Automatic Fitting of Digitised Contours at Multiple Scales through Curvature Scale Space

Farzin Mokhtarian *and* Yoke Khim Ung

Centre for Vision, Speech and Signal Processing, Department of Elec. Engineering, University of Surrey, Guildford, UK

Abstract

The Curvature Scale Space (CSS) technique has been used in conjunction with Hermite curves for automatic fitting of digitised contours at multiple scales.
CSS is a powerful contour shape descriptor which is expected to be in the MPEG-7 standard. A parametric representation of the input contour is convolved with Gaussian functions in order to obtain multi-scale descriptions of the contour. Curvature can be computed directly at each point of the smoothed contours. As a result, a set of curvature zero-crossing points can be recovered from each smoothed contour.
Hermite curves were used since each Hermite curve is defined by two endpoints and the tangent vectors at those points. No points external to the input contour are required for Hermite curves. Hermite endpoints are defined as consecutive curvature zero-crossing points extracted at multiple scales using the CSS method. Hermite tangent vectors can also be determined using the CSS technique. The only data stored are the endpoints and the tangent vectors needed by the Hermite curves in order to arrive at an approximate reconstruction of the original contour.
Approximation Error and Compression Ratio are computed at each scale. The graph of compression ratio as a function of approximation error is smoothed to remove noise and small fluctuations. The bending point *of that function is then defined as the largest maximum of its second derivative. The bending point can be considered as the boundary between the mostly vertical and the mostly horizontal segments of the graph. It can be used for automatic selection of an optimal scale.*

1. Introduction

The CSS image has been used for contour shape representation and feature extraction at multiple scales [13]. It is a powerful contour shape descriptor which is expected to be in the MPEG-7 standard. It was shown that the CSS image uniquely represents a 2-D contour modulo a rigid motion [8]. The CSS method has also been extended to space curves. That generalisation is referred to as the *torsion scale space* representation [9, 10].

Existing techniques for contour data reconstruction suffer from a number of shortcomings. Polygons have been used to approximate the shape of free-form contours by several researchers [14, 1, 6]. The vertices of the approximating polygons are then stored for later reconstruction of the shape. This approach works best when the corners of the input shapes are detected and chosen as polygon vertices. Robust corner detection is itself a challenging problem that needs to be addressed first. Another problem with polygon approximation is that polygons are not suitable for description of free-form contours, and would therefore require a large number of sides for a good approximation. A spline approach would yield a better approximation using less data. Fourier descriptors [15, 16] are another class of methods which can be utilized for contour data reconstruction. The first k Fourier coefficients can be computed and stored for later reconstruction of the contour. Naturally, a larger k yields a better reconstruction. A shortcoming of Fourier descriptors is that a large number of them will be needed to obtain an accurate reconstruction of the input contour. Furthermore, it is not suited to CAD applications. An algorithm by Schneider [17] fits splines to digital contours but the procedure used for control point selection is quite ad hoc and will not, in general, yield the best results.

Applications of this technique include efficient contour data compression as well as Computer Aided Design. It is usually assumed in CAD work that the user will supply all the control points required to generate the desired shapes. In this case, the user will have to start the design work from scratch. However, often the user may wish to start from a known shape which exists in digital form and modify that to obtain the final desired shape. The proposed technique will enable the user to obtain a spline approximation to that starting digital shape. The control points can then be adjusted by the user to produce the desired shape.

This paper presents a new method based on the Curvature Scale Space technique for automatic fitting of digitised contours. This technique utilises the CSS method to recover curvature zero-crossings and tangent vectors at those zero-crossings which are then used for Hermite curve fitting.

Section 2 presents a brief review of spline fitting techniques. Section 3 contains a short introduction to the CSS method. Section 4 is on contour data reconstruction through CSS and Hermite curves. Section 5 discusses the computation of the approximation error and the compression ratio at multiple scales. Section 6 presents the results and discussion, and section 7 contains the concluding remarks.

2. Spline Fitting Techniques

Spline fitting techniques [2,3] have been used widely in computer graphics, computer vision and image processing. They are useful since they can model free-form shapes in a compact way: A relatively small number of points and/or tangent vectors are sufficient for accurate reconstruction of the original shape.

