1-1-1997

Edge image description using fractal interpolation

P Motallebi
University of Wollongong

P O. Ogunbona
University of Wollongong, philipo@uow.edu.au

Abstract

Edge images derived from compressed image databases are described using fractal techniques. The proposed method is able to give affine transformation-invariant description suitable for use in a query-by-example database application. Comparison among the proposed method, polynomial interpolation and spline interpolation is given. It is concluded that fractal interpolation can give a compact description of image contours and is able to cope with random perturbation of the coordinates of the contour points by as much as 25 percent.

Keywords

edge, fractal, image, interpolation, description

Disciplines

Physical Sciences and Mathematics

Publication Details

Motallebi, P. & Ogunbona, P. (1997). Edge image description using fractal interpolation. IEEE Region 10th Annual International Conference: Speech Image Technologies for Computing and Telecommunications Proceedings (pp. 553-556). IEEE.

EDGE IMAGE DESCRIPTION USING FRACTAL INTERPOLATION

P. Motallebi and P. O. Ogunbona
School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
NSW 2522
payam@uow.edu.au, pogna@elec.uow.edu.au

ABSTRACT

Edge images derived from compressed image databases are described using fractal techniques. The proposed method is able to give affine transformation-invariant description suitable for use in a query-by-example database application. Comparison among the proposed method, polynomial interpolation and spline interpolation is given. It is concluded that fractal interpolation can give a compact description of image contours and is able to cope with random perturbation of the coordinates of the contour points by as much as 25 percent.

1. INTRODUCTION

The migration of telecommunications networks towards a wireless access paradigm for all services poses several challenges to such application as image and video databases. There is a need to efficiently access distributed image and video databases from hand-held low-resolution mobile terminals. An advantage of using a mobile terminal in accessing fixed telecommunication networks is the freedom given to the user to roam about and at the same time have access to required information. This brings about the concept of personal communication "anywhere anytime".

The amount of visual data available on the Internet begs for a human-intuitive approach to data discovery and manipulation. Currently, the storage of visual data is motivated by the need to economize bandwidth and storage requirements. However, in order to perform several of the functions of an image or video database a different paradigm is required. Storage techniques should allow high compression and also cater for image browsing, content-based query and possibly manipulation of the image content to suit users' requirements. Several of the currently available compression methods cannot support a content-based query without tremendous computation because the encoded data units are not the objects on which the query is based. It is still not clear how a compression method based on objects will serve all purposes, especially the fidelity of the reconstructed image. A possible model is to store the compressed images separately from an index of features that will facilitate database functions. The features may contain for example descriptions of objects in each image along with some information regarding the relationship among the objects. A disadvantage of this model is that a huge amount of storage may be required by the index itself. Furthermore, there remains the problem of how to quickly search through a large index. There are data structures that can be used to alleviate this problem. Santini and Jain [1] proposed a *fuzzy feature set contrast* similarity matching scheme. Alternatively, Arbter used *affine-invariant Fourier Descriptors* [2]. The work of Watt and Nimmo-Smith deals with modeling the image contours in a way similar to the human visual system [3]. In [4], wavelet image compression is proposed as a method that may satisfy the requirement of good compression and also facilitate image database functions without an explicit creation of an index. While this model may obviate the need for a separate index it restricts the query method to "query by example". Edge images are generated from the high frequency decompositions of the image and used as index on which searches are carried out. In the method described in [4], Hausdorff distance is crudely used to measure the similarity between a user query, which in this case is a sketch, and the index created from the high frequency image decompositions. A more appropriate solution should find a descriptor for the edge image derived from both the sketch query and the target image. Of course the edge image needs to be processed to remove the "noisy" streaks.

In this paper, interpolation methods are investigated as means of providing descriptors for an edge image in order to reduce the dimensionality of the search space and improve the flexibility of the search. Historically, polynomial and spline interpolation techniques have been used to interpolate data sets. However, fractal interpolation techniques have much to offer as will be shown in the following.

Polynomial and spline interpolation algorithms are presented in Sections 2 and 3 respectively, with Section 4 describing a fractal interpolation scheme. The results obtained are presented and discussed in Section 5 and concluding remarks are given in Section 6.

