape of the object is described using one or more global quantitative measurements of the object such as area, perimeter, compactness (perimeter²/area) and eccentricity (ratio of the principal axes of inertia). Some of these measurements, such as perimeter, can also be computed for space

curves. The uniqueness criterion is not satisfied by this class of representations since there is a drastic reduction in data. However, they may be useful when the global measurements are sufficient to distinguish among a class of objects.

- f. Slope density function [Nahin 1974]: Let ϕ be the angle made between a fixed vector and a tangent to a curve. The slope density function is the histogram or frequency distribution of ϕ collected over the entire length of the curve. Note that the SDF is flat for a circle or any curve with a monotonically varying ϕ . This method can be applied to space curves as well. The uniqueness criterion is not satisfied by this method since the mapping involves a loss of information. The stability criterion may also be violated if the input curve is noisy. However, smoothing the curve may remedy this problem.
- g. Polyline or polygonal approximation [Pavlidis 1972, 1973, 1975, 1977, Pavlidis & Horowitz 1974, Horowitz & Pavlidis 1976, Tomek 1974, Sklansky & Kibler 1976, Turner 1974, Roberts 1965, Shirai 1975]: A polyline or polygon (when the curve to be approximated is closed) is used to approximate the curve to any desired level of accuracy. The problem is to find corners or breakpoints that yield the "best" polyline or polygon. Various criteria have been used to solve that problem which can be characterized by the concepts of splitting and merging. In a merging algorithm, points along a curve are accepted into a linear segment as long as they fit sufficiently well. A concept of scale can be introduced by adjusting the number of breakpoints or the total approximation error allowed. This method can easily be applied to space curves by using three-dimensional polylines or polygons. This class of representations does not satisfy the invariance criterion since the approximating polyline or polygon rotates, scales and moves with the curve. The uniqueness criterion is satisfied well when the input curve consists mainly of segments with little or no curvature. Otherwise, the approximation error depends on the size (measured in terms of the number of vertices) of the approximating polygon.
- h. Strip trees [Davis 1977, Ballard 1981]: A strip tree is a set of approximating polygons or polylines ordered such that each polygon or polyline approximates the curve with less approximation error than the previous polygon or polyline. Initially, an open curve is approximated by a line segment which joins its endpoints and a closed curve is approximated by a line segment joining two points on the curve that are furthest away. At each level of refinement, a point on a curve segment that is furthest away from its approximating line segment is joined to the endpoints of that line segment to obtain better approximations. This method can be generalized to space curves by using three-dimensional polygons or polylines to approximate the curves. Strip trees do not satisfy the invariance criterion since they move, rotate and scale with the curve. They also do not satisfy the stability criterion since a small change in the shape of the curve can result in a large change in its representation.

- i. Splines [deBoor 1978, Barnhill & Riesenfeld 1974, Ballard & Brown 1982]: The curve is represented using a set of analytic and smooth curves. This method can also be generalized to space curves by making use of three-dimensional splines. Note that splines have been used primarily in computer graphics for object modelling. The invariance criterion is not satisfied since shape-preserving transformations of the curve and the choice of knot points for the approximating splines change the parameters of those splines. The uniqueness criterion is partially satisfied. The approximation error depends on the degree of approximating splines.
- j. Smoothing splines [Shahraray & Anderson 1989]: The curve is parametrized to obtain two coordinate functions. Cross-validated regularization [Wahba 1977] is then used to arrive at an "optimal" smoothing of each coordinate function. The smoothed functions together define a new smooth curve. This technique can also be generalized to space curves: three coordinate functions instead of two are obtained by parametrizing the curve. The invariance criterion is not satisfied since shape-preserving transformations of the curve also change the smoothing splines.
- k. Fourier descriptors [Persoon & Fu 1974, Schwartz et al. 1983, Richard & Hemami 1974]: The curve is represented by the coefficients of the Fourier expansion of a parametric representation of the curve. This method can be applied to both two-dimensional and three-dimensional curves. The invariance criterion is not satisfied by this class since shape-preserving transformations of the curve will change its Fourier coefficients.
- 1. Curvature primal sketch [Asada & Brady 1986]: The curve is approximated using a library of well-defined, analytic curves. Then the curvature function of the approximating curve is computed and convolved with a Gaussian of varying standard deviation. This technique can be generalized to space curves by computing either the curvature or the torsion function of those curves. This method may fail to satisfy the stability criterion. If the original curve is noisy, then computing its curvature function is an error-prone process and the computed representation may change significantly.
- m. Scale space blackboard [Saund 1989]: A scale space blackboard is constructed by adding a notion of scale to Marr's [1976] Primal Sketch. Two types of grouping operations are introduced: (a) Aggregation of fine scale primitiveedge tokens into coarser scale edge maps, and (b) pairwise grouping of symmetrically placed primitive-edges into primitive-partial-region type tokens denoting curved-contour, primitive-corner, and bar events. It is possible to compute a scale space blackboard for space curves by introducing grouping operations for three-dimensional edges. However, the method fails to satisfy the invariance criterion since the primitive-edges move, rotate and scale as the

curve to be represented moves, rotates and scales. It also does not fully satisfy the uniqueness criterion since the process of approximating curved objects using straight line segments involves some loss of information.

- n. Extended circular image [Horn & Weldon 1986]: This representation is the two-dimensional equivalent of the extended Gaussian image. In the extended circular image, one is given the radius of curvature as a function of normal direction. This technique can also be generalized to space curves. However, the resulting representation will be spherical rather than circular since the normal vectors will not all lie in the same plane. The invariance criterion is not fully satisfied by this method since the representation rotates as the original curve rotates. The uniqueness criterion is satisfied only for the class of simple and convex curves.
- o. Curve signatures [O'Rourke & Washington 1985]: The signature of a planar curve Γ in class C_1 is a function s(w) where w is the normalized arc length parameter on Γ . For each value w_0 of w, $s(w_0)$ is defined to be the total length of those segments of C which are to the left of the tangent vector to C at point $P=C(w_0)$. This technique is also generalizable to space curves. The stability criterion is not satisfied by curve signatures since noise on the curve can significantly alter the representation.
- p. Volumetric diffusion [Koenderink & van Doorn 1986]: A geometrical object is defined by way of its "characteristic function" $\chi(\mathbf{r})$ which equals unity when the point r belongs to the object and zero otherwise. The object is then blurred by requiring that its characteristic function satisfy the diffusion equation. The boundary of each blurred object is defined by the equation $\chi(\mathbf{r}) = 0.5$. Alternatively, the boundary can be extracted by applying a Laplacian operator to the blurred function. This method does not generalize to space curves. The invariance criterion is not satisfied by this method since shapepreserving transformations of the object also affect the blurred objects computed by this method. This problem may be remedied by locating the curvature zero-crossing points of those curves and mapping those points to a curvature scale space representation as described in chapter 3. The stability criterion may also be violated since a small change to the shape of an object could result in a significant change in its computed deformations. For example, the contour of a dumbbell may remain a connected curve or may split into two disconnected contours when the thickness of its neck varies by an infinitesimal amount [Babaud et al. 1986].
- q. Scale space image [Witkin 1983]: The scale space image was in fact introduced as a representation for one-dimensional signals and functions. The input function f(x) is convolved with a filter approximating the second derivative of the Gaussian. The standard deviation σ of the Gaussian is varied from a small to a

large value and the zero-crossing points of the convolved functions are located and marked in the x- σ space. A scale space volume can also be computed for a 2D function or image f(x,y) by convolving it with the Laplacian of a 2D Gaussian function as the standard deviation σ varies from a small to a large value and marking the zero-crossing points in the x-y- σ space.

r. Reactive and diffusive deformations of shape [Kimia et al. 1989]: Chapters 3 and 4 will introduce multi-scale theories of shape representation for planar and space curves which are based on computing specific deformations of the curves to be represented. Kimia, Tannenbaum and Zucker [1989] have proposed somewhat different deformations of shape. Let $\vec{\gamma}$ be the input curve and let \vec{N} be the normal vector. The deformation of $\vec{\gamma}$ is defined as:

$$rac{\partial \overrightarrow{\gamma}}{\partial t} = eta(\kappa) \, \overrightarrow{N}$$

where $\beta(\kappa)$ is a function of curvature and t is time and determines the amount of deformation. This method is somewhat similar to the technique described in this thesis and can also be applied to space curves. However, it fails to satisfy the stability criterion since a small change to the shape of an object could result in a significant change in its computed deformations. For example, the contour of a dumbbell may remain a connected curve or may split into two disconnected contours when the thickness of its neck varies by an infinitesimal amount. The method also fails to satisfy the invariance criterion since deformations of a curve are not invariant with respect to shape-preserving transformations of that curve. This shortcoming may be remedied by locating the curvature zero-crossing points of those curves and mapping those points to a curvature scale space representation as described in chapter 3.

As demonstrated, each of the existing shape representation techniques for planar and space curves in computer vision fails to satisfy one or more of the criteria considered useful for a general-purpose representation. Chapters 3 and 4 present theories of shape representation for planar and space curves which attempt to meet the criteria for shape representations proposed in chapter 1. An evaluation of those representations is presented in chapter 6.

Chapter 3:

A Theory of Multi-Scale Shape Representation for Planar Curves

Multi-scale approaches to various computer vision problems have been investigated by many vision researchers recently. Multi-scale analyses of those problems have proven more fruitful than attempts at solving those problems at a single scale. A multi-scale approach allows one to obtain an initial approximate solution at a stable higher scale. Such a solution can be obtained efficiently by searching a reduced search space. That initial solution can then be used to guide one through lower scales in order to obtain more accurate solutions at any desired level of accuracy. Examples are signal matching using scale space [Witkin *et al.* 1987], multi-resolution computation of surfaces [Terzopoulos 1984], organization of image curves at multiple scales [Lowe 1988], hierarchical models of three-dimensional objects [Koenderink & van Doorn 1986], hierarchical surface interpolation [Szeliski 1989] and solving the depth interpolation problem using a multi-grid approach [Choi & Kender 1988].

Similarly, multi-scale approaches to the shape representation problem have been proposed in the vision literature. Some examples can be found among the shape representation techniques reviewed in chapter 2. Using a multi-scale representation of an object, a matching algorithm can obtain a quick, approximate match at a coarse scale where the shapes of the objects are relatively simple and then use that initial match to guide the search through the finer scales and obtain more accurate matches.

A number of multi-scale shape representation techniques can be identified among methods reviewed in chapter 2. However, as it was pointed out in that chapter, each one fails to satisfy some of the criteria listed in chapter 1. This chapter introduces a multi-scale shape representation which satisfies nearly all the criteria proposed in chapter 1.

A multi-scale representation for one-dimensional functions was first proposed by Stansfield [1980] and later developed by Witkin [1983]. The function f(x) is convolved with a Gaussian function as its variance σ^2 varies from a small to a large value. The zero-crossings of the second derivative of each convolved function are extracted and marked in the $x-\sigma$ plane. The result is the scale space image of the function.

The curvature scale space image was introduced in [Mokhtarian & Mackworth 1986] as a new shape representation for planar curves. The representation is computed by convolving a path-based parametric representation of the curve with a Gaussian function, as the standard deviation of the Gaussian varies from a small to a large value, and extracting the curvature zero-crossing points of the resulting curves. The representation is invariant under rotation, uniform scaling and translation of the curve. This and a number of other properties makes it suitable for recognizing a noisy curve of arbitrary shape at any scale or orientation. The process of describing a curve at increasing levels of abstraction is referred to as the evolution of that curve. The evolution of a planar curve and the curvature scale space image are described in detail in section 3.1.

Mackworth and Mokhtarian [1988] introduced a modification of the curvature scale space image referred to as the *renormalized curvature scale space image*. This representation is computed in a similar fashion but the curve is reparametrized by arc length after convolution. As was demonstrated in [Mackworth & Mokhtarian 1988], the renormalized curvature scale space image is more suitable for recognizing a curve with non-uniform noise added to it. However, the renormalized curvature scale space image is more computationally intensive than the regular curvature scale space image. The renormalized curvature scale space image is described in detail in section 3.2.

The resampled curvature scale space image is a substantial refinement of the curvature scale space which is based on the concept of arc length evolution. It is shown that the resampled curvature scale space image is more suitable than the renormalized curvature scale space image for recognition of curves with added non-uniform noise or when local shape differences exist. The comparison table in chapter 6 lists the advantages and disadvantages of each representation. The arc length evolution of a planar curve and the resampled curvature scale space image are described in detail in section 3.3.

Section 3.4 contains a number of theorems which demonstrate a number of global and local properties of planar curves during evolution. Together, these theorems constitute a sound theoretical foundation for the representation technique presented in this chapter.

3.1. The curvature scale space image

The following is a review of those concepts of differential geometry of planar curves which are relevant to the theory presented in this chapter.

A planar curve is a set of points whose position vectors are the values of a continuous vector-valued function. It can be represented by the parametric vector equation

$$\mathbf{r}(u) = (x(u), y(u))$$
 (3.1.1)

The function $\mathbf{r}(u)$ is a parametric representation of the curve and $u \in [a,b]$ is the parametrization variable. A planar curve has an infinite number of distinct parametric representations. A parametric representation in which the parameter is the arc length s is called a *natural* parametrization of the curve. A natural parametrization can be computed from an arbitrary parametrization using the following equation

$$s = \int_{0}^{u} |\dot{\mathbf{r}}(v)| dv. \qquad (3.1.2)$$

where . represents the derivative, i.e.

$$\dot{\mathbf{r}}(v) = \frac{d\mathbf{r}}{dv}$$

For any parametrization

$$\dot{\mathbf{r}}(u) = (\dot{x}(u), \dot{y}(u))$$
$$|\dot{\mathbf{r}}(u)| = (\dot{x}^2 + \dot{y}^2)^{1/2}$$
$$\mathbf{t}(u) = \frac{\dot{\mathbf{r}}}{|\dot{\mathbf{r}}|} = \left(\frac{\dot{x}}{(\dot{x}^2 + \dot{y}^2)^{1/2}}, \frac{\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{1/2}}\right)$$

and

$$\mathbf{n}(u) = \Big(\frac{-\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{1/2}}, \frac{\dot{x}}{(\dot{x}^2 + \dot{y}^2)^{1/2}}\Big)$$

where $\mathbf{t}(u)$ and $\mathbf{n}(u)$ are the tangent and normal vectors at u respectively.

For any planar curve the vectors $\mathbf{t}(u)$ and $\mathbf{n}(u)$ must satisfy the simplified Serret-Frenet vector equations [Goetz 1970]:

$$\mathbf{t}(s) = \kappa(s) \mathbf{n}(s)$$

 $\dot{\mathbf{n}}(s) = -\kappa(s) \mathbf{t}(s)$

where $\kappa(s)$ is the curvature of the curve at s and is defined as the limit of the ratio of the angle ϕ between $\mathbf{t}(s)$ and $\mathbf{t}(s+h)$ to h as $h\to 0$. That is

$$\kappa(s) = \lim_{h \to 0} \frac{\phi}{h}.$$

A straight line has zero curvature everywhere. A circle is an example of a closed curve with constant curvature. An ellipse is another closed curve with positive (but not constant) curvature everywhere.

Now observe that:

$$\dot{\mathbf{t}}(s) = \frac{d\mathbf{t}}{ds} = \frac{d\mathbf{t}}{du}\frac{du}{ds}.$$

Therefore

$$\frac{d\mathbf{t}}{du} = \frac{ds}{du} \,\kappa \,\mathbf{n} = |\dot{\mathbf{r}}| \,\kappa \,\mathbf{n}.$$

Hence

$$\kappa(u) = \frac{\mathbf{t} \cdot \mathbf{n}}{|\mathbf{\dot{r}}|}.$$

Differentiating the expression for t(u), we obtain:

$$\dot{\mathbf{t}}(u) = \left(\frac{-\dot{y}(\dot{x}\ddot{y}-\ddot{x}\dot{y})}{(\dot{x}^2+\dot{y}^2)^{3/2}}, \frac{\dot{x}(\dot{x}\ddot{y}-\ddot{x}\dot{y})}{(\dot{x}^2+\dot{y}^2)^{3/2}}\right).$$

It now follows that:

$$\kappa(u) = rac{\dot{x}(u)\ddot{y}(u)-\dot{y}(u)\ddot{x}(u)}{(\dot{x}(u)^2+\dot{y}(u)^2)^{3/2}}.$$

Therefore it is possible to compute the curvature of a planar curve from its parametric representation.

Two special cases of the parametrization, of interest here, yield simplifications of these formulas. If we have a natural path representation with s, the arc length parameter, ranging over [0, L] then:

$$|\dot{\mathbf{r}}(s)| = |(\dot{x}(s), \dot{y}(s))| = (\dot{x}^2(s) + \dot{y}^2(s))^{1/2} = 1$$

To see this, note that it follows from equation (3.1.2) that:

$$\frac{ds}{du} = |\dot{\mathbf{r}}(u)|.$$

Therefore

$$|\dot{\mathbf{r}}(s)| = |\dot{\mathbf{r}}(u)| \frac{du}{ds} = \frac{|\dot{\mathbf{r}}(u)|}{|\dot{\mathbf{r}}(u)|} = 1.$$

Furthermore,

$$\mathbf{t}(s) = (\dot{x}(s), \dot{y}(s))$$

$$\dot{\mathbf{t}}(s) = (\ddot{x}(s), \ddot{y}(s))$$
$$\mathbf{n}(s) = (-\dot{y}(s), \dot{x}(s))$$
$$\kappa(s) = \dot{\mathbf{t}}(s) \cdot \mathbf{n}(s)$$

and

$$\kappa(s) = \dot{x}(s)\dot{y}(s) - \ddot{x}(s)\dot{y}(s).$$

Note also

$$\kappa^2(s) = |\mathbf{t}(s)|^2$$

 $\kappa^2(s) = \ddot{x}^2(s) + \ddot{y}^2(s).$

If the parameter is a linear rescaling of the arc length ranging over [0,1], the normalized path length parameter w, then

$$w = \frac{s}{L}$$
$$|\dot{\mathbf{r}}(w)| = L$$
$$\mathbf{t}(w) = \frac{1}{L}(\dot{x}(w), \dot{y}(w))$$
$$\mathbf{n}(w) = \frac{1}{L}(-\dot{y}(w), \dot{x}(w))$$
$$\kappa(w) = \frac{1}{L^3}(\dot{x}(w)\ddot{y}(w) - \ddot{x}(w)\dot{y}(w))$$

$$\kappa^{2}(w) = rac{1}{L^{4}}(\ddot{x}^{2}(w) + \ddot{y}^{2}(w)).$$

This concludes the review of differential geometry of planar curves.

Given a planar curve

$$\Gamma = \{(x(w), y(w)) | w \in [0,1]\}$$

where w is the normalized arc length parameter, an *evolved* version of that curve is defined by

$$\Gamma_{\sigma} = \{ (X(u,\sigma), Y(u,\sigma)) | u \in [0,1] \}$$

where

$$X(u,\sigma) = x(u) \circledast g(u,\sigma)$$
$$Y(u,\sigma) = y(u) \circledast g(u,\sigma)$$

and \circledast represents the convolution operator. Function $g(u,\sigma)$ denotes a Gaussian of width σ [Marr & Hildreth 1980] and is defined by

$$g(u,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-u^2}{2\sigma^2}}.$$

As it will become clear in appendices A and B, the use of the Gaussian filter results in a number of important properties of the representations which other smoothing filters do not provide. Functions $X(u,\sigma)$ and $Y(u,\sigma)$ are given explicitly by

$$X(u,\sigma) = \int_{-\infty}^{\infty} x(v) \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(u-v)^2}{2\sigma^2}} dv$$

and

$$Y(u,\sigma) = \int_{-\infty}^{\infty} y(v) \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(u-v)^2}{2\sigma^2}} dv.$$

The curvature of Γ_{σ} is given by

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma) Y_{uu}(u,\sigma) - X_{uu}(u,\sigma) Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{3/2}}$$

where

$$\begin{split} X_u(u,\sigma) &= \frac{\partial}{\partial u} (x(u) \circledast g(u,\sigma)) = x(u) \circledast g_u(u,\sigma) \\ X_{uu}(u,\sigma) &= \frac{\partial^2}{\partial u^2} (x(u) \circledast g(u,\sigma)) = x(u) \circledast g_{uu}(u,\sigma) \\ Y_u(u,\sigma) &= y(u) \circledast g_u(u,\sigma) \end{split}$$

and

$$Y_{uu}(u,\sigma) = y(u) \circledast g_{uu}(u,\sigma).$$

The process of generating the ordered sequence of curves $\{\Gamma_{\sigma}|\sigma\geq 0\}$ is referred to as the *evolution* of Γ .

Note that when a planar curve evolves according to the process defined above, its total arc length shrinks. The amount of shrinkage is directly proportional to the value of σ . In certain applications, this may be an undesirable feature. For example, the evolution process defined above may be used to smooth edges extracted from an image by an edge detector. However, it may be advantageous to have the smoothed edges close to the physical location of the original edges. This can be accomplished by estimating the amount of movement at each point on the smoothed edges and adding a vector to the location vector of that point to compensate for that movement [Lowe 1988]. The result is a smoothed curve which is physically close to the original curve.

The function defined implicitly by

 $\kappa(u,\sigma)=0$

is the curvature scale space image of Γ [Mokhtarian & Mackworth 1986].

3.2. The renormalized curvature scale space image

Mackworth and Mokhtarian [1988] observed that although w is the normalized arc length parameter on the original curve Γ , the parameter u is **not**, in general, the normalized arc length parameter on the evolved curve Γ_{σ} . In other words, arc length does not shrink uniformly everywhere on the curve during evolution. To correct this problem, Mackworth and Mokhtarian [1988] proposed the *renormalized* curvature scale space image.

Let

 $\mathbf{R}(\mathit{u},\!\sigma) = (\mathit{X}(\mathit{u},\!\sigma), \mathit{Y}\!(\mathit{u},\!\sigma))$

 $w = \Phi_{\sigma}(u)$

and

where

$$\Phi_{\sigma}(u) = \frac{\int\limits_{0}^{u} |\mathbf{R}_{v}(v,\sigma)| dv}{\int\limits_{0}^{1} |\mathbf{R}_{v}(v,\sigma)| dv}.$$

Now define

$$\hat{X}(w,\sigma) = X(\Phi_{\sigma}^{-1}(w),\sigma) \qquad \hat{Y}(w,\sigma) = Y(\Phi_{\sigma}^{-1}(w),\sigma).$$
(3.2.1)

That is, each evolved curve Γ_{σ} is reparametrized by its normalized arc length parameter w.

Notice that

 $\Phi_{\sigma}(0) = 0$ $\Phi_{\sigma}(1) = 1$

and

Also

 $\Phi_0(u) = u.$

 $\Phi_{\sigma}(u)$ deviates from the identity function $\Phi_{\sigma}(u) = u$ only to the extent to which the scale-related statistics deviate from stationarity along the original curve.

Once we have changed parameters according to equations (3.2.1), the curvature of the curve with the normalized path length parameter is given by

$$\kappa(w,\sigma) = \frac{1}{L^3} [\hat{X}_w(w,\sigma) \, \hat{Y}_{ww}(w,\sigma) - \hat{X}_{ww}(w,\sigma) \, \hat{Y}_w(w,\sigma)].$$

The function defined implicitly by

 $\kappa(w,\sigma)=0$

is the *renormalized* curvature scale space image of Γ .

3.3. The resampled curvature scale space image

Note that as a planar curve evolves according to the process defined in section 3.1, the parametrization of its coordinate functions x(u) and y(u) does not change. In other words, the function mapping values of the parameter u of the original coordinate functions x(u) and y(u) to the values of the parameter u of the smoothed coordinate functions $X(u,\sigma)$ and $Y(u,\sigma)$ is the identity function.

For both theoretical and practical reasons, it is interesting to generalize the definition of evolution so that the mapping function can be different from the identity function. Again let Γ be defined by:

$$\Gamma = \{ (x(w), y(w)) | w \in [0,1] \}.$$

The generalized evolution which maps Γ to Γ_{σ} is now defined by:

$$\Gamma \rightarrow \Gamma_{\sigma}$$

where

$$\Gamma_{\sigma} = \{ (X(W,\sigma), Y(W,\sigma)) | W \in [0,1] \}$$
$$X(W,\sigma) = x(W) \circledast g(W,\sigma)$$

and

$$Y(W,\sigma) = y(W) \circledast g(W,\sigma).$$

Note that

$$W = W(w,\sigma)$$

and

 $W(w,\sigma_0)$

where σ_0 is any value of σ , is a continuous and monotonic function of w. This condition is necessary to ensure physical plausibility since W is the parameter of the evolved curve Γ_{σ} .

An important case is when W always remains the arc length parameter as the curve evolves. When this criterion is satisfied, the evolution of Γ is referred to as arc length evolution. An explicit formula for W can be derived [Gage & Hamilton 1986].

Let

$$\mathbf{R}(W,\sigma) = (X(W,\sigma), Y(W,\sigma)).$$

The Frenet equations for a planar curve are given by

$$\frac{\partial \mathbf{t}}{\partial u} = \left| \frac{\partial \mathbf{R}}{\partial u} \right| \kappa \mathbf{n}$$
$$\frac{\partial \mathbf{n}}{\partial u} = -\left| \frac{\partial \mathbf{R}}{\partial u} \right| \kappa \mathbf{t}.$$

Let $t = \sigma^2/2$. Observe that

$$\frac{\partial}{\partial t}\left(|\frac{\partial \mathbf{R}}{\partial u}|^2\right) = \frac{\partial}{\partial t}\left(\frac{\partial \mathbf{R}}{\partial u} \cdot \frac{\partial \mathbf{R}}{\partial u}\right) = 2\left(\frac{\partial \mathbf{R}}{\partial u} \cdot \frac{\partial^2 \mathbf{R}}{\partial u \partial t}\right).$$

Note that

$$\frac{\partial \mathbf{R}}{\partial u} = \left| \frac{\partial \mathbf{R}}{\partial u} \right| \mathbf{t}$$

and

$$\frac{\partial \mathbf{R}}{\partial t} = \kappa \mathbf{n}$$

since the Gaussian function satisfies the heat equation. It follows that

$$\frac{\partial}{\partial t}(|\frac{\partial \mathbf{R}}{\partial u}|^2) = 2(|\frac{\partial \mathbf{R}}{\partial u}|\mathbf{t}.\frac{\partial}{\partial u}(\kappa \mathbf{n})) = 2(|\frac{\partial \mathbf{R}}{\partial u}|\mathbf{t}.(\frac{\partial \kappa}{\partial u}\mathbf{n} - |\frac{\partial \mathbf{R}}{\partial u}|\kappa^2\mathbf{t})) = -2|\frac{\partial \mathbf{R}}{\partial u}|^2\kappa^2.$$

Therefore

$$2|\frac{\partial \mathbf{R}}{\partial u}|\frac{\partial}{\partial t}|\frac{\partial \mathbf{R}}{\partial u}| = -2|\frac{\partial \mathbf{R}}{\partial u}|^2\kappa^2$$

or

$$\frac{\partial}{\partial t} \left| \frac{\partial \mathbf{R}}{\partial u} \right| = - \left| \frac{\partial \mathbf{R}}{\partial u} \right| \kappa^2.$$

Let L denote the length of the curve. Now observe that

$$\frac{\partial L}{\partial t} = \int_0^L \frac{\partial}{\partial t} \left| \frac{\partial \mathbf{R}}{\partial u} \right| du = -\int_0^L \left| \frac{\partial \mathbf{R}}{\partial u} \right| \kappa^2 du = -\int_0^1 \kappa^2 dw.$$

Since the value w_0 of the normalized arc length parameter w at a point P measures the length of the curve from the starting point to point P, it follows that

$$\frac{\partial W}{\partial t} = -\int_{0}^{W} \kappa^{2}(W,t) \, dW$$

and therefore

$$W(w,t) = -\int_{0}^{t} \int_{0}^{w} \kappa^{2}(W,t) \, dW \, dt + w.$$
(3.3.1)

Note that

$$W(w,0) = w.$$

The function defined implicitly by

$$\kappa(W,\sigma)=0$$

is the resampled curvature scale space of Γ .

Since the function $\kappa(W,t)$ in (3.3.1) is unknown, W(w,t) can not be computed directly from (3.3.1). However, the resampled curvature scale space can be computed in a simple way. A Gaussian filter based on a small value of the standard deviation is computed. The curve Γ is parametrized by the normalized arc length parameter and convolved with the filter. The resulting curve is reparametrized by the normalized arc length parameter and convolved again with the same filter. This process is repeated until the curve is convex and no longer has any curvature zero-crossing points. The curvature zero-crossings of each curve are marked in the resampled curvature scale space image. Note that the standard deviation of the Gaussian chosen above should be small enough so that the deviation from arc length parametrization after each iteration is negligible. Then the entire process can be considered to model arc length evolution.

3.4. Evolution and arc length evolution properties of planar curves

This section contains a number of important results on evolution and arc length evolution of planar curves as defined in sections 3.1 and 3.3. The proofs can be found in appendix A.

The first five theorems express a number of global properties of planar curves during evolution and arc length evolution.

Theorem 3.4.1. Evolution and arc length evolution of a planar curve are invariant under the shape preserving transformations (rotation, uniform scaling and translation) of the curve.

Theorem 3.4.2. A closed planar curve remains closed during its evolution and arc length evolution.

Theorem 3.4.3. A connected planar curve remains connected during its evolution and arc length evolution.

Theorem 3.4.4. The center of mass of a planar curve does not move during evolution and arc length evolution of that curve.

Theorem 3.4.5. Let Γ be a closed planar curve and let G be its convex hull. Every point on Γ remains in G during evolution and arc length evolution.

Theorem 3.4.6 shows that the mapping from a planar curve to its curvature scale space image is an invertible one.

Theorem 3.4.6. Let Γ be a planar curve in C_1 . The derivatives at a single point on one curvature zero-crossing contour in the regular, renormalized or resampled curvature scale space image of Γ determines Γ uniquely up to uniform scaling, rotation and translation (except on a set of measure zero).

Theorem 3.4.7 states that under certain conditions, new curvature zerocrossing points are not created during evolution and arc length evolution of planar curves.

Theorem 3.4.7. Let Γ be a planar curve in C_2 . If all evolved and arc length evolved curves Γ_{σ} are in C_2 , then all extrema of contours in the regular, renormalized and resampled curvature scale space images of Γ are maxima.

Theorem 3.4.8 locally characterizes the behaviour of planar curves during evolution and arc length evolution just before the creation of a cusp point.

Theorem 3.4.8. Let $\Gamma = (x(u), y(u))$ be a planar curve in C_1 and let x(u) and y(u) be polynomial functions of u. Let Γ_{σ} be an evolved or arc length evolved version of Γ with a cusp point at u_0 . There is a $\delta > 0$ such that $\Gamma_{\sigma-\delta}$ intersects itself in a neighborhood of point u_0 .

The following theorem holds only for arc length evolution.

Theorem 3.4.9. Simple curves remain simple during arc length evolution.

Theorem 3.4.10 locally characterizes the behaviour of a planar curve during evolution and arc length evolution just after the creation of a cusp point.

Theorem 3.4.10: Let $\Gamma = (x(u), y(u))$ be a planar curve in C_1 and let x(u) and y(u) be polynomial functions of u. Let Γ_{σ} be an evolved or arc length evolved version of Γ with a cusp point at u_0 . There is a $\delta > 0$ such that $\Gamma_{\sigma+\delta}$ has two new curvature zero-crossings in a neighborhood of u_0 .

Chapter 4:

A Theory of Multi-Scale Shape Representation for Space Curves

Almost all work done in computer vision on shape representation has focused either on planar curves and two-dimensional shapes, as shown in chapter 2, or on three-dimensional objects and surfaces [Bentley 1975, Voelcker & Requicha 1977, O'Rourke & Badler 1979, Jackins & Tanimoto 1980, Requicha 1980, Faugeras & Ponce 1983, Brady *et al.* 1985, Weiss 1985]. This chapter addresses the problem of describing the shape of three-dimensional curves. A space curve may be directly computed from an image [Watson & Shapiro 1982, Barnard & Pentland 1983]. It may also represent a surface reconstructed using stereo [Woodham 1984, Grimson 1985], "shape from" techniques [Ikeuchi & Horn 1981, Witkin 1981, Stevens 1982], or laser range finders [Faugeras *et al.* 1984], as described below.

Why study the problem of representing the shape of space curves? Space curves are useful to study for the following reasons:

- a. Trajectories of objects in outer space and paths taken by atomic particles are space curves. Often, such an object or particle can be recognized by studying the shape of its path when subjected to specific forces. Trajectories of robot arms and robot vehicles are also space curves.
- b. Axes of generalized cones and cylinders [Agin & Binford 1973] are also space curves. A generalized cone or cylinder representation of a three-dimensional object can itself be efficiently represented by its axes.

c. Bounding contours of objects that consist of flat or nearly flat surfaces are rich in information and can be used to represent the object effectively and economically. These bounding contours are space curves and can be extracted by *thinning* the object into lines and planes. An attempt to describe such objects using three-dimensional surfaces may not add much useful information but can significantly increase storage and processing requirements.