A number of spline fitting techniques with different properties are available. The most common of these are the family of parametric cubic curves. This family consists of three major types of curves:

Hermite curves Defined by two endpoints and two endpoint tangent vectors.

Bézier curves Defined by two endpoints and two other points (not on the contour) which control the endpoint tangent vectors.

B-splines Defined by four control points (not on the contour).

3. Curvature Scale Space

The curvature scale space technique is suitable for recovering invariant geometric features (curvature zero-crossing points and/or extrema) of a planar curve at multiple scales. To compute it, the curve Γ is first parametrised by the arc length parameter u. This yields two coordinate functions:

$$\Gamma(u) = (x(u), y(u)).$$

Parametrising a contour by the arc length parameter is equivalent to travelling along that contour and sampling it at equal-sized intervals. An *evolved version* Γ_σ of Γ can then be computed [13]:

$$\Gamma_\sigma = (X(u,\sigma), Y(u,\sigma))$$

where

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

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

where $\otimes$ is the convolution operator and $g(u,\sigma)$ denotes a Gaussian of width σ. Convolution of a function with a Gaussian filter produces, at each point of that function, a weighted average of neighboring points with the weights decreasing as one moves away from the centre of convolution. Note that σ is also referred to as the *scale* parameter. The process of generating evolved versions of Γ as σ increases from 0 to ∞ is referred to as the *evolution* of Γ. This technique is suitable for removing noise from and smoothing a planar curve as well as gradual simplification of its shape [7,12].

In order to find curvature zero-crossings or extrema from evolved versions of the input curve, one needs to compute curvature accurately and directly on an evolved version Γ_σ of that curve. Curvature κ on Γ_σ is given by:

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{1.5}}$$

where

$$X_u(u,\sigma) = \frac{\partial}{\partial u}(x(u) \otimes g(u,\sigma)) = x(u) \otimes g_u(u,\sigma)$$

$$X_{uu}(u,\sigma) = \frac{\partial^2}{\partial u^2}(x(u) \otimes g(u,\sigma)) = x(u) \otimes g_{uu}(u,\sigma)$$

$$Y_u(u,\sigma) = y(u) \otimes g_u(u,\sigma)$$

and

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

Note that g_u and g_{uu} denote the first and second derivatives of $g(u,\sigma)$ with respect to u. The function defined implicitly by $\kappa(u,\sigma) = 0$ is the CSS image of Γ. Figure 1 shows the coastline of Africa. Figures 2-4 show the evolution of Africa. Figure 5 shows the CSS image of the contour shown in figure 1. In the CSS image, the horizontal axis corresponds to the arc length parameter on the input contour, and the vertical axis corresponds to the scale parameter which controls the degree of smoothing applied to the input contour. Each of the arches in the CSS image corresponds to a pair of curvature zero-crossing points on the input contour. As the contour is smoothed, that pair will gradually move closer and eventually merge and disappear. This event generates the peak of the corresponding arch in the CSS image.