2. POLYNOMIAL INTERPOLATION

Any two points can be represented by a unique line, which passes through them. Any three points can be represented by a quadratic, which will pass through them. Thus, any N points can be represented by an interpolating polynomial of order (N-1). This polynomial can be obtained by solving Lagrange's classical formula:

$$P(x)=\frac{(x-x_2)(x-x_3)\dots(x-x_N)}{(x_1-x_2)(x_1-x_3)\dots(x_1-x_N)}y_1 + \frac{(x-x_2)(x-x_3)\dots(x-x_N)}{(x_2-x_2)(x_2-x_3)\dots(x_2-x_N)}y_2 + \dots + \frac{(x-x_2)(x-x_3)\dots(x-x_{N-1})}{(x_N-x_2)(x_N-x_3)\dots(x_N-x_{N-1})}y_N \quad (1)$$

In practice, it is often easier to implement *Neville's algorithm*, to produce the same interpolating polynomial [5].

Polynomial interpolation works well for small data sets, however, as the order of the polynomial, and correspondingly the complexity, increase linearly with the data set size, they begin to fluctuate wildly and are computationally intensive.

3. SPLINE INTERPOLATION

One solution to the ever increasing complexity of a single N-1 order polynomial interpolating N points is to use N-1 cubic polynomials interpolating all N-1 point pairs:

$$s_i = a_i x^3 + b_i x^2 + c_i x + d_i \tag{2}$$

Thus, after computing the required a, b, c and d coefficients for all spline segments from the given data set, the constraints of the spline give rise to a set of simultaneous equations which can be solved to obtain the coefficients.

Clearly, this solution results in a much more accurate descriptor for the data set while simultaneously reducing the computational complexity significantly.

4. FRACTAL INTERPOLATION

With the use of common Euclidean methods, as presented in the previous two sections, the randomness of image contours is not well modeled. Neither the use of straight line nor polynomial approximation of these image contours gives an exact match to the contour. Splines do result in a bettor descriptor, however, obtaining N-1 quadratics for N points is still computationally cumbersome.

As opposed to the approximate methods espoused by Euclidean geometry, fractal interpolation uses an exact approach. Rather than the interpolation function approximating the points it is meant to pass through, a fractal interpolation function actually passes through each and every point; the function is made to 'fit' the data [6]. This may not seem an important consideration, but when faced with the random nature of images (and in turn their contours), modelling these with Euclidean methods is not satisfactory. Fractal interpolation functions can moreover be represented by simple functions, thus reducing the computational overhead and possibly the bandwidth required for transmission of a query to a remote database.

4.1. Fractal Interpolation Functions

For a set of data points extracted from an image contour, of the form:

$$\{(x_i, F_i) \in R^2 : i = 0,1,2,\ldots,N\} \tag{3}$$

where

$$x_0 < x_1 < x_2 < x_3 < \cdots < x_N$$

The fractal interpolation function is a continuous function $f : [x_0, x_N] \to R$ such that [6]:

$$f(x_i) = F_i \qquad \text{for } i = 0,1,2,\ldots,N \tag{4}$$

Consider an iterative function system (IFS) of the form $\{R^2; w_n, n = 1,2,\ldots,N\}$ where the maps are affine transformations of the special structure:

$$w_n \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_n & 0 \\ c_n & d_n \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_n \\ f_n \end{pmatrix} \tag{5}$$

The maps (transformations) are specified by five real numbers - a, b, c, d, e and f. These must all obey the four linear equations:

$$\begin{aligned} a_n x_0 + e_n &= x_{n-1} \\ a_n x_N + e_n &= x_n \\ c_n x_0 + d_n F_0 + f_n &= F_{n-1} \\ c_n x_N + d_n F_N + f_n &= F_n \end{aligned} \tag{6}$$

By choosing d as the free parameter in the transform, the other variables can be obtained using:

$$\begin{aligned} a_n &= \frac{(x_n - x_{n-1})}{(x_N - x_0)} \\ e_n &= \frac{(x_N x_{n-1} - x_0 x_n)}{(x_N - x_0)} \\ c_n &= \frac{(F_n - F_{n-1})}{(x_N - x_0)} - \frac{d_n (F_N - F_0)}{(x_N - x_0)} \\ f_n &= \frac{(x_N F_n - x_0 F_{n-1})}{(x_N - x_0)} - \frac{d_n (x_N F_N - x_0 F_0)}{(x_N - x_0)} \end{aligned} \tag{7}$$

Coefficient d is called the vertical scaling factor, the reason it is chosen as the free variable is that it only affects the scale of the output of the transformation [6]. Thus the only coefficients actually forming a descriptor for the interpolating function are a, c, e and f. It can be shown that this IFS is in fact the same graph as the interpolation function f given in (4) over the given interval.