This chapter proposes a multi-scale theory of shape representation for space curves. This theory is a generalization of the multi-scale theory of shape representation for planar curves which was proposed in chapter 3. Section 4.1 develops the concept of curvature and torsion scale space representation for space curves. This representation was first proposed in [Mokhtarian 1988a]. Section 4.2 introduces a modification of the representation proposed in section 4.1 referred to as the renormalized curvature and torsion scale space representations. Section 4.3 proposes a significant further refinement of the earlier representations referred to as the resampled curvature and torsion scale space representations. Section 4.4 presents a number of important theoretical results on the evolution properties of space curves. These results further understanding of the evolution process and constitute a sound theoretical foundation for the representation techniques proposed in this chapter.

4.1. The curvature and torsion scale space images

This section introduces the parametric representation of space curves and describes the Frenet Trihedron for space curves. Curvature and torsion of a space curve are then defined and geometrical interpretations given to them. Next, it is shown how to compute curvature and torsion on a space curve at varying levels of detail. A multi-scale representation for a space curve which combines information about the curvature and torsion of the curve at varying levels of detail is then proposed.

i. The parametric representation of a space curve

The set of points of a space curve are the values of a continuous vectorvalued function [Goetz 1970]:

$$\mathbf{r} = \mathbf{r}(u) = (x(u), y(u), z(u)) \tag{4.1.1}$$

where x(u), y(u) and z(u) are the components of $\mathbf{r}(u)$ and u is a function of arclength s of the curve. s is also called the *natural parameter*. The function $\mathbf{r}(u)$ or the triple of functions (x(u), y(u), z(u)) is called a *parametric representation* of the curve.

ii. Frenet Trihedron and formulae for space curves

Let P_0 be a common point of a space curve and a plane and let P be a

variable point of the curve. Let h denote the arc length between P_0 and P and let d_h denote the distance of P from the plane. The curve and the plane are said to have a contact of order at least n [Goetz 1970] at P_0 if $d_h = O(h^n)$. The plane with the highest possible order of contact with the curve at P_0 is called the osculating plane at P_0 .

With every point P of a space curve of class C_2 is associated an orthonormal triple of unit vectors: the tangent vector \mathbf{t} , the principal normal vector \mathbf{n} and the binormal vector \mathbf{b} (Figure 4.1.1). The principal normal vector is the unit vector normal to the curve at P which lies in the osculating plane. The binormal vector is the unit vector perpendicular to the osculating plane such that the three vectors \mathbf{t} , \mathbf{n} and \mathbf{b} in that order form a positively oriented or a right-handed triple. The plane containing \mathbf{t} and \mathbf{n} is the osculating plane. The plane containing \mathbf{n} and \mathbf{b} is the normal plane and the one containing \mathbf{b} and \mathbf{t} is the rectifying plane.



Figure 4.1.1. The Frenet trihedron for a space curve

The derivatives of t, n and b with respect to the arc length parameter give us:

$$\frac{d\mathbf{t}}{ds} = \kappa \mathbf{n}, \qquad \frac{d\mathbf{n}}{ds} = -\kappa \mathbf{t} + \tau \mathbf{b}, \qquad \frac{d\mathbf{b}}{ds} = -\tau \mathbf{n}.$$

These formulae are called the *Frenet* or the *Serret-Frenet* formulae. The coefficients κ and τ are called the *curvature* and *torsion* of the curve respectively.

Curvature is the instantaneous rate of change of the tangent vector to the curve with respect to the arc length parameter. Curvature of a space curve has no sign. Torsion is the instantaneous rate of change of the binormal vector with respect to the arc length parameter. A sign is assigned to the absolute measure of torsion as following:

Let point P correspond to value s of the arc length parameter and let point Q correspond to value s+h. Let line l be the intersection of the osculating planes at P and Q. Note that l is parallel to the vector $\mathbf{b}(s) \ge \mathbf{b}(s+h)$ since $\mathbf{b}(s)$ and $\mathbf{b}(s+h)$ are perpendicular to the osculating planes at points P and Q respectively. Let \mathbf{w} be a vector on line l which points into the same half-space determined by the normal plane into which the vector $\mathbf{t}(s)$ is pointing. If $\mathbf{b}(s)$, $\mathbf{b}(s+h)$ and \mathbf{w} form a positively oriented or a right-handed triple, then torsion at point P has positive sign, otherwise it has negative sign.

A planar curve has zero torsion everywhere. The helix is a space curve with constant torsion. To see this, note that the parametric representation of the helix is given by:

 $x(u) = a\cos u$ $y(u) = a\sin u$ z(u) = bu.

Therefore

$\dot{x}(u) = -a\sin u$	$\dot{y}(u) = a\cos u$	$\dot{z}(u) = b$
$\ddot{x}(u) = -a\cos u$	$\ddot{y}(u) = -a\sin u$	$\ddot{z}(u)=0$
$\ddot{x}(u) = a\sin u$	$\ddot{y}(u) = -a\cos u$	$\ddot{z}(u)=0$
$A = a b \sin u$	$B = -ab\cos u$	$C = a^2$

and

$$\tau = \frac{b}{a^2 + b^2}.$$

Since the curve is represented in parametric form, in order to compute curvature and torsion at each point on the curve, we need to express those quantities in terms of the derivatives of its coordinate functions x(), y() and z(). Note that as in chapter 3, . represents the derivative.
iii. Curvature

In case of an arc-length parametrization, we have:

$$\kappa(s) = |\mathbf{\ddot{r}}(s)| = \sqrt{(\ddot{x})^2 + (\ddot{y})^2 + (\ddot{z})^2}.$$

Given an arbitrary parametrization of the curve:

$$\kappa(u) = |\mathbf{t}_s| = \frac{|\mathbf{t}_u|}{|\dot{\mathbf{r}}(u)|} = \frac{\left|\frac{d}{du}(\dot{\mathbf{r}}(u)/|\dot{\mathbf{r}}(u)|)\right|}{|\dot{\mathbf{r}}(u)|} = \frac{|\dot{\mathbf{r}}(u) \times \ddot{\mathbf{r}}(u)|}{|\dot{\mathbf{r}}(u)|^3}.$$

In coordinate form

$$\kappa(u) = \frac{\sqrt{A^2 + B^2 + C^2}}{((\dot{x})^2 + (\dot{y})^2 + (\dot{z})^2)^{3/2}}$$

where

$$A = \left| egin{array}{cc} \dot{y} & \dot{z} \ y & z \end{array}
ight| \quad B = \left| egin{array}{cc} \dot{z} & \dot{x} \ z & x \end{array}
ight| \quad C = \left| egin{array}{cc} \dot{x} & \dot{y} \ x & y \end{array}
ight|.$$

iv. Torsion

We will first derive an expression for the torsion of a space curve with arc-length parametrization [Goetz 1970]. Multiplying both sides of the third Frenet formula by n results in

$$\tau = -\mathbf{b}_s \mathbf{n} = -(\mathbf{t} \times \mathbf{n})_s \mathbf{n} = -(\mathbf{t}_s \times \mathbf{n})\mathbf{n} - (\mathbf{t} \times \mathbf{n}_s)\mathbf{n} = \mathbf{t} \mathbf{n} \mathbf{n}_s.$$

Note that tnn_s is the *mixed product* of vectors t, n and n_s and is equal to $(t \ge n)n_s$. We now make use of

$$\mathbf{t} = \dot{\mathbf{r}}(s), \quad \mathbf{n} = \frac{\ddot{\mathbf{r}}(s)}{|\kappa|}, \quad \mathbf{n}_s = \frac{\ddot{\mathbf{r}}(s)}{\kappa} - \frac{\kappa_s}{\kappa^2}\ddot{\mathbf{r}}(s)$$

to obtain

$$\tau(s) = \frac{\dot{\mathbf{r}}(s)\ddot{\mathbf{r}}(s)\ddot{\mathbf{r}}(s)}{\kappa(s)^2} = \frac{\dot{\mathbf{r}}(s)\ddot{\mathbf{r}}(s)\ddot{\mathbf{r}}(s)}{\ddot{\mathbf{r}}(s)^2} = \frac{\begin{vmatrix}\dot{x}&\dot{y}&\dot{z}\\\dot{x}&\dot{y}&\dot{z}\\\ddot{x}&\ddot{y}&\dot{z}\end{vmatrix}}{(\ddot{x})^2 + (\ddot{y})^2 + (\ddot{z})^2}.$$

In case of an arbitrary parametrization, we make use of:

$$\dot{\mathbf{r}}(s) = \dot{\mathbf{r}}(u) \frac{dt}{ds}, \qquad \ddot{\mathbf{r}}(s) = \ddot{\mathbf{r}}(u) \left(\frac{dt}{ds}\right)^2 + \dot{\mathbf{r}}(u) \frac{d^2t}{ds^2},$$

and

$$\ddot{\mathbf{r}}(s) = \ddot{\mathbf{r}}(u)\left(\frac{dt}{ds}\right)^3 + 3\dot{\mathbf{r}}(u)\left(\frac{dt}{ds}\right)\frac{d^2t}{ds^2} + \dot{\mathbf{r}}(u)\frac{d^3t}{ds^3}$$

to obtain

$$\tau(u) = \frac{\dot{\mathbf{r}}(u)\ddot{\mathbf{r}}(u)\ddot{\mathbf{r}}(u)}{|\dot{\mathbf{r}}(u)|^{6}} \cdot \frac{|\dot{\mathbf{r}}(u)|^{6}}{(\dot{\mathbf{r}}(u)\mathbf{x}\ddot{\mathbf{r}}(u))^{2}} = \frac{\dot{\mathbf{r}}(u)\ddot{\mathbf{r}}(u)\ddot{\mathbf{r}}(u)}{(\dot{\mathbf{r}}(u)\mathbf{x}\ddot{\mathbf{r}}(u))^{2}}.$$

In coordinate form

$$\tau = \frac{\begin{vmatrix} \dot{x} & \dot{y} & \dot{z} \\ \dot{x} & \dot{y} & \dot{z} \\ \dot{x} & \ddot{y} & \ddot{z} \end{vmatrix}}{A^2 + B^2 + C^2}$$

where A, B and C are as before.

v. Computing curvature and torsion at varying levels of detail

In order to compute κ and τ at varying levels of detail of the curve Γ , functions x(u), y(u) and z(u) are convolved with a Gaussian kernel $g(u,\sigma)$ of width σ :

$$g(u,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{u^2}{2\sigma^2}}.$$

The convolved functions together define the *evolved* curve Γ_{σ} . The convolution of a function f(u) and the Gaussian kernel is defined as:

$$F(u,\sigma) = f(u) \circledast g(u,\sigma) = \int_{-\infty}^{\infty} f(v) \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(u-v)^2}{2\sigma^2}} dv.$$

Furthermore, it is known that

$$\dot{F}(u,\sigma) = \frac{\partial F(u,\sigma)}{\partial u} = f(u) \circledast \frac{\partial g(u,\sigma)}{\partial u}$$
$$\ddot{F}(u,\sigma) = \frac{\partial^2 F(u,\sigma)}{\partial u^2} = f(u) \circledast \frac{\partial^2 g(u,\sigma)}{\partial u^2}$$

and

$$\ddot{F}(u,\sigma) = \frac{\partial^3 F(u,\sigma)}{\partial u^3} = f(u) \circledast \frac{\partial^3 g(u,\sigma)}{\partial u^3}.$$

These properties of convolution can be used to compute curvature and torsion on evolved versions of a space curve. Let

$$X(u,\sigma) = x(u) \circledast g(u,\sigma)$$

$$Y(u,\sigma) = y(u) \circledast g(u,\sigma)$$

and

where

$$Z(u,\sigma) = z(u) \circledast g(u,\sigma).$$

It follows that curvature on an evolved curve Γ_{σ} is given by

$$\kappa = \frac{\sqrt{D^2 + E^2 + F^2}}{(\dot{X}(u,\sigma)^2 + \dot{Y}(u,\sigma)^2 + \dot{Z}(u,\sigma)^2)^{3/2}}$$
$$D = \begin{vmatrix} \dot{Y}(u,\sigma) & \dot{Z}(u,\sigma) \\ \dot{Y}(u,\sigma) & \ddot{Z}(u,\sigma) \end{vmatrix}$$
$$E = \begin{vmatrix} \dot{Z}(u,\sigma) & \dot{X}(u,\sigma) \\ \ddot{Z}(u,\sigma) & \ddot{X}(u,\sigma) \end{vmatrix}$$

and

$$F = \begin{vmatrix} \dot{X}(u,\sigma) & \dot{Y}(u,\sigma) \\ \ddot{X}(u,\sigma) & \ddot{Y}(u,\sigma) \end{vmatrix}$$

and torsion on evolved curve Γ_{σ} is given by

$$\tau = \frac{\begin{vmatrix} \dot{X}(u,\sigma) & \dot{Y}(u,\sigma) & \dot{Z}(u,\sigma) \\ \ddot{X}(u,\sigma) & \dot{Y}(u,\sigma) & \ddot{Z}(u,\sigma) \\ \ddot{X}(u,\sigma) & \ddot{Y}(u,\sigma) & Z(u,\sigma) \end{vmatrix}}{D^2 + E^2 + F^2}$$

where D, E and F are as before.

vi. A multi-scale representation for space curves

The curvature and torsion functions of a space curve specify that curve uniquely up to rotation and translation [Do Carmo 1976]. We therefore propose a representation for a space curve that consists of the *curvature scale space* and *torsion scale space* images of the curve. This representation is a generalization of the curvature scale space representation proposed for planar curves in [Mokhtarian & Mackworth 1986].

The function defined implicitly by the level-crossings

$$\kappa(u,\sigma) = c$$

is the curvature scale space image of Γ and the function defined implicitly by

$$\tau(u,\sigma)=0$$

is the torsion scale space image of Γ .

To compute the torsion scale space image of a space curve $\Gamma = (x(u), y(u), z(u))$, evolved curves Γ_{σ} are computed as σ varies from a small to a large value. The torsion zero-crossing points of each Γ_{σ} are extracted and recorded in the $u-\sigma$ space. The smallest value of σ used is slightly larger than zero and the largest value of σ used in the smallest value of σ that results in a Γ_{σ} with very few torsion zero-crossing points.

The curvature scale space image of a space curve is constructed in a similar fashion. The only difference is that level-crossings rather than zero-crossings are searched for. This is because the curvature of a space curve has only magnitude and no sign. Some care should be given to choosing a suitable value for level c. If c is too large, the number of level-crossing points found on curves Γ_{σ} drops to zero quickly as σ increases and the resulting curvature scale space image will not be very rich and therefore not suitable for matching. If c is too small, the resulting curvature scale space image will contain excessive detail. The actual value used for c is the average of curvature values of points of Γ_{σ_0} where $\sigma_0 \in [0,\sigma_i]$ and σ_t is the largest value of σ used to compute the torsion scale space image will be sufficiently rich for matching and will represent roughly the same range of values of σ represented in the torsion scale space image of Γ .

4.2. The renormalized curvature and torsion scale space images

Mackworth and Mokhtarian [1988] observed that although w is the normalized arc length parameter on the original curve Γ , the parameter u is **not**, in general, the normalized arc length parameter on the evolved curve Γ_{σ} . To correct this problem, we propose the *renormalized* curvature and torsion scale space images.

Let

$$\mathbf{R}(u,\sigma) = (X(u,\sigma), Y(u,\sigma), Z(u,\sigma))$$

and

where

 $w = \Phi_{\sigma}(u)$

$$\Phi_{\sigma}(u) = \frac{\int\limits_{0}^{u} |\mathbf{R}_{v}(v,\sigma)| dv}{\int\limits_{0}^{1} |\mathbf{R}_{v}(v,\sigma)| dv}.$$

Now define

Functions $\hat{X}(w,\sigma)$, $\hat{Y}(w,\sigma)$ and $\hat{Z}(w,\sigma)$ defined by equations (4.2.1) define the renormalized version of each Γ_{σ} . That is, each evolved curve Γ_{σ} is reparametrized by its normalized arc length parameter w.

Notice that

 $\Phi_{\sigma}(0) = 0$ $\Phi_{\sigma}(1) = 1$

and

$$\frac{d\Phi_{\sigma}(u)}{du} = \frac{|\mathbf{R}_{u}(u,\sigma)|}{\int\limits_{0}^{1} |\mathbf{R}_{v}(v,\sigma)| dv} > 0 \quad \text{ at non-singular points.}$$

Also

 $\Phi_0(u)=u.$

 $\Phi_{\sigma}(u)$ deviates from the identity function $\Phi_{\sigma}(u) = u$ only to the extent to which the scale-related statistics deviate from stationarity along the original curve.

The function defined implicitly by

 $\kappa(w,\sigma) = c$

is the *renormalized* curvature scale space image of Γ and the function defined implicitly by

 $\tau(w,\sigma)=0$

is the renormalized torsion scale space image of Γ .

4.3. The resampled curvature and torsion scale space images

Note that as a space curve evolves according to the process defined in section **B**, the parametrization of its coordinate functions x(u), y(u) and z(u) does not change. In other words, the function mapping values of the parameter u of the original coordinate functions x(u), y(u) and z(u) to the values of the parameter u of the smoothed coordinate functions $X(u,\sigma)$, $Y(u,\sigma)$ and $Z(u,\sigma)$ is the identity function.

For both theoretical and practical reasons, it is interesting to generalize the definition of evolution so that the mapping function can be different from the identity function. Again let Γ be defined by:

 $\Gamma = \{ (x(w), y(w), z(w)) | w \in [0,1] \}.$

The generalized evolution which maps Γ to Γ_{σ} is now defined by:

$$\Gamma \longrightarrow \Gamma_{\sigma} = \{ (X(W,\sigma), Y(W,\sigma), Z(W,\sigma)) | W \in [0,1] \}$$

where

 $X(W,\sigma) = x(W) \circledast g(W,\sigma)$ $Y(W,\sigma) = y(W) \circledast g(W,\sigma)$

and

 $Z(W,\sigma) = z(W) \circledast g(W,\sigma).$

Note that

 $W = W(w,\sigma)$

and

 $W(w,\sigma_0)$

where σ_0 is any value of σ , is a continuous and monotonic function of w. This condition is necessary to ensure physical plausibility since W is the parameter of the evolved curve Γ_{σ} .

A specially interesting case is when W always remains the arc length parameter as the curve evolves. When this criterion is satisfied, the evolution of Γ is referred to as *arc length* evolution. An explicit formula for W can be derived [Gage & Hamilton 1986].

Let

$$\mathbf{R}(W,\sigma) = (X(W,\sigma), Y(W,\sigma), Z(W,\sigma)).$$

The Frenet equations for a planar curve are given by

$$\frac{\partial \mathbf{t}}{\partial u} = \left| \frac{\partial \mathbf{R}}{\partial u} \right| \kappa \, \mathbf{n}$$

and

$$\frac{\partial \mathbf{n}}{\partial u} = -\left|\frac{\partial \mathbf{R}}{\partial u}\right| \kappa \mathbf{t} + \left|\frac{\partial \mathbf{R}}{\partial u}\right| \tau \mathbf{b}.$$

Let $t = \sigma^2/2$. Observe that

$$\frac{\partial}{\partial t}(|\frac{\partial \mathbf{R}}{\partial u}|^2) = \frac{\partial}{\partial t}(\frac{\partial \mathbf{R}}{\partial u} \cdot \frac{\partial \mathbf{R}}{\partial u}) = 2(\frac{\partial \mathbf{R}}{\partial u} \cdot \frac{\partial^2 \mathbf{R}}{\partial u \partial t}).$$

Note that

$$rac{\partial \mathbf{R}}{\partial u} = |rac{\partial \mathbf{R}}{\partial u}|$$
t

and

$$\frac{\partial \mathbf{R}}{\partial t} = \kappa \mathbf{n}$$

since the Gaussian function satisfies the heat equation. It follows that

$$\frac{\partial}{\partial t}(|\frac{\partial \mathbf{R}}{\partial u}|^2) = 2(|\frac{\partial \mathbf{R}}{\partial u}|\mathbf{t}.\frac{\partial}{\partial u}(\kappa \mathbf{n})) = 2(|\frac{\partial \mathbf{R}}{\partial u}|\mathbf{t}.(\frac{\partial \kappa}{\partial u}\mathbf{n} - |\frac{\partial \mathbf{R}}{\partial u}|\kappa^2\mathbf{t} + |\frac{\partial \mathbf{R}}{\partial u}|\kappa\tau\mathbf{b})) = -2|\frac{\partial \mathbf{R}}{\partial u}|^2\kappa^2.$$

Therefore

$$2\left|\frac{\partial \mathbf{R}}{\partial u}\right| \frac{\partial}{\partial t} \left|\frac{\partial \mathbf{R}}{\partial u}\right| = -2\left|\frac{\partial \mathbf{R}}{\partial u}\right|^2 \kappa^2$$

Or

$$\frac{\partial}{\partial t} \left| \frac{\partial \mathbf{R}}{\partial u} \right| = - \left| \frac{\partial \mathbf{R}}{\partial u} \right| \kappa^2.$$

Let L denote the length of the curve. Now observe that

$$\frac{\partial L}{\partial t} = \int_0^L \frac{\partial}{\partial t} \left| \frac{\partial \mathbf{R}}{\partial u} \right| du = -\int_0^L \left| \frac{\partial \mathbf{R}}{\partial u} \right| \kappa^2 du = -\int_0^1 \kappa^2 dw.$$

Since the value w_0 of the normalized arc length parameter w at a point P measures the length of the curve from the starting point to point P, it follows that

$$\frac{\partial W}{\partial t} = -\int_{0}^{W} \kappa^{2}(W,t) \, dW$$

and therefore

$$W(w,t) = -\iint_{0}^{tW} \kappa^{2}(W,t) \, dW \, dt + w. \tag{4.3.1}$$

Note that

W(w,0)=w.

The function defined implicitly by

$$\kappa(W,\sigma)=c$$

is the resampled curvature scale space of Γ and the function defined implicitly by

$$\tau(W,\sigma)=0$$

is the resampled torsion scale space of Γ .

Since the function $\kappa(W,t)$ in (4.3.1) is unknown, W(w,t) can not be computed directly from (4.3.1). However, the resampled curvature and torsion scale

space images can be computed in a simple way: A Gaussian filter based on a small value of the standard deviation is computed. The curve Γ is parametrized by the normalized arc length parameter and convolved with the filter. The resulting curve is reparametrized by the normalized arc length parameter and convolved again with the same filter. This process is repeated until the curve has very few torsion zero-crossing points. The curvature level-crossings of each curve are marked in the resampled curvature scale space image and the torsion zero-crossings of each curve are marked in the resampled torsion scale space image.

4.4. Evolution and arc length evolution properties of space curves

This section contains a number of results on evolution and arc length evolution of space curves as defined in sections 4.1 and 4.3. The proofs can be found in appendix B.

The first five theorems express a number of global properties of space curves during evolution and arc length evolution.

Theorem 4.4.1. Evolution and arc length evolution of a space curve are invariant under rotation, uniform scaling and translation of the curve.

Theorem 4.4.2. A closed space curve remains closed during evolution and arc length evolution.

Theorem 4.4.3. A connected space curve remains connected during evolution and arc length evolution.

Theorem 4.4.4. The center of mass of a space curve is invariant during evolution and arc length evolution.

Theorem 4.4.5. Let Γ be a closed space curve and let G be its convex hull. Γ remains inside G during evolution and arc length evolution.

The following theorem also appeared in Mokhtarian [1989]. It concerns the uniqueness properties of the torsion scale space image.

Theorem 4.4.6. Let Γ be a space curve in C_3 . Let $\tau(u)$ and $\kappa(u)$ represent the torsion and curvature functions of Γ respectively. The derivatives at a single point on one torsion zero-crossing contour in the regular, resampled or renormalized torsion scale space image of Γ determines the function $\beta(u) = \tau(u)\kappa^2(u)$ uniquely modulo a scale factor (except on a set of measure zero).

The following theorem makes explicit the conditions under which new torsion zero-crossings will not be observed at the higher scales of torsion scale space images.

Theorem 4.4.7. Let Γ be a space curve in C_3 . If all evolved and arc length evolved curves Γ_{σ} are in C_3 and torsion is bounded at every point of Γ during evolution, then all extrema of contours in the regular, renormalized and resampled torsion scale space images of Γ are maxima.

Theorems 4.4.8 and 4.4.9 first appeared in [Mokhtarian 1988b]. They concern the local behaviour of space curves just before and just after the formation of cusp points during evolution.

Theorem 4.4.8. Let $\Gamma = (x(w), y(w), z(w))$ be a space curve in C_1 and let x(w), y(w) and z(w) be polynomial functions of w. Let $\Gamma_{\sigma} = (X(W,\sigma), Y(W,\sigma), Z(W,\sigma))$ be an evolved or arc length evolved version of Γ with a cusp point at W_0 . There is a $\delta > 0$ such that either $\Gamma_{\sigma-\delta}$ intersects itself in a neighborhood of point W_0 or two projections of $\Gamma_{\sigma-\delta}$ intersect themselves in a neighborhood of point W_0 .

Theorem 4.4.9: Let $\Gamma = (x(w), y(w), z(w))$ be a space curve in C_1 and let x(w), y(w) and z(w) be polynomial functions of w. Let $\Gamma_{\sigma} = (X(W, \sigma), Y(W, \sigma), Z(W, \sigma))$ be an evolved or arc length evolved version of Γ with a cusp point at W_0 , then either $\Gamma_{\sigma+\delta}$ has two new torsion zero-crossings in a neighborhood of W_0 or a torsion zero-crossing point exists at W_0 on $\Gamma_{\sigma-\delta}$ and $\Gamma_{\sigma+\delta}$.

The last theorem defines other conditions under which new torsion zero-crossings can appear on a space curve.

Theorem 4.4.10: New torsion zero-crossings can appear on a smooth space curve during evolution or arc length evolution in a neighborhood of a point of zero curvature.



Chapter 5:

Algorithms and Implementation

This chapter contains a number of key algorithms which have been used to implement the theories presented in chapters 3 and 4. Each algorithm is followed by a brief complexity analysis. It also contains a discussion of efficiency issues raised when implementing those theories. Chapter 5 ends by proposing a new scale space matching algorithm which has not been implemented yet.

5.1. The convolution masks

The convolution masks are approximations to the Gaussian function

$$G(u,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-u^2}{2\sigma^2}}$$

its first derivative

$$G_{u}(u,\sigma) = \frac{-u}{\sigma^{3}\sqrt{2\pi}} e^{\frac{-u^{2}}{2\sigma^{2}}}$$

its second derivative

$$G_{uu}(u,\sigma) = rac{u^2 - \sigma^2}{\sigma^5 \sqrt{2\pi}} \; e^{rac{-u^2}{2\sigma^2}}$$

and its third derivative

$$G_{uuu}(u,\sigma) = \frac{3u\sigma^2 - u^3}{\sigma^7 \sqrt{2\pi}} e^{\frac{-u^2}{2\sigma^2}}.$$

The Gaussian function and its derivatives have infinite extent but in order to compute convolutions, they must be approximated by finite masks. For a given value of σ , the width of the Gaussian function, the size of the corresponding mask is set to 12σ . As a result, the ratio of the height of the function at the cutoff point to its height at its center point is:

$$e^{\frac{-(6\sigma)^2}{2\sigma^2}} = e^{\frac{-36\sigma^2}{2\sigma^2}} = e^{-18\sigma^2}$$

which is a very small ratio and negligible. Furthermore, the size of the mask is always set to an odd number so that it will be symmetric. Integer values of u are used to compute the mask elements. The mask elements are stored in a *double* C array and all computations are carried out using *double* arithmetic. As a result, the mask elements add up to values very close to one and no adjustments to their values were considered necessary. In fact, for $\sigma = 1.0$, the sum of mask elements computed using algorithm MASK was equal to 1.000000. Our experience shows that computations which use the first two derivatives, such as computation of curvature, are stable for values of σ larger than 1.0 (corresponding to a mask which covers 13 points on the sampled input curve) and computations which use the first three derivatives, such as computation of torsion, are stable for values of σ higher than 2.0 (corresponding to a mask which covers 25 points on the sampled input curve). The following is an algorithm for a function that computes a mask approximating the Gaussian function (or one of its derivatives):

Algorithm: MASK

- 1. Let size = 12σ
- 2. If size is even then size = size +1
- 3. Let start = (1 size) / 2
- 4. Let finish = (size 1) / 2
- 5. For i = start to finish do $mask[i] = Gauss(i, \sigma)$

End of Algorithm: MASK

Note that algorithm MASK runs in time linearly proportional to σ or in $O(\sigma)$ time since the size of the computed mask is a linear function of the value of σ .

To convolve a mask with an array of data at element number k, the center element of the mask (corresponding to u = 0, where u is the first parameter in $G(u,\sigma)$) is multiplied by array element k, the mask element corresponding to u=1is multiplied by array element k+1, the mask element corresponding to u=-1 is multiplied by array element k-1, and so on. This process is repeated until there are no more mask elements left to multiply. If the first or last array element is reached and the input curve is closed, there will be a wrap around: the array element after the last element is the first element. If the first or last array element is reached and the input curve is open, the first or last array element is repeated as many times as necessary (This is the method that was used in this thesis. Note that other methods such as the extrapolation of the curve beyond the endpoint are also possible.) All products are then added up. The result is the value of convolution at element number k.

5.2. The sampling algorithm

The input curve, whether two-dimensional or three-dimensional, is represented by a sequence of points. Consecutive points of that sequence are assumed to be connected by straight line segments. However, the lengths of these line segments are not necessarily equal. It is therefore necessary to sample the input curve at equal-length intervals so that its evolved versions or its multi-scale representations can be computed. The sampling function takes the number of points to be sampled as an input parameter, determines the size of the sampling interval, Δs , and returns a list of points such that each pair of consecutive points on that list have a distance equal to Δs on the original curve.

Let P_1, P_2, \ldots, P_m be the sequence of points representing the original curve and let Q_1, Q_2, \ldots, Q_n be the sequence of sampled points on that curve. Note that $P_1 = Q_1$. If the input curve is open, $P_m = Q_n$. The following is an algorithm which describes how the sampling function generates its output. Note that function d returns the Euclidean distance between two points.

Algorithm: SAMPLE

1. Let
$$Q_1 = P_1$$

2. Let $k = 2$
3. For $i = 2$ to n do
3.1. Let $old_k = k$
3.2. Let $D = d_{Q_{i-1}P_k}$
3.3. while ($D < \Delta s$) do
3.3.1. $k = k+1$
3.3.2. $D = D + d_{P_kP_{k-1}}$
3.4. If ($k > old_k$) then do
3.4.1. $w = (D - \Delta s) / d_{P_kP_{k-1}}$
3.4.2. $Q_i = w P_{k-1} + (1-w) P_k$
else do
3.4.3. $w = (D - \Delta s) / D$
3.4.4. $Q_i = w Q_{i-1} + (1-w) P_k$

End of Algorithm: SAMPLE

Note that algorithm SAMPLE runs in time linearly proportional to the larger of m and n.

5.3. Algorithms for representations of planar curves

This section contains algorithms for functions which compute the regular, renormalized and resampled curvature scale space images of planar curves.

Note that an evolved version of a planar curve is computed by first sampling it according to algorithm SAMPLE and then convolving each of its x and ycoordinate functions with a mask corresponding to the input value of σ .

The following is an algorithm to compute the curvature scale space image of a planar curve. Note that this image is stored in array *css* which is assumed to be initialized to zero.

Algorithm: CURVATURE_SCALE_SPACE

```
1. Read(input curve)
2. Call SAMPLE \rightarrow n (* no. of sampled points *)
3. Let \sigma = \sigma_0 (* \sigma_0 is relatively small; usually one *)
4. Let p=1 (* p holds the no. of curvature zero-crossings found *)
5. While (p > 0) do
  5.1. Let p = 0
  5.2. Call MASK_1ST_DERIV(\sigma) (* computes mask based on 1st deriv. of Gaussian *)
  5.3. Call MASK_2ND_DERIV(\sigma) (* computes mask based on 2nd deriv. of Gaussian *)
  5.4. Call CONVOLVE(x[],mask_1st[],1) \rightarrow x'[1]
  5.5. Call CONVOLVE(y/], mask_1st/],1) \rightarrow y'[1]
  5.6. Call CONVOLVE(x/],mask_2nd[],1) \rightarrow x''[1]
  5.7. Call CONVOLVE(y[], mask_2nd[],1) \rightarrow y"[1]
  5.8. Let \kappa[1] = x'[1]y''[1] - x''[1]y'[1]
  5.9. For i = 2 to n do
        5.9.1. Call CONVOLVE(x[],mask_1st[],i) \rightarrow x'[i]
        5.9.2. Call CONVOLVE(y/],mask_1st/],i) \rightarrow y'[i]
        5.9.3. Call CONVOLVE(x] ], mask_2nd[],i) \rightarrow x''[i]
        5.9.4. Call CONVOLVE(y[], mask_2nd[], i) \rightarrow y"[i]
        5.9.5. Let \kappa[i] = x'[i]y''[i] - x''[i]y'[i]
        5.9.6. If ((\kappa[i] > 0) and (\kappa[i-1] < 0)) or ((\kappa[i] < 0) and (\kappa[i-1] > 0)) then
              5.9.6.1. If (|\kappa/i|| > |\kappa/i-1||) then css/i-1,\sigma = 1
                       else css[i,\sigma] = 1
             5.9.6.2. p = p + 1
       5.9.7. \sigma = \sigma + \Delta \sigma
```

End of Algorithm: CURVATURE_SCALE_SPACE

Function CONVOLVE takes $O(\sigma)$ time to return a value. Therefore each row of the curvature scale space image takes $O(n\sigma)$ time to compute. Since the number of rows of a curvature scale space image is linearly related to σ , it takes $O(n\sigma^2)$ to compute a curvature scale space image using algorithm CURVATURE_SCALE_SPACE.

