[16] Weak Continuity Constraints Generate Uniform Scale-Space Descriptions of Plane Curves

Andrew Blake, Andrew Zisserman and Andreas V Papoulias

Department of Computer Science
University of Edinburgh, Edinburgh EH9 3JZ, Scotland, UK

Reprinted, with permission of Elsevier Science Publishers, from *Advances in Artificial Intelligence II*, 1987, 587-597.

Abstract

Scale-space filtering (Witkin 83) is a recently developed technique, both powerful and general, for segmentation and analysis of signals. Asada and Brady (84) have amply demonstrated the value of scale-space for description of curved contours from digitised images. Weak continuity constraints (Blake 83a,b, Blake and Zisserman 85,87) furnish novel, powerful, non-linear filters, to use in place of gaussians, for scale-space filtering. This has some striking advantages (fig 1). First, *scale-space is uniform*, so that tracking across scale is a trivial task. *Structure need not be preserved* to indefinitely fine scale; this leads to an enrichment of the concept of scale - a rounded corner, for example, can be represented as a discontinuity at coarse scale but smooth at fine scale. And finally boundary conditions at ends of curves are handled satisfactorily - it is as easy to analyse open curves as closed ones.

1 Weak continuity constraints

Weak continuity constraints are a principled and effective treatment of the localisation of discontinuities in discrete data. Detailed discussions are given in (Blake 83a, Blake 83b, Blake and Zisserman 85, Blake and Zisserman 87). Applications in computer vision include curve description, edge detection, reconstruction of $2\frac{1}{2}$D surfaces from stereo or laser-rangefinder data, and others. This paper deals with the application of weak continuity constraints to description of plane curves. First a brief summary of weak continuity constraints is given for problems like curve description, in which the data is a 1D array. Data may be obtained from a plane curve as an array θ_i of tangent angle values at equal spacings in arc-length s.

The problem is to localise discontinuities in noisy, discrete data. The notion of a discontinuity applies to functions, not to discrete arrays so the problem is ill-posed, and this is exacerbated by the presence of noise. One solution is to interpolate the data by a smooth function such as a gaussian, whose 1st derivative can then be examined. Of course this is common practice in edge detection and in spline interpolation (e.g. de Boor 78). Such smoothing can be regarded as fitting a function $u(s)$ which tends to seek a minimum of some elastic energy P. Energy P is traded off against a sum of squares error measure D, defined as:

$$D = \Sigma_i (u(s_i) - \theta_i)^2$$

by minimising variationally the total energy (or cost) $P + D$. The result is a function $u(s)$ that is both fairly smooth and is a fair approximation to the data θ_i. The simplest form of the energy P is that of a horizontal stretched string (approximately):

$$P = \lambda^2 \int (u')^2 \, ds,$$

where the parameter λ governs the stiffness of the string. If λ is large then the tendency to smoothness overwhelms the tendency (from D) to approximate the data well. In the extremes, if λ is very large, the fitted function is simply $u = const$, the least squares regression of a constant function to the data θ_i; but if $\lambda \sim 0$ then u interpolates the data, linking the θ_i by straight lines.

Weak continuity constraints can be applied to a scheme like the one above, to incorporate discontinuities *explicitly* into the fitting of u above. Rather than fitting a u that is smooth everywhere and then examining the gradient u', the function u is allowed to break (at knots, in spline jargon) - it is piecewise continuous. The number and position of the discontinuities is chosen optimally, by using an augmented form of cost function $E = D + P + S$, where the additional term S embodies weak continuity constraints:

$$S = \alpha \times (\text{number of discontinuities})$$

– a fixed penalty α is paid for each discontinuity allowed. This has the effect of discouraging discontinuities; u is continuous "almost everywhere". But an occasional discontinuity may be allowed if there is sufficient benefit in terms of smoothness (P) and faithfulness to data (D) in so doing. Clearly α is some kind of measure of reluctance to allow a discontinuity.

