One-dimensional B–B polynomial and Hilbert scan for graylevel image coding

Sambhunath Biswas*

Machine Intelligence Unit, Indian Statistical Institute, 203 Barrackpore Trunk Road, Calcutta 700 108, India

Received 4 October 2002; accepted 12 September 2003

Abstract

Effectiveness of one-dimensional Bezier–Bernstein polynomial has been studied for compression of graylevel images. Both raster and Hilbert scan have been used for this purpose. Two different algorithms for one-dimensional approximation and encoding of Hilbert images provide evidence of better compression ratio compared to that for raster scanned images. Comparison with the result of an existing algorithm has also been performed to examine the effectiveness of our methods.

Keywords: Bezier–Bernstein polynomial; Hilbert scan; Raster scan; Approximation; Compression; Huffman coding

1. Introduction

Bezier approximation technique [1] which uses Bernstein polynomial as the blending function is well known in the field of computer graphics for its speed of computation and axis independence property. The present work is an investigation about the use of one-dimensional Bezier–Bernstein (B–B) polynomial in gray tone image data compression using the conventional raster scanned and the nonconventional Hilbert scanned images. To carry out this investigation we, first of all, have examined if the conventional way of approximating an image, in a raster scan, by B–B polynomial provides any advantage from the data compression standpoint. For this, we have considered an entire row (or column) of an image as a single segment for its approximation. From the approximation theorem of Bernstein [2] it is evident that, for a given error, the order of the polynomial increases with the maximum gray value present in the segment. Therefore, if the maximum gray value in an image is very large, the order of the polynomial also becomes large. Consequently, it introduces a large number of control or guiding pixels for approximation. As a result, approximation becomes computationally expensive and the segment generation also becomes slow. This makes the conventional way of approximating an image for its compression inconvenient.

To serve our purpose, we have developed a modified version of the approximation technique. Here, we emphasize on the local control of data points (pixels) instead of minimizing the global squared error. An absolute error criterion has been developed to keep the absolute error within a bound. Also, for the sake of data compression, we have chosen the second-order polynomial.

Based on the modified concept of approximation, we have proposed two algorithms for selecting segments from an image for approximation. The first algorithm uses an error bound to partition the pixel data set into different segments through approximation. The partition can be made into either arc or line segments depending on the choice of approximation for both the raster and Hilbert scanned images. These segments are then coded in the subsequent stage. The second algorithm, on the other hand, considers, for a raster scanned image, a row (or column) of pixels as a space curve on an intensity surface and separates out the small deflection curve segments on the basis of a homogeneity criterion. A

Hilbert scanned image, however, for a raster scanned square graylevel image can be considered as a space curve with length equal to the square of the length or width of the raster scanned image. Note that the size of the image provides the resolution of the Hilbert curve. Due to the neighborhood property of the Hilbert scan, long homogeneous segments are found to be approximated; resulting in less number of segments for encoding than that for a raster scanned image. Consequently, the compression ratio, is found to be higher.

The performance of the algorithms is tested on a set of input images. The discriminating features of the proposed two algorithms are also discussed. Finally, the results are compared to that of the algorithm proposed by Kamata et al. [3].

2. Hilbert scanned image

Hilbert curve is one of the space filling curves, published by G. Peano in 1890. The Hilbert curve has a one-to-one mapping between an n-dimensional space and a one-dimensional space which preserves point neighborhoods as much as possible. There are many applications of this curve. A review on the applications of Hilbert curve can be found in Refs. [4,5]. Some of the researchers have already used this curve in the area of image processing. Reported works in the area of image compression can be found in Refs. [6–10,3,11–14].

Let R^n be an n-dimensional space. The Peano curve published in 1890 is a locus of points $(y_1, y_2, \ldots, y_n) \in R^n$ defined by continuous functions $y_1 = \chi_1(v)$, $y_2 = \chi_2(v), \ldots, y_n = \chi_n(v)$, $(v \in R^1)$ where $0 \leqslant y_1, y_2, \ldots, y_n < 1$ and $0 \leqslant v < 1$. It was an analytical solution of a space filling curve. In 1891, Hilbert drew a curve having the space filling property in R^2. Hilbert found a one-to-one mapping between segments on the line and quadrants on the square. Fig. 1 shows the Hilbert curve with different resolutions. Hilbert scan considers the positions on the square through which the curve passes. Therefore, a Hilbert scanned image or simply a Hilbert image is a one-dimensional image with its pixels identical to those through which the curve passes. The merit of this curve is to pass through all points in a quadrant and always to move to the neighboring quadrant. Thus, it maintains the neighborhood property.