The algorithm to compute the renormalized curvature scale space image is similar to algorithm CURVATURE_SCALE_SPACE and therefore only the differences will be pointed out:

Algorithm RENORMALIZED_CURVATURE_SCALE_SPACE has a few extra steps between steps 5.1 and 5.2 to compute the total length of the curve and the length up to each point of the curve as following:

a.1. Let dist[1] = 0 (* array dist holds the distances up to each point *)
a.2. Let L = 0 (* L holds the total length of the sampled curve *)
a.3. Call MASK_GAUSS(σ) (* computes a mask based on the Gaussian fn *)
a.4. Call CONVOLVE(x[],mask_gauss[],1) → X[1]
a.5. Call CONVOLVE(y[],mask_gauss[],1) → Y[1]
a.6. For i = 2 to n do

a.6.1. Call CONVOLVE(x[],mask_gauss[],i) → X[i]
a.6.2. Call CONVOLVE(x[],mask_gauss[],i) → Y[i]
a.6.3. L = L + d_{((X[i],Y[i]),(X[i-1],Y[i-1]))}
a.6.4. dist[i] = L

Step 5.9.6.1 in algorithm CURVATURE_SCALE_SPACE should also be modified as following:

5.9.6.1. If $(|\kappa[i]| > |\kappa[i-1]|)$ then $css[(dist[i-1]/L)n,\sigma] = 1$ else $css[(dist[i]/L)n,\sigma] = 1$

Algorithm RENORMALIZED_CURVATURE_SCALE_SPACE also takes $O(n\sigma^2)$ time to compute a curvature scale space image, however, in practice it takes longer than algorithm CURVATURE_SCALE_SPACE due to the extra time needed to compute the length of the evolved curve in each iteration.

Note that an arc length evolved version of a planar curve is computed by first sampling it using function SAMPLE and then convolving its x and y coordinate functions with a mask corresponding to a small σ . The output curve is sampled again using SAMPLE and its x and y coordinate functions are convolved again with the same mask. This process is repeated for as many iterations as necessary.

The third algorithm in this section computes the resampled curvature scale space image of a planar curve. Again only the differences between this algorithm and algorithm CURVATURE_SCALE_SPACE will be pointed out:

Algorithm RESAMPLED_CURVATURE_SCALE_SPACE makes only one call to MASK_1ST_DERIV and MASK_2ND_DERIV and therefore these two calls come just before step 5. Furthermore, step 2 should now be:

2. Call SAMPLE $\rightarrow n_0$

step 5.9.7 should be changed to

5.9.7. Call SAMPLE \rightarrow n

and step 5.9.6.1 should be modified as following:

5.9.6.1. If $(|\kappa[i]| > |\kappa[i-1]|)$ then $css[((i-1)/n)n_0,\sigma] = 1$ else $css[(i/n)n_0,\sigma] = 1$

Algorithm RESAMPLED_CURVATURE_SCALE_SPACE always uses masks of constant size and therefore takes O(n) time to compute each row of a resampled curvature scale space image and O(rn) time to compute an entire image with r rows.

5.4. Algorithms for representations of space curves

This section contains algorithms for functions which compute the regular, renormalized and resampled torsion scale space images of space curves.

Note that an evolved version of a space curve is computed by first sampling it according to algorithm SAMPLE and then convolving each of its x, y and z coordinate functions with a mask corresponding to the input value of σ .

Our experience showed that choosing a threshold value to be used when computing the curvature scale space image of a space curve was not straightforward and that variations in its magnitude significantly affected the structure of the curvature scale space image. Therefore the curvature scale space representation of a space curve is not invariant with respect to scaling since scaling changes curvature values on a space curve. Since the torsion scale space image of a space curve is in fact invariant with respect to scaling and is usually sufficient to distinguish the curve from other curves it is being compared to, it was decided not to compute the curvature scale space image of the space curves shown in chapter 6. It is therefore not necessary to present an algorithm for this class of representations. Note however, that the algorithm would be quite similar to algorithm CURVATURE_SCALE_SPACE in section 5.3.

The following is an algorithm to compute the torsion scale space image of a space curve. Note that this image is stored in array *tss* which is assumed to be initialized to zero.

Algorithm: TORSION_SCALE_SPACE

1. Read(input curve) 2. Call SAMPLE $\rightarrow n$ (* no. of sampled points *) 3. Let $\sigma = \sigma_0$ (* σ_0 is relatively small; usually two *) 4. Let p=1 (* p holds the no. of torsion zero-crossings found *) 5. While $(p > p_0)$ do (* where p_0 is a small number *) 5.1. Let p = 05.2. Call MASK_1ST_DERIV(σ) (* computes mask based on 1st deriv. of Gaussian *) 5.3. Call MASK_2ND_DERIV(σ) (* computes mask based on 2nd deriv. of Gaussian *) 5.4. Call MASK_3RD_DERIV(σ) (* computes mask based on 3rd deriv. of Gaussian *) 5.5. Call CONVOLVE(x[],mask_1st[],1) $\rightarrow x'[1]$ 5.6. Call CONVOLVE(y/], mask_1st/],1) \rightarrow y'[1] 5.7. Call CONVOLVE(z[],mask_1st[],1) $\rightarrow z'[1]$ 5.8. Call CONVOLVE(x/],mask_2nd/],1) $\rightarrow x''[1]$ 5.9. Call CONVOLVE(y/],mask_2nd[],1) \rightarrow y"[1] 5.10. Call CONVOLVE(z[],mask_2nd[],1) $\rightarrow z''$ [1] 5.11. Call CONVOLVE(x[],mask_3rd[],1) $\rightarrow x'''[1]$ 5.12. Call CONVOLVE(y/], mask_3rd/], 1) $\rightarrow y'''$ [1] 5.13. Call CONVOLVE($z[], mask_3rd[], 1$) $\rightarrow z'''[1]$ 5.14. Let $\tau[1] = z'''x'y'' - z'''x''y' + y'''z'x'' - y'''x'z'' + x'''y'z'' - x'''z'y''$ 5.15. For i = 2 to n do 5.15.1. Call CONVOLVE(x[],mask_1st[],i) $\rightarrow x'/i$] 5.15.2. Call CONVOLVE(y/], mask_1st/],i) $\rightarrow y'[i]$ 5.15.3. Call CONVOLVE(z[],mask_1st[],i) $\rightarrow z'[i]$ 5.15.4. Call CONVOLVE(x/],mask_2nd/],i) $\rightarrow x''[i]$ 5.15.5. Call CONVOLVE(y/],mask_2nd[],i) $\rightarrow y''[i]$ 5.15.6. Call CONVOLVE(z[],mask_2nd[],i) $\rightarrow z''[i]$ 5.15.7. Call CONVOLVE(x/], mask_3rd[],i) $\rightarrow x'''[i]$ 5.15.8. Call CONVOLVE(y/],mask_3rd[],i) $\rightarrow y'''[i]$ 5.15.9. Call CONVOLVE(z[], mask_3rd[],i) $\rightarrow z'''[i]$ 5.15.10. Let $\tau[i] = z'''x'y'' - z'''x''y' + y'''z'x'' - y'''x'z'' + x'''y'z'' - x'''z'y''$ 5.15.11. If $((\tau | i | > 0) \text{ and } (\tau | i - 1 | < 0))$ or $((\tau | i | < 0) \text{ and } (\tau | i - 1 | > 0))$ then 5.15.11.1. If $(|\tau/i|| > |\tau/i-1||)$ then $tss/i-1,\sigma = 1$ else $tss[i,\sigma] = 1$ 5.15.11.2. p = p + 15.15.12. $\sigma = \sigma + \Delta \sigma$

End of Algorithm: TORSION_SCALE_SPACE

The algorithm to compute the renormalized torsion scale space image is similar to algorithm TORSION_SCALE_SPACE and therefore only the differences will be pointed out:

Algorithm RENORMALIZED_TORSION_SCALE_SPACE has a few extra steps between steps 5.1 and 5.2 to compute the total length of the curve and the length up to each point of the curve as following:

a.1. Let dist[1] = 0 (* array dist holds the distances up to each point *) a.2. Let L = 0 (* L holds the total length of the sampled curve *) a.3. Call MASK_GAUSS(σ) (* computes a mask based on the Gaussian fn *) a.4. Call CONVOLVE(x[],mask_gauss[],1) \rightarrow X[1] a.5. Call CONVOLVE(y[],mask_gauss[],1) \rightarrow Y[1] a.6. Call CONVOLVE(x[],mask_gauss[],1) \rightarrow Z[1] a.7. For i = 2 to n do a.7.1. Call CONVOLVE(x[],mask_gauss[],i) \rightarrow X[i] a.7.2. Call CONVOLVE(x[],mask_gauss[],i) \rightarrow Y[i] a.7.3. Call CONVOLVE(y[],mask_gauss[],i) \rightarrow Z[i] a.7.4. $L = L + d_{((X[i], Y[i], Z[i]), (X[i-1], Y[i-1], Z[i-1]))}$ a.7.5. dist[i] = L

Step 5.15.11.1 in algorithm TORSION_SCALE_SPACE should also be modified as following:

5.15.11.1. If $(|\tau[i]| > |\tau[i-1]|)$ then $tss[(dist[i-1]/L)n,\sigma] = 1$ else $tss[(dist[i]/L)n,\sigma] = 1$

Note that an arc length evolved version of a space curve is computed by first sampling it using function SAMPLE and then convolving its x, y and z coordinate functions with a mask corresponding to a small σ . The output curve is sampled again using SAMPLE and its x, y and z coordinate functions are convolved again with the same mask. This process is repeated for as many iterations as necessary.

The third algorithm in this section computes the resampled torsion scale space image of a space curve. Again only the differences between this algorithm and algorithm TORSION_SCALE_SPACE will be pointed out:

Algorithm RESAMPLED_TORSION_SCALE_SPACE makes only one call to MASK_1ST_DERIV, MASK_2ND_DERIV and MASK_3RD_DERIV and therefore these two calls come just before step 5. Furthermore, step 2 should now be: 2. Call SAMPLE $\rightarrow n_0$

step 5.15.12 should be changed to

5.15.12. Call SAMPLE $\rightarrow n$

and step 5.15.11.1 should be modified as following:

5.15.11.1. If $(|\tau[i]| > |\tau[i-1]|)$ then $tss[((i-1)/n)n_0,\sigma] = 1$ else $tss[(i/n)n_0,\sigma] = 1$

Similar analyses show that algorithms TORSION_SCALE_SPACE and RENORMALIZED_TORSION_SCALE_SPACE take $O(n\sigma^2)$ time to compute torsion scale space images and algorithm RESAMPLED_TORSION_SCALE_SPACE takes O(rn) time to compute a torsion scale space image.

5.5. Efficiency issues

Since the computation of curvature and torsion scale space images calls for the computation of a large number of convolutions, it is appropriate to investigate ways of rendering those computations more efficient:

1. The computation of curvature and torsion scale space images can be made more efficient by *tracking* the zero-crossing points across scales. A small change in the value of σ , the width of the Gaussian, will result in a small change in the location of a curvature or torsion zero-crossing point. It follows that if the location of curvature or torsion zero-crossing points of a curve are known at level σ_0 , then the location of those points at level $\sigma_0 + \Delta \sigma$ can be determined by searching only a neighborhood of the zero-crossing points at level σ_0 . This method is specially useful when it is known that no new zero-crossing points will be encountered at the higher levels of the scale space image, for example when computing the resampled curvature scale space image of a simple curve. This method was used to speed up the computation of several curvature scale space images shown in chapter 6.

Let n and c be the number of sampled points and the number of curvature zero-crossings on the input curve respectively. It follows that the curvature scale space image can be computed in $O(c\sigma^2)$ time. Note, however, that in general after a small number of iterations, the number of zero-crossings drops to a small number. If it is assumed that after a constant number of iterations, the number of zero-crossings drops below a constant, then it follows that the curvature scale space image can be computed in $O(n) + O(\sigma)$ time. 2. Fast Fourier transforms can also be used to speed up the computation of curvature and torsion scale space images when σ becomes large. This method is an alternative to tracking when there is reason to believe that new curvature or torsion zero-crossing points will be encountered at the higher scales of the curvature or torsion scale space image.

Since using the FFT method, it takes $O(n\log n)$ time to compute each row [Aho *et al.* 1974], computation of the entire curvature scale space image takes $O(\sigma n\log n)$ time.

3. The computation of curvature and torsion scale space images can also be made more efficient by using an *incremental* method. Note that algorithm CURVATURE_SCALE_SPACE in section 5.3 uses the original data in each iteration and a larger value of σ to compute the next higher level of description of the input curve. This can lead to quite large values of σ and a long computation time. The computation time can be significantly reduced by using the data obtained at the last scale level to compute the next level of description. To see why this is possible, note that if f(x) is a continuous function and $g_1(x)$ and $g_2(x)$ are Gaussian functions with widths σ_1 and σ_2 , then

$$(f \circledast g_1) \circledast g_2 = (f \circledast g_2) \circledast g_1 = f \circledast g_3$$

where g_3 is another Gaussian function with width σ_3 and

$$\sigma_3{}^2 = \sigma_1{}^2 + \sigma_2{}^2.$$

The computation of the curvature scale space image again takes $O(n\sigma^2)$ time, however, this technique results in much smaller values of σ used to compute the curvature or torsion scale space image. This method was used to speed up the computations of the torsion scale space images shown in chapter 6.

4. Another way to speed up the computation of curvature and torsion scale space images is to use parallelism and specialized hardware such as convolution chips. This method can also be combined with methods 1,2 and 3 for maximum efficiency. For example, suppose we wish to compute the curvature scale space image of a planar curve, and that n processors are available for parallel use. The curvature scale space image can be divided into n segments using lines perpendicular to the σ -axis or lines perpendicular to the u-axis and each processor given the job of computing the curvature scale space image of the segment assigned to it. If segmentation is done along the σ -axis, fast Fourier transforms can be used to compute the evolved curve at the starting scale level of each segment and the incremental method or the tracking method can be used to compute the rest of the segment.

If it is assumed that enough processors are available such that each processor computes a constant number of rows of the curvature or torsion scale space image, then the entire curvature or torsion scale space image can be computed in $O(n\log n)$ time.

Note that almost all the curvature and torsion scale space images shown in chapter 6 have been computed at high resolution so that small details as well as the basic structure of the scale space images can be recognized. In practice, one may wish to compute the curvature or torsion scale space image at a considerably lower resolution by increasing the length of the sampling interval and the magnitude of the step size in σ . The low-resolution scale space image will be considerably faster to compute and should be adequate for most recognition tasks. For example if the length of the sampling interval is doubled, the number of sampled points and the largest value of σ used will be reduced by a factor of two. It follows the required computation time that using algorithm CURVATURE_SCALE_SPACE or algorithm TORSION_SCALE_SPACE will be reduced by a factor of eight.

5.6. Matching scale space images

The scale space matching algorithm in [Mokhtarian & Mackworth 1986] makes assumptions about curvature scale space representations which are not always true. For example, it assumes that CSS representations always have a hierarchical structure, i.e., curvature zero-crossing contours never intersect and that two curves which are close in shape always have the same hierarchical structure. The following is a scale space matching algorithm which should run considerably faster than the algorithm in Mokhtarian & Mackworth [1986]. It also does not make possibly incorrect assumptions about the structure of a scale space representation. This algorithm has not been implemented yet.

Note that the extrema of the curvature or torsion zero-crossing contours in a curvature or torsion scale space representation are important and distinguishable points. They occur at different scales and provide us with an abundance of features to be used for the matching process. They can be located easily and they are the natural candidate points to use in order to determine the correspondence between the scale space images to be matched since, unlike other zero-crossings, each extremum is isolated in a region which contains it. We therefore proceed by extracting all extrema of the zero-crossing contours from both scale space representations R_1 and R_2 to be matched. Let E_1 and E_2 be the set of points thus obtained. Let P_1, P_2, \dots, P_n be the extrema in E_1 ordered by height and let Q_1, Q_2, \dots, Q_n be the extrema in E_2 also ordered by height. The problem now is to assign each point of E_1 to a distinct point of E_2 which will minimize the total matching cost. The cost of assigning a point P_i to point Q_j is equal to the Euclidean distance between those two points multiplied by the height of those points since matches at higher scales are considered more important. The actual algorithm then is as follows:

Algorithm: SCALE_SPACE_MATCHER

- 1. Create n nodes corresponding to the possible match of P_1 and each of the extrema in E_2 . Assign a cost of 0 to each node.
- 2. For each node (P_1, Q_i) , compute the scale space transformation parameters which will transform the scale space coordinates of Q_i to the scale space coordinates of P_1 . If the node (P_1, Q_i) represents a correct match, then these transformation parameters will cause the remaining extrema of E_1 to overlap with the remaining extrema of E_2 in a one-to-one fashion.
- 3. For each node, apply the transformation parameters to the coordinates of Q_j $(1 \le j \le n)$ to compute their new coordinates for that node.
- 4. Expand each node one step: for each node find the closest unassigned extremum in E_2 to P_2 and assign them to each other. Note that the two extrema that are assigned to each other should be of the same type: minima or maxima. Compute the cost of that assignment and add it to the node cost.
- 5. Find the lowest cost node and expand it one step: assign the next unassigned extremum in E_1 to its closest unassigned extremum in E_2 and add the assignment cost to the node cost.
- 6. If there are no more unassigned extrema from either E_1 or E_2 in the lowest cost node, then STOP: the lowest cost match has been found. Otherwise, go to step 5.

End of Algorithm: SCALE_SPACE_MATCHER

Note that in general, if representation R_1 is a good match for representation R_2 , then point P_1 will usually match point Q_1 , P_2 will match Q_2 , etc. and the algorithm will terminate quickly. However, the algorithm above has been designed to have the capability to handle situations in which two or more of the highest extrema of R_1 or R_2 have roughly the same height.

Chapter 6:

Experimental Results and Evaluation

This chapter shows a number of planar and space curves which were used as input data to the programs described in chapter 5 along with the output obtained from those programs. It also describes a number of experiments which were carried out on the representations. The chapter also contains a discussion of the significance of the theorems of chapters 3 and 4. It ends with an evaluation of the curvature and torsion scale space representations proposed in chapters 3 and 4 according to the criteria proposed in chapter 1.

6.1. Planar Curves

Algorithm CURVATURE_SCALE_SPACE in section 5.3 was used to compute the curvature scale space image of three different planar curves. Figure 6.1.1 shows the coastline of Africa. This curve was used as the first input curve. Figure 6.1.2 shows several evolved version of the coastline of Africa for increasing values of σ , the width of the Gaussian filter. It can be observed that as σ increases, the small-scale features on the curve disappear. Larger values of σ perform a very good job of filtering out the detail on the curve and bringing out its basic structure. A very large value of sigma filters out nearly all of the structure and the curve tends to a circle. Figure 6.1.3 shows the curvature scale space image of Africa. Horizontal lines have been drawn across that image to indicate the values of σ which were used to compute the evolved curves of figure 6.1.2. It can be observed that the smaller features show up as contours which are confined to the fine scales of the curvature scale space image whereas the basic features result in contours which last until the larger scales.







Figure 6.1.2. Africa during evolution

51



Figure 6.1.3. The curvature scale space image of Africa

The second input curve is shown in figure 6.1.4. This is Koch's snowflake curve. Several evolved versions of that curve are shown in figure 6.1.5 and the curvature scale space image of the curve is shown in figure 6.1.6. Horizontal lines have been drawn across that image to indicate the values of σ which were used to compute the curves of figure 6.1.5. Note that Koch's snowflake curve is a fractal curve and its curvature scale space image also has a fractal property.

The third input curve is the curve of figure 6.1.7. This curve is the outline of a design taken from a Persian carpet. Figure 6.1.8 shows the carpet design during evolution and figure 6.1.9 shows the curvature scale space image of the carpet design. Horizontal lines have been drawn across that image to indicate the values of σ which were used to compute the curves of figure 6.1.8. Note that since the carpet design is a symmetric curve, its curvature scale space image is also symmetric.

We then carried out a number of experiments to test the suitability of the representations under various conditions. The purpose of the first experiment was to test the behaviour of the curvature scale space image when a considerable amount of noise corrupts the shape of the input curve. Figure 6.1.10 shows the coastline of Africa with a significant amount of random and uniform noise added to it. The noise at each point of the curve is added along the direction of the vector normal to the curve at that point. Note that the curve can intersect itself. It follows from theorems 3.4.3 and 3.4.5 that new curvature zero-crossings can appear during its evolution. Figure 6.1.11 shows the curvature scale space image of the original Africa figure. As expected, the images shown in figure 6.1.11 show differences in detail. However, a remarkable similarity in the basic structures of the two images can be observed. This experiment shows that the curvature scale space image space image is very stable and reliable even when a significant amount of uniform noise corrupts the shape of the input curve.

The next experiment tested the behaviour of the curvature scale space image under severe noise conditions. Figure 6.1.12 shows the coastline of Africa with severe, uniform noise added to it. Figure 6.1.13 shows the curvature scale space image of the curve of figure 6.1.12 superimposed on the curvature scale space image of the original Africa figure. Even with the presence of severe noise, a very close similarity can be observed between the two images.

The next experiment examined the behaviour of the proposed representations when non-uniform noise corrupts the shape of the input curve. Figure 6.1.14 shows the coastline of Africa with random, non-uniform noise added to its lower half. Note that the curve remains simple after the addition of noise. Figure 6.1.15 shows the curvature scale space image of Africa with non-uniform noise.



Figure 6.1.4. Koch's snowflake curve



(a) $\sigma = 2$



(b) $\sigma = 5$







(d) $\sigma = 20$





Figure 6.1.6. The curvature scale space image of the snowflake

55



Figure 6.1.7. Design from a Persian carpet







(b) $\sigma = 10$







Figure 6.1.8. Carpet design during evolution



Figure 6.1.9. The curvature scale space image of the carpet design



Figure 6.1.10. Coastline of Africa with uniform, random noise



Figure 6.1.11. The CSS image of Africa with uniform noise superimposed on the CSS image of Africa



Figure 6.1.12. Coastline of Africa with severe, uniform noise


Figure 6.1.13. The CSS image of Africa with severe noise superimposed on the CSS image of Africa



Figure 6.1.14. Coastline of Africa with non-uniform, random noise





When compared to the curvature scale space image of Africa shown in figure 6.1.3, it is clear that there does not exist a good overall match of the two images. Figure 6.1.16 shows the renormalized curvature scale space image of Africa and figure 6.1.17 shows the renormalized curvature scale space image of Africa with non-uniform noise. Figure 6.1.18 shows the superposition of the images shown in figures 6.1.16 and 6.1.17. It can be seen that the degree of match shown in figure 6.1.18 is much better than the degree of match of figure 6.1.3 to figure 6.1.19 shows the resampled curvature scale space image of Africa and figure 6.1.20 shows the resampled curvature scale space image of Africa with non-uniform noise. Figure 6.1.21 shows the superposition of the images shown in figure 6.1.19 and 6.1.20. Note that a very close match exists between those two images.

Three different multi-scale representation techniques for planar curves were described in this thesis. These three are: the regular curvature scale space image, the renormalized curvature scale space image and the resampled curvature scale space image. Each representation technique is suitable for a class of specific applications. When low to moderate, uniform noise exists on the curve, the regular curvature scale space image can be used. However, when there is non-uniform or severe noise on the curve or when there are local shape differences between the model curves and the image curves, either the renormalized or the resampled curvature scale space images should be used. Note that the renormalized curvature scale space image is the most computationally intensive. Observations indicate that when there are local shape differences, the resampled curvature scale space images show the best overall match whereas the renormalized curvature scale space images match well at high scales but are more influenced by the shape differences at lower scales. Therefore the choice of the representation technique should depend on the scale level of the curve features that one wishes to emphasize. Table 6.1.1 summarizes the advantages and disadvantages of each representation technique.

This concludes the experiments which were carried out on planar curves. The following is a discussion of the practical significance of the theorems of chapter 3.

Theorem 3.4.1 showed that evolution and arc length evolution of a planar curve are invariant under rotation, uniform scaling and translation of the curve. This shows that the regular, renormalized and resampled curvature scale space images of a planar curve have the *invariance* property [Mokhtarian & Mackworth 1986]. The invariance property is essential since it makes it possible to match a planar curve to another of similar shape which has undergone a transformation consisting of arbitrary amounts of rotation, uniform scaling and translation.



Figure 6.1.16. The renormalized curvature scale space image of Africa



Figure 6.1.17. The renormalized curvature scale space image of Africa with non-uniform noise



Figure 6.1.18. The Renormalized CSS image of Africa with non-uniform noise superimposed on the Renormalized CSS image of Africa



Figure 6.1.19. The Resampled curvature scale space image of Africa



Figure 6.1.20. The Resampled curvature scale space image of Africa with non-uniform noise



Figure 6.1.21. The Resampled CSS image of Africa with non-uniform noise superimposed on the Resampled CSS image of Africa

Theorems 3.4.2 and 3.4.3 showed that connectedness and closedness of a planar curve are preserved during evolution and arc length evolution. These theorems demonstrate that evolution and arc length evolution of a planar curve do not change the physical interpretation of that curve as the boundary of a two-dimensional object.

Representation technique	Advantages	Disadvantages
The Regular Curvature Scale Space Image	 Suitable for transformations consisting of uniform scaling, rotation and translation. Suitable when uniform, low-intensity noise has corrupted the curve. 	• Non-uniform noise or local difference in shape can cause problems.
The Renormalized Curvature Scale Space Image	• More suitable when there is non-uniform noise on the curve or local shape differences exist.	• Most computationally intensive.
The Resampled Curvature Scale Space Image	• Most suitable when there is high-intensity, non-uniform noise or local shape differences exist.	• De-emphasizes shape differences at fine scales.

Table 6.1.1. Comparison of Regular, Renormalizedand Resampled Curvature Scale Space Images.

Theorem 3.4.4 showed that the center of mass of a planar curve does not move as the curve evolves and theorem 3.4.5 showed that a planar curve remains inside its convex hull during evolution and arc length evolution. Together, theorems 3.4.4 and 3.4.5 impose constraints on the physical location of a planar curve as it evolves. These constraints become useful whenever the physical location of curves in an image or their locations with respect to each other is important. A possible application area is stereo matching in which it is advantageous to carry out matching at coarser levels of detail first and then match at fine detail levels to increase accuracy.

Theorem 3.4.6 showed that the curvature scale space images of a planar curve determines that curve uniquely modulo uniform scaling, rotation and translation. This shows that the curvature scale space images satisfy the *uniqueness* property [Mokhtarian & Mackworth 1986]. This property ensures that curves of different shapes do not have the same representation.

Theorems 3.4.8 and 3.4.10 together locally characterize the behaviour of a planar curve just before and just after the formation of a cusp point during evolution and arc length evolution. This behaviour can be used to detect any cusp points that form during evolution or arc length evolution of a planar curve. Such cusp points can then be used effectively to facilitate matching since they provide us with a set of distinctive and easily recognizable features. These theorems also show that self-intersecting curves are described in a natural way by our representation technique. The self-intersection loop gradually grows smaller until it turns into a cusp point and vanishes. In contrast, Asada and Brady's method [1986] enlarges the smaller loop until it becomes as large as the larger loop. Figure 6.1.22 shows a self-intersecting curve during evolution. The self-intersection is resolved through the formation of a cusp point after which the curve becomes simple. Figure 6.1.23 shows the curvature scale space image of the curve of figure 6.1.22. Horizontal lines have been drawn across that image to indicate the values of σ which were used to compute the curves of figure 6.1.22. Figure 6.1.24 shows a convex, self-crossing curve during evolution and figure 6.1.25 shows the curvature scale space image of the curve of figure 6.1.24. Horizontal lines have again been drawn across that image to indicate the values of σ which were used to compute the curves of figure 6.1.24.

Theorem 3.4.7 showed that if a planar curve remains smooth during evolution or arc length evolution, then no new curvature zero-crossings will be observed in its curvature scale space images. Theorem 3.4.8 showed that every planar curve intersects itself during evolution or arc length evolution just before the formation of a cusp point and theorem 3.4.9 showed that simple curves remain simple during arc length evolution. Combining theorems 3.4.7, 3.4.8 and 3.4.9, we conclude that no new curvature zero-crossing points are created during arc length evolution of simple curves. This is an important result since it indicates that new "structure" is not created in the curvature scale space representations of simple curves [Marr & Nishihara 1978]. Note that a subclass of selfcrossing curves also shares this property.

The result stated by theorem 3.4.9 is also very important. Simple planar curves usually represent the boundaries of two-dimensional objects. Arc length evolved versions of those curves can only have physical interpretation as boundaries of two-dimensional objects if they are also simple. Theorem 3.4.9 shows that this is in fact the case. Figure 6.1.26 shows a simple curve and its evolved versions. It can be seen that the curve intersects itself during evolution. Figure 6.1.27 shows the same curve and its arc length evolved versions. As expected, the curve remains simple during arc length evolution.



(a) Original curve



(b) $\sigma = 16$







(d) $\sigma = 40$





Figure 6.1.23. The CSS image of the curve of figure 6.1.22



(a) Original curve



(b) $\sigma = 20$







(d) $\sigma = 40$





Figure 6.1.25. The CSS image of the curve of figure 6.1.24









(d) $\sigma = 25$





Figure 6.1.26. A simple curve during (regular) evolution



(f) After 50 iterations



6.2. Space Curves

We used algorithm TORSION_SCALE_SPACE given in section 5.4 to compute the torsion scale space images of several input space curves. Figure 6.2.1 shows a space curve depicting a fork. Figure 6.2.2 shows several evolved versions of the fork and figure 6.2.3 shows the torsion scale space image of the fork. Horizontal lines have been drawn across that image to indicate the values of σ which were used to compute the curves of figure 6.2.2. Again note that as σ , the width of the Gaussian function, increases, small-scale features on the curve disappear but the more basic features are preserved. Figure 6.2.4 shows a space curve depicting a bottle-opener. Several evolved versions of the bottle-opener are shown in figure 6.2.5 and its torsion scale space image is shown in figure 6.2.6. Horizontal lines have been drawn across that image to indicate the values of σ which were used to compute the curves of figure 6.2.5.

The third example is the curve shown in figure 6.2.7. This curve depicts an armchair. Figure 6.2.8 shows several evolved versions of the armchair and figure 6.2.9 shows the torsion scale space image of the armchair. Horizontal lines have been drawn across that image to indicate the values of σ which were used to compute the curves of figure 6.2.8. Note that since the fork, the bottle-opener and the armchair are all depicted by symmetric curves, their torsion scale space images are all symmetric as well.

Experiments were also carried out to examine the behaviour of the proposed torsion scale space representations when input curves are corrupted by noise. Figure 6.2.10 shows the armchair with a significant amount of noise added to it. The direction as well as the magnitude of the noise is random. Figure 6.2.11 shows the torsion scale space image of armchair with noise and figure 6.2.12 shows the renormalized torsion scale space image of armchair with noise. It can be observed that despite the addition of a considerable amount of noise, the torsion scale space image retains its basic structure very well. Figure 6.2.13 shows the superposition of the image of figure 6.2.11 and the image of figure 6.2.9. A very close match can be observed between the two images. Figure 6.2.14 shows the armchair corrupted with severe random noise. The torsion scale space image of armchair with severe noise, shown in figure 6.2.15, no longer matches well with the torsion scale space image of the original armchair shown in figure 6.2.9. However, the resampled torsion scale space image of the armchair, shown in figure 6.2.16, and the resampled torsion scale space image of the armchair with severe noise, shown in figure 6.2.17, show a very close match. Figure 6.2.18 shows the superposition of the images shown in figures 6.2.16 and 6.2.17.



Figure 6.2.1. Two views of a space curve depicting a fork



Figure 6.2.2. The fork during evolution



Figure 6.2.3. The torsion scale space image of the fork









(b) $\sigma = 2$









Figure 6.2.6. The torsion scale space image of the bottle-opener



Figure 6.2.7. A space curve depicting an armchair



(a) $\sigma = 10$



(b) $\sigma = 20$



Figure 6.2.8. The armchair during evolution

÷.



Figure 6.2.9. The torsion scale space image of the armchair



Figure 6.2.10. The armchair with random noise



Figure 6.2.11. The torsion scale space image of the armchair with noise



Figure 6.2.12. The Renormalized TSS image of armchair with noise



Figure 6.2.13. The TSS image of armchair with noise superimposed on the TSS image of armchair



Figure 6.2.14. The armchair with severe random noise







Figure 6.2.16. The Resampled torsion scale space image of the armchair






Figure 6.2.18. The Resampled TSS image of armchair with severe noise superimposed on the Resampled TSS image of armchair

For a comparison of regular, renormalized and resampled torsion scale space images and the advantages and disadvantages of each see the discussion of section 6.1 on the regular, renormalized and resampled curvature scale space images and table 6.1.1 on their advantages and disadvantages. The same arguments apply fully to torsion scale space representations.