In fact the two parameters α, λ interact in a rather interesting way. Far from being "fudge factors" that must be empirically set, they have clear interpretations in terms of

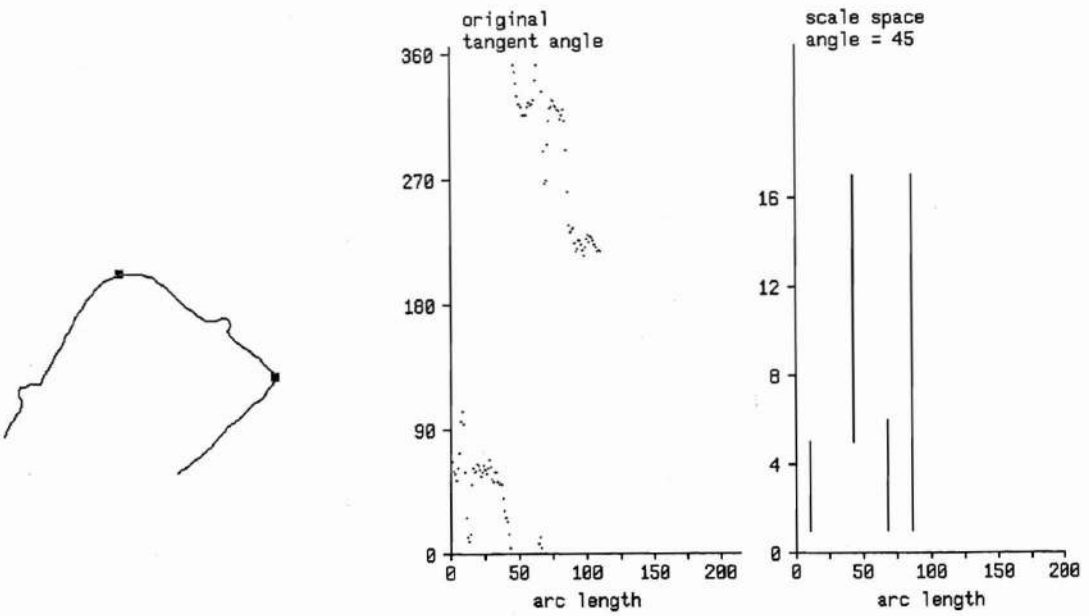

Fig 1 Curve segmentation by weak continuity constraints. From left to right: a hand drawn curve; angle/arc-length data (note quantisation noise); scale-space, at angle sensitivity of 45° - note: 1. vertical lines (uniformity) 2. the rounded corner disappears from scale-space at fine scale (non-preservation of structure) and 3. structure near curve ends causes no problems for segmentation (there are no spurious discontinuities near the ends).

scale and sensitivity (Blake and Zisserman 85,87). Here is what they signify, in the context of plane-curve segmentation:

- $\Phi_0 = \sqrt{(2\alpha/\lambda)}$ is a measure of angular sensitivity. If plane curve data θ_i contains an isolated discontinuity of magnitude Φ (e.g. two long straight line segments, joined at a vertex making an exterior angle Φ) then the fitted function $u(s)$ will have a discontinuity there if and only if $\Phi > \Phi_0$

- λ is a characteristic scale. "Events" (e.g. steps) in the data that are separated by more than λ are treated as effectively independent by the fitting process. But events spaced less that λ apart may interact and small "glitches" in the data whose total extent is much less than λ may well be ignored - filtered out. This mechanism removes both noise and small-scale structure.

- $\kappa_0 = \Phi_0/2\lambda$ is a curvature limit. If an extended arc in the data has curvature $k > k_0$ then there will be a discontinuity in the fitted function u somewhere on that arc. This can be regarded as a limitation on performance - the inability to discriminate between high curvature arcs and angular discontinuities - to be traded off against the previous 2 performance measures Φ_0, λ

- α itself is a measure of resistance to noise. If α is large compared with the variance σ^2 of noise in the angle data, then there will be no spurious discontinuities due to noise.