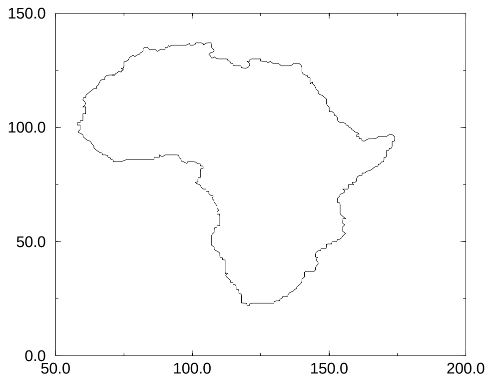

Figure 1: *Test contour: Africa*

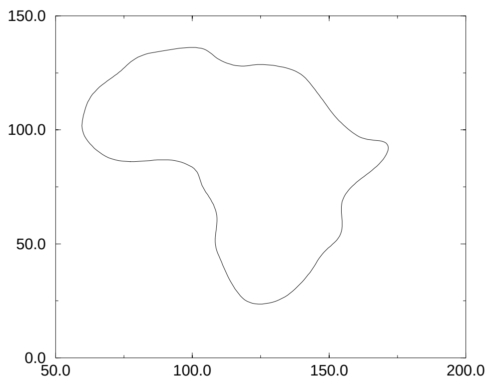

Figure 2: *Africa smoothed at* $\sigma = 4$

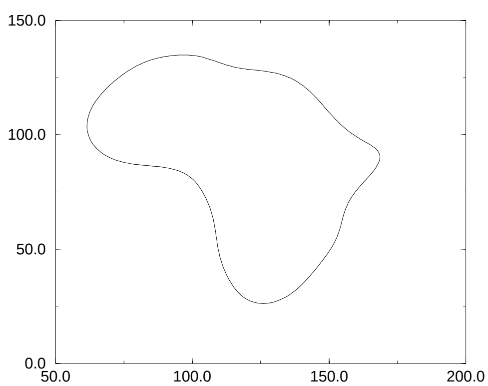

Figure 3: *Africa smoothed at* $\sigma = 8$

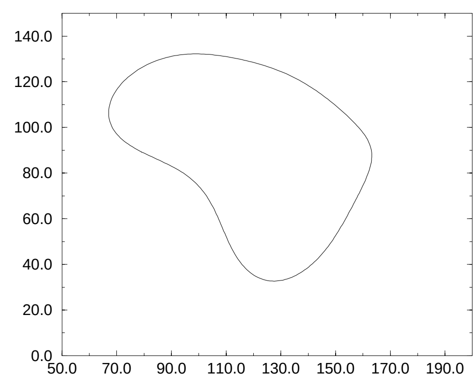

Figure 4: *Africa smoothed at* $\sigma = 16$

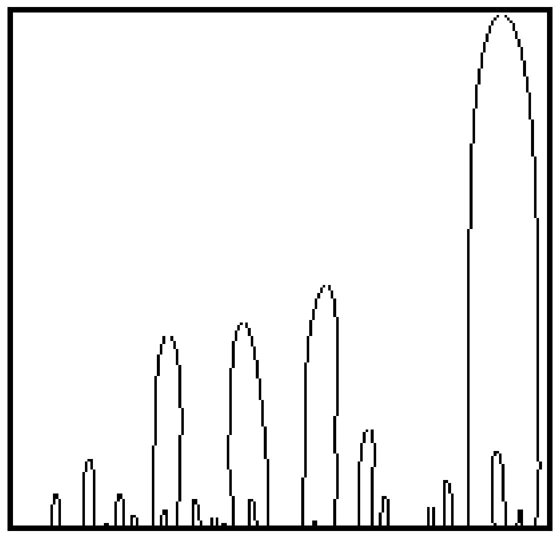

Figure 5: *Curvature Scale Space image of Africa*

4. Contour Data Reconstruction through CSS and Hermite Curves

This section explains how contour data reconstruction can be achieved at multiple scales through a combination of Hermite curves and the CSS technique.

Hermite curves were chosen since they do not require any points external to the input contour. Each Hermite curve segment requires two endpoints and the tangent vectors at those endpoints as input. All of these can be supplied automatically and robustly by the CSS method at multiple scales.

Suppose that $P(x_p, y_p)$ and $Q(x_q, y_q)$ are the endpoints of a Hermite segment. Assume that $u(x_u, y_u)$ and $v(x_v, y_v)$ are the tanget vectors at P and Q respectively. The Hermite segment is given by the following equations [3]:

$$x(w) = (2w^3 - 3w^2 + 1)x_p + (-2w^3 + 3w^2)x_q + (w^3 - 2w^2 + w)x_u + (w^3 - w^2)x_v$$

$$y(w) = (2w^3 - 3w^2 + 1)y_p + (-2w^3 + 3w^2)y_q + (w^3 - 2w^2 + w)y_u + (w^3 - w^2)y_v$$

where $w \in [0,1]$.