This concludes the experiments which were carried out on space curves. The following is a discussion of the practical significance of the theorems of chapter 4.

Theorem 4.4.1 showed that evolution and arc length evolution of a space curve are invariant under rotation, uniform scaling and translation of the curve. This shows that the regular, renormalized and resampled torsion scale space images of a space curve have the *invariance* property. The invariance property is essential since it makes it possible to match a space curve to another of similar shape which has undergone a transformation consisting of arbitrary amounts of rotation, uniform scaling and translation.

Theorems 4.4.2 and 4.4.3 showed that closedness and connectedness of a space curve are preserved during evolution and arc length evolution. These theorems demonstrate that evolution and arc length evolution of a space curve do not change the physical interpretation of that curve as the boundary of a three-dimensional object.

Theorem 4.4.4 showed that the center of mass of a space curve does not move as the curve evolves and theorem 4.4.5 showed that a space curve remains inside its convex hull during evolution and arc length evolution. Together, theorems 4.4.4 and 4.4.5 impose constraints on the physical location of a space curve as it evolves. These constraints become useful whenever the physical location of curves in a scene or their locations with respect to each other is important for example when two or more space curves are used to represent a threedimensional object.

Theorem 4.4.6 showed that the torsion scale space images of a space curve determine that curve modulo the class defined by the function $\beta(u) = \kappa^2(u) \tau(u)$ where $\kappa(u)$ and $\tau(u)$ are the curvature and torsion functions of that curve respectively. This shows that the torsion scale space images are often sufficient to distinguish a space curve from other space curves it is being compared to. In such cases, the torsion scale space can be said to satisfy the uniqueness property. This property ensures that curves of different shapes do not have the same representation.

Theorems 4.4.8 and 4.4.9 together locally characterize the behaviour of a space curve just before and just after the formation of a cusp point during evolution and arc length evolution. This behaviour can be used to detect any cusp

points that form during evolution or arc length evolution of a space curve. Such cusp points can then be used effectively to facilitate matching since they provide us with a set of distinctive and easily recognizable features.

Theorem 4.4.7 showed that if a space curve remains smooth during evolution or arc length evolution and torsion remains bounded at every point of that curve, then no new torsion zero-crossings will be observed in its torsion scale space images and theorem 4.4.10 showed that new torsion zero-crossings can indeed occur on a space curve that remains smooth during evolution at points of zero curvature. Together, theorems 4.4.9 and 4.4.10 describe all situations that can lead to creation of new torsion zero-crossings on a space curve during evolution. This enables one to make inferences about a space curve when new torsion zero-crossings are observed in its torsion scale space image.

6.3. Evaluation of the Representations

The following is an evaluation of the curvature and torsion scale space representations according to the criteria proposed in chapter 1.

Criterion: Invariance

Recall that by *invariance*, we meant that the representation for the shape of a curve should not change when shape-preserving transformations (rotation, uniform scaling and translation) are applied to that curve.

Translation of the curve causes no change in the curvature and torsion scale space representations proposed here. Uniform scaling causes the curvature and torsion scale space representations to undergo uniform scaling as well. If the represented curves are closed, then their curvature and torsion scale space representations can be normalized and invariance with respect to uniform scaling will also be achieved. If the represented curves are open, changes due to uniform handled scaling can be by a matching algorithm such as SCALE_SPACE_MATCHER in chapter 5.

Rotation causes only a horizontal shift in the curvature and torsion scale space representations. However, due to the multi-scale nature of those representations, an algorithm such as SCALE_SPACE_MATCHER can determine the shift difference between two matching CSS or TSS representations.

Criterion: Uniqueness

The uniqueness criterion required that two curves with different shapes be mapped to different representations. This is necessary in order to be able to recognize two or more curves with the same shape by observing that their representations are the same. As argued earlier, theorem 3.4.6 shows that a planar curve can be reconstructed from any of its curvature scale space representations and therefore the curvature scale space representations satisfy the uniqueness criterion. Theorem 4.4.6 shows that a space curve can be constructed modulo an equivalence class from any of its torsion scale space representations therefore the torsion scale space representations satisfy the uniqueness criterion with respect to the class of curves defined by theorem 4.4.6. Nevertheless, since torsion scale space representations are very rich representations, it is believed that in practical shape representation and matching tasks, they will be more than enough to distinguish space curves of different shapes from each other.

The only arbitrary choice to be made when computing the curvature and torsion scale space representations is the starting point for parametrization on a closed curve. This only causes a horizontal shift in the curvature and torsion scale space representations but it causes no structural change.

Criterion: Stability

The stability criterion requires that s small change in the shape of a curve lead to a small change in its representation and vice versa. Theorems 3.4.3 and 4.4.3 show that planar and space curves respectively remain connected during evolution and arc length evolution and therefore their curvature and torsion scale space representations can always be constructed. Furthermore, while a planar or space curve evolves, a small change in the standard deviation of the Gaussian filter always results in a small change in the locations of the curvature or torsion zero-crossings on that curve. The experiments of chapter 6 also show that the curvature and torsion scale space representations are stable with respect to significant uniform and non-uniform noise on the curves they represent and therefore satisfy the stability criterion.

Since the representation methods proposed in this thesis involve identification of curvature zero-crossing points on planar curves, it may be conjectured that they are not suitable for application to curves with straight segments on them. However, it should be noted that while the presence of straight line segments on a curve might introduce instabilities at the finest scale levels, after a small number of iterations the originally straight segments will have nonzero curvature and the computations will stabilize. Figure 6.3.1 shows a planar curve made up of straight line segments and figure 6.3.2 shows the curve of figure 6.3.1 with added random noise. Figure 6.3.3 shows the curvature scale space representation of the curve of figure 6.3.1 and figure 6.3.4 shows the curvature scale space representation of the curve of figures 6.3.3 and 6.3.4. It can be seen that while there is disagreement between the two representations at the finest scale levels, a very close match exists at the higher levels of the representations.



Figure 6.3.1. A planar curve made up of straight line segments



Figure 6.3.2. The curve of figure 6.3.1 with added random noise







Figure 6.3.4. The curvature scale space image of the curve of figure 6.3.2



Figure 6.3.5. The CSS image shown in figure 6.3.3 superimposed on the CSS image shown in figure 6.3.4

The error present on a space curve may be non-isotropic since most sensors provide 3D data that is much more accurate in the x and y directions than in the z direction. This type of error can be handled by applying more smoothing to the z-coordinate than to the x- and y-coordinates. The amount of smoothing for each coordinate can be determined by the variance of the error present on each of the x-, y- and z-coordinates.

Criterion: Efficiency

The computation of the representations proposed here calls for evaluation of a large number of convolutions. As discussed in chapter 5, this process can be rendered efficient using one or more of the following techniques:

i. Fast Fourier transforms

ii. Parallelism

- iii. Expression of convolutions involving Gaussians of large widths in terms of convolutions involving Gaussians of small widths only
- iv. Tracking the curvature and torsion zero-crossing points across multiple scales: when it is known that new curvature zero-crossings will not be created at higher scales, convolutions can be carried out only in a small neighborhood of the existing zero-crossings in order to find the zero-crossings at the next higher level.

Furthermore, curvature and torsion scale space representations can be stored very efficiently as encoded binary images. An alternative is to store only a selected subset of points from those scale space representations which will be used for matching. In general, all algorithms proposed are efficient in that their complexities are low order polynomials in the size of the input.

Criterion: Ease of implementation

Specific algorithms for computing the representations discussed in this thesis were given in chapter 5. The procedures needed to compute the curvature and torsion scale space images are not difficult to implement. Convolutions with Gaussian filters are at the heart of the computations. These are standard and well-understood procedures in the computational vision area. It follows that this criterion is also satisfied.

Criterion: Computation of shape properties

The curvature and torsion scale space representations of symmetric shapes are also symmetric since two symmetric shapes have curvature or torsion zerocrossings at the same locations across scales. Therefore the symmetry criterion is satisfied. Furthermore, curvature and torsion scale space computations are carried out using finite Gaussian filters and making use of finite-sized neighborhoods. Therefore a curvature or torsion scale space representation can also be computed for an open curve and, except near the endpoints of the curve, will resemble the corresponding representation for a closed curve of which it is a part. It is therefore believed that the representations also satisfy the part/whole criterion.

Finally note that Volumetric diffusion [Koenderink & van Doorn 1986] and Reactive and diffusive deformations of shape [Kimia *et al.* 1989] are shape representation techniques which are the most similar to ours since they also compute deformations of shapes of two-dimensional and three-dimensional curves. It may therefore be suggested that an alternative way to compute the curvature and torsion scale space representations is to use one of the above techniques to compute deformations of the curves to be represented and then locate curvature and torsion zero-crossing points on each deformed curve and map them to the appropriate representations. However, as noted in chapter 2, the application of each of the techniques mentioned above might result in disconnected curves. In such cases, it will no longer be possible to construct the curvature or torsion scale space representations. Furthermore, our technique combines the curve deformation and the computation of curvature or torsion into one step and is therefore more numerically accurate than the aforementioned techniques which separate the processes of curve deformation and computation of curvature or torsion.

It follows that the curvature and torsion scale space representations satisfy nearly all the criteria for general-purpose shape representation methods proposed in chapter 1.

Chapter 7:

Summary, Conclusions and Future Directions

This chapter contains a summary and the conclusions of this thesis and a discussion of directions for further research.

7.1. Summary

This thesis dealt with the area of shape representation and recognition which is an important area of research within computer vision. The problem of shape representation was considered here for two-dimensional and threedimensional curves. Two-dimensional curves are boundaries of two-dimensional objects and three-dimensional curves can be considered to be abstract representations of some three-dimensional objects as argued in chapter 4.

Chapter 1 proposed a number of criteria considered necessary or useful for any shape representation technique in computer vision. Chapter 2 reviewed several previously proposed shape representation methods for planar curves and discussed ways of generalizing those methods to compute representations of space curves. Each method was also evaluated according to the criteria proposed in chapter 1. Each method fails to satisfy one or more of those criteria.

Chapter 3 proposed a novel theory of multi-scale shape representation for planar curves. A path-based parametric representation of the curve was computed and convolved with a Gaussian function of varying width to obtain descriptions of the curve at multiple levels of detail. The curvature zero-crossing points on each curve were identified and mapped to the u- σ space where u is the parameter along the the curve and σ is the width of the Gaussian function. Chapter 3 proposed three different versions of the representation, each suitable for particular applications and developed a theory for the proposed representations which provides a sound, theoretical foundation for those representations. The proofs of the theorems of chapter 3 are given in appendix A.

Chapter 4 generalized the theory of chapter 3 to propose a theory of multi-scale shape representation for space curves. Multiple descriptions of the curve at varying levels of scale were computed in a similar manner and torsion zero-crossing and curvature level-crossing points on each curve were located and again mapped to $u - \sigma$ spaces. Chapter 4 proposed three different versions of the representation as well and developed a parallel theory for the proposed representations which again provides a sound, theoretical foundation for those representations. The proofs of the theorems of chapter 4 are given in appendix B.

Chapter 5 discussed the implementation issues associated with the task of implementing the shape representation theories proposed in chapters 3 and 4. Procedures to sample the input curves and to compute the Gaussian masks for convolution were described and algorithms for computing the regular, renormalized and resampled curvature and torsion scale space images were given. Complexity analyses for those algorithms were also presented.

Finally, chapter 6 presented a number of examples of planar and space curves used as input to the programs and the curvature and torsion scale space representations of those curves. Experiments were carried out to study the behaviour of the representations when considerable noise was added to the input curves and to illustrate the particular applications for which each representation is useful. The significance of the theoretical results of chapters 3 and 4 were also demonstrated. Chapter 6 ended with an evaluation of the curvature and torsion scale space representations proposed in this thesis according to the criteria proposed in chapter 1. It was shown that the proposed representations satisfy nearly all of the criteria.

7.2. Conclusions

The contribution of this thesis is to present a theory of shape representation for planar and space curves. The representations are referred to as the curvature and torsion scale space images and are general-purpose because they satisfy a number of criteria that are considered useful for general-purpose shape representations. They are nearly invariant with respect to the shape-preserving transformations of the curves since they make use of invariant geometric properties of the curves and experiments indicate that they are stable with respect to even severe uniform or non-uniform random noise on the curves because they combine information about the curves at a continuum of scales. Furthermore, they represent curves uniquely since it is possible in theory to reconstruct those curves using their representations (modulo uniform scaling and a rigid motion in case of planar curves and modulo a larger class in case of space curves). These properties make the representations very useful for shape recognition tasks. A new and efficient scale space matching algorithm was also presented.

The theoretical results of this thesis significantly contribute to a proper understanding of the proposed representations. These results define the properties of both the representations and the underlying processes of evolution and arc length evolution. It follows from theorems 3.4.1 and 4.4.1 that the representations are in fact invariant with respect to shape preserving transformations of the curves they represent. Theorems 3.4.2 and 4.4.2 show that (arc length) evolution preserves closedness of planar and space curves and therefore does not change the physical interpretation of those curves as boundaries of objects. Theorems 3.4.3 and 4.4.3 are very important since they show that connectedness of planar and space curves is preserved by (arc length) evolution. As a result, it is always possible to construct the curvature and torsion scale space representations. Theorems 3.4.4, 3.4.5, 4.4.4 and 4.4.5 impose strong constraints on the location of planar and space curves during (arc length) evolution. These constraints are useful for matching of image curves at multiple scales. Theorems 3.4.6 and 4.4.6 show that the curvature and torsion scale space images in fact uniquely represent the curves they are computed from. This property is crucial for matching and distinguishing shapes from each other. Theorems 3.4.7 and 4.4.7 make explicit the conditions under which new curvature or torsion zero-crossings will not be created in scale space representations. They can be used to speed up the computation of those representations by tracking the zero-crossings across scales. Theorem 3.4.9 shows that arc length evolution has further physical plausibility since it preserves the simplicity of planar curves. Theorems 3.4.8, 3.4.10, 4.4.8 and 4.4.9 together locally characterize the behaviour of planar and space curves just before and after the creation of cusp points during (arc length) evolution. Theorems 3.4.7, 3.4.8 and 3.4.9 also combine to conclude that no new curvature zero-crossings are created during arc length evolution of simple curves. Finally, theorem 4.4.10 shows that new torsion zero-crossings can form on smooth space curves during (arc length) evolution. Together with theorem 4.4.9, they describe all conditions under which new torsion zero-crossings can appear on space curves during (arc length) evolution. This information can also be used to speed up the compution of the torsion scale space representation by tracking of torsion zero-crossings whenever possible.

7.3. Directions for future research

The following are a number of possible areas of research that are related to the curvature and torsion scale space representations for planar and space curves proposed in this thesis.

• In order to complete the theoretical results which form a sound foundation for

the representations, it would be useful to show that the curvature and torsion scale space representations proposed here have desirable convergence properties. Observations show that a planar curve eventually becomes simple and convex after undergoing a process of evolution or arc length evolution and that once such a curve becomes simple and convex, it will always remain simple and convex. This implies that the search for curvature zero-crossing points can stop as soon as the curve becomes simple and convex. Gage and Hamilton [1986] have shown that simple and convex planar curves remain simple and convex during arc length evolution. Other properties remain to be shown formally. Similarly, observations show that a space curve eventually enters a state in which each of the two-dimensional curves obtained by disregarding one of its coordinate functions, is simple and convex and in which it has few torsion zero-crossing points after undergoing a process of evolution or arc length evolution. Observations further show that once such a curve enters that state, it will always remain in that state. Again, this implies that the search for torsion zero-crossing points can stop as soon as that state is reached. These properties also remain to be shown formally.

- For both theoretical and practical reasons, it would be interesting to fully generalize the shape representation theory proposed in this thesis and apply it to three-dimensional surfaces. In such an approach, the surface would be parametrized in an appropriate way and convolved with a two-dimensional Gaussian function to obtain descriptions of the surface at multiple levels of detail. Contours of zero curvature can then be identified on each surface and mapped to a suitable generalized scale space. The result would be a multiscale, geometric-based representation of the shape of that surface. Results about the local, global and convergence properties of the surface during evolution as well as the uniqueness properties of its representation would constitute a sound theoretical foundation for such a representation. A matching algorithm to find good matches of two such representations would also be useful.
- The aim of this thesis has been to solve the curve representation problem but not the segmentation problem. In other words, it is assumed that the curves to be represented have been satisfactorily segmented beforehand. However, the tools developed in this thesis can also be used to tackle the segmentation problem. For example, Lowe [1988] computed curvature on Gaussian smoothed image curves and segmented those curves at points where the rate of change of curvature was high. More work should be done towards the application of the tools developed in this thesis to the segmentation problem.
- Theorem 3.4.6 states that the derivatives at a single point on a curvature zerocrossing contour in the curvature scale space image of a planar curve

112

determine that curve uniquely up to uniform scaling and a rigid motion. The proof of that theorem provides us with a constructive algorithm. It would be interesting to actually attempt such a reconstruction and determine the conditions under which it can be made stable with respect to noise. A similar argument applies to theorem 4.4.6.

• The curve description and recognition methods developed in this thesis should be further applied to practical vision problems. Recognition of characters, shorelines in aerial images [Mokhtarian & Mackworth 1986] and organisms in microscopic images as well as handwriting analysis are examples of problems which can be attacked using the tools developed here. Stereo matching should also be easier by first using an edge detector to extract image edges at a fine level of detail and then using our method to describe the candidate curves at coarse levels of scale where unnecessary detail has been removed and matching is more efficient and more reliable. This method would have clear advantages over blurring the image since image-blurring followed by edge detection totally removes many of the candidate curves and results in the joining of curves from unrelated parts of the image which complicates the matching problem.

It would also be advantageous to develop a new matching algorithm to match curvature and torsion scale space images since the matching algorithm in [Mokhtarian & Mackworth 1986] makes assumptions about curvature scale space representations which are not always true as pointed out in chapter 5. The matching algorithm in [Witkin *et al.* 1987] provides an example of a multi-scale signal matching algorithm. It may be possible to generalize some of the concepts employed in that algorithm to the matching of curves. The matching algorithm in [Kishon & Wolfson 1987] provides an example of a curve matching algorithm but it can have problems when scale changes exist between an observed curve and a model curve and when noise exists on the observed curve. Algorithm SCALE_SPACE_MATCHER proposed in chapter 5 should run considerably faster than the algorithm in Mokhtarian and Mackworth [1986]. It also does not make possibly incorrect assumptions about the structure of a scale space representation.

In summary, this thesis proposes multi-scale, geometric-based, shape representations for planar and space curves. A number of important theoretical issues about those representations have been investigated and as a result, their properties are well-understood. A number of experiments have also been carried out on them in order to illustrate their practical utility under various noise conditions. It has also been shown that the proposed representations satisfy a number of criteria considered useful for shape recognition tasks. Finally, several directions for future research have been outlined.



References

- Agin, G. J. and T. O. Binford, "Computer description of curved objects," Proc. IJCAI, pp. 629-640, Palo Alto, CA, 1973.
- Aho, A. V., J. E. Hopcroft, and J. D. Ullman, The design and analysis of computer algorithms, Addison-Wesley, Reading, MA, 1974.
- Asada, H. and M. Brady, "The curvature primal sketch," *IEEE PAMI*, vol. 8, pp. 2-14, 1986.
- Babaud, J., A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernel for scale space filtering," IEEE PAMI, vol. 8, pp. 26-33, 1986.
- Ballard, D. H., "Generalizing the Hough transform to detect arbitrary shapes," Pattern Recognition, vol. 13, pp. 111-122, 1981.
- Ballard, D. H. and C. M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 1982.
- Barnard, S. T. and A. P. Pentland, "Three-dimensional shape from line drawings," Proc. IJCAI, pp. 1062-1064, Vancouver, B.C., 1983.
- Barnhill, R. E. and R. F. Riesenfeld, Computer Aided Geometric Design, Academic Press, New York, 1974.
- Bentley, J. L., "Multidimensional search trees used for associative searching," Comm. ACM, vol. 18, no. 9, pp. 509-517, 1975.
- Blum, H., "Biological shape and visual science (part I)," J. Theoretical Biology, vol. 38, pp. 205-287, 1973.
- Brady, M., J. Ponce, A. L. Yuille, and H. Asada, "Describing surfaces," AI memo 822, MIT AI Lab, Cambridge, MA, 1985.
- Brown, C. M., "Two descriptions and a two-sample test for 3-d vector data," TR 49, Computer Science Dept., Univ. Rochester, 1979.
- Choi, D. J. and J. R. Kender, "Solving the depth interpolation problem on a parallel architecture with a multigrid approach," Proc. CVPR, pp. 189-194,

Ann Arbor, Michigan, 1988.

Danielsson, P. E., "A new shape factor," CGIP, vol. 7, pp. 292-299, 1978.

- Davis, L. S., "Understanding shape: Angles and sides," IEEE Trans. Computers, vol. C-26, pp. 236-242, 1977.
- DeBoor, C., A practical guide to splines, Springer-Verlag, New York, 1978.
- DoCarmo, M., Differential geometry of curves and surfaces, Prentice-Hall, Englewood Cliffs, NJ, 1976.
- Duda, R. O. and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," Comm. ACM, vol. 15, pp. 11-15, 1972.
- Faugeras, O. D., et al., "Object representation, identification and positioning from range data," in *First International Symposium on Robotics Research*, ed. R. Paul, pp. 425-446, MIT press, 1984.
- Faugeras, O. D. and J. Ponce, "Prism trees: a hierarchical representation for 3-D objects," Proc. IJCAI, pp. 982-988, Vancouver, B.C., 1983.
- Freeman, H., "Computer processing of line-drawing images," Comput. Surveys, vol. 6, 1974.
- Gage, M. and R. S. Hamilton, "The heat equation shrinking convex plane curves," J. Differential Geometry, vol. 23, pp. 69-96, 1986.
- Gallus, G. and P. W. Neurath, "Improved computer chromosome analysis incorporating preprocessing and boundary analysis," *Physics in Medicine and Biology*, vol. 15, p. 435, 1970.
- Goetz, A., Introduction to differential geometry, Addison-Wesley, Reading, MA, 1970.
- Grimson, W. E. L., "Computational experiments with feature based stereo algorithm," IEEE PAMI, vol. 7, pp. 24-27, 1985.
- Haralick, R. M., A. K. Mackworth, and S. L. Tanimoto, "Computer vision update," *Handbook of AI*, vol. IV, Ed. A. Barr, Addison-Wesley, N. Y., 1989.
- Hong, J. and X. Tan, "Recognize the similarity between shapes under affine transformation," Proc. ICCV, pp. 489-493, Tarpon Springs, Florida, 1988.
- Horn, B. K. P. and E. J. Weldon, "Filtering closed curves," *IEEE PAMI*, vol. 8, pp. 665-668, 1986.
- Horowitz, S. L. and T. P. Pavlidis, "Picture segmentation by a tree traversal algorithm," J. ACM, vol. 23, pp. 368-388, 1976.

- Hough, P. V. C., "Method and means for recognizing complex patterns," U.S. patent 3 069 654, 1962.
- Hummel, R. A., B. Kimia, and S. W. Zucker, "Deblurring Gaussian blur," CVGIP, vol. 38, pp. 66-80, 1987.
- Ikeuchi, K. and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," Artificial Intelligence, vol. 17, pp. 141-184, 1981.
- Jackins, C. L. and S. L. Tanimoto, "Oct-trees and their use in representing three-dimensional objects," CGIP, vol. 14, no. 3, pp. 249-270, 1980.
- Kecs, W., The convolution product and some applications, D. Reidel, Boston, MA, 1982.
- Kimia, B. B., A. Tannenbaum, and S. W. Zucker, "Toward a computational theory of shape: an overview," TR-CIM-89-13, McGill University, 1989.
- Kishon, E. and H. Wolfson, "3-D curve matching," TR 283, Courant Inst. Math. Sciences, New York University, 1987.
- Koenderink, J. J. and A. J. van Doorn, "Dynamic shape," Biological Cybernetics, vol. 53, pp. 383-396, 1986.
- Lowe, D., "Organization of smooth image curves at multiple scales," Proc. ICCV, pp. 558-567, Tampa, Florida, 1988.
- Mackworth, A. K. and F. Mokhtarian, "The renormalized curvature scale space and the evolution properties of planar curves," Proc. IEEE CVPR, pp. 318-326, Ann Arbor, Michigan, 1988.
- Marr, D., "Early processing of visual information," Proc. R. Soc. Lond. B, vol. 275, pp. 483-519, 1976.
- Marr, D., "Representing visual information," AI memo 415, MIT AI Lab, 1977.
- Marr, D. and E. C. Hildreth, "Theory of edge detection," Proc. R. Soc. Lond. B, vol. 207, pp. 187-217, 1980.
- Marr, D. and H. K. Hishihara, "Representation and recognition of the spatial organization of 3D structures," Proc. R. Soc. Lond. B, vol. 200, pp. 269-294, 1978.
- McKee, J. W. and J. K. Aggarwal, "Computer recognition of partial views of curved objects," *IEEE Trans. Computers*, vol. C-26, pp. 790-800, 1977.
- Mokhtarian, F. and A. K. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE PAMI*, vol. 8, pp. 34-43, 1986.

- Mokhtarian, F., "Multi-scale description of space curves and three-dimensional objects," Proc. IEEE CVPR, pp. 298-303, Ann Arbor, Michigan, 1988a.
- Mokhtarian, F., "Evolution properties of space curves," Proc. ICCV, pp. 100-105, Tarpon Springs, Florida, 1988b.
- Mokhtarian, F., "Fingerprint theorems for curvature and torsion zero-crossings," Proc. IEEE CVPR, pp. 269-275, San Diego, California, 1989.
- Nahin, P. J., "The theory and measurement of a silhouette descriptor for image preprocessing and recognition," *Pattern Recognition*, vol. 6, no. 2, 1974.
- Nishihara, H. K., "Intensity, visible-surface, and volumetric representations," Artificial Intelligence, vol. 17, pp. 265-284, 1981.
- O'Rourke, J. and N. I. Badler, "Decomposition of three-dimensional objects into spheres," *IEEE PAMI*, vol. 1, 1979.
- O'Rourke, J. and R. Washington, "Curve similarity via signatures," in *Computational geometry*, G. Toussaint (Ed.), North-Holland, Amsterdam, 1985.
- Pavlidis, T., "Segmentation of pictures and maps through functional approximations," CGIP, vol. 1, pp. 360-372, 1972.
- Pavlidis, T., "Waveform segmentation through functional approximation," IEEE Trans. Comput., vol. C-22, pp. 689-697, 1973.
- Pavlidis, T., "Optimal piecewise polynomial L₂ approximation of functions of one and two variables," *IEEE Trans. Comput.*, vol. C-24, pp. 98-102, 1975.
- Pavlidis, T., "Polygonal approximations by Newton's method," IEEE Trans. Computers, vol. C-26, pp. 800-807, 1977.
- Pavlidis, T., Structural Pattern Recognition, Springer-Verlag, New York, 1977.
- Pavlidis, T. and S. L. Horowitz, "Segmentation of plane curves," IEEE Trans. Comput., vol. C-23, pp. 860-870, 1974.
- Persoon, E. and K. S. Fu, "Shape discrimination using Fourier descriptors," Proc. IJCPR, pp. 126-130, Copenhagen, Denmark, 1974.
- Requicha, A. A. G., "Representations of rigid solid objects," Computer Surveys, vol. 12, no. 4, 1980.
- Richard, C. W. and H. Hemami, "Identification of 3D objects using Fourier descriptors of the boundary curve," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-4, no. 4, pp. 371-378, 1974.
- Roberts, L. G., "Machine perception of three-dimensional solids," In Optical and electro-optical information processing, Eds. J. P. Tippett et al., MIT press,

Cambridge, MA, 1965.

- Samet, H., "Region representation: quadtrees from boundary codes," Comm. ACM, vol. 23, pp. 163-170, 1980.
- Saund, E., "Adding scale to the primal sketch," Proc. IEEE CVPR, pp. 70-78, San Diego, California, 1989.
- Schneier, M., "Linear time calculations of geometric properties using quadtrees," TR 770, Computer Science Center, Univ. Maryland, 1979.
- Schwartz, E. L., R. Desimone, T. Albright and C. G. Gross, "Shape recognition and inferior temporal neurons," Proc. of the National Academy of Sciences, 1983.
- Shahraray, B. and D. J. Anderson, "Optimal estimation of contour properties by cross-validated regularization," *IEEE PAMI*, vol. 11, no. 6, pp. 600-610, 1989.
- Shirai, Y., "Analyzing intensity arrays using knowledge about scenes," In PCV, 1975.
- Sklansky, J. and D. P. Kibler, "A theory of non-uniformly digitizing binary pictures," IEEE Trans. SMC, vol. 6, no. 9, pp. 637-647, 1976.
- Stansfield, J. L., "Conclusions from the commodity expert project," AI memo 601, MIT AI Lab, Cambridge, MA, 1980.
- Stevens, K. A., "Implementation of a theory for inferring surface shape from contours," AI memo 676, MIT AI Lab, Cambridge, MA, 1982.
- Szeliski, R., "Fast surface interpolation using hierarchical basis functions," Proc. CVPR, pp. 222-228, San Diego, CA, 1989.
- Terzopoulos, D., Multiresolution computation of visible-surface representations, PhD thesis, MIT, Cambridge, MA, 1984.
- Tomek, I., "Two algorithms for piecewise linear continuous approximation of functions of one variable," *IEEE Trans. Computers*, vol. 23, no. 4, pp. 445-448, 1974.
- Turner, K. J., "Computer perception of curves objects using a television camera," Ph.D. dissertation, Univ. Edinburgh, 1974.
- Voelcker, H. B. and A. A. G. Requicha, "Geometric modeling of mechanical parts and processes," *Computer*, vol. 10, pp. 48-57, 1977.
- Wahba, G., "A survey of some smoothing problems and the method of generalized cross-validation for solving them," in Applications of Statistics, P. R. Krishnaiah, Ed., North-Holland, Amsterdam, The Netherlands, pp. 507-523,

1977.

- Watson, L. T. and L. G. Shapiro, "Identification of space curves from twodimensional perspective views," *IEEE PAMI*, vol. 4, pp. 469-475, 1982.
- Weiss, I., "3-D shape representation by contours," Proc. IJCAI, pp. 969-972, Los Angeles, CA, 1985.
- Witkin, A. P., "Recovering surface shape and orientation from texture," Artificial Intelligence, vol. 17, pp. 17-45, 1981.
- Witkin, A. P., "Scale space filtering," Proc. IJCAI, pp. 1019-1023, Karlsruhe, W. Germany, 1983.
- Witkin, A., D. Terzopoulos, and M. Kass, "Signal matching through scale space," IJCV, vol. 1, pp. 133-144, 1987.
- Woodham, R. J., "Photometric method for determining shape from shading," in Image Understanding 1984, ed. W. Richards, pp. 97-125, Ablex publishing corp., Norwood, NJ, 1984.
- Yuille, A. L. and T. A. Poggio, "Fingerprint theorems for zero crossings," AI Memo 730, MIT AI Lab., Cambridge, MA, 1983.
- Yuille, A. L. and T. A. Poggio, "Fingerprint theorems," Proc. AAAI, pp. 362-365, Austin, Texas, 1984.
- Yuille, A. L. and T. A. Poggio, "Scaling theorems for zero-crossings," IEEE PAMI, vol. 8, pp. 15-25, 1986.

Appendix A:

Proofs of Results on Planar Curves

This appendix contains the proofs of the theorems of chapter 3. The proofs of the theorems 3.4.1 - 3.4.5 have been given for arc length evolution only since the proofs for regular evolution are similar and simpler. Theorems 3.4.6 and 3.4.7 have been shown to hold for regular, renormalized and resampled curvature scale space images. Furthermore, theorems 3.4.8 and 3.4.10 have been shown to hold for both regular and arc length evolution.

2

Proof of theorem 3.4.1: It will be shown that arc length evolution is invariant under a general affine transform. Let $\Gamma_{\sigma} = (X(W,\sigma), Y(W,\sigma))$ be an arc length evolved version of $\Gamma = (x(w), y(w))$. If Γ_{σ} is transformed according to an affine transform, then at its new coordinates, X_1 and Y_1 , are given by

 $X_1(W,\sigma) = a X(W,\sigma) + b Y(W,\sigma) + c$ $Y_1(W,\sigma) = d X(W,\sigma) + e Y(W,\sigma) + f.$

Now suppose Γ is transformed according to an affine transform and then evolved. The coordinates X_2 and Y_2 of the new curve are

$$\begin{aligned} X_2(W,\sigma) &= (a x(W) + b y(W) + c) \circledast g(W,\sigma) \\ Y_2(W,\sigma) &= (d x(W) + e y(W) + f) \circledast g(W,\sigma). \end{aligned}$$

Since the convolution operator is distributive [Kecs 1982], it follows that

 $X_{2}(W,\sigma) = X_{1}(W,\sigma)$ $Y_{2}(W,\sigma) = Y_{1}(W,\sigma)$

and the theorem follows.