2 Algorithms: graduated non-convexity (GNC) and dynamic programming

Given a curve as a stream of coordinates (x_i, y_i), the first step is to convert it to $\theta_i(s)$ form. This is best done by dividing the stream of (x_i, y_i) into a sequence of strokes (Perkins 78) of equal lengths Δs. A stroke is formed by least squares fitting a straight line segment to the (x_i, y_i) that fall within the particular stroke. Length Δs should be chosen as short as possible to avoid blurring, but just long enough to avoid undue quantisation error. In practice, quantisation errors around $\pm 10°$ may be acceptable. The θ_i are not restricted to the range $[0°, 360°]$ but include a "winding number", so that curves with loops can be correctly represented (fig 2). Of course, it is not quite possible in practice to fit a function $u(s)$ to the data, but only a discrete representation of $u(s)$. So $u(s)$ is approximated, in accordance with the usual practice of finite elements (see Terzopoulos (83) for applications of finite elements to computer vision). For the simple stretched string energy P above, linear elements are sufficient. The function $u(s)$ is represented by a sequence of points u_i (at positions along the curve corresponding to the θ_i), interpolated linearly. The variational problem of the previous section becomes a discrete optimisation problem, to minimise:

$$F = \Sigma_i (u_i - \theta_i)^2 + \Sigma_i g(u_i - u_{i-1})$$

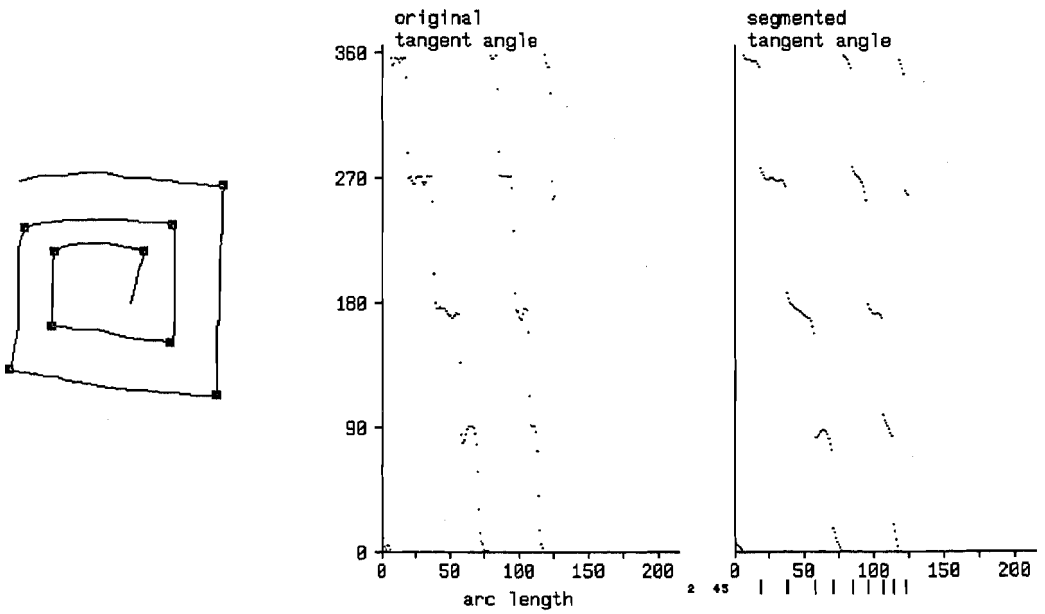

Fig 2. The $\theta - s$ representation of a curve includes winding number, so that even spirals can happily be segmented.

where the function

$$g(t) = \lambda^2 t^2 \text{ if } |t| < \sqrt{\alpha}/\lambda,\ \alpha \text{ otherwise}$$

is an interaction function between neighbouring u_i (see fig 3a) that incorporates both the membrane energy and weak continuity constraint penalties. The term $\Sigma_i g(...)$ above is the discrete representation of the functional $S+P$ above. Details of the discretisation, and derivation of g are given in (Blake and Zisserman 85,87).