At a specific scale, the input contour is smoothed, and curvature is computed at each point using the formula given in section 3. This step is followed by the recovery of curvature zero-crossing points. Each pair of adjacent curvature zero-crossings are used as the endpoints of a Hermite curve. The advantage of using curvature zero-crossing points is that they are invariant to many transforms and therefore constitute a natural set of feature points. Furthermore, tangent vector estimation is more robust at inflection points since the contour is locally straight at those points. The directions of the tangent vectors at those endpoints are given by:

$$(X_u, Y_u)$$

where X_u and Y_u are as defined in section 3. Note, however, that in general the lengths of those tangent vectors have to be adjusted in order to obtain the optimal shape for the Hermite curve segment that best fits the input contour segment.

The tangent vector directions must be estimated from a smoothed contour in order to remove the influence of noise on the estimation process. However, their lengths are optimised using the original input data since our intention is to approximate the original contour as best as possible.

This optimization is carried out using an iterative procedure. The initial tangent vectors are multiplied by a real number n. Hermite curve fitting then takes place. The average distance between the Hermite curve segment and the input contour segment is then defined as following:

The distance from each point on the Hermite curve segment to the closest point on the input contour segment is computed. All such distances are added up and divided by the total number of points on the Hermite curve segment to determine the average distance.

The value of n is increased by a step size and Hermite curve fitting is repeated. The average distance is then recomputed. This process continues as long as the average distance continues to decrease. When the process terminates, the optimal length tangents and therefore the optimal Hermite curve has been found.

To enhance efficiency, it is possible to start with a relatively large step size, and to reduce it as the process approaches the optimal value of n. In this approach, if a larger step size causes the average distance to increase, the process backtracks and attempts a smaller step size.

Our experiments indicate that this optimisation procedure does converge to a global minimum. We believe the reason is that we can initialise the procedure at a point sufficiently close to the global minimum.

5. Approximation Error and Compression Ratio

When all Hermite curve segments have been fitted, the total Approximation Error is defined as the mean of the average distances for all the Hermite curve segments. Furthermore, Compression Ratio is defined as the size of the data after compression divided by the size of the original data.

Contour data compression can be carried out at multiple scales. This allows the user to find an appropriate trade-off between approximation error and compression ratio. Clearly, reducing the approximation error would also result in less compression and more accuracy whereas allowing the approximation error to rise would result in more compression and less accuracy.

The graph of compression ratio as a function of approximation error can be smoothed to remove noise and small fluctuations. The *bending point* of that function after smoothing can then be defined as the largest maximum of its second derivative. The bending point can be considered as the boundary between the mostly vertical and the mostly horizontal segments of the graph. It can be used for automatic selection of an optimal scale for contour data reconstruction. As a result, the user will not have to set any parameters in order to use this technique.

6. Results and Discussion

This section presents results on contour reconstruction through the CSS image. The test data consisted of three contours: Africa, Hokaido, and an abstract design. The Africa contour can be seen in figure 1. The Hokaido and abstract contour are shown in figures 6 and 7 respectively.

Reconstruction by spline fitting was implemented next. Figures 8-11 show the contour reconstruction results for Africa at multiple scales. As the scale increases, the number of curvature zero-crossing points and therefore the number of spline segments decreases, but this is accompanied by an increase in approximation error. Figures 12-15 show the corresponding results for Hokaido and figures 16-19 show the corresponding results for the abstract design.

Finally figures 20-22 show the graphs of compression ratio as a function of approximation error for Africa, Hokaido, and abstract design corresponding to spline reconstruction. They demonstrate that as reconstruction accuracy decreases, greater compression of the input data can be achieved. The point marked with a + on the Africa graph indicates the bending point of that graph which corresponds to the optimal scale for reconstruction.