Proof of theorem 3.4.2: Let $\Gamma = (x(w), y(w))$ be a closed curve and let $\Gamma_{\sigma} = (X(W, \sigma), Y(W, \sigma))$ be an arc length evolved version of Γ . On Γ :

$$(x(0), y(0)) = (x(1), y(1)).$$

On Γ_{σ} :

$$(X(0,\sigma), Y(0,\sigma)) = (X(1,\sigma), Y(1,\sigma)).$$

It follows that Γ_{σ} is closed.

Proof of theorem 3.4.3: Let $\Gamma = (x(w), y(w))$ be a connected planar curve and $\Gamma_{\sigma} = (X(W, \sigma), Y(W, \sigma))$ be an arc length evolved version of that curve. Since Γ is connected, x(w) and y(w) are continuous functions and therefore $X(W, \sigma)$ and $Y(W, \sigma)$ are also continuous. Let W_0 be any value of parameter W and let x_0 and y_0 be the values of $X(W, \sigma)$ and $Y(W, \sigma)$ at W_0 respectively. If W goes through an infinitesimal change, then $X(W, \sigma)$ and $Y(W, \sigma)$ will also go through infinitesimal changes:

$$X(W_0,\sigma) \to x_0 + \delta$$
$$Y(W_0,\sigma) \to y_0 + \xi.$$

As a result, point $P(x_0,y_0)$ on Γ_{σ} goes to point $P'(x_0+\delta,y_0+\xi)$. Let the distance between P and P' be D. Then

$$D = \sqrt{\delta^2 + \xi^2} \le \delta \sqrt{2}$$

assuming $|\delta|$ is the larger of $|\delta|$ and $|\xi|$. It follows that an infinitesimal change in parameter W also results in an infinitesimal change in the value of the vector-valued function Γ_{σ} . Therefore Γ_{σ} is a connected curve.

Proof of theorem 3.4.4: Let *M* be the center of mass of $\Gamma = (x(w), y(w))$ with coordinates (x_M, y_M) . Then

$$x_M = \frac{\int\limits_0^1 x(w) dw}{\int\limits_0^1 dw} = \int\limits_0^1 x(w) dw$$

$$y_M = \frac{\int\limits_0^1 y(w) dw}{\int\limits_0^1 dw} = \int\limits_0^1 y(w) dw.$$

Let $\Gamma_{\sigma} = (X(W,\sigma), Y(W,\sigma))$ be an arc length evolved version of Γ with $N = (X_N, Y_N)$ as its center of mass. Observe that

$$X_{N} = \int_{0}^{1} X(W,\sigma) dW = \int_{0}^{1} \int_{-\infty}^{\infty} g(v,\sigma) x(W-v) dv dW = \int_{-\infty}^{\infty} g(v,\sigma) (\int_{0}^{1} x(W-v) dW) dv$$

W covers Γ_{σ} exactly once. Therefore

$$\int_{0}^{1} x(W-v) \, dW = x_{M.}$$

So

 $X_N = x_M$

Similarly

$$Y_N = y_M$$

It follows that M and N are the same point.

Proof of theorem 3.4.5: Since G is simple and convex, every line L tangent to G contains that curve in the left (or right) half-plane it creates. Since Γ is inside G, Γ is also contained in the same half-plane. Now rotate L and Γ so that L becomes parallel to the y axis. L is now described by the equation x = c. Since L does not intersect Γ , it follows that $x(w_0) \ge c$ for every point w_0 on Γ . Let Γ_{σ} be an arc length evolved version of Γ . Every point of Γ_{σ} is a weighted average of all the points of Γ . Therefore $X(W_0, \sigma) \ge c$ for every point W_0 on Γ_{σ} and Γ_{σ} is also contained in the same half-plane. This result holds for every line tangent to G therefore Γ_{σ} is contained inside the intersection of all the left (or right) half-planes created by the tangent lines of G. It follows that Γ_{σ} is also inside G.

Proof of theorem 3.4.6: The proof will first be given for the regular curvature scale space of Γ , then the modifications needed to apply the same proof to the resampled and renormalized curvature scale space images will be explained. The proof of this theorem for the regular curvature scale space image only was first given in [Mokhtarian 1989].

Section i shows that the derivatives at a point on a curvature zerocrossing contour provide homogeneous equations in the moments of the Fourier transforms of the coordinate functions of the curve. Section ii shows that the

moments are related to the coefficients of expansion of the coordinate functions of the curve in functions related to the Hermite polynomials. Section **iii** shows that the moments at one curvature zero-crossing point can be related to the moments at other curvature zero-crossing points. Section **iv** shows that the quadratic equations obtained in section **i** can be converted into a homogeneous linear system of equations which can be solved uniquely for the curvature function of the curve.

i. Constraints from the curvature zero-crossing contours

Let $\Gamma = (x(u), y(u))$ be the arc-length parametrization of the curve with Fourier transform $\tilde{\Gamma} = (\tilde{x}(\omega), \tilde{y}(\omega))$. The Fourier transform of the Gaussian filter $G(u, t) = \frac{1}{\sqrt{2t}}e^{-u^2/4t}$ is $\tilde{G}(\omega) = e^{-\omega^2 t}$.

Let $\Gamma_{t_0} = (x(u, t_0), y(u, t_0))$ be a curve obtained by convolving x(u) and y(u) with $G(u, t_0)$. Assume that Γ_{t_0} is in C_{∞} . Such a t_0 exists since Γ is in C_1 . The curvature zero-crossings in a neighborhood of t_0 are given by solutions of $\alpha(u, t) = 0$ where

$$\alpha(u,t) = \dot{x}(u,t)\ddot{y}(u,t) - \dot{y}(u,t)\ddot{x}(u,t)$$

where . represents derivative with respect to u. Using the convolution theorem, the terms in $\alpha(u, t)$ can be expressed as following:

$$\begin{split} \dot{x}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{x}(\omega) d\omega \\ \dot{y}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (-\omega^2) \tilde{y}(\omega) d\omega \\ \dot{y}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{y}(\omega) d\omega \\ \dot{x}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (-\omega^2) \tilde{x}(\omega) d\omega. \end{split}$$

The Implicit Function Theorem guarantees that the contours u(t) are C_{∞} in a neighborhood of t_0 . Let ξ be a parameter of the curvature zero-crossing contour. Then

$$\frac{d}{d\xi} = \frac{du}{d\xi}\frac{\partial}{\partial u} + \frac{dt}{d\xi}\frac{\partial}{\partial t}.$$

On the curvature zero-crossing contour, $\alpha = 0$ and $\frac{d^k}{d\xi^k}\alpha = 0$ for all integers k. Furthermore, since the curvature zero-crossing contour is known, all the deriva-

tives of u and t with respect to ξ are known as well.

We can now compute the derivatives of α with respect to ξ at (u_0, t_0) . The first derivative is given by:

$$\begin{split} \frac{d}{d\xi} \alpha(u_0, t_0) &= \frac{du}{d\xi} \Big(\int e^{-\omega^2 t} e^{i\omega u} (i\omega)^3 \tilde{y}(\omega) d\omega \int e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{x}(\omega) d\omega & (A.1) \\ &- \int e^{-\omega^2 t} e^{i\omega u} (i\omega)^3 \tilde{x}(\omega) d\omega \int e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{y}(\omega) d\omega \Big) \\ &+ \frac{dt}{d\xi} \Big(\int e^{-\omega^2 t} e^{i\omega u} (i\omega)^3 \tilde{x}(\omega) d\omega \int e^{-\omega^2 t} e^{i\omega u} (-\omega^2) \tilde{y}(\omega) d\omega \\ &+ \int e^{-\omega^2 t} e^{i\omega u} (\omega^4) \tilde{y}(\omega) d\omega \int e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{x}(\omega) d\omega \\ &- \int e^{-\omega^2 t} e^{i\omega u} (i\omega)^3 \tilde{y}(\omega) d\omega \int e^{-\omega^2 t} e^{i\omega u} (-\omega^2) \tilde{x}(\omega) d\omega \\ &- \int e^{-\omega^2 t} e^{i\omega u} (i\omega)^3 \tilde{y}(\omega) d\omega \int e^{-\omega^2 t} e^{i\omega u} (-\omega^2) \tilde{x}(\omega) d\omega \end{split}$$

Note that the moment of order k of function $f(\omega) = e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{x}(\omega)$ is defined by:

$$M_{k} = \int_{-\infty}^{\infty} (i\omega)^{k} e^{-\omega^{2}t} e^{i\omega u} (i\omega) \tilde{x}(\omega) d\omega$$

and the moment of order k of function $f'(\omega) = e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{y}(\omega)$ is defined by:

$$M'_{k} = \int_{-\infty}^{\infty} (i\omega)^{k} e^{-\omega^{2}t} e^{i\omega u} (i\omega) \tilde{y}(\omega) d\omega.$$

As a result, equation (A.1) can be re-written as:

$$\frac{d}{d\xi}\alpha(u_0, t_0) = \frac{du}{d\xi}(M'_2M_0 - M_2M'_0)$$

$$+ \frac{dt}{d\xi}(M_2M'_1 + M'_3M_0 - M'_2M_1 - M_3M'_0).$$
(A.2)

Furthermore, the second derivative is given by:

$$\frac{d^2}{d\xi^2}\alpha(u_0, t_0) = \frac{d^2u}{d\xi^2}(M_2'M_0 - M_2M_0')$$
(A.3)

$$+ \frac{d^{2}t}{d\xi^{2}}(M_{2}M_{1}' + M_{3}'M_{0} - M_{2}'M_{1} - M_{3}M_{0}')$$

$$+ \left(\frac{du}{d\xi}\right)^{2}(M_{3}'M_{0} + M_{1}M_{2}' - M_{3}M_{0}' - M_{1}'M_{2})$$

$$+ 2\frac{du}{d\xi}\frac{dt}{d\xi}(M_{4}'M_{0} - M_{4}M_{0}')$$

$$+ \left(\frac{dt}{d\xi}\right)^{2}(M_{4}M_{1}' + 2M_{3}'M_{2} + M_{5}'M_{0} - M_{4}'M_{1} - 2M_{3}M_{2}' - M_{5}M_{0}')$$

Since the parametric derivatives along the curvature zero-crossing contours are zero, equations (A.2) and (A.3) are equal to zero. Note that equation (A.2) is a quadratic equation in the first four moments of functions $f(\omega)$ and $f'(\omega)$ and equation (A.3) is a quadratic equation in the first six moments of those functions. In general, the k+1st equation, $\frac{d^k}{d\xi^k}\alpha(u,t) = 0$ is a quadratic equation in the first 2k+2 moments of each of the functions $f(\omega)$ and $f'(\omega)$ or a total of 4k+4 moments. Our axes are chosen such that $u_0 = 0$. The next section shows that the moments of $f(\omega)$ and $f'(\omega)$ are the coefficients a_k and b_k in the expression of functions $\dot{x}(u)$ and $\dot{y}(u)$ in functions related to the Hermite polynomials. Therefore, having computed the first 4n+4 coefficients a_k and b_k . To determine the a_k and b_k , we need 3n+3 additional and independent equations which can be provided by considering three neighboring curvature zero-crossing contours at $(u_1, t_0), (u_2, t_0),$ and (u_3, t_0) .

ii. The moments and the coefficients of expansion of $\dot{x}(u)$ and $\dot{y}(u)$

This section shows that the moments and the moment-pairs in equations $\frac{d^k}{d\xi^k}\alpha(u,t)$ are related respectively to the coefficients of the expression of the functions $\dot{x}(u)$, $\dot{y}(u)$ and the curvature function of Γ , $\kappa(u)$, in functions related to the Hermite polynomials. Expand function

$$\dot{x}(u) = \frac{d}{du}x(u)$$

in terms of the functions $\phi_k(u,\sigma)$ related to the Hermite polynomials $H_k(u)$ by

$$\phi_k(u,\sigma) = (-1)^k \frac{\sigma^{k-1}}{(\sqrt{2})^{k+1}\sqrt{\pi}} H_k(\frac{u}{\sigma\sqrt{2}})$$

$$\begin{split} H_k(u) &= (-1)^k e^{u^2} \frac{d^k}{du^k} e^{-u^2} \\ \dot{x}(u) &= \sum a_k(\sigma) \phi_k(u,\sigma). \end{split}$$

The coefficients $a_k(\sigma)$ of the expansion are given by

$$a_k(\sigma) = \langle w_k(u,\sigma), \dot{x}(u) \rangle$$

where <,> denotes inner product in L^2 and $\{w_k(u,\sigma)\}$ is the set of functions biorthogonal to $\{\phi_k(u,\sigma)\}$. The $\{\phi_k(u,\sigma)\}$ are given explicitly by

$$\phi_k(u,\sigma) = \frac{\sigma^{2k-1}}{k!\sqrt{2\pi}} e^{\frac{u^2}{2\sigma^2}} \frac{d^k}{du^k} e^{-\frac{u^2}{2\sigma^2}}$$

and the $w_k(u,\sigma)$ by

$$w_k(u,\sigma) = (-1)^k \frac{d^k}{du^k} e^{-\frac{u^2}{2\sigma^2}}.$$

Since

$$\dot{x}(u) = \frac{1}{\sqrt{2\pi}} \int e^{i\omega u} (i\omega) \tilde{x}(\omega) d\omega$$

the a_k are given by

$$a_k(\sigma) = \frac{1}{\sqrt{2\pi}} (-1)^k \int \leq \frac{d^k}{du^k} e^{-\frac{u^2}{2\sigma^2}}, e^{i\omega u} \geq (i\omega) \tilde{x}(\omega) d\omega.$$

The inner product is just the inverse Fourier transform of $w_k(u, \sigma)$. Therefore

$$a_k(\sigma) = \int (i\omega)^k e^{\frac{-\omega^2 \sigma^2}{2}} (i\omega) \,\tilde{x}(\omega) \,d\omega$$

which is equal to M_k modulus a factor $e^{i\omega u}$, since $t = \frac{\sigma^2}{2}$.

Similarly, the function

$$\dot{y}(u) = \frac{d}{du}y(u)$$

can be expanded in terms of the functions $\phi_k(u,\sigma)$ by

$$\dot{y}(u) = \sum b_k(\sigma) \phi_k(u,\sigma)$$

and it again follows that

$$b_k(\sigma) = \int (i\omega)^k e^{\frac{-\omega^2 \sigma^2}{2}} (i\omega) \, \tilde{y}(\omega) \, d\omega$$

which is equal to M'_k modulus a factor $e^{i\omega u}$.

Furthermore, $a'_k(\sigma)$ and $b'_k(\sigma)$, the coefficients of expansion of functions $\ddot{x}(u)$ and $\ddot{y}(u)$ in terms of the functions $\phi_k(u,\sigma)$, can be seen to be related to $a_k(\sigma)$ and $b_k(\sigma)$ according to the following relationships:

$$a'_{k-1} = a_k(\sigma)$$
$$b'_{k-1} = b_k(\sigma).$$

Therefore $\kappa(u)$, the curvature function of Γ can be expressed as:

$$\begin{split} \kappa(u) &= \dot{x}(u) \ddot{y}(u) - \dot{y}(u) \ddot{x}(u) \\ &= \sum a_k(\sigma) \phi_k(u, \sigma) \sum b'_k(\sigma) \phi_k(u, \sigma) - \sum b_k(\sigma) \phi_k(u, \sigma) \sum a'_k(\sigma) \phi_k(u, \sigma) \\ &= \sum \sum a_j(\sigma) b'_k(\sigma) \phi_j(u, \sigma) \phi_k(u, \sigma) - \sum \sum b_j(\sigma) a'_k(\sigma) \phi_j(u, \sigma) \phi_k(u, \sigma) \\ &= \sum \sum (a_j(\sigma) b_{k+1}(\sigma) - b_j(\sigma) a_{k+1}(\sigma)) \phi_j(u, \sigma) \phi_k(u, \sigma). \end{split}$$

It follows that if the pairs $a_j(\sigma)b_k(\sigma)$, $j,k=0, \cdots, 2n+1$, are all known, the curvature function of Γ can be reconstructed.

iii. Combining information from more than one contours

To solve the system of equations obtained in section **i**, we need to obtain additional equations from other points of the curvature scale space image and relate them to the equations obtained from the first point. Suppose additional equations are obtained in the moments of functions $e^{-\omega^2 t} e^{i\omega u'}(i\omega)\tilde{x}(\omega)$ and $e^{-\omega^2 t} e^{i\omega u'}(i\omega)\tilde{y}(\omega)$ at point (u', t_0) . We have

$$\dot{x}(u+u') = \int e^{i\omega u} e^{i\omega u'}(i\omega) \tilde{x}(\omega) d\omega = \sum c_k \phi_k(u)$$

and

$$\dot{y}(u+u') = \int e^{i\omega u} e^{i\omega u'}(i\omega) \tilde{y}(\omega) d\omega = \sum d_k \phi_k(u).$$

Now observe that

$$\sum c_k \phi_k(u) = \sum a_k \phi_k(u+u')$$

and

$$\sum d_k \phi_k(u) = \sum b_k \phi_k(u+u').$$

That is, $\phi_k(u+u')$ can be expressed as a linear combination of $\phi_j(u)$ with $j \le k$ as has been shown in [Yuille and Poggio 1983].

iv. Reconstructing the curvature function

It was shown in section I that four points from four curvature scale space contours give us 4n+4 equations in the first 2n+2 moments of each of the functions $f(\omega)$ and $f'(\omega)$. The first n+1 equations form a system of homogeneous qua- $M_0(u), \cdots, M_{2n+1}(u)$ equations in the unknowns: dratic and $M'_0(u), \dots, M'_{2n+1}(u)$. The other points, $u+u_k$, $1 \le k \le 3$, provide additional $M_0(u+u_k), \cdots, M_{2n+1}(u+u_k)$ equations in the unknowns: and $M'_0(u+u_k), \cdots, M'_{2n+1}(u+u_k)$. As shown in section iii, the moments at $u+u_k$ can be expressed as a linear combination of the moments at u. Therefore it is possible to express all the equations in terms of the moments at u. The result is a system of 4n+4 homogeneous quadratic equations in 4n+4 unknowns. That system has at least one solution since the moments of order higher than 2n+1 of $f(\omega)$ and $f'(\omega)$ are assumed to be zero. However, the solution obtained from a quadratic system of equations is in general not unique.

Equations (A.2) and (A.3) can be converted into homogeneous linear equations by assuming that each moment-pair appearing in those equations is a new variable. Table (A.1) shows the moment-pairs in equations (A.2) and (A.3). The + signs designate the moment-pairs in equation (A.2) and the + and x signs together designate the moment-pairs in equation (A.3).

	M_0'	M_1'	M_2'	M'_3	M'_4	M_5'
M_0			+	+	x	x
M_1			+		x	
M_2	+	+		x		
M_3	+		x			
M_4	x	x				
M_5	,x					

Table A.1. Moment-pairs in equations (A.2) and (A.3)

Note that all other moment-pairs in table (A.1) can be computed from the existing ones using the following relationships:

$$M_{i}M_{j}' = \frac{M_{i-1}M_{j}' \cdot M_{i}M_{j+1}'}{M_{i-1}M_{j+1}'} = \frac{M_{i+1}M_{j}' \cdot M_{i}M_{j+1}'}{M_{i+1}M_{j+1}'} = \frac{M_{i}M_{j-1}' \cdot M_{i-1}M_{j}'}{M_{i-1}M_{j-1}'} = \frac{M_{i}M_{j-1}' \cdot M_{i+1}M_{j}'}{M_{i+1}M_{j-1}'}.$$

As before, we proceed to compute the first n derivatives at point (u_0, t_0) on one of the curvature zero-crossing contours. We now obtain n+1 homogeneous linear equations in some of the moment-pairs $M_iM'_j$ by assuming that each moment-pair is a new variable. Since this system is in terms of the first 2n+2 moments of functions $f(\omega)$ and $f'(\omega)$, it will contain $O(n^2)$ moment-pairs. Therefore additional equations are required to constrain the system. To obtain those equations, we proceed as follows:

Assume that moments of order higher than 2n+2 are zero. Compute derivatives of order higher than n at (u_0, t_0) but set moments of order higher than 2n+2 to zero in the resulting equations. If a sufficient number of derivatives are computed at (u_0, t_0) , the number of equations obtained will be equal to the number of moment-pairs and our linear system will be constrained.

It follows from an assumption of generality that the system will have a unique zero eigenvector and therefore a unique solution modulus scaling. Once the moment-pairs in the system are known, all other moment-pairs can be computed from the known ones using the relationships given above. Since all the moment-pairs $M_iM'_j$ together determine the curvature function of the curve, it follows that the curve can be determined modulus a rigid motion and constant scaling.

Yuille and Poggio [1983] have shown that a 1-D signal can be reconstructed using two points from its scale space image. Note that our result implies that only one point is sufficient for the reconstruction of that ignal.

The theorem has now been proven for the regular curvature scale space image. To prove the same result about the resampled curvature scale space, recall that derivatives at one point (at any scale) on any curvature zero-crossing contour in the curvature scale space of Γ were computed and it was shown that the resulting equations can be solved for the coefficients of expansion of the curvature function of Γ in functions related to the Hermite polynomials.

As before, we choose a point on a zero-crossing contour at any scale of the resampled curvature scale space image of Γ and compute the necessary derivatives. The value of σ in the resulting equations is then set to zero. Consequently, the arc length evolved curve Γ_{σ} , where σ corresponds to the scale at which the derivatives were computed, is reconstructed modulus uniform scaling, rotation and translation.

The next step is to recover the original curve Γ . This is done by applying reverse arc length evolution to Γ_{σ} . Let the arc length evolved curve Γ_{σ} be defined by:

 $\Gamma_{\sigma} = \{ (X(W,\sigma), Y(W,\sigma)) | W \in [0,1] \} .$

A reverse arc length evolved curve Γ is defined by:

 $\Gamma = \{ (x(w), y(w)) | w \in [0,1] \}$

where

$$\begin{aligned} x(w) &= X(w,\sigma) \circledast D_N(w,\sigma) \\ y(w) &= Y(w,\sigma) \circledast D_N(w,\sigma) \end{aligned}$$

where D_N is a deblurring operator defined in [Hummel et al. 1987] and

$$w(W,\sigma) = \int_{0}^{\sigma} \int_{0}^{w} \kappa^{2}(w,\sigma) dw d\sigma.$$

As a result, Γ is recovered modulo uniform scaling, rotation and translation.

To prove the same result about the renormalized curvature scale space image, evolved curve Γ_{σ} is again reconstructed. Then each of its coordinate functions is deblurred by convolving it with the deblurring operator D_N . Once again Γ is recovered modulo uniform scaling, rotation and translation.

Proof of theorem 3.4.7: Since by assumption all evolved and arc length evolved curves Γ_{σ} are in C_2 , the conditions of the implicit function theorem are satisfied on contours $\kappa(u,\sigma)$, $\kappa(w,\sigma)$ and $\kappa(W,\sigma) = 0$ in the regular, renormalized and resampled curvature scale space images of Γ . Since the proofs are identical, the theorem will be proven here for the regular curvature scale space image.

On any contour in the curvature scale space image

$$\kappa(u,\sigma)=0.$$

Since all Γ_{σ} are in C_2 this is equivalent to:

$$\dot{X}(u,\sigma) \ddot{Y}(u,\sigma) - \ddot{X}(u,\sigma) \dot{Y}(u,\sigma) = 0.$$

To exploit the properties of the heat equation (Hummel et al, 1987), it is convenient to change variables and let

$$t = \frac{1}{2}\sigma^2.$$

So we define

$$\begin{aligned} x(u,t) &= X(u,\sigma) \\ y(u,t) &= Y(u,\sigma) \\ \alpha(u,t) &= x_u(u,t)y_{uu}(u,t) - x_{uu}(u,t)y_u(u,t). \end{aligned} \tag{A.4}$$

The functions x(u,t) and y(u,t) are obtained by convolving $\frac{1}{\sqrt{4\pi t}}e^{-(1/4t)u^2}$ with the original curve coordinates x(u) and y(u) respectively, and so they satisfy the heat equation:

$$x_{uu}(u,t) = x_t(u,t) \tag{A.5}$$

$$y_{uu}(u,t) = y_t(u,t).$$
 (A.6)

On contours $\alpha(u,t) = 0$:

$$\dot{t} = t(u)$$
$$\dot{t}(u) = \frac{dt}{du} = \frac{-\alpha_u}{\alpha_t}.$$

The theorem will be proven if we can show that for all points such that $\dot{t}(u) = 0$ we have $\ddot{t}(u) < 0$. Now,

$$t(u) = 0 \tag{A.7}$$

if and only if

$$\alpha_u(u,t)=0.$$

At an extremum where (A.7) holds, we have

$$\ddot{t}(u) = \frac{d}{du} \left(\frac{-\alpha_u}{\alpha_t} \right) = \frac{\partial}{\partial u} \left(\frac{-\alpha_u}{\alpha_t} \right) + \frac{\partial}{\partial t} \left(\frac{-\alpha_u}{\alpha_t} \right) \frac{dt}{du} = \frac{-\alpha_{uu}}{\alpha_t}.$$

So we must show that if

$$\alpha(u,t)=\alpha_u(u,t)=0$$

then

$$\frac{\alpha_{uu}}{\alpha_t} > 0.$$

We shall show that these conditions require $\frac{\alpha_{uu}}{\alpha_t} = 1$ which proves the theorem.

From (A.4), (A.5) and (A.6) we have

$$\alpha = x_u y_t - x_t y_u$$

$$\alpha_u = x_{uu}y_t + x_uy_{ut} - x_{ut}y_u - x_ty_{uu}.$$

But using (A.5)

$$\alpha_u = x_u y_{ut} - x_{ut} y_u. \tag{A.8}$$

Similarly

$$\alpha_{uu} = (x_u y_{tt} - x_{tt} y_u) + (x_t y_{ut} - x_{ut} y_t).$$

If $\alpha = \alpha_u = 0$ then using (A.4) and (A.8)
$$x_{t}y_{ut} - x_{ut}y_{t} = x_{t}\left(y_{ut} - x_{ut}\frac{y_{t}}{x_{t}}\right) = x_{t}\left(y_{ut} - x_{ut}\frac{y_{u}}{x_{u}}\right) = \frac{x_{t}}{x_{u}}(x_{u}y_{ut} - x_{ut}y_{u}) = 0$$

so

$$\alpha_{uu} = x_u y_{tt} - x_{tt} y_u.$$

We also have

$$\alpha_t = (x_u y_{tt} - x_{tt} y_u) - (x_t y_{ut} - x_{ut} y_t)$$

so

 $\alpha_t = x_u y_{tt} - x_{tt} y_u$

and hence $\alpha_{uu} = \alpha_t$ as claimed.

Notice, incidentally, that $\alpha(u,t)$ satisfies the diffusion equation at the maxima of the contours and that all such contours have a curvature of -1 at their maxima in (u,t) curvature scale space.

Proof of theorem 3.4.8: It will be shown that this theorem holds for an arbitrary parametrization of Γ_{σ} . Therefore it must also be true of arc length parametrization or close approximations.

Let $(X(u,\sigma), Y(u,\sigma))$ be an arbitrary parametrization of Γ_{σ} . Since the class of polynomial functions is closed under convolution with a Gaussian [Hummel *et al.*, 1987], it follows that $X(u,\sigma)$ and $Y(u,\sigma)$ are also polynomial functions:

$$X(u,\sigma) = a_0 + a_1 u + a_2 u^2 + a_3 u^3 + \cdots$$

$$Y(u,\sigma) = b_0 + b_1 u + b_2 u^2 + b_3 u^3 + \cdots$$

Suppose that Γ_{σ} goes through the origin of the coordinate system at u=0. It follows that $a_0=b_0=0$. Assume further that there is a singularity on Γ_{σ} at u=0. We have:

$$X_{u}(u,\sigma) = a_{1} + 2a_{2}u + 3a_{3}u^{2} + 4a_{4}u^{3} + \cdots$$
$$Y_{u}(u,\sigma) = b_{1} + 2b_{2}u + 3b_{3}u^{2} + 4b_{4}u^{3} + \cdots$$

Since $X_u(u,\sigma)$ and $Y_u(u,\sigma)$ are zero at a singular point, it also follows that $a_1=b_1=0$.

We will now perform a case analysis of the singular point at u = 0 to determine when it corresponds to a cusp point. Since we will examine a small neighborhood of point u = 0, we will approximate the curve using the lowest degree terms in $X(u, \sigma)$ and $Y(u, \sigma)$:

$$\Gamma_{\sigma} = (u^m, u^n).$$

Assume w.l.o.g. that n > m. From above we know that m > 1.

Using

$$\mathbf{r}_{u}(u,\sigma) = (X_{u}(u,\sigma), Y_{u}(u,\sigma)) = (m u^{m-1}, n u^{n-1})$$

it follows that

$$\mathbf{r}_{u}(\epsilon,\sigma) = (m \, \epsilon^{m-1}, \, n \, \epsilon^{n-1}) = \epsilon^{m-1}(m, \, n \, \epsilon^{n-m})$$

and

$$\mathbf{r}_{u}(-\epsilon,\sigma)=(m(-\epsilon)^{m-1},\ n(-\epsilon)^{n-1}).$$

We can now analyze the singular point in each of the four possible cases:

1. m and n are both even numbers.

m-1 and n-1 are both odd numbers. Therefore

$$\mathbf{r}_u(-\epsilon,\sigma) = (-m\epsilon^{m-1}, -n\epsilon^{n-1}) = -\epsilon^{m-1}(m, n\epsilon^{n-m}).$$

A comparison of $\mathbf{r}_u(\epsilon, \sigma)$ and $\mathbf{r}_u(-\epsilon, \sigma)$ shows that an infinitesimal change in the parameter u results in a large change in the direction of the tangent vector. Therefore the singular point *is* also a cusp point in this case.

2. m and n are both odd.

m-1 and n-1 are both even. Hence

$$\mathbf{r}_{u}(-\epsilon,\sigma) = (m \,\epsilon^{m-1}, n \,\epsilon^{n-1}) = \epsilon^{m-1}(m, n \,\epsilon^{n-m}).$$

Comparing $\mathbf{r}_u(\epsilon,\sigma)$ to $\mathbf{r}_u(-\epsilon,\sigma)$ now shows that the tangent direction does not change with u in a small neighborhood of the singular point. Therefore this singular point is *not* a cusp point.

3. m is odd and n is even.

m-1 is even and n-1 is odd. Hence

$$\mathbf{r}_{u}(-\epsilon,\sigma)=(m\,\epsilon^{m-1},\,-n\,\epsilon^{n-1})=\epsilon^{m-1}(m,\,-n\,\epsilon^{n-m}).$$

An infinitesimal change in u also results in an infinitesimal change in the tangent direction. Again, this singular point is *not* a cusp point.

4. m is even and n is odd.

m-1 is odd and n-1 is even. So

$$\mathbf{r}_{u}(-\epsilon,\sigma) = (-m\epsilon^{m-1}, n\epsilon^{n-1}) = \epsilon^{m-1}(-m, n\epsilon^{n-m}).$$

An infinitesimal change in u now results in a large change in the tangent direction. Therefore this singular point is a cusp point.

It follows from the case analysis above that only the singular points in

cases 1 and 4 are cusp points. We will now derive analytical expressions for the curve $\Gamma_{\sigma-\delta}$ so that it can be analyzed in a small neighborhood of the cusp point.

To deblur function $f(u) = u^k$, we convolve a rescaled version of that function with the function $\frac{2}{\sqrt{\pi}}e^{-x^2}(1-x^2)$, an approximation to the deblurring operator derived in (Hummel *et al.*, 1987) good for small amounts of deblurring, as follows:

$$(D_t f)(y) = \int_{-\infty}^{\infty} \frac{2}{\sqrt{\pi}} e^{-x^2} (1-x^2) f(y+2x\sqrt{t}) dx$$

Or

$$(D_t f)(y) = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-x^2} (1-x^2)(y+2x\sqrt{t})^k dx$$

where t is the scale factor and controls the amount of deblurring. Solving the integral above yields

$$(D_t f)(y) = \sum_{\substack{p=0\\(p \text{ even})}}^k 1.3.5 \cdots (p-1) \frac{(2t)^{p/2} k(k-1) \cdots (k-p+1)}{p!} (1-p) y^{k-p}.$$
(A.9)

The following are four functions of the form $f(u) = u^k$ and their deblurred versions:

$$\begin{array}{ll} \mathbf{a}. \ f(u) = u^2 & (D_t f)(u) = u^2 - 2t \\ \mathbf{b}. \ f(u) = u^3 & (D_t f)(u) = u^3 - 6tu \\ \mathbf{c}. \ f(u) = u^4 & (D_t f)(u) = u^4 - 12tu^2 - 36t^2 \\ \mathbf{d}. \ f(u) = u^5 & (D_t f)(u) = u^5 - 20tu^3 - 180t^2u \ . \end{array}$$

We can now analyze the cusp points identified in cases 1 and 4 above. In case 1, the curve Γ_{σ} is approximated by (u^m, u^n) where m and n are both even numbers. We now deblur the curve to obtain:

$$(D_t x)(u) = u^m - c_1 t u^{m-2} - c_2 t^2 u^{m-4} - \cdots - c_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u^2 - c_{\frac{m}{2}} t^{\frac{m}{2}}$$
$$(D_t y)(u) = u^n - c'_1 t u^{n-2} - c'_2 t^2 u^{n-4} - \cdots - c'_{\frac{n-2}{2}} t^{\frac{n-2}{2}} u^2 - c'_{\frac{n}{2}} t^{\frac{n}{2}}.$$

Note that all powers of u are even and the constants c_j and c'_j are all positive as follows from an examination of (A.9). It follows that

$$(D_t \dot{x})(u) = mu^{m-1} - (m-2)c_1 tu^{m-3} - \cdots - 2c_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u$$
$$(D_t \dot{y})(u) = nu^{n-1} - (n-2)c'_1 tu^{n-3} - \cdots - 2c'_{\frac{m-2}{2}} t^{\frac{n-2}{2}} u$$

contain only odd powers of u and $(D_t \dot{\mathbf{r}})(\epsilon) = -(D_t \dot{\mathbf{r}})(-\epsilon)$. Hence there is also a cusp point on the curve $\Gamma_{\sigma-\delta}$ at $u_0 = 0$. In fact, the cusp point must also exist on Γ itself. This is a contradiction of the assumption that Γ is in C_2 . It follows that Γ_{σ} can not have a cusp point of this kind at u_0 .