Now a fundamental property of F is that it is non-convex, and so may have many local minima. Its global minimum cannot generally be found by naive downhill search over the u_i. A rather general way of dealing with such non-convexity is to use "simulated annealing" (Metropolis et al 53), a stochastic method, which works for cost functions like F above (Geman and Geman 84) but is rather expensive computationally (Marroquin 84). Here two efficient algorithms are described; both have been implemented successfully on modest serial machines.

Graduated non-convexity

"Graduated non-convexity" (GNC) is fully described in (Blake 83a,83b, Blake and Zisserman 85,87). Whereas simulated annealing uses random processes to jump out of local minima, GNC constructs a function F^* which is convex (and hence is free of spurious local minima) and approximates F well. Then a family of functions $F^{(p)}$ $p \in [0,1]$ is constructed with $F^{(1)} = F^*$ and $F^{(0)} = F$, and $F^{(p)}$ varying gradually between the two as p varies from 1 to 0. The function $F^{(p)}$ is defined as F above, but with $g^{(p)}$ in place of g, where

$$g^p(t) = \alpha - c(|t| - r)^2 \text{ if } q < |t| < r,\ g(t) \text{ otherwise,}$$

where $c = 1/2p$, $r = \sqrt{\alpha}\sqrt{2/c + 1/\lambda^2}$, and $q = \alpha/(\lambda^2 r)$

- see fig 3b. The algorithm is to minimise a sequence of $F^{(p)}$, by direct descent on each one, starting with $F^{(1)}$ and ending with $F^{(0)}$. A sequence of 11 values of p 1.0,0.9,..0 proves to be more than adequate in practice. In fact, less work is needed for small λ (e.g. $\lambda \leq 4$, where the finite element between u_i, u_{i+1} has unit length) and it may be sufficient to use the convex approximation $F^* = F^{(1)}$ without bothering to descend on the remaining 10 $F^{(p)}$. But for large λ (e.g. λ=20) the whole sequence is needed. GNC is an approximate method - it finds u_i close to the optimum of F. It has been shown, however, to be exact under certain conditions (Blake 83b, Blake and Zisserman 87). Although execution times are relatively long for large λ, multigrid methods (Terzopoulos 83) might effect a considerable improvement.

Dynamic programming

An alternative to GNC is to treat the minimisation of F as an integer programming problem, by quantising angle measures as a range of M values u_i (say to 1° or 2° accuracy), and applying dynamic programming (Bellman and Dreyfus 62). Details of this method are given in Papoulias (85). It is applicable because the 1D vector u_i can be expressed, for all i, as a union of two sets $\{u_0, ..u_i\}$ and $\{u_i, ..u_N\}$ which have precisely *one* element u_i in common. Note that no equivalent family of simple decompositions exists for 2D arrays u_{ij} and hence dynamic programming is not usable for applying weak continuity

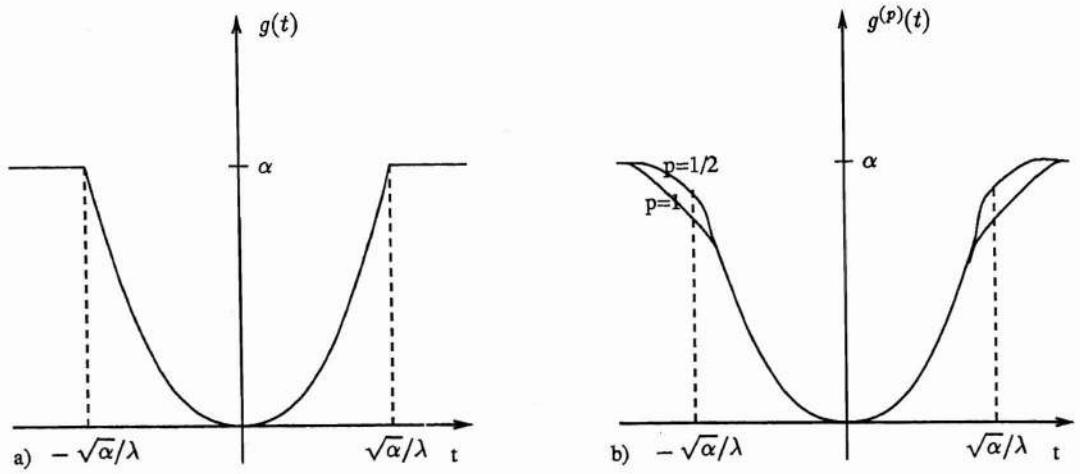

Fig 3. Neighbourhood interaction functions: (a) g for the cost function F; (b) $g^{(p)}$ for the sequence of functions $F^{(p)}$ that approximate F.