At a specific scale, the complexity of the fitting process is

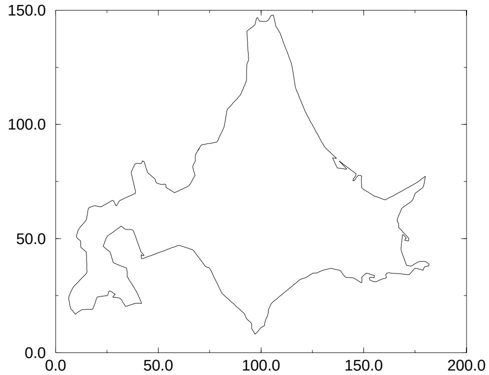

Figure 6: *Test contour: Hokaido*

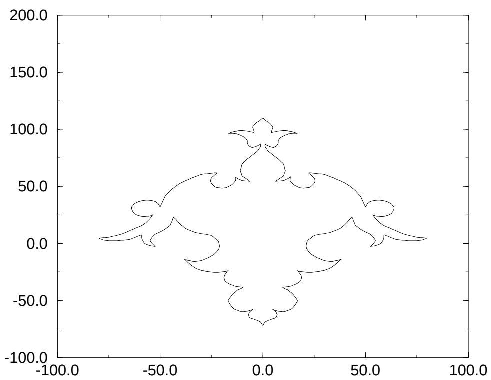

Figure 7: *Test contour: abstract design*

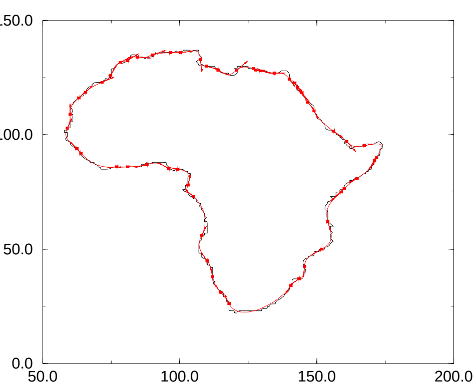

Figure 8: *Reconstruction of Africa at* $\sigma = 3$

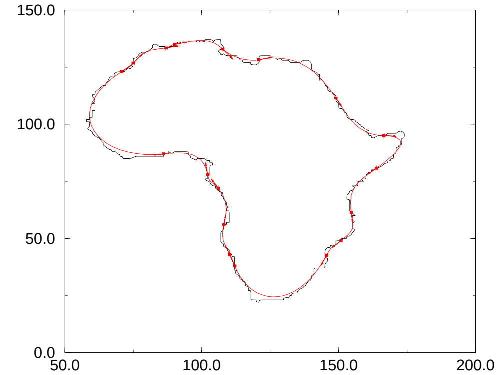

Figure 9: *Reconstruction of Africa at* $\sigma = 6$

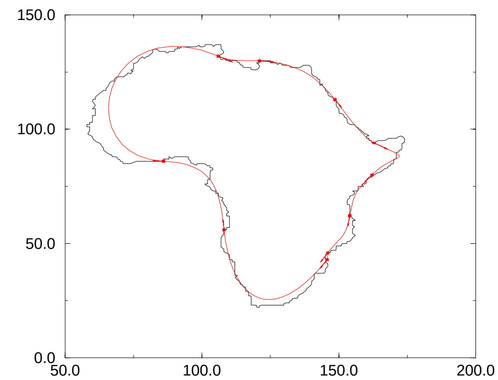

Figure 10: *Reconstruction of Africa at* $\sigma = 12$

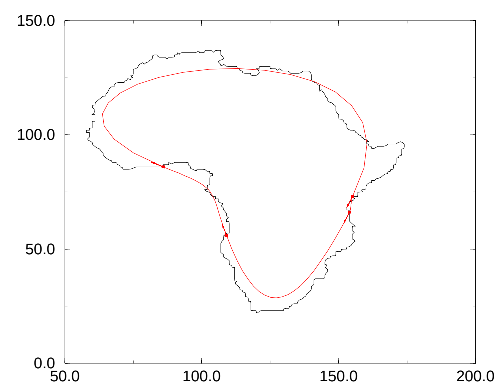

Figure 11: *Reconstruction of Africa at* $\sigma = 25$

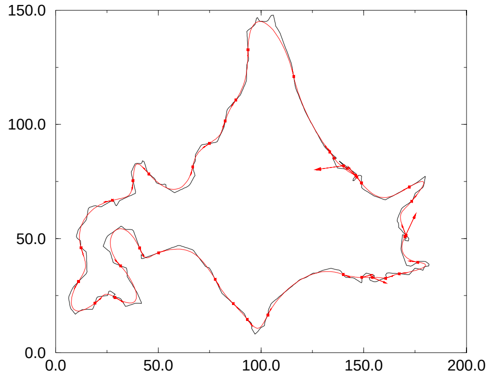

Figure 12: *Reconstruction of Hokaido at* $\sigma = 5$

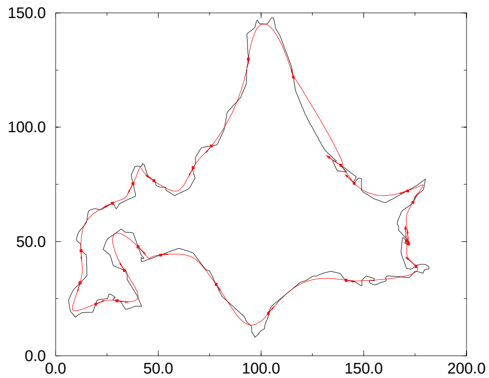

Figure 13: *Reconstruction of Hokaido at* $\sigma = 9$

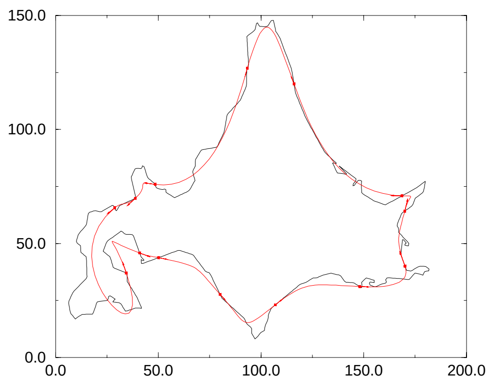

Figure 14: *Reconstruction of Hokaido at* $\sigma = 15$

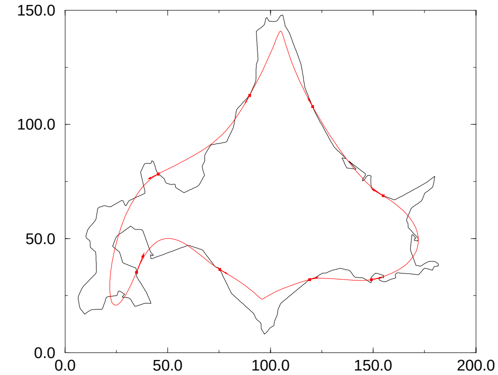

Figure 15: *Reconstruction of Hokaido at* $\sigma = 30$

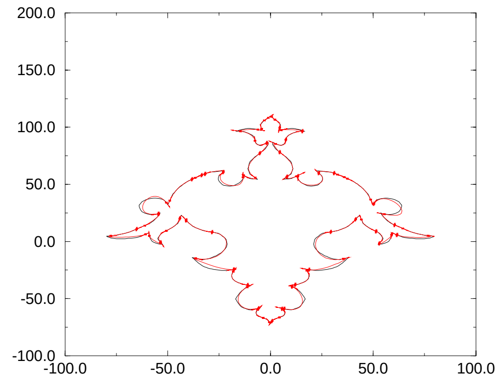