We shall now turn to the cusp points encountered in case 4. Recall that, in that case, the curve Γ_{σ} is approximated, in a small neighborhood of the cusp point, by (u^m, u^n) where *m* is even and *n* is odd. Again we deblur the curve to obtain:

$$(D_t x)(u) = u^m - c_1 t u^{m-2} - c_2 t^2 u^{m-4} - \cdots - c_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u^2 - c_{\frac{m}{2}} t^{\frac{m}{2}}$$
$$(D_t y)(u) = u^n - c'_1 t u^{n-2} - c'_2 t^2 u^{n-4} - \cdots - c'_{\frac{n-1}{2}} t^{\frac{n-1}{2}} u.$$

Again note that constants c_i and c'_i are all positive.

The deblurred curve intersects itself if there are two values of u, u_1 and u_2 , such that

$$x(u_1) = x(u_2)$$
 (A.10)

$$y(u_1) = y(u_2).$$
 (A.11)

Since $(D_t x)(u)$ contains even powers of u only, it follows from (A.10) that $u_1 = -u_2$. Since $(D_t y)(u)$ contains odd powers of u only, substituting $u_1 = -u_2$ in (A.11) and simplifying yields:

$$u_1^n - c'_1 t u_1^{n-2} - c'_2 t^2 u_1^{n-4} - \cdots - c'_{\frac{n-1}{2}} t^{\frac{n-1}{2}} u_1 = 0.$$

Since $\Gamma_{\sigma-\delta}$ is of interest to us, we let $t = \delta$. We now obtain

$$u_1^{n} - c'_1 \delta u_1^{n-2} - c'_2 \delta^2 u_1^{n-4} - \cdots - c'_{\frac{n-1}{2}} \delta^{\frac{n-1}{2}} u_1 = 0.$$
 (A.12)

 $u_1 = 0$ is one of the roots of this equation. For very small values of u_1 , the LHS of (A.12) is negative since the first term will be smaller than each of the other terms (which are negative). As u_1 grows larger, the first term becomes larger than the

sum of all other terms and therefore the LHS of (A.12) becomes positive. It follows that there exists a positive value of u_1 at which (A.12) is satisfied. Therefore $\Gamma_{\sigma-\delta}$ is self-intersecting in a small neighborhood of the cusp point of Γ_{σ} .

Proof of theorem 3.4.9: Assume by contradiction that Γ is a simple curve which intersects itself during arc length evolution. Γ must touch itself at point P before self-intersection. Let Γ_{σ_0} be such a curve. There are two distinct neighborhoods of Γ_{σ_0} which contain point P. Let these neighborhoods be S_1 and S_2 . Note that S_1 and S_2 have collinear tangents at P. Let L be the line of those tangents. The tangents exist since it follows from theorem 3.4.8 that P can not be a cusp point because Γ_{σ} does not self-intersect for $\sigma \leq \sigma_0$.

Recall that the infinitesimal movement of each point of S_1 and S_2 is determined by the equation:

$$\frac{\partial \mathbf{R}}{\partial t} = \kappa \,\mathbf{n}$$

Therefore during arc length evolution every point will move in the direction of the normal vector by an amount equal to the curvature at that point. Similarly, during reverse arc length evolution, every point will move in the opposite direction of the normal vector by an amount equal to the curvature at that point.

It follows that if S_1 and S_2 are on opposite sides of L, after an infinitesimal amount of reverse arc length evolution, they will intersect. This is a contradiction of the assumption that Γ was simple before touching itself. Assume then that S_1 and S_2 are on the same side of L. Note that S_1 and S_2 can not be overlapping since they would still be overlapping after an inifinitesimal amount of reverse arc length evolution which is also a contradiction of the assumption that Γ was simple before touching itself. Let S_1 be the segment *inside* S_2 , i.e., the tangent to S_2 always has S_1 to the same side. It can be seen that S_1 has a larger curvature at Pthan S_2 . It follows that after an infinitesimal amount of reverse arc length evolution, S_1 and S_2 will intersect which is again a contradiction. It follows that Γ remains simple during arc length evolution.

Proof of theorem 3.4.10: It will be shown that this theorem holds for an arbitrary parametrization of Γ_{σ} . Therefore it must also be true of arc length parametrization or close approximations.

Let (x(u), y(u)) be an arbitrary parametrization of Γ_{σ} with a cusp point at u_0 . Using a case analysis similar to the one in the proof of theorem 3.4.8 to characterize all possible kinds of singularities of Γ_{σ} at u_0 , we can again conclude that only the singular points in cases 1 and 4 are cusp points. In case 1, the curve is approximated by (u^m, u^n) in a neighborhood of u_0 where m and n are both even. This type of cusp point can not arise on Γ_{σ} if Γ is in C_1 . We now

turn to the cusp points of case 4. Recall that in case 4, the curve Γ_{σ} is approximated, in a neighborhood of u_0 , by (u^m, u^n) where m is even and n is odd. Observe that

$$\dot{x}(u) = m u^{m-1}$$
$$\ddot{x}(u) = m(m-1) u^{m-2}$$
$$\dot{y}(u) = n u^{n-1}$$
$$\dddot{y}(u) = n(n-1) u^{n-2}$$

and

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \dot{y}(u)\ddot{x}(u)}{(\dot{x}(u)^{2} + \dot{y}(u)^{2})^{3/2}} = \frac{mn(n-1)u^{m+n-3} - m(m-1)nu^{m+n-3}}{(m^{2}u^{2m-2} + n^{2}u^{2n-2})^{3/2}}$$

Since n > m, $\kappa(u)$ is always positive on either side of the cusp point in a neighborhood of u_0 . Therefore no curvature zero-crossings exist in that neighborhood on Γ_{σ} .

We now derive analytical expressions for $\Gamma_{\sigma+\delta}$ so that it can be analyzed in a neighborhood of u_0 . To blur function $f(u) = u^k$, we convolve a rescaled version of that function with the function $\frac{1}{\sqrt{\pi}}e^{-x^2}$, the deblurring operator, as follows:

$$F(u) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} e^{-x^2} f(u + 2x\sqrt{t}) dx$$

or

$$F(u) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-x^2} (u + 2x\sqrt{t})^k dx$$

where t is the scale factor and controls the amount of blurring. Solving the integral above yields

$$F(u) = \sum_{\substack{p=0 \\ (p \text{ even})}}^{k} 1.3.5 \cdots (p-1) \frac{(2t)^{p/2} k(k-1) \cdots (k-p+1)}{p!} u^{k-p}.$$

The following are four functions of the form $f(u) = u^k$ and their blurred versions:

a.
$$f(u) = u^2$$
 $F(u) = u^2 + 2t$
b. $f(u) = u^3$ $F(u) = u^3 + 6tu$
c. $f(u) = u^4$ $F(u) = u^4 + 12tu^2 + 12t^2$

d.
$$f(u) = u^5$$
 $F(u) = u^5 + 20tu^3 + 60t^2u$.

An expression for $\Gamma_{\sigma+\delta}$ in a neighborhood of the cusp point can be obtained by by blurring each of its coordinate functions:

$$X(u) = u^{m} + c_{1}tu^{m-2} + c_{2}t^{2}u^{m-4} + \cdots + c_{\frac{m-2}{2}}t^{\frac{m-2}{2}}u^{2} + c_{\frac{m}{2}}t^{\frac{m}{2}}$$
$$Y(u) = u^{n} + c'_{1}tu^{n-2} + c'_{2}t^{2}u^{n-4} + \cdots + c'_{\frac{n-1}{2}}t^{\frac{n-1}{2}}u.$$

Note that all constants are positive, all powers of u in X(u) are even and all powers of u in Y(u) are odd. It follows that all powers of u in

$$\dot{X}(u) = mu^{m-1} + (m-2)c_1tu^{m-3} + \cdots + 2c_{\frac{m-2}{2}}t^{\frac{m-2}{2}}u^{m-2}$$

are odd, all powers of u in

$$\ddot{X}(u) = m(m-1)u^{m-2} + (m-2)(m-3)c_1tu^{m-4} + \cdots + 2c_{\frac{m-2}{2}}t^{\frac{m-2}{2}}$$

are even, all powers of u in

$$\dot{Y}(u) = nu^{n-1} + (n-2)c'_{1}tu^{n-3} + \cdots + c'_{\frac{n-1}{2}}t^{\frac{n-1}{2}}$$

are even and all powers of u in

$$\ddot{Y}(u) = n(n-1)u^{n-2} + (n-2)(n-3)c'_{1}tu^{n-4} + \cdots + c'_{\frac{n-3}{2}}t^{\frac{n-3}{2}}$$

are odd.

The curvature of $\Gamma_{\sigma+\delta}$ in a neighborhood of u_0 is given by

$$\kappa(u) = rac{\dot{X}(u) \, \ddot{Y}(u) - \dot{Y}(u) \ddot{X}(u)}{(\dot{X}(u)^2 + \dot{Y}(u)^2)^{3/2}}.$$

Since the denominator of $\kappa(u)$ never goes to zero in a neighborhood of u_0 , the zero-crossings of $\kappa(u)$ are the same as those of

$$\kappa'(u) = \dot{X}(u)\ddot{Y}(u) - \dot{Y}(u)\ddot{X}(u).$$

Observe that the term with the highest power of u in $\dot{X}(u)\ddot{Y}(u)$ is $mn(n-1)u^{m+n-3}$ and the term with the highest power of u in $\dot{Y}(u)\ddot{X}(u)$ is $m(m-1)nu^{m+n-3}$ and that in both $\dot{X}(u)\ddot{Y}(u)$ and $\dot{Y}(u)\ddot{X}(u)$, all powers of u are even and all constants are positive. Furthermore, note that at u=0, $\dot{X}(u)\ddot{Y}(u)$ is zero and $\dot{Y}(u)\ddot{X}(u) > 0$. Therefore at u=0, $\kappa < 0$. As u grows larger in absolute value, the terms in $\dot{X}(u) \ddot{Y}(u)$ and $\dot{Y}(u)\ddot{X}(u)$ with highest powers of u become dominant (all other terms have positive powers of $t=\delta$ in them). Since the dominant terms have equal powers of u, the one with the larger coefficient becomes the larger term. Since n > m, the largest term in $\dot{X}(u) \dot{Y}(u)$ becomes larger than the largest term in $\dot{Y}(u)\ddot{X}(u)$. Therefore as u grows in absolute value, κ becomes positive. It follows that there are two curvature zero-crossings in the neighborhood of u_0 on $\Gamma_{\sigma+\delta}$. These zero-crossings are new since it was shown that no zero-crossings exist in the neighborhood of u_0 on Γ_{σ} .

Appendix B:

Proofs of Results on Space Curves

This appendix contains the proofs of the theorems of chapter 4. The proofs of theorems 4.4.1 - 4.4.5 have been given for arc length evolution only since the proofs for regular evolution are similar and simpler. Theorems 4.4.6 and 4.4.7 have been shown to hold for regular, renormalized and resampled torsion scale space images. Furthermore, theorems 4.4.8, 4.4.9 and 4.4.10 have been shown to hold for both regular and arc length evolution.

Proof of theorem 4.4.1: It will be shown that arc length evolution is invariant under a general affine transform. Let $\Gamma_{\sigma} = (X(W,\sigma), Y(W,\sigma), Z(W,\sigma))$ be an arc length evolved version of $\Gamma = (x(w), y(w), z(w))$. If Γ_{σ} is transformed according to an affine transform, then its new coordinates, X_1 , Y_1 and Z_1 are given by

 $X_1(W,\sigma) = a X(W,\sigma) + b Y(W,\sigma) + c Z(W,\sigma) + d$ $Y_1(W,\sigma) = e X(W,\sigma) + f Y(W,\sigma) + g Z(W,\sigma) + h$ $Z_1(W,\sigma) = i X(W,\sigma) + j Y(W,\sigma) + k Z(W,\sigma) + l.$

Now suppose Γ is transformed according to an affine transform and then evolved. The coordinates X_2 , Y_2 and Z_2 of the new curve are

$$\begin{aligned} X_2(W,\sigma) &= (ax(W) + by(W) + cz(W) + d) \circledast g(W,\sigma) \\ Y_2(W,\sigma) &= (ex(W) + fy(W) + gz(W) + h) \circledast g(W,\sigma) \\ Z_2(W,\sigma) &= (ix(W) + jy(W) + kz(W) + l) \circledast g(W,\sigma). \end{aligned}$$

Since the convolution operator is distributive [Kecs 1982], it follows that

$$X_{2}(W,\sigma) = X_{1}(W,\sigma)$$
$$Y_{2}(W,\sigma) = Y_{1}(W,\sigma)$$
$$Z_{2}(W,\sigma) = Z_{1}(W,\sigma)$$

and the theorem follows.

Proof of theorem 4.4.2: Let $\Gamma = (x(w), y(w), z(w))$ be a closed curve and let $\Gamma_{\sigma} = (X(W, \sigma), Y(W, \sigma), Z(W, \sigma))$ be an arc length evolved version of Γ . On Γ :

$$(x(0), y(0), z(0)) = (x(1), y(1), z(1)).$$

On Γ_{σ} :

$$(X(0,\sigma), Y(0,\sigma), Z(0,\sigma)) = (X(1,\sigma), Y(1,\sigma), Z(1,\sigma)).$$

It follows that Γ_{σ} is closed.

Proof of theorem 4.4.3: Let $\Gamma = (x(w), y(w), z(w))$ be a connected planar curve and $\Gamma_{\sigma} = (X(W, \sigma), Y(W, \sigma), Z(W, \sigma))$ be an arc length evolved version of that curve. Since Γ is connected, x(w), y(w) and z(w) are continuous functions and therefore $X(W, \sigma)$, $Y(W, \sigma)$ and $Z(W, \sigma)$ are also continuous. Let W_0 be any value of parameter W and let x_0 , y_0 and z_0 be the values of $X(W, \sigma)$, $Y(W, \sigma)$ and $Z(W, \sigma)$ at W_0 respectively. If W goes through an infinitesimal change, then $X(W, \sigma)$, $Y(W, \sigma)$ and $Z(W, \sigma)$ will also go through infinitesimal changes:

$$X(W_0,\sigma) \rightarrow x_0 + \delta$$

 $Y(W_0,\sigma) \rightarrow y_0 + \xi$
 $Z(W_0,\sigma) \rightarrow z_0 + \epsilon.$

As a result, point $P(x_0, y_0)$ on Γ_{σ} goes to point $P'(x_0 + \delta, y_0 + \xi, z_0 + \epsilon)$. Let the distance between P and P' be D. Then

$$D = \sqrt{\delta^2 + \xi^2 + \epsilon^2} \le \delta\sqrt{2}$$

assuming $|\delta|$ is the largest of $|\delta|$, $|\xi|$ and $|\epsilon|$. It follows that an infinitesimal change in parameter W also results in an infinitesimal change in the value of the vector-valued function Γ_{σ} . Therefore Γ_{σ} is a connected curve.

Proof of theorem 4.4.4: Let *M* be the center of mass of $\Gamma = (x(w), y(w), z(w))$ with coordinates (x_M, y_M, z_M) . Then

$$x_M = \frac{\int\limits_0^1 x(w) dw}{\int\limits_0^1 dw} = \int\limits_0^1 x(w) dw.$$

Let $\Gamma_{\sigma} = (X(W,\sigma), Y(W,\sigma), Z(W,\sigma))$ be an arc length evolved version of Γ with $N = (X_N, Y_N, Z_N)$ as its center of mass. Observe that

$$X_N = \int_0^1 X(W,\sigma) dW = \int_{0-\infty}^{1\infty} \int_{-\infty}^{\infty} g(v,\sigma) x(W-v) dv dW = \int_{-\infty}^{\infty} g(v,\sigma) \left(\int_0^1 x(W-v) dW\right) dv.$$

W covers Γ_{σ} exactly once. Therefore

$$\int_{0}^{1} x(W-v) dW = x_{M}.$$

So

$$X_N = x_M$$

 $Y_N = y_M$

Similarly

and

 $Z_N = z_M$

It follows that M and N are the same point.

Proof of theorem 4.4.5: Since G is simple and convex, every plane P tangent to G contains that curve in the left (or right) half-space it creates. Since Γ is inside G, Γ is also contained in the same half-space. Now rotate P and Γ so that P becomes parallel to the YZ-plane. P is now described by the equation x = c. Since P does not intersect Γ , it follows that $x(w_0) \ge c$ for every point w_0 on Γ . Let Γ_{σ} be an arc length evolved version of Γ . Every point of Γ_{σ} is a weighted average of all the points of Γ . Therefore $X(W_0, \sigma) \ge c$ for every point W_0 on Γ_{σ} and Γ_{σ} is also contained in the same half-space. This result holds for every plane tangent to G therefore Γ_{σ} is contained inside the intersection of all the left (or right) halfspaces created by the tangent planes of G. It follows that Γ_{σ} is also inside G. \Box

Proof of theorem 4.4.6: The proof will first be given for the regular torsion scale space of Γ , then the modifications needed to apply the same proof to the resampled and renormalized torsion scale space images of Γ will be explained.

Section i shows that the derivatives at a point on a torsion zero-crossing contour provide homogeneous equations in the moments of the coordinate

functions of the curve. Section ii shows that the moments are related to the coefficients of expansion of the coordinate functions of the curve in functions related to the Hermite polynomials. Section iii shows that the moments at one torsion zero-crossing point can be related to the moments at other torsion zero-crossing points. Section iv shows that the cubic equations obtained in section i can be converted into a homogeneous linear system of equations which can be solved uniquely for function $\tau(u) \kappa^2(u)$.

i. Constraints from the torsion zero-crossing contours

Let $\Gamma = (x(u), y(u), z(u))$ be the arc-length parametrization of the curve with Fourier transform $\tilde{\Gamma} = (\tilde{x}(\omega), \tilde{y}(\omega), \tilde{z}(\omega))$. The Fourier transform of the Gaussian filter $G(u, t) = \frac{1}{\sqrt{2t}}e^{-u^2/4t}$ is $\tilde{G}(\omega) = e^{-\omega^2 t}$.

Let $\Gamma_{t_0} = (x(u, t_0), y(u, t_0), z(u, t_0))$ be a curve obtained by convolving x(u), y(u) and z(u) with $G(u, t_0)$. Assume that Γ_{t_0} is in C_{∞} . Such a t_0 exists since Γ is in C_1 . Assume that $\kappa(u, t) \neq 0$ on the torsion zero-crossing contours in a neighborhood of t_0 . It follows that the torsion zero-crossings are given by solutions of $\beta(u, t) = 0$ where [Goetz 1970]

$$\beta(u,t) = \dot{x}(\ddot{y}\ddot{z} - \ddot{y}\ddot{z}) - \dot{y}(\ddot{x}\ddot{z} - \ddot{x}\ddot{z}) + \dot{z}(\ddot{x}\ddot{y} - \ddot{x}\ddot{y})$$
(B.1)

where . represents derivative with respect to u. Note that on Γ (t=0), $\beta(u,t)$ is given by

$$\beta(u,t) = \tau(u,t) \kappa^2(u,t). \tag{B.2}$$

Using the convolution theorem, $\dot{x}(u,t)$, $\dot{y}(u,t)$ and $\dot{z}(u,t)$ can be expressed as following:

$$\dot{x}(u,t) = \int e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{x}(\omega) d\omega$$
$$\dot{y}(u,t) = \int e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{y}(\omega) d\omega$$
$$\dot{z}(u,t) = \int e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{z}(\omega) d\omega$$

and therefore

$$\begin{split} \ddot{x}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (-\omega^2) \, \tilde{x}(\omega) \, d\omega \\ \ddot{y}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (-\omega^2) \, \tilde{y}(\omega) \, d\omega \end{split}$$

$$\begin{split} \ddot{z}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (-\omega^2) \, \tilde{z}(\omega) \, d\omega \\ \ddot{x}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (-i\omega^3) \, \tilde{x}(\omega) \, d\omega \\ \ddot{y}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (-i\omega^3) \, \tilde{y}(\omega) \, d\omega \\ \ddot{z}(u,t) &= \int e^{-\omega^2 t} e^{i\omega u} (-i\omega^3) \, \tilde{z}(\omega) \, d\omega. \end{split}$$

Note that the moment of order k of the function $f(\omega) = e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{x}(\omega)$ is defined by:

$$M_{k} = \int_{-\infty}^{\infty} (i\omega)^{k} e^{-\omega^{2}t} e^{i\omega u} (i\omega) \tilde{x}(\omega) d\omega$$

the moment of order k of the function $f'(\omega) = e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{y}(\omega)$ is defined by:

 \sim

$$M'_{k} = \int_{-\infty}^{\infty} (i\omega)^{k} e^{-\omega^{2}t} e^{i\omega u} (i\omega) \,\tilde{y}(\omega) \,d\omega$$

and the moment of order k of the function $f''(\omega) = e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{z}(\omega)$ is defined by:

$$M_k'' = \int_{-\infty}^{\infty} (i\omega)^k e^{-\omega^2 t} e^{i\omega u} (i\omega) \tilde{z}(\omega) d\omega.$$

Therefore equation (B.2) can be written as:

$$\beta(u,t) = M_0 M_1' M_2'' - M_0 M_1'' M_2' - M_0' M_1 M_2'' + M_0' M_2 M_1'' + M_0'' M_1 M_2' - M_0'' M_2 M_1'.$$

The Implicit Function Theorem guarantees that the contours u(t) are C_{∞} in a neighborhood of t_0 . Let ξ be a parameter of the torsion zero-crossing contour. Then

$$\frac{d}{d\xi} = \frac{du}{d\xi}\frac{\partial}{\partial u} + \frac{dt}{d\xi}\frac{\partial}{\partial t}.$$

On the torsion zero-crossing contour, $\beta = 0$ and $\frac{d^k}{d\xi^k}\beta = 0$ for all integers k. Furthermore, since the torsion zero-crossing contour is known, all the derivatives of u and t with respect to ξ are known as well. We now compute the derivatives of β with respect to ξ at (u_0, t_0) . The first derivative is given by:

$$\frac{d}{d\xi}\beta(u_0, t_0) = \frac{du}{d\xi}\frac{\partial\beta(u_0, t_0)}{\partial u} + \frac{dt}{d\xi}\frac{\partial\beta(u_0, t_0)}{\partial t}$$
(B.3)

where

$$\begin{split} \frac{\partial \beta(u_0, t_0)}{\partial u} &= M_3'' M_0 M_1' - M_3' M_0 M_1'' - M_3' M_0' M_1 + M_3 M_0' M_1'' + M_3' M_0' M_1 - M_3 M_0' M_1'' \\ \text{and} \\ \frac{\partial \beta(u_0, t_0)}{\partial t} &= M_3' M_0 M_2'' + M_4'' M_0 M_1' - M_3'' M_0 M_2' - M_4' M_0 M_1'' - M_3 M_0' M_2'' - M_4'' M_0' M_1 \\ &+ M_4 M_0' M_1'' + M_3' M_0' M_2 + M_3 M_0' M_2' + M_4' M_0' M_1 - M_4 M_0' M_1' - M_3' M_0' M_2 \\ \text{and the second derivative is given by:} \\ \frac{\partial^2 \beta}{\partial \xi^2} &= \frac{d^2 u}{d\xi^2} \frac{\partial \beta}{\partial u} + \frac{d^2 t}{d\xi^2} \frac{\partial \beta}{\partial t} + \left(\frac{d u}{d\xi}\right)^2 \frac{\partial^2 \beta}{\partial u^2} + 2 \frac{d u}{d\xi} \frac{d t}{d\xi} \frac{\partial^2 \beta}{\partial u \partial t} + \left(\frac{d t}{d\xi}\right)^2 \frac{\partial^2 \beta}{\partial t^2} \quad (B.4) \\ \text{where} \\ \frac{\partial^2 \beta}{\partial u^2} &= M_4'' M_0 M_1' + M_2' M_3'' M_0 - M_4' M_0 M_1'' - M_2' M_3' M_0 - M_4'' M_0' M_1 - M_2 M_3' M_0' \\ &+ M_4 M_0' M_1'' + M_2'' M_3 M_0' + M_4' M_0'' M_1 + M_2 M_3' M_0'' - M_4 M_0'' M_1' - M_2' M_3 M_0' \\ &+ M_4 M_0' M_1'' + M_2' M_3' M_1' - M_5' M_0 M_1'' - M_2 M_3' M_1'' - M_5'' M_0' M_1 - M_2' M_3' M_1' \\ &+ M_5 M_0' M_1'' + M_2' M_3' M_1' - M_5' M_0' M_1' + M_2' M_3' M_1 - M_5' M_0' M_1' - M_2'' M_3' M_1 \\ &+ M_5 M_0' M_1'' + M_2' M_3' M_0' + M_6' M_0 M_1'' + M_2' M_3' M_1' - M_5' M_0' M_1' - M_2'' M_3' M_1 \\ &+ M_6 M_0' M_1'' + M_2' M_3' M_0' + M_6' M_0 M_1' + M_2' M_3' M_0' - M_6'' M_0' M_1' - M_2'' M_3' M_1 \\ &+ M_6 M_0' M_1'' + M_2' M_3' M_0' + M_6' M_0 M_1' + M_2' M_3' M_0' - M_6'' M_0' M_1' - M_2'' M_3' M_0 \\ &- M_6' M_0 M_1'' - M_2 M_4' M_1'' - M_5 M_0' M_1'' + M_2'' M_3' M_0' - M_6'' M_0' M_1' - M_2'' M_3' M_0' \\ &+ M_6' M_0' M_1'' + M_2'' M_4' M_1'' + 2 M_3'' M_4' M_0' + M_5'' M_0' M_2' + 2 M_4' M_3 M_0'' \\ &+ M_6' M_0' M_1'' + M_2'' M_4' M_1'' + 2 M_3'' M_4' M_0' + M_5'' M_0' M_2' + M_5 M_0'' M_2' + 2 M_4' M_3 M_0'' \\ &+ M_6' M_0'' M_1' + M_2'' M_4' M_1' - M_6 M_0' M_1' - M_2'' M_4' M_1' - 2 M_3' M_0' - M_5'' M_0'' M_2' . \end{split}$$

Since the parametric derivatives along the torsion zero-crossing contours are zero, equation (B.3) is equal to zero. Note that equation (B.3) is in the first five

145

moments of functions $f(\omega)$, $f'(\omega)$ and $f''(\omega)$ and equation (B.4) is in the first seven moments of those functions. In general, the k+1st equation, $\frac{d^k}{d\xi^k}\beta(u,t) = 0$ is a cubic equation in the first 2k+3 moments of each of the functions $f(\omega)$, $f'(\omega)$ and $f''(\omega)$.

It follows that the first n+1 equations at (u_0, t_0) are in a total of 3(2n+3) = 6n+9 moments. Our axes are again chosen such that $u_0 = 0$. The next section shows that the moments of $f(\omega)$, $f'(\omega)$ and $f''(\omega)$ are the coefficients a_k , b_k and c_k in the expression of functions $\dot{x}(u)$, $\dot{y}(u)$ and $\dot{z}(u)$ in functions $\phi_k(u,\sigma)$ related to Hermite polynomials. Therefore we have n+1 equations in the first 6n+9 coefficients a_k , b_k and c_k . To determine the a_k , b_k and c_k , we need 5n+8 additional and independent equations which can be provided by considering six neighboring torsion zero-crossing contours at (u_1, t_0) , (u_2, t_0) , (u_3, t_0) , (u_4, t_0) , (u_5, t_0) and (u_6, t_0) .

ii. The moments and the coefficients of expansion of $\dot{x}(u)$, $\dot{y}(u)$ and $\dot{z}(u)$

This section shows that the moments and the moment-triples in equations $\frac{d^k}{d\xi^k}\beta(u,t)$ are related respectively to the coefficients of the expression of the functions $\dot{x}(u)$, $\dot{y}(u)$ and $\dot{z}(u)$ and function $\beta(u)$ in functions related to the Hermite polynomials.

Expand function

$$\dot{x}(u) = \frac{d}{du}x(u)$$

in terms of the functions $\phi_k(u,\sigma)$ related to the Hermite polynomials $H_k(u)$ by

$$\phi_k(u,\sigma) = (-1)^k \frac{\sigma^{k-1}}{(\sqrt{2})^{k+1}\sqrt{\pi}} H_k(\frac{u}{\sigma\sqrt{2}})^k$$
$$H_k(u) = (-1)^k e^{u^2} \frac{d^k}{du^k} e^{-u^2}$$
$$\dot{x}(u) = \sum a_k(\sigma)\phi_k(u,\sigma)$$

The coefficients $a_k(\sigma)$ of the expansion are given by

$$a_k(\sigma) = \langle w_k(u,\sigma), \dot{x}(u) \rangle$$

where \langle , \rangle denotes inner product in L^2 and $\{w_k(u,\sigma)\}$ is the set of functions biorthogonal to $\{\phi_k(u,\sigma)\}$. The $\{\phi_k(u,\sigma)\}$ are given explicitly by

$$\phi_k(u,\sigma) = \frac{\sigma^{2k-1}}{k!\sqrt{2\pi}} e^{\frac{u^2}{2\sigma^2}} \frac{d^k}{du^k} e^{-\frac{u^2}{2\sigma^2}}$$

and the $w_k(u,\sigma)$ by

$$w_k(u,\sigma) = (-1)^k \frac{d^k}{du^k} e^{-\frac{u^k}{2\sigma^2}}$$

Since

$$\dot{x}(u) = \frac{1}{\sqrt{2\pi}} \int e^{i\omega u} (i\omega) \tilde{x}(\omega) \, d\omega$$

the a_k are given by

$$a_k(\sigma) = \frac{1}{\sqrt{2\pi}} (-1)^k \int \leq \frac{d^k}{du^k} e^{-\frac{u^2}{2\sigma^2}}, e^{i\omega u} \geq (i\omega) \tilde{x}(\omega) d\omega$$

The inner product is just the inverse Fourier transform of $w_k(u,\sigma)$. Therefore

$$a_k(\sigma) = \int (i\omega)^k e^{\frac{-\omega^2 \sigma^2}{2}} (i\omega) \tilde{x}(\omega) d\omega$$

which is equal to M_k modulo a factor $e^{i\omega u}$, since $t = \sigma^2/2$.

Similarly, the functions

$$\dot{y}(u) = \frac{d}{du}y(u)$$
$$\dot{z}(u) = \frac{d}{du}z(u)$$

can be expanded in terms of the functions $\phi_k(u,\sigma)$ by

$$\dot{y}(u) = \sum b_k(\sigma) \phi_k(u, \sigma)$$
$$\dot{z}(u) = \sum c_k(\sigma) \phi_k(u, \sigma)$$

and it again follows that

$$b_{k}(\sigma) = \int (i\omega)^{k} e^{\frac{-\omega^{2}\sigma^{2}}{2}} (i\omega) \tilde{y}(\omega) d\omega$$
$$c_{k}(\sigma) = \int (i\omega)^{k} e^{\frac{-\omega^{2}\sigma^{2}}{2}} (i\omega) \tilde{z}(\omega) d\omega$$

which are equal to M'_k and M''_k respectively modulo a factor $e^{i\omega u}$.