4.1.1. *Calculation Methods*

The calculation of the fractal interpolation functions from the edge contour segments can be done using either the Random Iteration algorithm or the Deterministic algorithm. The random iteration algorithm is used here due to its simplicity and the ease with which it can be visualized.

4.1.2. Random Iteration Algorithm

Once the IFS maps have been calculated, the Random Iteration Algorithm can be used to calculate the points, through which the interpolating IFS pass.

5. RESULTS AND DISCUSSION

Simulation results based on the edge contour "Car", shown in Figure 1 are presented in Tables 1 through 3.

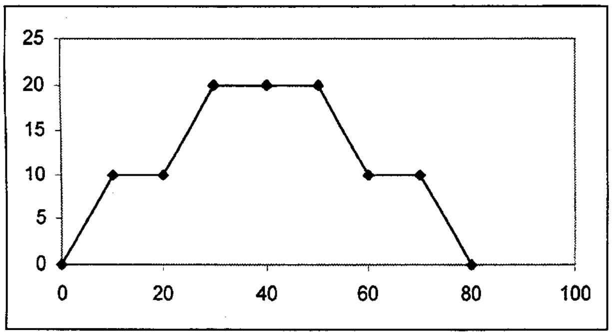

Figure 1: Test Contour "Car"

Table 1: Polynomial Coefficients derived from Contour "Car"

Original Contour	X and y Offset	Y Offset	Random Perturbation
-0.003630	-768.381592	9.996026	-374.638275
15.252504	196.559830	15.272578	144.610748
-3.288407	-19.344919	-3.293361	-18.992006
0.286451	0.998568	0.286915	1.213959
-0.012810	-0.030013	-0.012832	-0.043205
0.000322	0.000545	0.000323	0.000902
-0.000005	-0.000006	-0.000005	-0.000011
0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000

Table 2: Spline Coefficients derived from Contour "Car"

Original Contour	x and y offset	y offset	Random Perturbation
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.20797	0.20797	0.20797	0.082734
-0.231879	-0.231879	-0.231879	-0.150938
0.119547	0.119547	0.119547	0.041017
-0.246307	-0.246307	-0.246307	-0.07313
0.265682	0.265682	0.265682	0.011501
-0.21642	-0.21642	-0.21642	0.027125
0.0	0.0	0.0	0.0

Table 3: Fractal Coefficients derived from Contour "Car"

Coeff	Original Contour	x and y Offset	y Offset	Randomly Perturbed
C	8.89E-09	8.89E-09	8.89E-09	8.89E-09
D	0	-0.05111	-0.05111	-0.01674
E	31.11111	40	31.11111	31.11111
F	11.11111	19.48889	18.46667	14.05333

Table 1 is the result obtained from using polynomial interpolation. It can be seen that with a small offset in the y direction, the system retains a similar description. However, with only a small perturbation or an offset in both x and y directions, the scheme fails. Figure 2 presents the polynomial descriptor for the contour. It is seen that even with the original descriptor, the system fails to accurately match the data points.

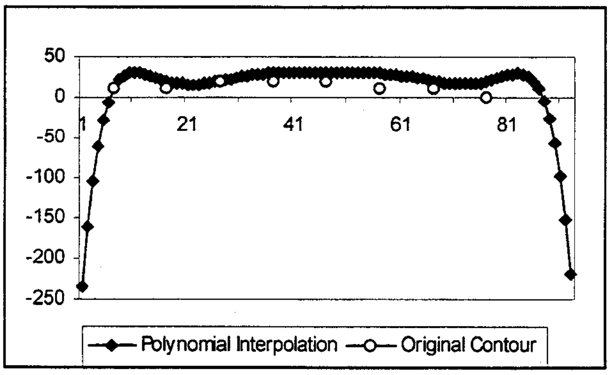

Figure 2: Polynomial Interpolation of the contour

As would be expected in the spline scheme, (see Table 2), an offset in either/both direction does not affect the quadratic equation coefficients. Unfortunately, a small perturbation causes the scheme to deteriorate. As seen in