Figure 16: *Reconstruction of abstract design at* $\sigma = 1$

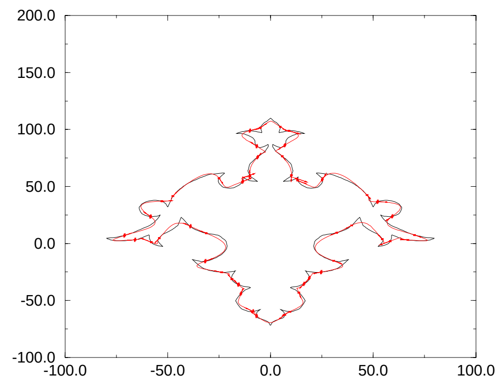

Figure 17: *Reconstruction of abstract design at* $\sigma = 7$

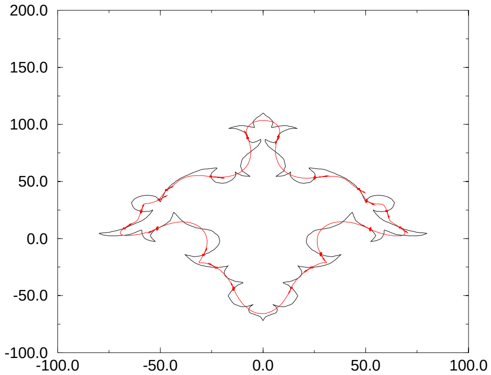

Figure 18: *Reconstruction of abstract design at* $\sigma = 15$

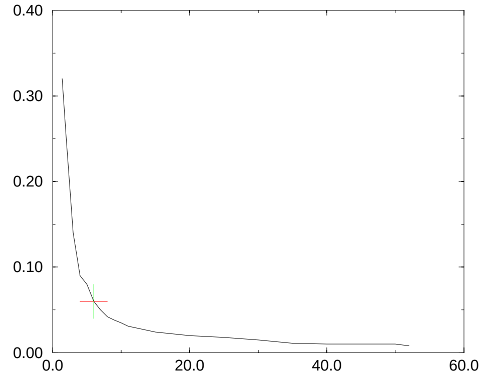

Figure 20: *Graph of compression-ratio as a function of approximation-error for spline reconstruction of Africa*

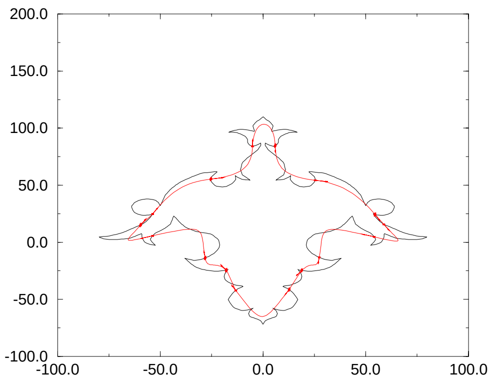

Figure 19: *Reconstruction of abstract design at* $\sigma = 22$

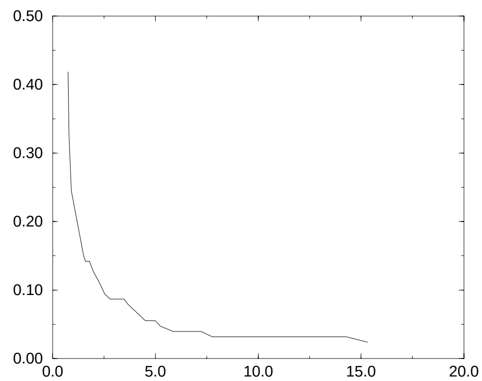

Figure 21: *Graph of compression-ratio as a function of approximation-error for spline reconstruction of Hokaido*

$O(nk)$ where k is the size of the convolution filter and n is the number of points on the input contour.

7. Conclusions

A novel technique was presented for automatic fitting of digitised contours at multiple scales through the curvature scale space technique used in conjunction with Hermite curves.

Hermite curves were used since each Hermite curve is defined by two endpoints and the tangent vectors at those points. No points external to the input contour are required for Hermite curves. Hermite endpoints were defined as consecutive curvature zero-crossing points recovered at multiple scales using the CSS method. Hermite tangent vectors were also determined using the CSS technique. Contour data compression was achieved by storing only the endpoints and the tangent vectors needed by the Hermite curves in order to arrive at an approximate reconstruction of the original contour.

Approximation Error and Compression Ratio were also computed at each scale. The graph of compression ratio as a function of approximation error was smoothed to remove noise and small fluctuations. The *bending point* of that function was then defined as the largest maximum of its second derivative. The bending point was considered as the boundary between the mostly vertical and the mostly horizontal segments of the graph. It was used for automatic selection of an optimal scale for contour data reconstruction. As a result, the user will not have to set any parameters to use this technique.