Furthermore, $a'_k(\sigma)$, $b'_k(\sigma)$ and $c'_k(\sigma)$, the coefficients of expansion of functions $\ddot{x}(u)$, $\ddot{y}(u)$ and $\ddot{z}(u)$ in $\phi_k(u,\sigma)$ respectively, can be seen to be related to $a_k(\sigma)$, $b_k(\sigma)$ and $c_k(\sigma)$ according to the following relationships:

$$a'_{k-1}(\sigma) = a_k(\sigma)$$

$$b'_{k-1}(\sigma) = b_k(\sigma)$$

$$c'_{k-1}(\sigma) = c_k(\sigma)$$

(B.5)

and $a_k''(\sigma)$, $b_k''(\sigma)$ and $c_k''(\sigma)$, the coefficients of expansion of functions $\ddot{x}(u)$, $\ddot{y}(u)$ and $\ddot{z}(u)$ in $\phi_k(u,\sigma)$ respectively, can be seen to be related to $a_k(\sigma)$, $b_k(\sigma)$ and $c_k(\sigma)$ by the following relationships:

$$a_{k-2}''(\sigma) = a_k(\sigma)$$

$$b_{k-2}''(\sigma) = b_k(\sigma)$$

$$c_{k-2}''(\sigma) = c_k(\sigma).$$

(B.6)

Now observe that the function $\tau(u) \kappa^2(u)$ can be expressed as:

$$\begin{aligned} \tau(u) \,\kappa^2(u) &= \dot{x}(u)(\ddot{y}(u)\ddot{z}(u) - \ddot{y}(u)\dot{z}(u)) \\ &- \dot{y}(u)(\ddot{x}(u)\ddot{z}(u) - \ddot{x}(u)\dot{z}(u)) \\ &+ \dot{z}(u)(\ddot{x}(u)\ddot{y}(u) - \ddot{x}(u)\ddot{y}(u)) \\ &= \dot{x}(u)\ddot{y}(u)\ddot{z}(u) - \dot{x}(u)\ddot{y}(u)\ddot{z}(u) - \dot{y}(u)\ddot{x}(u)\ddot{z}(u) \\ &+ \dot{y}(u)\ddot{x}(u)\ddot{z}(u) + \dot{z}(u)\ddot{x}(u)\ddot{y}(u) - \dot{z}(u)\ddot{x}(u)\ddot{y}(u) \\ &= \sum a_i(\sigma)\phi_i(u,\sigma)\sum b_i'(\sigma)\phi_i(u,\sigma)\sum c_i''(\sigma)\phi_i(u,\sigma) \\ &- \sum a_i(\sigma)\phi_i(u,\sigma)\sum a_i'(\sigma)\phi_i(u,\sigma)\sum c_i''(\sigma)\phi_i(u,\sigma) \\ &+ \sum b_i(\sigma)\phi_i(u,\sigma)\sum a_i'(\sigma)\phi_i(u,\sigma)\sum c_i'(\sigma)\phi_i(u,\sigma) \\ &+ \sum c_i(\sigma)\phi_i(u,\sigma)\sum a_i''(\sigma)\phi_i(u,\sigma)\sum b_i''(\sigma)\phi_i(u,\sigma) \\ &- \sum c_i(\sigma)\phi_i(u,\sigma)\sum a_i''(\sigma)\phi_i(u,\sigma)\sum b_i''(\sigma)\phi_i(u,\sigma) \\ &- \sum \sum \sum a_i(\sigma)b_j'(\sigma)c_k'(\sigma)\phi_i(u,\sigma)\phi_j(u,\sigma)\phi_k(u,\sigma) \\ &- \sum \sum \sum b_i(\sigma)a_j'(\sigma)c_k'(\sigma)\phi_i(u,\sigma)\phi_j(u,\sigma)\phi_k(u,\sigma) \end{aligned}$$

$$\begin{split} &+\sum\sum \sum b_i(\sigma)a_j''(\sigma)c_k'(\sigma)\phi_i(u,\sigma)\phi_j(u,\sigma)\phi_k(u,\sigma) \\ &+\sum\sum \sum c_i(\sigma)a_j'(\sigma)b_k''(\sigma)\phi_i(u,\sigma)\phi_j(u,\sigma)\phi_k(u,\sigma) \\ &-\sum\sum \sum c_i(\sigma)a_j''(\sigma)b_k'(\sigma)\phi_i(u,\sigma)\phi_j(u,\sigma)\phi_k(u,\sigma) \\ &=\sum\sum \sum (a_i(\sigma)b_j'(\sigma)c_k''(\sigma) + b_i(\sigma)a_j''(\sigma)c_k'(\sigma) + c_i(\sigma)a_j'(\sigma)b_k''(\sigma) \\ &-a_i(\sigma)b_j''(\sigma)c_k'(\sigma) - b_i(\sigma)a_j'(\sigma)c_k''(\sigma) - c_i(\sigma)a_j''(\sigma)b_k'(\sigma)) \\ &\phi_i(u,\sigma)\phi_j(u,\sigma)\phi_k(u,\sigma). \end{split}$$

Using
$$(B.5)$$
 and $(B.6)$ we obtain

It follows that if the triples $a_i(\sigma)b_j(\sigma)c_k(\sigma)$ in the equation above are all known, the function $\beta(u) = \tau(u)\kappa^2(u)$ can be reconstructed.

iii. Combining information from more than one contours

To solve the system of equations obtained in section i, we need to obtain additional equations from other points of the torsion scale space image and relate them to the equations obtained from the first point. Suppose additional equations are obtained in the moments of functions $e^{-\omega^2 t} e^{i\omega u'}(i\omega)\tilde{x}(\omega)$, $e^{-\omega^2 t} e^{i\omega u'}(i\omega)\tilde{y}(\omega)$ and $e^{-\omega^2 t} e^{i\omega u'}(i\omega)\tilde{z}(\omega)$ at point (u', t_0) . We have

$$\dot{x}(u+u') = \int e^{i\omega u} e^{i\omega u'}(i\omega) \tilde{x}(\omega) d\omega = \sum d_k \phi_k(u)$$
$$\dot{y}(u+u') = \int e^{i\omega u} e^{i\omega u'}(i\omega) \tilde{y}(\omega) d\omega = \sum e_k \phi_k(u)$$

and

$$\dot{z}(u+u') = \int e^{i\omega u} e^{i\omega u'}(i\omega) \tilde{z}(\omega) d\omega = \sum f_k \phi_k(u)$$

Now observe that

$$\sum d_k \phi_k(u) = \sum a_k \phi_k(u+u')$$
$$\sum e_k \phi_k(u) = \sum b_k \phi_k(u+u')$$

and

$$\sum f_k \phi_k(u) = \sum c_k \phi_k$$

That is, $\phi_k(u+u')$ can be expressed as a linear combination of $\phi_j(u)$ with $j \le k$ as has been shown in [Yuille and Poggio 1983].

iv. Reconstructing the function $\tau(u) \kappa^2(u)$

It was shown in section i that seven points from seven torsion scale space contours give us 6n+9 cubic equations in the first 2n+3 moments of each of the functions $f(\omega)$, $f'(\omega)$ and $f''(\omega)$. Section iii showed that the moments of order k of any function at u+u' can be expressed as a linear combination of the moments of order less than or equal to k of that function at u. Therefore we obtain a system of homogeneous cubic equations in the first 6n+9 coefficients of functions $\dot{x}(u)$, $\dot{y}(u)$ and $\dot{z}(u)$ using seven points from the torsion scale space image of Γ (Note that only three equations from the seventh point need be used). That system has at least one solution since the moments of order higher than 2n+2 of $f(\omega)$, $f'(\omega)$ and $f''(\omega)$ are assumed to be zero. However, the solution obtained from a cubic system of equations is in general not unique.

Equations (B.3) and (B.4) can be converted into homogeneous linear equations by assuming that each moment-triple appearing in those equations is a new variable. Tables B.1-B.7 show the moment-triples in equations (B.3) and (B.4). The + signs designate the moment-triples in equation (B.3) and the + and the x signs together designate the moment-triples in equation (B.4). Each table shows those moment-triples which share the same $M_{k}^{"}$, $0 \le k \le 6$.

_	M_0'	M_1'	M_2'	M'_3	M'_4	M_5'	M'_6
M_0							
M_1				+	+	x	x
M_2				+		x	
M_3		+	+		x		
M_4		+		x			
M_5		x	x				
M_6		x					

Table B.1. Moment-triples with M_0''

	M_0'	M_1'	M_2'	M'_3	M_4'	M_5'	M_6'
M_0				+	+	x	x
M_1							
M_2				x	х		
M_3	+		x				
M_4	+		x				
M_5	x						
M_6	x						

Table B.2. Moment-triples with M_1''

	M_0'	M_1'	M_2'	M'_3	M'_4	M_5'	M_6'
M_0				+		x	
M_1				x	x		
M_2 M_2	+	x					
M_4		x					
M_5	x						
M_6							

Table	B.3.	Moment-triples with	M''
		and and and an and a state	/

	M_0'	M_1'	M_2'	M'_3	M'_4	M_5'	M_6'
M_0		+		x			
M_1	+		x				
M_2		х					
M_3	x						
M_4							
M_5							
M_6							

Table B.5. Moment-triples with M_4''

 M_0' $M'_2 M'_3 M'_4 M'_5 M'_6$ M M_0 + +x M_1 + х M_2 + х M_3 M_4 х M_5 M_6

Table B.4. Moment-triples with M''_3

	M_0'	M_1'	M_2'	M'_3	M'_4	M_5'	M_6'
M_0		x	x				
M_1	x						
M_2	x						
M_3							
M_4							
M_5							
M_6							

Table B.6. Moment-triples with M_5''

	M_0'	M_1'	M_2'	M_3'	M_4'	M_5'	M_6'
M_0		x					
M_1	x						
M_2							
M_3							
M_4							
M_5							
M_6							

Table B.7. Moment-triples with M_6''

Note that all other moment-triples in tables B.1-B.7 can be computed from the existing ones using the following relationships:

$$M_{i}M_{j}'M_{k}'' = \frac{M_{i}M_{j-1}'M_{k}'' \cdot M_{i+1}M_{j}'M_{k}''}{M_{i+1}M_{j-1}'M_{k}''} = \frac{M_{i}M_{j-1}'M_{k}'' \cdot M_{i-1}M_{j}'M_{k}''}{M_{i-1}M_{j-1}'M_{k}''}$$

$$= \frac{M_{i+1}M'_{j}M''_{k} \cdot M_{i}M'_{j+1}M''_{k}}{M_{i-1}M'_{j+1}M''_{k}} = \frac{M_{i+1}M'_{j}M''_{k} \cdot M_{i}M'_{j+1}M''_{k}}{M_{i+1}M'_{j+1}M''_{k}}$$

$$= \frac{M_{i}M'_{j-1}M''_{k} \cdot M_{i}M'_{j}M''_{k-1}}{M_{i}M'_{j-1}M''_{k-1}} = \frac{M_{i}M'_{j+1}M''_{k} \cdot M_{i}M'_{j}M''_{k-1}}{M_{i}M'_{j+1}M''_{k-1}}$$

$$= \frac{M_{i}M'_{j-1}M''_{k} \cdot M_{i}M'_{j}M''_{k+1}}{M_{i}M'_{j-1}M''_{k+1}} = \frac{M_{i}M'_{j}M''_{k+1} \cdot M_{i}M'_{j+1}M''_{k}}{M_{i}M'_{j+1}M''_{k+1}}$$

$$= \frac{M_{i}M'_{j}M''_{k-1} \cdot M_{i-1}M'_{j}M''_{k}}{M_{i-1}M'_{j}M''_{k-1}} = \frac{M_{i-1}M'_{j}M''_{k} \cdot M_{i}M'_{j}M''_{k+1}}{M_{i-1}M'_{j}M''_{k-1}}$$

$$= \frac{M_{i}M'_{j}M''_{k+1} \cdot M_{i+1}M'_{j}M''_{k}}{M_{i+1}M'_{j}M''_{k}} = \frac{M_{i+1}M'_{j}M''_{k} \cdot M_{i}M'_{j}M''_{k-1}}{M_{i+1}M'_{j}M''_{k-1}}.$$

Again we proceed to compute the first *n* derivatives at point (u_0, t_0) on one of the torsion zero-crossing contours. We now obtain n+1 homogeneous linear equations in some of the moment-triples $M_i M'_j M''_k$ by assuming that each moment-triple is a new variable.

Since this system is in terms of the first 2n+3 moments of functions $f(\omega)$, $f'(\omega)$ and $f''(\omega)$, it will contain $O(n^3)$ moment-triples. Therefore additional equations are required to constrain the system. To obtain those equations, we proceed as follows:

Assume that moments of order higher than 2n+2 are zero. Compute derivatives of order higher than n at (u_0, t_0) but set moments of order higher than 2n+2 to zero in the resulting equations. If a sufficient number of derivatives are computed at (u_0, t_0) , the number of equations obtained will be equal to the number of moment-triples and our linear system will be constrained.

It follows from an assumption of generality that the system will have a unique zero eigenvector and therefore a unique solution modulo scaling. Once the moment-triples in the system are known, all other moment-triples can be computed from the known ones using the relationships given above. Since all the moment-triples $M_i M'_j M''_k$ together determine function of $\beta(u)$, it follows that function $\beta(u)$ can be determined modulo constant scaling.

Yuille and Poggio [1983] have shown that a 1-D signal can be reconstructed using two points from its scale space image. Note that our result implies that only one point is sufficient for the reconstruction of that signal.

The theorem has now been proven for the regular torsion scale space image. To prove the same result about the resampled torsion scale space image, recall that derivatives at one point (at any scale) on any torsion zero-crossing 153

contour in the torsion scale space of Γ were computed and it was shown that the resulting equations can be solved for the coefficients of expansion of the function $\beta(u)$ in functions related to the Hermite polynomials.

As before, we choose a point on a zero-crossing contour at any scale of the resampled curvature scale space image of Γ and compute the necessary derivatives. The value of σ in the resulting equations is then set to zero. Consequently, the arc length evolved curve Γ_{σ} , where σ corresponds to the scale at which the derivatives were computed, is reconstructed modulus uniform scaling, rotation and translation.

The next step is to recover the original curve Γ modulo function $\beta(u)$. This is done by applying *reverse* arc length evolution to Γ_{σ} . Let the arc length evolved curve Γ_{σ} be defined by:

$$\Gamma_{\sigma} = \{ (X(W,\sigma), Y(W,\sigma), Z(W,\sigma) | W \in [0,1] \}$$

A reverse arc length evolved curve Γ is defined by:

$$\Gamma = \{(x(w), y(w), z(w)) | w \in [0,1]\}$$

where

$$\begin{aligned} x(w) &= X(w,\sigma) \circledast D_N(w,\sigma) \\ y(w) &= Y(w,\sigma) \circledast D_N(w,\sigma) \end{aligned}$$

and

$$z(w) = Z(w,\sigma) \circledast D_N(w,\sigma)$$

where D_N is a deblurring operator defined in [Hummel et al. 1987] and

$$w(W,t) = \int_0^t \int_0^w \kappa^2(w,t) dw dt.$$

where $t = \sigma^2/2$. As a result, Γ is recovered modulo function $\beta(u)$.

To prove the same result about the renormalized torsion scale space image, evolved curve Γ_{σ} is again reconstructed, then each of its coordinate functions is deblurred by convolving it with the deblurring operator D_N . Once again, Γ is recovered modulo function $\beta(u)$.

Proof of theorem 4.4.7: Since by assumption all evolved and arc length evolved curves Γ_{σ} are in C_3 , the conditions of the implicit function theorem are satisfied on contours $\tau(u,\sigma)$, $\tau(w,\sigma)$ and $\tau(W,\sigma) = 0$ in the regular, renormalized and resampled torsion scale space images of Γ . Since the proofs are identical, the theorem will be proven here for the regular torsion scale space image.

The torsion of each evolved curve $\Gamma_{\sigma} = (x(u,\sigma), y(u,\sigma), z(u,\sigma))$ is given by:

$$\tau(u,\sigma) = \frac{\ddot{z}\,\dot{x}\,\ddot{y} - \ddot{z}\,\dot{y}\ddot{x} + \ddot{y}\,\dot{z}\ddot{x} - \ddot{y}\,\dot{x}\ddot{z} + \ddot{x}\,\dot{y}\ddot{z} - \ddot{x}\,\dot{z}\ddot{y}}{\left(\dot{y}\,\ddot{z} - \dot{z}\,\ddot{y}\right)^2 + \left(\dot{z}\,\ddot{x} - \dot{x}\,\ddot{z}\right)^2 + \left(\dot{x}\,\ddot{y} - \dot{y}\,\ddot{x}\right)^2}$$

where . represents derivative with respect to u. On any contour in the torsion scale space image of Γ :

$$\tau(u,\sigma)=0.$$

It follows from the assumption that all Γ_{σ} are in C_3 that:

$$\beta(u,t) = \overline{z}\,\overline{x}\,\overline{y} - \overline{z}\,\overline{y}\,\overline{x} + \overline{y}\,\overline{z}\,\overline{x} - \overline{y}\,\overline{x}\,\overline{z} + \overline{x}\,\overline{y}\,\overline{z} - \overline{x}\,\overline{z}\,\overline{y}$$

where again . represents derivative with respect to u and $t = \sigma^2/2$. The functions x(u,t), y(u,t) and z(u,t) satisfy the heat equation:

$$x_{uu}(u,t) = x_t(u,t)$$
$$y_{uu}(u,t) = y_t(u,t)$$
$$z_{uu}(u,t) = z_t(u,t).$$

Since evolved curves Γ_{σ} are all in C_3 , the conditions of the implicit function theorem are satisfied on contours $\beta(u,t) = 0$:

$$\dot{t} = t(u)$$
$$\dot{t}(u) = \frac{dt}{du} = \frac{-\beta_u}{\beta_t}$$

The theorem will be proven if it is shown that if t(u) = 0 at any point on a torsion zero-crossing contour, then t(u) < 0 at that point. Observe that t(u) = 0 if and only if $\beta_u(u,t) = 0$. It follows that at a point where t(u) = 0:

$$\ddot{t}(u) = \frac{d}{du} \left(\frac{-\beta_u}{\beta_t} \right) = \frac{\partial}{\partial u} \left(\frac{-\beta_u}{\beta_t} \right) + \frac{\partial}{\partial t} \left(\frac{-\beta_u}{\beta_t} \right) \frac{dt}{du} = \frac{-\beta_{uu}}{\beta_t}.$$

Therefore it must be shown that if $\beta(u,t) = \beta_u(u,t) = 0$ then $\beta_{uu}/\beta_t > 0$.

We will now derive explicit expressions for β_{uu} and β_t . Differentiating the expression for $\beta(u,t)$ with respect to u and simplifying yields:

 $\beta_u(u,t) = z_{tt}x_uy_t - z_{tt}y_ux_t + y_{tt}z_ux_t - y_{tt}x_uz_t + x_{tt}y_uz_t - x_{tt}z_uy_t.$

Differentiating the expression for β_u with respect to u and simplifying yields:

$$\beta_{uu} = \Psi_1 + \Psi_2$$

where

$$\Psi_1 = z_{ttu}x_uy_t - z_{ttu}y_ux_t + y_{ttu}z_ux_t - y_{ttu}x_uz_t + x_{ttu}y_uz_t - x_{ttu}z_uy_t$$

and

$$\Psi_2 = y_{tu}z_{tt}x_u - x_{tu}z_{tt}y_u + x_{tu}y_{tt}z_u - z_{tu}y_{tt}x_u + z_{tu}x_{tt}y_u - y_{tu}x_{tt}z_u.$$

Differentiating the expression for $\beta(u,t)$ with respect to t and simplifying yields:

 $\beta_t = \Psi_1 - \Psi_2.$

Let P be a point on an evolved curve Γ_{σ} where $\beta(u,t) = \beta_u(u,t) = 0$. The coordinate functions of Γ_{σ} can be locally approximated at P using polynomial functions. Furthermore, assume that u=0 at point P. It follows that (u^m, u^n, u^p) is a local approximation to Γ_{σ} around P where m, n and p are the lowest non-zero powers of the polynomials approximating functions x(u,t), y(u,t) and z(u,t) respectively. Also assume without loss of generality that p > n > m. Observe that

$$\begin{split} x_u &= m \, u^{m-1} \\ x_t &= m(m-1) \, u^{m-2} \\ x_{tu} &= m(m-1)(m-2) \, u^{m-3} \\ x_{tt} &= m(m-1)(m-2)(m-3) \, u^{m-4} \\ x_{ttu} &= m(m-1)(m-2)(m-3)(m-4) \, u^{m-5} \end{split}$$

and that

$$y_u = n u^{n-1}$$

$$y_t = n(n-1) u^{n-2}$$

$$y_{tu} = n(n-1)(n-2) u^{n-3}$$

$$y_{tt} = n(n-1)(n-2)(n-3) u^{n-4}$$

$$y_{ttu} = n(n-1)(n-2)(n-3)(n-4) u^{n-5}$$

and that

$$\begin{split} z_u &= p \, u^{p-1} \\ z_t &= p(p-1) \, u^{p-2} \\ z_{tu} &= p(p-1)(p-2) \, u^{p-3} \\ z_{tt} &= p(p-1)(p-2)(p-3) \, u^{p-4} \end{split}$$

$$z_{ttu} = p(p-1)(p-2)(p-3)(p-4) u^{p-5}.$$

It now follows that at point P:

$$\frac{\beta_{uu}}{\beta_t} = \frac{\Xi_1 u^{m+n+p-8} + \Xi_2 u^{m+n+p-8}}{\Xi_1 u^{m+n+p-8} - \Xi_2 u^{m+n+p-8}} = \frac{\Xi_1 + \Xi_2}{\Xi_1 - \Xi_2}$$

where

$$\begin{split} \Xi_1 &= (p-1)(p-2)(p-3)(p-4)(n-1) - (p-1)(p-2)(p-3)(p-4)(m-1) \\ &+ (n-1)(n-2)(n-3)(n-4)(m-1) - (n-1)(n-2)(n-3)(n-4)(p-1) \\ &+ (m-1)(m-2)(m-3)(m-4)(p-1) - (m-1)(m-2)(m-3)(m-4)(n-1) \end{split}$$

and

$$\begin{split} \Xi_2 &= (p-1)(p-2)(p-3)(n-1)(n-2) - (p-1)(p-2)(p-3)(m-1)(m-2) \\ &+ (n-1)(n-2)(n-3)(m-1)(m-2) - (n-1)(n-2)(n-3)(p-1)(p-2) \\ &+ (m-1)(m-2)(m-3)(p-1)(p-2) - (m-1)(m-2)(m-3)(n-1)(n-2). \end{split}$$

It can be shown that:

 $\Xi_1 = (p-n)(p-m)(n-m)(p^2 + (n+m-10)p + n^2 + m^2 + mn - 10n - 10m + 35)$ and that:

$$\Xi_2 = (p-n)(p-m)(n-m)(p(n+m) - 3(n+m) - 3p + mn + 7).$$

It can now be concluded that to prove the theorem, it must be shown that:

 $|\Xi_1| \ge |\Xi_2|$

or

 $|\Delta_1| \ge |\Delta_2|$

where

$$\Delta_1 = p^2 + n^2 + m^2 + np + mp + mn - 10p - 10n - 10m + 35$$

and

$$\Delta_2 = np + mp + mn - 3p - 3n - 3m + 7.$$

We shall now use a case analysis to prove that the inequality above holds for all valid triples of values of m, n and p. The analysis below shows that only triples of values which satisfy the inequality:

p > m+n

are valid:

lows that in a neighborhood of P, torsion is given by:

$$\tau(u) = \frac{\lambda u^{p+n+m-6}}{\lambda_1 u^{2(p+n-3)} + \lambda_2 u^{2(p+m-3)} + \lambda_3 u^{2(m+n-3)}}$$

where λ , λ_1 , λ_2 and λ_3 are constants. The expression above is ambiguous at u=0. To resolve the ambiguity, l'Hopital's rule can be applied repeatedly. Since both the numerator and the denominator are polynomials, to have $\tau(u) = 0$ at u=0, repeated application of l'Hopital's rule should result in:

$$\lim_{u\to 0} \tau(u) = \frac{\psi \, u^{i}}{\xi + f(u)}$$

where ψ and ξ are constants, i > 0 and f(u) = 0 at u = 0. This can only happen when one of the following three conditions are met:

a.
$$p+n+m-6 > 2(p+n-3)$$

b. $p+n+m-6 > 2(p+m-3)$
c. $p+n+m-6 > 2(m+n-3)$.

Conditions **a** and **b** are not possible since they violate the assumption that p > n > m. However, condition **c** is possible. It follows from this condition that:

$$p > m+n$$
.

We can now proceed with the case analysis. All triples of values for m, n and p in which m is even correspond to cusp points which are excluded by the assumption that all evolved curves Γ_{σ} are in C_3 . Therefore we will consider only odd values of m.

Case 1. Suppose $m \ge 7$. Recall that p > n > m. It is easily seen that both Δ_1 and Δ_2 are positive. So the absolute value signs can be dropped and the inequality:

$$\Delta_1 \geq \Delta_2$$

can be simplified. As a result, we must now show that the following inequality holds:

$$p^2 + n^2 + m^2 \ge 7p + 7n + 7m - 28.$$

Note that $m^2 \ge 7m$, $n^2 > 7n$ and $p^2 > 7p$. It follows that the inequality does hold.

Case 2. Suppose m=5. Again, it can be seen that both Δ_1 and Δ_2 are positive. We must again show that: $p^2 + n^2 + m^2 \ge 7p + 7n + 7m - 28.$

Substitute m=5 in the above inequality. We now have:

 $p^2 + n^2 \ge 7p + 7n - 18.$

Since $n \ge 6$, $n^2 \ge 7n-18$ and since p > 11, $p^2 > 7p$. Hence the inequality again holds.

Case 3. Suppose m=3. Substitute this value for m in Δ_1 . As a result, Δ_1 simplifies to:

$$p^2 + n^2 + np - 7p - 7n + 14$$
.

Note that $n \ge 4$ and $p \ge 8$. So $p^2 > 7p$. Hence to show that Δ_1 is positive, it is sufficient to show that:

$$n^2 + np - 7n + 14 > 0.$$

Since $p \ge 8$, $np \ge 8n$. Therefore:

 $n^{2}+np-7n+14 \ge n^{2}+8n-7n+14 = n^{2}+n+14 > 0.$

Now substitute m=3 in Δ_2 . As a result Δ_2 simplifies to:

np+3p+3n-3p-3n-9+7=np-2

which is always positive. Therefore we must again show that:

 $\dot{p}^2 + n^2 + m^2 \ge 7p + 7n + 7m - 28.$

Substituting m=3 in the above inequality yields:

$$p^2 + n^2 \ge 7p + 7n - 16.$$

Since $p \ge 8$, $p^2 > 7p$ and it is sufficient to show that:

 $n^2 \ge 7n-16.$

It is easily seen that this inequality is satisfied for $n \ge 4$.

Case 4. Suppose m=1. Substituting this value in Δ_1 simplifies it to:

 $p^2 + n^2 + np - 9p - 9n + 26.$

Since $p \ge 4$, $p^2 - 9p \ge -20$. Hence to show that Δ_1 is non-negative, it is sufficient to show that:

$$n^2 + np - 9n + 6 \geq 0$$

Again since $p \ge 4$:

$$n^{2}+np-9n+6 \ge n^{2}+4n-9n+6 = n^{2}-5n+6$$

which is non-negative for $n \ge 2$. Now substitute for m=1 in Δ_2 to obtain:

$$pn-2p-2n+4 = p(n-2)-2n+4.$$

Since $p \ge 4$

$$p(n-2)-2n+4 \ge 4(n-2)-2n+4 = 4n-8-2n+4 = 2n-4$$

which is non-negative since $n \ge 2$. So Δ_2 is also non-negative. Therefore we must again show that:

$$p^2 + n^2 + m^2 \ge 7p + 7n + 7m - 28$$

Substitute for m=1 in the above expression to obtain:

 $p^2 + n^2 \ge 7p + 7n - 22$

which is equivalent to:

$$(p^2-7p) + (n^2-7n)+22 \ge 0.$$

If n=2, then $n^2-7n = -10$ and $p \ge 4$. It follows from $p \ge 4$ that $p^2-7p \ge -12$. As a result, the inequality above is satisfied. If n>2, then $n^2-7n \ge -12$ and $p\ge 5$. It follows from $p\ge 5$ that $p^2-7p \ge -10$. Therefore, the inequality above is again satisfied.

This completes the case analysis. We have shown that the inequality:

 $|\Delta_1| \ge |\Delta_2|$

and therefore the inequality:

 $|\Xi_1| \ge |\Xi_2|$

is satisfied for all valid triples of values of m, n and p. Therefore β_{uu}/β_t is always positive. Hence all extrema of contours in all torsion scale space images of Γ are maxima.

Proof of theorem 4.4.8: It will be shown that this theorem holds for an arbitrary parametrization of Γ_{σ} . Therefore it must also be true of arc length parametrization or close approximations.

Let $(X(u,\sigma), Y(u,\sigma), Z(u,\sigma))$ be an arbitrary parametrization of Γ with a cusp point at u_0 . It has been shown by [Hummel *et al.* 1987] that the class of polynomial functions is closed under convolution with a Gaussian. Therefore $X(u,\sigma)$, $Y(u,\sigma)$ and $Z(u,\sigma)$ are also polynomial functions:

 $X(u,\sigma) = a_0 + a_1u + a_2u^2 + a_3u^3 + \cdots$ $Y(u,\sigma) = b_0 + b_1u + b_2u^2 + b_3u^3 + \cdots$ $Z(u,\sigma) = c_0 + c_1u + c_2u^2 + c_3u^3 + \cdots$

Let Γ_{σ} go through the origin of the coordinate system at $u_0=0$. It follows that

 $a_0=b_0=c_0=0$. Every cusp point is also a singular point of the curve. Therefore Γ_{σ} has a singularity at u_0 . Now observe that

$$X_{u}(u,\sigma) = a_{1} + 2a_{2}u + 3a_{3}u^{2} + 4a_{4}u^{3} + \cdots$$
$$Y_{u}(u,\sigma) = b_{1} + 2b_{2}u + 3b_{3}u^{2} + 4b_{4}u^{3} + \cdots$$
$$Z_{u}(u,\sigma) = c_{1} + 2c_{2}u + 3c_{3}u^{2} + 4c_{4}u^{3} + \cdots$$

 $X_u(u,\sigma)$, $Y_u(u,\sigma)$ and $Z_u(u,\sigma)$ are zero at a singular point. It follows that $a_1=b_1=c_1=0$. As a result, all powers of u in $X(u,\sigma)$, $Y(u,\sigma)$ and $Z(u,\sigma)$ are larger than one.

The following case analysis identifies the cases in which the singular point at u_0 is also a cusp point. Since Γ_{σ} is examined in a small neighborhood of point $u_0=0$, it can be approximated using the lowest degree terms in $X(u,\sigma)$, $Y(u,\sigma)$ and $Z(u,\sigma)$:

$$\Gamma_{\sigma} = (u^m, u^n, u^p)$$

Assume without loss of generality that p > n > m. Observe that

$$\mathbf{r}_{\mathbf{u}}(u,\sigma) = (X_{\mathbf{u}}(u,\sigma), Y_{\mathbf{u}}(u,\sigma), Z_{\mathbf{u}}(u,\sigma)) = (m u^{m-1}, n u^{n-1}, p u^{p-1})$$

Therefore

$$\mathbf{r}_{\mathbf{u}}(\epsilon,\sigma) = (m \, \epsilon^{m-1}, n \, \epsilon^{n-1}, p \, \epsilon^{p-1}) = \epsilon^{m-1}(m, n \, \epsilon^{n-m}, p \, \epsilon^{p-m})$$

and

$$\mathbf{r}_{\mathbf{u}}(-\epsilon,\sigma) = (m(-\epsilon)^{m-1}, n(-\epsilon)^{n-1}, p(-\epsilon)^{p-1})$$

Since m, n and p can be odd or even, the singular point at u_0 must be analyzed in each of eight possible cases:

1. m, n and p are even.

m-1, n-1 and p-1 are odd. So

$$\mathbf{r}_{u}(-\epsilon,\sigma) = (-m\,\epsilon^{m-1}, -n\,\epsilon^{n-1}, -p\,\epsilon^{p-1}) = -\epsilon^{m-1}(m,n\,\epsilon^{n-m}, p\,\epsilon^{p-m})$$

Comparing $\mathbf{r}_{u}(\epsilon,\sigma)$ to $\mathbf{r}_{u}(-\epsilon,\sigma)$ shows that an infinitesimal change in parameter u in a neighborhood of the singular point results in a large change in the direction of the tangent vector. Therefore this singularity is a cusp point.

2. m and n are even, p is odd.

m-1 and n-1 are odd and p-1 is even. Therefore

$$\mathbf{r}_{\mathbf{u}}(-\epsilon,\sigma) = (-m\,\epsilon^{m-1}, -n\,\epsilon^{n-1}, p\,\epsilon^{p-1}) = \epsilon^{m-1}(-m, -n\,\epsilon^{n-m}, p\,\epsilon^{p-m})$$

A comparison of $\mathbf{r}_u(\epsilon,\sigma)$ and $\mathbf{r}_u(-\epsilon,\sigma)$ again shows that an infinitesimal change in

u causes a large change in the tangent direction. Hence this singular point is also a cusp point.

3. m is even, n is odd and p is even.

m-1 is odd, n-1 is even and p-1 is odd. Hence

 $\mathbf{r}_{\mathbf{u}}(-\epsilon,\sigma) = (-m\,\epsilon^{m-1},n\,\epsilon^{n-1},-p\,\epsilon^{p-1}) = \epsilon^{m-1}(-m,n\,\epsilon^{n-m},-p\,\epsilon^{p-m})$

An infinitesimal change in u again results in a large change in the tangent direction. This singularity is a cusp point as well.

4. m is even, n and p are odd.

m-1 is odd, n-1 and p-1 are even. So

 $\mathbf{r}_{u}(-\epsilon,\sigma) = (-m\,\epsilon^{m-1}, n\,\epsilon^{n-1}, p\,\epsilon^{p-1}) = \epsilon^{m-1}(-m, n\,\epsilon^{n-m}, p\,\epsilon^{p-m})$

A large change in the tangent direction is caused by an infinitesimal change in u. Therefore this singularity also corresponds to a cusp point.