constraints to 2D data, as in edge detection or surface reconstruction. Although a dynamic programming algorithm for the 2D problem could be defined, in theory, it would involve the use of tables with up to M^N entries! For similar reasons, it is not practicable to use dynamic programming for higher order energies P, involving 2nd or higher derivatives of u. Tables of size $O(M^2)$ would be required (for P involving u''). GNC is quite usable, however, both for 2D data and for 2nd order P.

Following normal dynamic programming practice, the algorithm consists largely of constructing a pair of tables (the return function f_i and the policy function p_i) for each i, each of length M. Total storage required is therefore $O(NM)$ units. The value of $f_i(u_{i+1})$ is the minimal partial cost for $u_0..u_i$ for a given value of u_{i+1}, and $p_i(u_{i+1})$ is the value of u_i at that minimum. Having constructed the tables, there remains the task (requiring relatively insignificant computation time) of tracing back through the tables, from f_N down to f_0, to recover the optimal u_i. The complexity of the algorithm is $O(NM^2)$ - so extra precision in angle quantisation (large M) is expensive. The expense can be mitigated to some degree by "table reduction" (Papoulias 85), which works as follows. For the cost function F for the weak elastic string, it transpires that each table f_i contains a non-constant interval flanked by entries all of the same constant value. Those constant entries can be treated for computational purposes as one entry. This effectively reduces the value of M. The effective M appears, in practice, to be proportional to $\sqrt{\alpha}$ (independent of λ); so the reduction may be effective even at large λ when GNC is least efficient. In practice, reduction by a factor of up to 4 was obtained, reducing execution time by a factor of up to 16. More recently, an exact dynamic programming algorithm has been constructed, that requires no quantisation at all (Blake 89).

Comparison of GNC and dynamic programming

It has been mentioned that GNC is an approximate method, whereas dynamic programming is exact. In practice, no qualitative difference between solutions obtained from the two methods is observed (see also Blake 89); this is, in itself, a confirmation that solutions from GNC are good approximations. As for efficiency, each method has its advantages. For large values of λ GNC is slow, but (for a given α) dynamic programming continues to work well. Finally, GNC requires high precision arithmetic, unlike quantised dynamic programming. In practice (for modest values of λ) it seems that GNC is faster on a Motorola 68000, for example, if it has adequate hardware floating-point support. For smaller values of λ, GNC runs in about 1 second (SUN 2, SKY floating point, vector length N=50, λ=2). This could be expected to improve by an order of magnitude with the new 68000 floating point co-processor.

3 Scale-space properties

This section discusses the properties of scale-space descriptions of curves, under weak continuity constraints. An example was displayed in fig 1. Several notable properties are illustrated: the most striking is the uniformity of the scale-space - the locus of each discontinuity in scale space is plumb vertical. Moreover, in this scale-space, unlike gaussian scale space (Witkin, 83) in which the fingerprint theorem holds (Yuille and Poggio 84), structure is *not* preserved - discontinuities *may* be created as scale increases. We argue here that this *lack* of structure preser-

vation is a desirable property. Four other issues are considered: how to achieve an invariant parametrisation of the curve, detection of curvature discontinuities and how to treat the "curvature limit" described in section 1, and boundary conditions for open-ended curves. Finally it is worth noting that the new scale-space has an extra parameter in addition to scale, namely angular sensitivity (Φ_0 in section 1). Plots of scale-space shown here are at fixed values of Φ_0(e.g. 45^o in fig 1.).