Note that this approach can also be extended to the 3-D

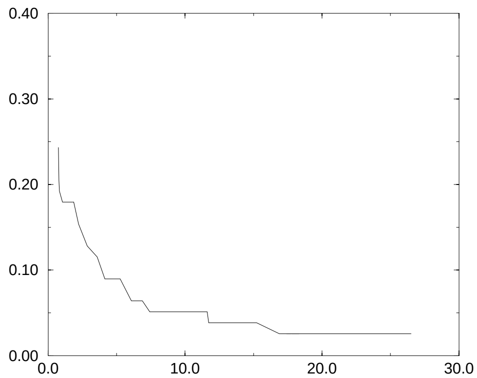

Figure 22: *Graph of compression-ratio as a function of approximation-error for spline reconstruction of abstract design*

case since the curvature scale space method has been generalised to 3-D surfaces recently [11, 18, 4, 5].

References

1. L. S. Davis. Understanding shape: Angles and sides. *IEEE Trans Computers*, C-26:236–242, 1977. 1

2. J. D. Foley and A. VanDam. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley, 1984. 2

3. J. D. Foley, A. VanDam, S. K. Feiner, J. F. Hughes, and R. L. Phillips. *Introduction to Computer Graphics*. Addison-Wesley, 1997. 2, 3

4. N. Khalili, F. Mokhtarian, and P. Yuen. Free-form surface description in multiple scales: Extension to incomplete surfaces. In *Proc International Conference on Computer Analysis of Images and Patterns*, pages 293–300, Ljubljana, Slovenia, 1999. 7

5. N. Khalili, F. Mokhtarian, and P. Yuen. Recovery of curvature and torsion features from free-form 3-D meshes at multiple scales. In *Proc Asian Conference on Computer Vision*, pages 1070–1075, Taipei, Taiwan, 2000. 7

6. L. J. Latecki and R. Lakamper. Contour-based shape similarity. In *Proc International Conference on Visual Information Systems*, pages 617–624, Amsterdam, 1999. 1

7. A. K. Mackworth and F. Mokhtarian. Scale-based description of planar curves. In *Proc Canadian Society for Computational Studies of Intelligence*, pages 114–119, London, Ontario, 1984. 2

8. F. Mokhtarian. Fingerprint theorems for curvature and torsion zero-crossings. In *Proc IEEE Conference on Computer Vision and Pattern Recognition*, pages 269–275, San Diego, CA, 1989. 1

9. F. Mokhtarian. Multi-scale, torsion-based shape representations for space curves. In *Proc IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–661, New York City, 1993. 1

10. F. Mokhtarian. A theory of multi-scale, torsion-based shape representation for space curves. *Computer Vision and Image Understanding*, 68(1):1–17, 1997. 1

11. F. Mokhtarian, N. Khalili, and P. Yuen. Multi-scale 3-d free-form surface smoothing. In *Proc British Machine Vision Conference*, pages 730–739, Southampton, UK, 1998. 7

12. F. Mokhtarian and A. K. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans Pattern Analysis and Machine Intelligence*, 8(1):34–43, 1986. 2

13. F. Mokhtarian and A. K. Mackworth. A theory of multi-scale, curvature-based shape representation for planar curves. *IEEE Trans Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992. 1, 2

14. T. Pavlidis. Polygonal approximations by newton's method. *IEEE Trans Computers*, C-26:800–807, 1977. 1

15. E. Persoon. Shape discrimination using fourier descriptors. *IEEE Trans Systems, Man and Cybernetics*, 7:170–179, 1977. 1

16. Y. Rui, A. C. She, and T. S. Huang. Modified fourier descriptors for shape representation - a practical approach. In *International Workshop on Image DataBases and Multi-Media Search*, pages 115–122, Amsterdam, The Netherlands, 1996. 1

17. P. J. Schneider. An algorithm for automatically fitting digitized curves. In A. S. Glassner, editor, *Graphics Gems*, pages 612–626. Academic Press Professional, 1990. 1

18. P. Yuen, F. Mokhtarian, and N. Khalili. Multi-scale 3-d surface description: Open and closed surfaces. In *Proc Scandinavian Conference on Image Analysis*, pages 303–310, Greenland, 1999. 7