5. m is odd, n and p are even.

m-1 is even, n-1 and p-1 are odd. Therefore

 $\mathbf{r}_{\mathbf{u}}(-\epsilon,\sigma) = (m \epsilon^{m-1}, -n \epsilon^{n-1}, -p \epsilon^{p-1}) = \epsilon^{m-1}(m, -n \epsilon^{n-m}, -p \epsilon^{p-m})$

A comparison of $\mathbf{r}_u(\epsilon,\sigma)$ and $\mathbf{r}_u(-\epsilon,\sigma)$ now shows that an infinitesimal change in u in the neighborhood of the singular point also results in an infinitesimal change in the tangent direction. Hence, this singular point is *not* a cusp point.

6. m is odd, n is even, p is odd.

m-1 is even, n-1 is odd and p-1 is even. So

 $\mathbf{r}_{\mathbf{u}}(-\epsilon,\sigma) = (m \, \epsilon^{m-1}, -n \, \epsilon^{n-1}, p \, \epsilon^{p-1}) = \epsilon^{m-1}(m, -n \, \epsilon^{n-m}, p \, \epsilon^{p-m})$

The tangent direction changes only infinitesimally in the neighborhood of the singular point. Therefore this singularity is *not* a cusp point either.

7. m and n are odd, p is even.

m-1 and n-1 are even and p-1 is odd. Hence

$$\mathbf{r}_{u}(-\epsilon,\sigma) = (m \,\epsilon^{m-1}, n \,\epsilon^{n-1}, -p \,\epsilon^{p-1}) = \epsilon^{m-1}(m, n \,\epsilon^{n-m}, -p \,\epsilon^{p-m})$$

This singularity is again *not* a cusp point since the tangent direction changes only infinitesimally in its neighborhood.

8. m, n and p are odd.

m-1, n-1 and p-1 are even. Therefore

$$\mathbf{r}_{u}(-\epsilon,\sigma) = (m \epsilon^{m-1}, n \epsilon^{n-1}, p \epsilon^{p-1}) = \epsilon^{m-1}(m, n \epsilon^{n-m}, p \epsilon^{p-m})$$

The last singular point is *not* a cusp point either since the changes in the tangent direction are again infinitesimal.

It follows from the case analysis above that only the singular points in cases 1-4 are cusp points. We next derive analytical expressions for the curve $\Gamma_{\sigma-\delta}$ so that it can be analyzed in a small neighborhood of the cusp point.

To deblur function $f(u) = u^k$, a rescaled version of that function is convolved with the function $\frac{2}{\sqrt{\pi}}e^{-u^2}(1-u^2)$. This function is an approximation to the deblurring operator derived in [Hummel *et al.* 1987] and is good for small amounts of deblurring. The convolution can be expressed as

$$(D_t f)(u) = \int_{-\infty}^{\infty} \frac{2}{\sqrt{\pi}} e^{-v^2} (1 - v^2) f(u + 2v\sqrt{t}) dv$$

$$(D_t f)(u) = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-v^2} (1-v^2)(u+2v\sqrt{t})^k dv$$

where t is the scale factor and controls the amount of deblurring. Solving the integral above yields

$$(D_t f)(u) = \sum_{\substack{p=0\\(p \text{ even})}}^{k} 1.3.5 \cdots (p-1) \frac{(2t)^{p/2} k(k-1) \cdots (k-p+1)}{p!} (1-p) u^{k-p} (B.7)$$

 $\Gamma_{\sigma-\delta}$ can now be analyzed in each of the cases 1-4:

Case 1: Γ_{σ} is approximated by (u^m, u^n, u^p) where m, n and p are even.

To obtain analytical expressions for $\Gamma_{\sigma-\delta}$, we deblur each of its coordinate functions:

$$(D_t x)(u) = u^m - c_1 t u^{m-2} - \cdots - c_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u^2 - c_{\frac{m}{2}} t^{\frac{m}{2}}$$
$$(D_t y)(u) = u^n - c_1' t u^{n-2} - \cdots - c_{\frac{n-2}{2}} t^{\frac{n-2}{2}} u^2 - c_{\frac{m}{2}} t^{\frac{n}{2}}$$

$$(D_t z)(u) = u^p - c_1'' t u^{p-2} - \cdots - c_{\frac{p-2}{2}}'' t^{\frac{p-2}{2}} u^2 - c_{\frac{p}{2}}'' t^{\frac{p}{2}}$$

Note that all powers of u are even and all constants are positive. It follows that

$$(D_t \dot{x})(u) = mu^{m-1} - (m-2)c_1tu^{m-3} - \cdots - 2c_{\frac{m-2}{2}}t^{\frac{m-2}{2}}u$$
$$(D_t \dot{y})(u) = nu^{n-1} - (n-2)c_1tu^{n-3} - \cdots - 2c_{\frac{n-2}{2}}t^{\frac{n-2}{2}}u$$
$$(D_t \dot{z})(u) = pu^{p-1} - (p-2)c_1''tu^{p-3} - \cdots - 2c_{\frac{p-2}{2}}t^{\frac{p-2}{2}}u$$

contain only odd powers of u and $(D_t \dot{\mathbf{r}})(\epsilon) = -(D_t \dot{\mathbf{r}})(-\epsilon)$. Hence there is also a cusp point on $\Gamma_{\sigma-\delta}$ at u_0 . Since that cusp point is of the same kind as the cusp point on Γ_{σ} , it follows that a cusp point must also exist on Γ . This is a contradiction of the assumption that Γ is in C_1 . Therefore Γ_{σ} can not have a cusp point of this kind at u_0 .

Case 2: Γ_{σ} is approximated by (u^m, u^n, u^p) where m and n are even and p is odd.

 $\Gamma_{\sigma-\delta}$ is obtained by deblurring each of its coordinate functions:

$$(D_t x)(u) = u^m - c_1 t u^{m-2} - \cdots - c_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u^2 - c_{\frac{m}{2}} t^{\frac{m}{2}}$$
$$(D_t y)(u) = u^n - c_1' t u^{n-2} - \cdots - c_{\frac{m-2}{2}}' t^{\frac{n-2}{2}} u^2 - c_{\frac{m}{2}}' t^{\frac{n}{2}}$$
$$(D_t z)(u) = u^p - c_1'' t u^{p-2} - \cdots - c_{\frac{m-1}{2}}' t^{\frac{p-1}{2}} u$$

Note that $(D_t x)$ and $(D_t y)$ contain even powers of u only, $(D_t z)$ contains odd powers of u only and all constants are positive.

The deblurred curve intersects itself if there are two values of u, u_1 and u_2 , such that

$$\begin{aligned} x(u_1) &= x(u_2) \\ y(u_1) &= y(u_2) \end{aligned}$$

$$z(u_1) = z(u_2)$$

It follows from the first two constraints above that $u_1 = -u_2$. Substituting for u_2 in the third constraint and simplifying yields:

$$u_1^{p} - c_1'' t u_1^{p-2} - \cdots - c_{\frac{p-1}{2}}'' t^{\frac{p-1}{2}} u_1 = 0$$

Now let $t = \delta$ to obtain

$$u_1^{p} - c_1'' \delta u_1^{p-2} - \cdots - c_{\frac{p-1}{2}}'' \delta u_1^{\frac{p-1}{2}} u_1 = 0$$
 (B.8)

The LHS of (B.8) is negative for very small values of u_1 since the first term will be smaller than all other terms, which are negative. As u_1 grows, the first term becomes larger than the sum of all other terms and the LHS of (B.8) becomes positive. Therefore there is a positive value of u_1 at which (B.8) is satisfied. Hence $\Gamma_{\sigma-\delta}$ intersects itself in a neighborhood of u_0 .

Case 3: Γ_{σ} is approximated by (u^m, u^n, u^n) where *m* is even, *n* is odd and *p* is even.

As in the previous case, we obtain analytical expressions for $\Gamma_{\sigma-\delta}$:

$$(D_t x)(u) = u^m - c_1 t u^{m-2} - \cdots - c_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u^2 - c_{\frac{m}{2}} t^{\frac{m}{2}}$$
$$(D_t y)(u) = u^n - c_1' t u^{n-2} - \cdots - c_{\frac{n-1}{2}}' t^{\frac{n-1}{2}} u$$
$$(D_t z)(u) = u^p - c_1'' t u^{p-2} - \cdots - c_{\frac{p-2}{2}}' t^{\frac{p-2}{2}} u^2 - c_{\frac{p}{2}}' t^{\frac{p}{2}}$$

In this case, $(D_t x)$ and $(D_t z)$ contain only even powers of u and $(D_t y)$ contains only odd powers of u. Again, $\Gamma_{\sigma-\delta}$ can be shown to intersect itself if there are two values of u, u_1 and u_2 , such that

$$x(u_1) = x(u_2)$$
$$y(u_1) = y(u_2)$$
$$z(u_1) = z(u_2)$$

It now follows from the first and third constraints above that $u_1 = -u_2$. Substituting for u_2 in the second constraint, letting $t = \delta$ and simplifying yields

$$u_1^n - c_1' \delta u_1^{n-2} - \cdots - c_{\frac{n-1}{2}}' \delta^{\frac{n-1}{2}} u_1 = 0$$
 (B.9)

An argument similar to the one used in the previous case shows that there exists a positive value of u_1 at which (B.9) is satisfied. Therefore $\Gamma_{\sigma-\delta}$ again intersects itself in a neighborhood of u_0 .

Case 4: Γ_{σ} is approximated by (u^m, u^n, u^p) where m is even and n and p are odd.

An analytical expression for $\Gamma_{\sigma-\delta}$ in a neighborhood of u_0 is given by

$$(D_t x)(u) = u^m - c_1 t u^{m-2} - \cdots - c_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u^2 - c_{\frac{m}{2}} t^{\frac{m}{2}}$$
$$(D_t y)(u) = u^n - c_1' t u^{n-2} - \cdots - c_{\frac{n-1}{2}}' t^{\frac{n-1}{2}} u$$
$$(D_t z)(u) = u^p - c_1'' t u^{p-2} - \cdots - c_{\frac{p-1}{2}}' t^{\frac{p-1}{2}} u$$

All powers of u in $(D_t x)$ are even and all powers of u in $(D_t y)$ and $(D_t z)$ are odd. As before, $\Gamma_{\sigma-\delta}$ intersects itself if the three constraints

$$x(u_1) = x(u_2)$$
$$y(u_1) = y(u_2)$$
$$z(u_1) = z(u_2)$$

are satisfied simultaneously. It follows from the first constraint that $u_1 = -u_2$. Now substitute for u_2 in the second and third constraints, let $t = \delta$ and simplify:

$$u_1^{n} - c_1' \delta u_1^{n-2} - \cdots - c_{\frac{n-1}{2}}' \delta^{\frac{n-1}{2}} u_1 = 0$$
 (B.10)

$$u_1^{p} - c_1'' \delta u_1^{p-2} - \cdots - c_{\frac{p-1}{2}}'' \delta^{\frac{p-1}{2}} u_1 = 0$$
 (B.11)

Each of the equations (B.10) and (B.11) is satisfied at a positive value of u_1 but, in general, the same value of u_1 will not satisfy both. It follows that, in this case, $\Gamma_{\sigma-\delta}$ does not intersect itself. However, an argument similar to the ones in the previous two cases shows that the planar curves defined by $(D_t x)$ and $(D_t y)$ and by $(D_t x)$ and $(D_t z)$, that is, the projections of $\Gamma_{\sigma-\delta}$ on the XY and XZ planes respectively, do intersect themselves in a neighborhood of u_0 . This completes the proof of theorem 4.4.8.

Proof of theorem 4.4.9: It will be shown that this theorem holds for an arbitrary parametrization of Γ_{σ} . Therefore it must also be true of arc length parametrization or close approximations.

Let (x(u), y(u), z(u)) be an arbitrary parametrization of Γ_{σ} with a cusp point at u_0 . Using a case analysis similar to the one in the proof of theorem 4.4.8 to characterize all the possible singularities of Γ_{σ} at u_0 , we again conclude that only the singular points in cases 1-4 are cusp points.

We now derive analytical expressions for $\Gamma_{\sigma+\delta}$ so that it can be analyzed in a neighborhood of u_0 . To blur function $f(u) = u^k$, we convolve a rescaled version of that function with the function $\frac{1}{\sqrt{\pi}}e^{-u^2}$, the blurring operator, as follows:

$$F(u) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} e^{-v^2} f(u+2v\sqrt{t}) dv$$

or

$$F(u) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-v^2} (u + 2v\sqrt{t})^k dv$$

where t is the scale factor and controls the amount of blurring. Solving the integral above yields

$$F(u) = \sum_{\substack{p=0\\(p \ even)}}^{k} 1.3.5 \cdots (p-1) \frac{(2t)^{p/2} k(k-1) \cdots (k-p+1)}{p!} u^{k-p}$$
(B.12)

An expression for $\Gamma_{\sigma+\delta}$ in a neighborhood of the cusp point can be obtained by blurring each of its coordinate functions. Furthermore, expressions for $\Gamma_{\sigma-\delta}$ in a neighborhood of the cusp point can be obtained by deblurring each of its coordinate functions according to (B.7).

Each of the cases 1-4 can now be analyzed in turn:

Case 1: Γ_{σ} is approximated by (u^m, u^n, u^p) where m, n and p are even.

An argument similar to the one used in case 1 of theorem 4.4.8 shows that this kind of cusp point can not arise during evolution of Γ .

Case 2: Γ_{σ} is approximated by (u^m, u^n, u^p) where m and n are even and p is odd.

Observe that

$$\begin{aligned} \dot{x}(u) &= mu^{m-1} & \ddot{x}(u) = m(m-1)u^{m-2} & \ddot{x}(u) = m(m-1)(m-2)u^{m-3} \\ \dot{y}(u) &= nu^{n-1} & \ddot{y}(u) = n(n-1)u^{n-2} & \ddot{y}(u) = n(n-1)(n-2)u^{n-3} \\ \dot{z}(u) &= pu^{p-1} & \ddot{z}(u) = p(p-1)u^{p-2} & \ddot{z}(u) = p(p-1)(p-2)u^{p-3} \end{aligned}$$

Torsion on Γ_{σ} is given by:

$$\tau(u) = \frac{\ddot{x}\ddot{x}\ddot{y} - \ddot{z}\ddot{y}\ddot{x} + \ddot{y}\ddot{z}\ddot{x} - \ddot{y}\ddot{x}\ddot{z} + \ddot{x}\ddot{y}\ddot{z} - \ddot{x}\ddot{z}\ddot{y}}{(\ddot{y}\ddot{z} - \dot{z}\ddot{y})^2 + (\ddot{z}\ddot{x} - \dot{x}\ddot{z})^2 + (\dot{x}\ddot{y} - \dot{y}\ddot{x})^2}$$

or

$$\tau(u) = \frac{mnp \, K \, u^{p+n+m-6}}{A+B+C} \tag{B.13}$$

where

$$egin{aligned} B &= ((pm(m-1) - mp(p-1)) \, u^{p+m-3})^2 \ C &= ((mn(n-1) - nm(m-1)) \, u^{m+n-3})^2 \end{aligned}$$

 $A = ((np(p-1) - pn(n-1))u^{p+n-3})^2$

and

$$K = (p-1)(p-2)(n-m) + (n-1)(n-2)(m-p) + (m-1)(m-2)(p-n).$$

At u = 0 (cusp point), τ is undefined. When u is positive or negative, the sign of $\tau(u)$ depends on the sign of K. Observe that

$$\begin{split} K &= (p-1)(p-2)(n-m) + (n-1)(n-2)(m-p) + (m-1)(m-2)(p-n) \\ &= (p^2 - 3p + 2)(n-m) + (n^2 - 3n + 2)(m-p) + (m^2 - 3m + 2)(p-n) \\ &= np^2 - mp^2 - 3pn + 3pm + 2n - 2m + mn^2 - 3mn + 2m - pn^2 \\ &+ 3pn - 2p + pm^2 - 3pm + 2p - nm^2 + 3mn - 2n \\ &= (n-m)p^2 + (m^2 - n^2)p + mn^2 - nm^2 \\ &= (n-m)p^2 + (m+n)(m-n)p + mn(n-m) \\ &= (n-m)(p^2 - (m+n)p + mn) \\ &= (n-m)(p-m)(p-n) \end{split}$$

which is positive because of the assumption that p > n > m. Since p+n+m-6, the power of u in the numerator, is odd, it follows that $\tau(u)$ is positive for positive u and negative for negative u.
We now investigate $\tau(u)$ on $\Gamma_{\sigma+\delta}$. It follows from (B.12) that $\Gamma_{\sigma+\delta}$ is given

$$X(u) = u^{m} + c_{1}tu^{m-2} + \cdots + c_{\frac{m-2}{2}}t^{\frac{m-2}{2}}u^{2} + c_{\frac{m}{2}}t^{\frac{m}{2}}$$
$$Y(u) = u^{n} + c_{1}'tu^{n-2} + \cdots + c_{\frac{n-2}{2}}t^{\frac{n-2}{2}}u^{2} + c_{\frac{n}{2}}t^{\frac{n}{2}}$$
$$Z(u) = u^{p} + c_{1}''tu^{p-2} + \cdots + c_{\frac{p-1}{2}}''t^{\frac{p-1}{2}}u$$

where all constants are positive, all powers of u in X(u) and Y(u) are even, all powers of u in Z(u) are odd and t equals δ , a small constant. Note also that the last terms in X(u) and Y(u) do not contain any positive powers of u but all terms in Z(u) contain positive powers of u. It follows that the last terms in X(u), Y(u), Z(u) and Z(u) do not contain positive powers of u whereas all terms in X(u), X(u), Y(u), Y(u) and Z(u) contain positive powers of u. Therefore, at u = 0, X(u) = X(u) = Y(u) = Y(u) = Z(u) = 0 and $\tau = 0$. As u grows, the terms in X(u), X(u), X(u), Y(u), Y(u), Y(u), Y(u), Z(u), Z(u) and Z(u) with the largest power of u (which are also the only terms without δ) become dominant and torsion is again given by (B.13). It follows that $\tau(u)$ is positive for positive u and negative for negative u on $\Gamma_{\sigma+\delta}$. Since τ is zero at u = 0, $\Gamma_{\sigma+\delta}$ has a torsion zero-crossing point at u = 0.

We next investigate $\tau(u)$ on $\Gamma_{\sigma-\delta}$. From (B.7) it follows that $\Gamma_{\sigma-\delta}$ is given by:

$$(D_t x)(u) = u^m - d_1 t u^{m-2} - \cdots - d_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u^2 - d_{\frac{m}{2}} t^{\frac{m}{2}}$$
$$(D_t y)(u) = u^n - d_1' t u^{n-2} - \cdots - d_{\frac{m-2}{2}} t^{\frac{n-2}{2}} u^2 - d_{\frac{m}{2}} t^{\frac{n}{2}}$$
$$(D_t z)(u) = u^p - d_1'' t u^{p-2} - \cdots - d_{\frac{p-1}{2}} t^{\frac{p-1}{2}} u$$

where all constants are positive, all powers of u in $D_t x$ and $D_t y$ are even, all powers of u in $D_t z$ are odd and t equals δ , a small constant. It again follows that $\tau = 0$ at u = 0, τ is positive for positive u and negative for negative u. Therefore there is also a torsion zero-crossing point at u = 0 on $\Gamma_{\sigma-\delta}$. It follows that there is a torsion zero-crossing point at u_0 on $\Gamma_{\sigma-\delta}$ before the formation of the cusp point and on $\Gamma_{\sigma+\delta}$ after the formation of the cusp point.

168

by:

Case 3: Γ_{σ} is approximated by (u^m, u^n, u^p) where m is even, n is odd and p is even.

The proof is analogous to that of case 2, and the same result follows.

Case 4: Γ_{σ} is approximated by (u^m, u^n, u^p) where m is even, and n and p are odd.

At u = 0, the cusp point, τ is undefined. At all other points, $\tau(u)$ is given by (B.13). Since the coefficient of the numerator of (B.13) is positive (as shown in the proof of case 2) and p+n+m-6, the power of u in the numerator, is even, $\tau(u)$ is positive for positive and negative values of u in the neighborhood of u_0 on Γ_{σ} . Therefore there are no torsion zero-crossing points in the neighborhood of u_0 on Γ_{σ} .

We now investigate $\tau(u)$ on $\Gamma_{\sigma+\delta}$. It follows from (B.12) that $\Gamma_{\sigma+\delta}$ is given by:

$$X(u) = u^{m} + c_{1}tu^{m-2} + \cdots + c_{\frac{m-2}{2}}t^{\frac{m-2}{2}}u^{2} + c_{\frac{m}{2}}t^{\frac{m}{2}}$$
$$Y(u) = u^{n} + c_{1}'tu^{n-2} + \cdots + c_{\frac{n-1}{2}}t^{\frac{n-1}{2}}u$$
$$Z(u) = u^{p} + c_{1}''tu^{p-2} + \cdots + c_{\frac{p-1}{2}}'t^{\frac{p-1}{2}}u$$

where all constants are positive, all powers of u in X(u) are even, all powers of uin Y(u) and Z(u) are odd and t equals δ , a small constant. Furthermore, note that the last term in X(u) does not contain a positive power of u but all terms in Y(u) and Z(u) contain positive powers of u. Therefore the last terms in X(u), Y(u), Y(u), Z(u) and Z(u) do not contain positive powers of u whereas all terms in X(u), X(u), X(u), Y(u) and Z(u) contain positive powers of u. Hence at u=0, X(u) = X(u) = Y(u) = Z(u) = 0 and

$$\tau(u) = \frac{\ddot{Y}(u)\dot{Z}(u)\ddot{X}(u) - \ddot{Z}(u)\dot{Y}(u)\ddot{X}(u)}{(\dot{Z}(u)\ddot{X}(u))^{2} + (\dot{Y}(u)\ddot{X}(u))^{2}} = \frac{\ddot{X}(u)(\ddot{Y}(u)\dot{Z}(u) - \ddot{Z}(u)\dot{Y}(u))}{(\dot{Z}(u)\ddot{X}(u))^{2} + (\dot{Y}(u)\ddot{X}(u))^{2}}$$

Since the denominator is positive and $\ddot{X}(u)$ is positive, to determine the sign of $\tau(u)$, we must determine the sign of the expression: $\ddot{Y}(u)\dot{Z}(u) - \ddot{Z}(u)\dot{Y}(u)$. At u=0, using (B.12) we conclude that the non-zero term of $\dot{Y}(u)$ is:

$$c_{\frac{n-1}{2}}t^{\frac{n-1}{2}} = 1.3.5 \cdots (n-2)\frac{(2t)^{\frac{n-1}{2}}n!}{(n-1)!} = 1.3.5 \cdots n \ 2^{\frac{n-1}{2}}t^{\frac{n-1}{2}}$$

Similarly, at u=0, the non-zero term of $\dot{Z}(u)$ is:

$$c''_{\frac{p-1}{2}}t^{\frac{p-1}{2}} = 1.3.5 \cdots p \ 2^{\frac{p-1}{2}}t^{\frac{p-1}{2}}$$

Using (B.12), it follows that at u=0, the non-zero term of Y(u) is:

$$6c_{\frac{n-3}{2}}t^{\frac{n-3}{2}} = 6(1.3.5\cdots(n-4))\frac{(2t)^{\frac{n-3}{2}}n!}{6(n-3)!} = (1.3.5\cdots n)(n-1)2^{\frac{n-3}{2}}t^{\frac{n-3}{2}}$$

Similarly, at u=0, the non-zero term of $\ddot{Z}(u)$ is:

$$6c''_{\frac{p-3}{2}}t^{\frac{p-3}{2}} = (1.3.5\cdots p)(p-1)2^{\frac{p-3}{2}}t^{\frac{p-3}{2}}$$

Therefore

$$\ddot{Y}(u)\dot{Z}(u) - \ddot{Z}(u)\dot{Y}(u) = (1.3.5\cdots n)(n-1)2^{\frac{n-3}{2}}t^{\frac{n-3}{2}}(1.3.5\cdots p)2^{\frac{p-1}{2}}t^{\frac{p-1}{2}}$$
$$- (1.3.5\cdots n)2^{\frac{n-1}{2}}t^{\frac{n-1}{2}}(1.3.5\cdots p)(p-1)2^{\frac{p-3}{2}}t^{\frac{p-3}{2}}$$
$$= (2t)^{\frac{p+n-4}{2}}(1.3.5\cdots n)(1.3.5\cdots p)(n-p)$$

and it follows that $\ddot{Y}(u)\dot{Z}(u) - \ddot{Z}(u)\dot{Y}(u) < 0$ since n < p. Therefore $\tau(u)$ is negative at u=0 on $\Gamma_{\sigma+\delta}$. As u grows the terms in $\dot{X}(u)$, $\ddot{X}(u)$, $\ddot{X}(u)$, $\ddot{Y}(u)$, $\ddot{Z}(u)$ and $\ddot{Z}(u)$ with the largest power of u (which are also the only terms without δ) become dominant and $\tau(u)$ is again given by (B.13). Since p+n+m-6, the power of u in the numerator, in now even, $\tau(u)$ becomes positive as u grows in absolute value. Therefore there exist two new torsion zero-crossings in a neighborhood of u_0 on $\Gamma_{\sigma+\delta}$.

This completes the proof of theorem 4.4.9.

Proof of theorem 4.4.10: It will be shown that this theorem holds for an arbitrary parametrization of Γ_{σ} . Therefore it must also be true of arc length parametrization or close approximations.

Let $\Gamma = (x(u), y(u), z(u))$ be a space curve and let x(u), y(u) and z(u) be polynomial functions of u. Let $\Gamma_{\sigma} = (X(u,\sigma), Y(u,\sigma), Z(u,\sigma))$ be an evolved version of Γ with a point of zero curvature at u_0 . Assume without loss of generality that $u_0 = 0$ and that at $u_0 \Gamma_{\sigma}$ goes through the origin of the coordinate system. It follows that Γ_{σ} can be approximated in a neighborhood of u_0 by:

$$\Gamma_{\sigma} = (u^m, u^n, u^p) \tag{B.14}$$

where u^m , u^n and u^p are the lowest degree terms in $X(u,\sigma)$, $Y(u,\sigma)$ and $Z(u,\sigma)$ respectively. Assume without loss of generality that p > n > m.

Since m, n and p can be odd or even, point u_0 must be analyzed in each of eight possible cases. The analysis in the proof of theorem 4.4.8 showed that when m is even, a cusp point exists on Γ_{σ} at u_0 . We will therefore look at the remaining four cases in which m is odd:

Case 1. m is odd and n and p are even.

Torsion on Γ_{σ} is given by equation (B.13). Since p+n+m-6 is odd, torsion is positive for positive u and negative for negative u in a neighborhood of u_0 . We now investigate torsion on $\Gamma_{\sigma+\delta}$ where δ is a small, positive number. Expressions for $X(u,\sigma)$, $Y(u,\sigma)$ and $Z(u,\sigma)$ can be obtained using equation (B.12). Note that all powers of u in $X(u,\sigma)$ are odd and all powers of u in $Y(u,\sigma)$ and $Z(u,\sigma)$ are even. It follows that all powers of u in $\dot{X}(u)$, $\ddot{X}(u)$, $\ddot{Y}(u)$ and $\ddot{Z}(u)$ are even and all powers of u in $\ddot{X}(u)$, $\dot{Y}(u)$, $\ddot{Y}(u)$, $\dot{Z}(u)$ and $\ddot{Z}(u)$ are odd. Note also that those terms in which all powers of u are odd, are equal to zero at u_0 . Therefore torsion is zero at u_0 on $\Gamma_{\sigma+\delta}$. As u grows, u^m , u^n and u^p , that is the terms in $X(u,\sigma)$, $Y(u,\sigma)$ and $Z(u,\sigma)$ with the largest powers of u, become dominant and torsion is again given by equation (B.13). It follows that torsion is positive for positive u and negative for negative u on $\Gamma_{\sigma+\delta}$ in a neighborhood of u_0 . Hence no new torsion zero-crossings have been created.

Case 2. m is odd, n is even and p is odd.

Torsion on Γ_{σ} is again given by (B.13). Since p+n+m-6 is even, torsion is positive for positive and negative u on Γ_{σ} . We now investigate torsion on $\Gamma_{\sigma+\delta}$. Note that all powers of u in $X(u,\sigma)$ are odd, all powers of u in $Y(u,\sigma)$ are even and all powers of u in $Z(u,\sigma)$ are odd. It follows that all powers of u in $\dot{X}(u)$, $\ddot{X}(u)$, $\ddot{Y}(u)$, $\dot{Z}(u)$ and $\ddot{Z}(u)$ are even and all powers of u in $\dot{X}(u)$, $\ddot{Y}(u)$, $\ddot{Y}(u)$ and $\ddot{Z}(u)$ are odd. Note also that those terms in which all powers of u are odd, are equal to zero at u_0 . It follows that torsion on $\Gamma_{\sigma+\delta}$ at u_0 is given by:

$$\tau(u) = \frac{\ddot{Z} \dot{X} \ddot{Y} - \ddot{X} \ddot{Z} \ddot{Y}}{(\dot{Z} \ddot{Y})^2 + (\dot{X} \ddot{Y})^2} = \frac{\ddot{Y}(\ddot{Z} \dot{X} - \ddot{X} \dot{Z})}{(\dot{Z} \ddot{Y})^2 + (\dot{X} \ddot{Y})^2}.$$

Since the denominator of the expression above is positive and \ddot{Y} is positive, the sign of $\tau(u)$ is the same as the sign of the expression: $\ddot{Z}\dot{X} - \ddot{X}\dot{Z}$. At u_0 , using (B.12) it can be shown that:

$$\ddot{Z}\dot{X} - \ddot{X}\dot{Z} = (2t)^{\frac{p+m-4}{2}}(1.3.5\cdots p)(1.3.5\cdots m)(p-m)$$

which is positive at u_0 . As u grows larger, torsion is again given by (B.13) in a neighborhood of u_0 and is therefore positive in that neighborhood. Again no new torsion zero-crossings have been created.

Case 3. m and n are odd and p is even.

Torsion is again given by (B.13) on Γ_{σ} . Since p+n+m-6 is even, torsion is positive for positive and negative u on Γ_{σ} . We now investigate torsion on $\Gamma_{\sigma+\delta}$. Note that all powers of u in $X(u,\sigma)$ and $Y(u,\sigma)$ are odd and all powers of u in $Z(u,\sigma)$ are even. Hence all powers of u in $\dot{X}(u)$, $\ddot{X}(u)$, $\dot{Y}(u)$, $\ddot{Y}(u)$ and $\ddot{Z}(u)$ are even and all powers of u in $\ddot{X}(u)$, $\ddot{Y}(u)$, $\ddot{Z}(u)$ and $\ddot{Z}(u)$ are odd. Note also that those terms in which all powers of u are odd, are equal to zero at u_0 . Therefore torsion on $\Gamma_{\sigma+\delta}$ at u_0 is given by:

$$\tau(u) = \frac{\ddot{X}\dot{Y}\ddot{Z} - \ddot{Y}\dot{X}\ddot{Z}}{(\dot{Y}\ddot{Z})^2 + (\dot{X}\ddot{Z})^2} = \frac{\ddot{Z}(\ddot{X}\dot{Y} - \ddot{Y}\dot{X})}{(\dot{Y}\ddot{Z})^2 + (\dot{X}\ddot{Z})^2}.$$

Since the denominator of the expression above is positive and Z is positive, the sign of $\tau(u)$ is the same as the sign of the expression: $X \dot{Y} - \dot{Y} \dot{X}$. At u_0 , using (B.12) it can be shown that:

$$\ddot{X}\dot{Y} - \ddot{Y}\dot{X} = (2t)^{\frac{n+m-4}{2}}(1.3.5\cdots n)(1.3.5\cdots m)(m-n)$$

which is negative since n > m. Therefore torsion is negative at u_0 on $\Gamma_{\sigma+\delta}$. As u grows larger, torsion is again given by (B.13) in a neighborhood of u_0 and is therefore positive for positive and negative u. It follows that there are two new torsion zero-crossings in a neighborhood of u_0 on $\Gamma_{\sigma+\delta}$.

Case 4. m, n and p are odd.

Torsion on Γ_{σ} is again given by equation (B.13). Since p+n+m-6 is odd, torsion is positive for positive u and negative for negative u on Γ_{σ} . We now investigate torsion on $\Gamma_{\sigma+\delta}$. Note that all powers of u in $X(u,\sigma)$, $Y(u,\sigma)$ and $Z(u,\sigma)$ are odd. Hence all powers of u in $\dot{X}(u)$, $\dot{Y}(u)$, $\dot{Z}(u)$, $\ddot{X}(u)$, $\ddot{Y}(u)$ and $\ddot{Z}(u)$ are even and all powers of u in $\ddot{X}(u)$, $\ddot{Y}(u)$ and $\ddot{Z}(u)$ are odd. Note also that those terms in which all powers of u are odd, are equal to zero at u_0 . It follows that torsion is unbounded at u_0 on $\Gamma_{\sigma+\delta}$. As u grows larger, torsion is again given by (B.13) in a neighborhood of u_0 and is therefore positive for positive u and negative for negative u. Hence there are no new torsion zero-crossings in a neighborhood of u_0 on $\Gamma_{\sigma+\delta}$.