Uniformity

It is apparent in fig 1 that the locus in scale-space of an individual feature (corner) is uniformly vertical, unlike gaussian scale spaces. This is a consequence of the theoretically predicted, spatial stability with respect to scale, that is inherent in optimal function fitting under weak continuity constraints (Blake and Zisserman 87). It arises because the extra cost in F, if a corner were slightly misplaced, is very large - far greater than the relatively modest extra costs introduced by spatially incoherent noise, or by extended but gentle curves (figure 4). Hence corners do not get misplaced.

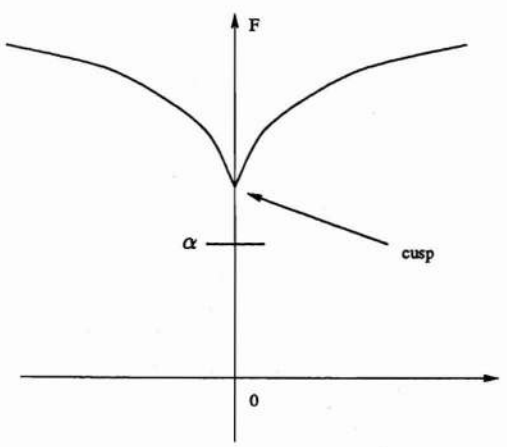

Fig.4 The uniformity property is a consequence of the sharp, cusp-like minimum in the energy F, plotted as a function of edge position. For a displacement ϵ in edge position it can be shown (Blake and Zisserman, 87) that the corresponding F is as plotted above. Hence there is a strong attraction towards $\epsilon = 0$, the true edge position.

Alternatively, in the terms of Canny's (83) performance measurers, the *localisation* is very good - as good, in fact, as a difference of boxes operator. (But it doesn't have that operator's multiple response problem!) A consequence of uniformity is that any one connected contour in scale-space must belong to only *one* physical feature on the curve. This is untrue of gaussian scale-space, as fig 5. shows.

Preservation of structure

Under weak continuity constraints, structure is not preserved as scale increases. There is an example of this in figure 1, in which a rounded corner is represented in scale space by a line that is present at large scale, but absent at small scale ($\lambda\epsilon[0,5]$). This is absolutely as it should be. The rounded corner appears smooth at small scale. It seems that the ability to represent this fact is important. Whereas structure preservation is a must with gaussian filters because it guarantees a successful tracking algorithm - tracking from fine to coarse scale picks up all zero-crossings - it is redundant under weak continuity constraints. Tracking is trivial, due to uniformity.

Invariant parametrisation

A problem with any scheme that uses arclength s to parametrise curves is that the parametrisation is defined with respect to the data rather than the *interpreted* curve $u(s)$. At small scale, this could mean extreme sensitivity to sensor and quantisation noise; in a practical vision system, this would result in curve descriptions that were unstable over time. A simple solution to this is adopted by Asada and Brady (84): they obtain their data from images, by means of an edge detector that inherently supresses noise. However there remains the lesser problem, that intermediate structure could generate distortions of scale (fig 6). An elegant solution to this problem, in the context of gaussian scale space filtering, proposed by Porril (85), subjects the curve to a simulated diffusion process. Under weak continuity constraints, an invariant scheme could conceivably be attainable by fitting a curve to data supplied as a sequence of coordinate vectors X_i, minimising curvature. Further work may be needed here.

Curvature limit and detection of discontinuities

It was explained in section 1 that $\kappa_0 = \Phi_0/2\lambda$ is a curvature limit, such that curves of curvature $\kappa > \kappa_0$ are segmented, even if there is no curvature maximum (fig 7).

Moreover the actual point of segmentation need not be particularly spatially stable. This seems to be a limitation of the scheme, for which two partial remedies are proposed. One is to note that such segmentation points exist only at large scale - but of course there may be "genuine" structure too that exists only at large scale. A better remedy is to use a higher order scheme, in which $P = \int u''^2$. This allows both *tangent* and *curvature* discontinuities to be detected, rather than tangent only. It also pushes the "spurious" segmentation problem to higher order (i.e. spurious curvature discontinuities) - but at some extra computational expense.