Hilbert curve has a one-to-one mapping between an n-dimensional space and a one-dimensional space which preserves point neighborhoods as much as possible. A Hilbert image or a Hilbert scanned image is a set of ordered pixels which can be obtained by scanning the positions of pixels through which this curve passes.

2.1. Construction of Hilbert curve

Construction of Hilbert curve, following Hilbert's ideas, considers a square which is filled by the curve. Since our objective is to scan a gray tone image and produce a Hilbert scanned image for the study of image compression, we shall explain the basic philosophy behind construction of the curve and provide a scheme through which real life images can be converted into Hilbert scanned images. We also provide a scheme for inverse mapping to get back gray tone images from the Hilbert scanned images.

First of all, we divide the square as shown in Fig. 2 into four quarters. The construction starts with a curve H_0 which connects the centers of the quadrants by three line segments. Let us assume the size of the segments to be 1. In the next step we produce four copies (reduced by $\frac{1}{2}$) of this initial stage and place the copies into the quarters as shown. Thereby we rotate the first copy clockwise and the last one counterclockwise by 90°. Then we connect the start and end points of these four curves using three line segments

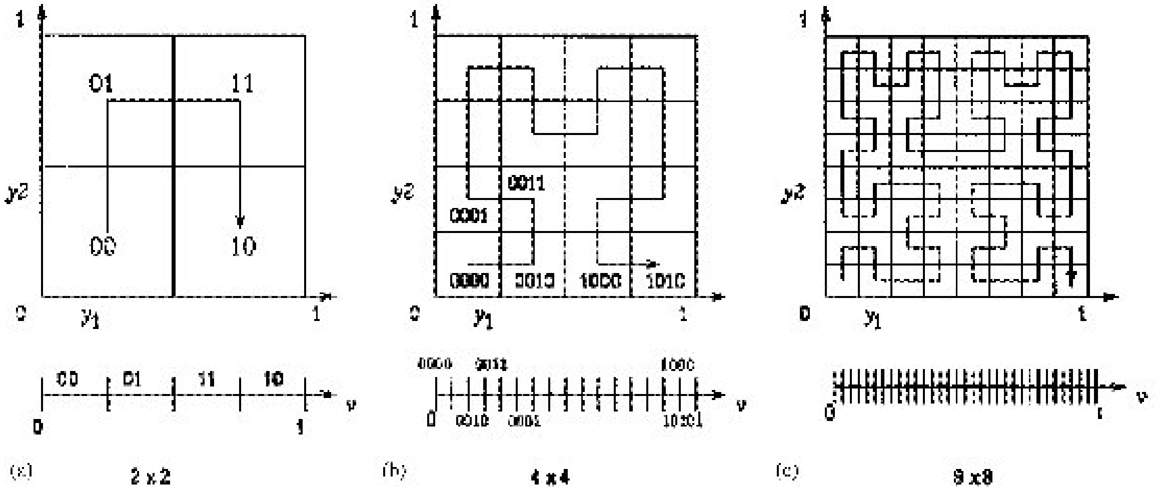

Fig. 1. Hilbert curve with different resolutions.

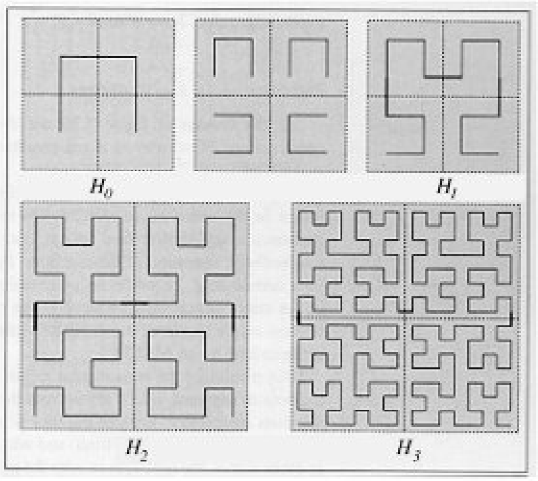

Fig. 2. Four stages of the Hilbert curve.