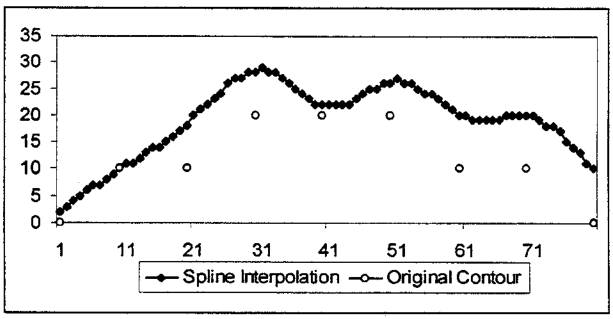

Figure 3: Spline Interpolation of the contour

Figure 3, the spline interpolant does mach the data set closely. It is also far more stable within the data range, unlike the wild fluctuation present in the polynomial interpolant.

Table 3 presents the fractal interpolation results. Each row consists of the mean value of the respective coefficient (a, c, e or f). Although no correlation between the actual coefficients has been taken into account when considering the mean values of each, the results obtained point to as flexible yet stable system. It should be pointed out that the perturbation applied to the coordinates of the contour was as much as 25% in all cases.

It is seen from Figure 4 that the rendering of the Fractal interpolant (the attractor of the IFS), looks nothing like the original contour. However, this attractor is unique to this set of data points, thus providing an accurate descriptor for the contour.

6. CONCLUSION

The paper has investigated the use of fractal geometry in describing image contours. Results obtained so far indicate its promise and elucidate the difficulty associated with using Euclidean geometry in describing image contours. Both polynomial and spline interpolation techniques failed in coping with perturbation; a condition that is most likely to occur in practice. The fractal interpolation approach has been shown to better model the random nature of image contour. The descriptors are more flexible and stable than the Euclidean methods considered. Further, at lower sampling rates, the fractal approach continues to provide good results.

Further work needs to be done in order to make fractal interpolation a practical image contour descriptor. Work is continuing in adapting the two-dimensional fractal coding technique of Jacquin [7] to an image contour description.

This approach to image database searches should find applications in interactive multimedia searches. Due to the added bonus of low information content within the search descriptor, the system also lends itself to mobile systems.

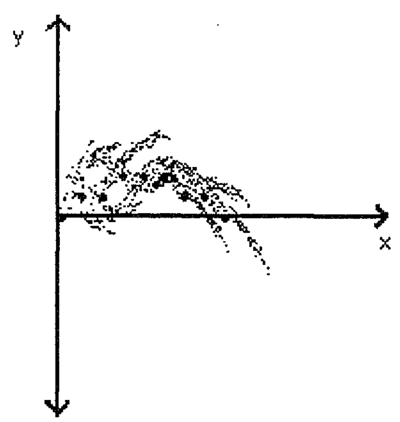

Figure 4: Fractal Interpolation of the contour

7. REFERENCES

[1] S. Santini and R. Jain, "The Graphical Specification of Similarity Queries," Technical Report, Department of Computer Science and Engineering, University of California, San Diego.

[2] K. Arbter, "Affine-Invariant Fourier Descriptors", from Pixels to Features, J.C. Simon (ed.), Elsevier Science Publishers, 1989

[3] R. J. Watt and I. Nimmo-Smith, "Representing the Shape of Image Contours", from Pixels to Features, J.C. Simon (ed.), Elsevier Science Publishers, 1989

[4] P. Ogunbona and P. Sangassapaviriya, "Content-based Retrieval from Compressed Image Databases," IEEE Data Compression Conference, 1997

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, "Numerical Recipes in C", Second Edition, Cambridge Press, 1992.

[6] M. Barnsley, *Fractals Everywhere*, Academic Press, Inc., London, 1988.

[7] A. Jacquin, "Fractal Image Coding: A Review," IEEE Proceedings, Vol. 81. No. 10, pp 1451 – 1465, 1993.