Boundary conditions

A very attractive property of the proposed scheme is that boundary conditions on open-ended contours are dealt with naturally. Naive gaussian filtering generates spurious discontinuities near ends of contours, which may mask genuine features near ends. Cures are of course possible, such as using modified convolution masks near ends (thus

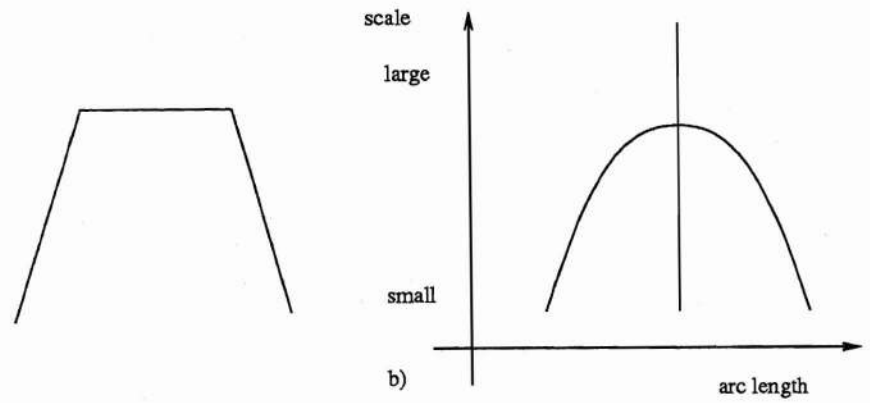

Fig 5. The gaussian scale-space generated by a polygonal contour (a), contains a bifurcation (b) which is unstable (non-transverse). It is therefore uncertain to which fine scale zero-crossing the single coarse scale zero-crossing belongs.

losing the gaussian's time-saving factorisability) or a diffusion process as above. Fig 1 illustrates the correct handling of boundary conditions: the small feature near the end is treated much like the one in the middle.

Acknowledgements

The authors thank the University of Edinburgh for provision of facilities, and the SERC for funding. A Blake is grateful for the support of the IBM Research Fellowship from the Royal Society.

References

1. Asada,H. and Brady,J.M. (1986). The Curvature Primal Sketch. *IEEE PAMI,* 8, 1, 2-14.

2. Bellman, R. and Dreyfus, S. (1962) *Applied Dynamic Programming.* Princeton University Press, Princeton, USA.

3. Blake,A. (1983a). The least disturbance principle and weak constraints. *Pattern Recognition Letters, 1, 393-399.*

4. Blake,A. (1983b). *Parallel computation in low-level vision.* Ph.D. Thesis, University of Edinburgh.

5. Blake,A. and Zisserman,A. (1985). Using weak continuity constraints. Report CSR-186-85, Dept. Computer Science, Edinburgh University, Edinburgh, Scotland. Also *Pattern Recognition Letters 6, 51-60, 1987.*

6. Blake A. and Zisserman A. *Visual Reconstruction.* MIT Press, 1987.

7. Blake A. Comparison of the efficiency of stochastic and deterministic algorithms for visual reconstruction. *PAMI 11, 1, 2-12.* Also reprinted in this volume.

8. Canny,J.F. (1983). *Finding edges and lines in images.* S.M. thesis, MIT, Cambridge, USA.

9. de Boor,C. (1978). *A practical guide to splines.* Springer-Verlag, New York.

10. Geman,S. and Geman,D. (1984). Stochastic Relaxation, Gibbs distribution, and Bayesian restoration of images. *IEEE PAMI, Nov 1984.*

11. Marroquin,J. (1984). Surface reconstruction preserving discontinuities. Memo 792, AI Laboratory, MIT, Cambridge, USA.

12. Metropolis,N., Rosenbluth,A.W., Rosenbluth,M.N., Teller,A.H. and Teller,E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics,* 6, 1087.

13. Papoulias,A. (1985). *Curve segmentation using weak continuity constraints.* M.Sc. thesis, Dept. Computer Science, University of Edinburgh.

14. Perkins,W.A. (1978). A model-based system for industrial parts. *IEEE Trans. Comp.,* 27, 2, 126 - 143.

15. Porril, J., Mayhew, J.E.W. and Frisby, J.P. (1986). Scale space diffusion: plane and space curves. Research Memo 018, AIVRU, Sheffield University.

16. Terzopoulos,D. (1983). Multilevel computational processes for visual surface reconstruction, *Computer Vision Graphics and Image Processing,* 24, 52-96.

17. Witkin,A.P. (1983). Scale-space filtering. *Proc. IJCAI 1983,* 1019-1022.

18. Yuille,A.L. and Poggio,T. (1984). Fingerprints theorems. *Proc. AAAI 1984,* 362-365.

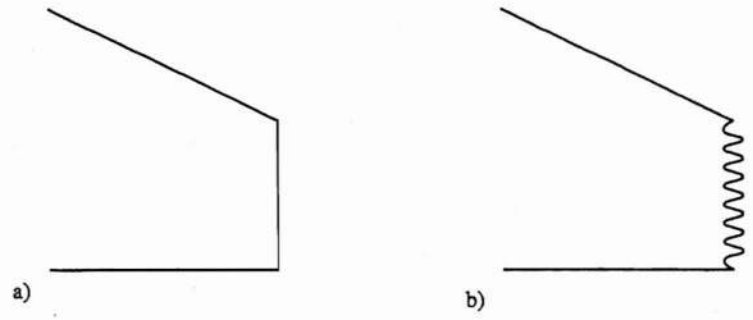

Fig 6 Non-invariance in s-parametrisation of curves. Curve (a) has two corners. Curve (b) very similar "at large scale", but has some detail between the two corners. As a result it acquires a great deal of extra arc-length between those corners, which distorts its scale-space diagram.

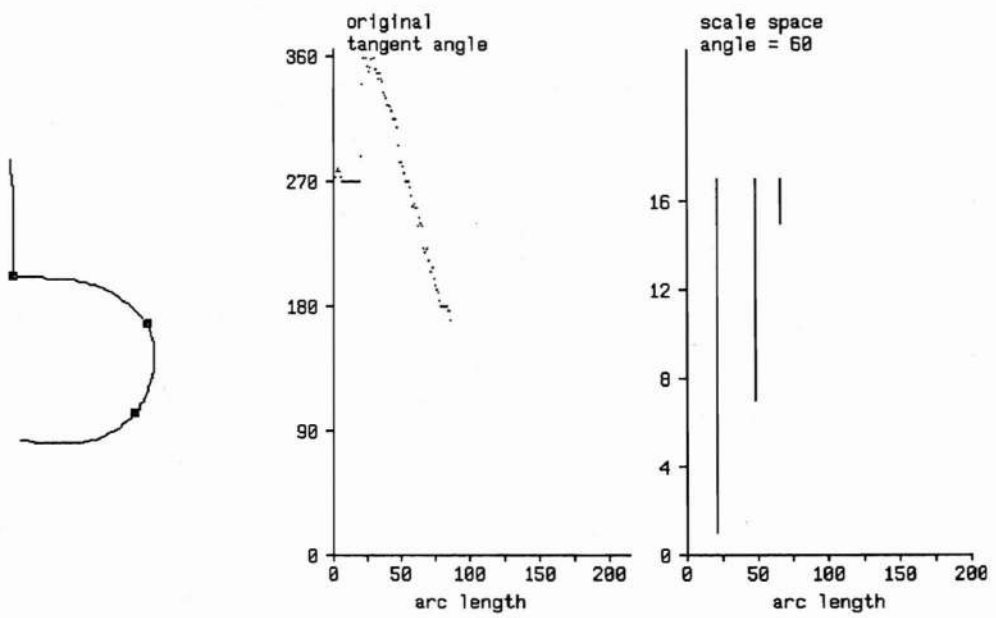

Fig 7 The gradient limit: arcs above a certain threshold curvature $\kappa_0 = \Phi_0/2\lambda$ may be segmented spuriously at larger scales.