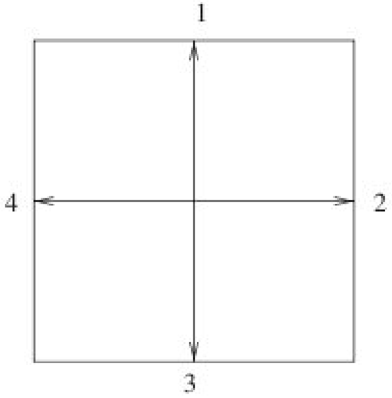

Fig. 3. 4-connected chain code.

(of size $\frac{1}{2}$) as shown and call the resulting curve H_1. In the next step we scale H_1 by $\frac{1}{2}$ and place four copies into the quadrants of the square as in step one. Again we connect using three line segments (now of size $\frac{1}{4}$) and obtain H_2. This curve contains 16 copies of H_0, each of size $\frac{1}{4}$. As a general rule, in step n we obtain H_n from four copies of H_{n-1} which are connected by three line segments of length $1/2^n$ and this curve contains $4n$ copies of H_0 (scaled by $1/2^n$). A different approach (normally known as the L-system approach) for construction of the Hilbert curve can be found in Ref. [15].

Since, in image compression problem we are concerned with mapping gray tone images of different size into corresponding Hilbert scanned images, we construct Hilbert curve with different resolutions using Freeman's four connected chain code. The chain code is as shown in Fig. 3. Using this chain code the curves H_0, H_1 and H_2 are, respectively, as given below:

H_0: 123.
H_1: 214,1,123,2,123,3,432.
H_2: 123221412144341,1,214112321233432,2,
214112321233432, 3,341443234322123.

2.1.1. *Inverse mapping*

Since the size of the raster scanned graylevel image is known, one can always generate the 4-connected chain code for the corresponding Hilbert image. Now given the Hilbert image and the chain code, one can quickly get back the original graylevel image.

3. Shortcomings of Bernstein polynomial and error of approximation

Bernstein polynomial is a powerful tool to approximate a continuous function within any degree of accuracy. It uses the global information while approximating a function and the order of the polynomial increases with accuracy in approximation. The Bernstein polynomial of degree p is

$$B_{ip}(t) = \sum_{i=0}^{p} f\left(\frac{i}{p}\right)\phi_{ip}(t) \tag{1}$$

for approximating a function $f(t)$. Here $f(t)$ is defined and finite on the closed interval [0, 1]. Also

$$\phi_{ip}(t) = \begin{pmatrix} p \\ i \end{pmatrix} t^i(1-t)^{p-i}$$

and

$$\begin{pmatrix} p \\ i \end{pmatrix} = \frac{p!}{(p-i)!(i)!}$$

with $i = 1, 2, \ldots, p$.

The order p of the Bernstein polynomial $B_{ip}(t)$ satisfies the inequality

$$\frac{k_m}{\varepsilon\,\delta^2} < p \tag{2}$$

in order to have the error of approximation less than ε, where k_m is the maximum value of the approximating function $f(t)$ in the interval [0, 1]. δ is a positive number such that for points $t_1, t_2 \in (0, 1)$

$$|f(t_1) - f(t_2)| < \frac{\varepsilon}{2} \quad \text{whenever} \quad |t_1 - t_2| < \delta.$$

Since a graylevel image in a raster scan can be approximated either row wise or column wise, it appears from the inequality (2) that the order of the approximating polynomial may be different for different rows (or columns) depending on the value of k_m (assuming ε and δ do not change appreciably).

As an illustration, let us consider the case of approximating, row wise, a 32 level $(0,1,\ldots,31)$, image of size 32×32. If a row has its maximum value $k_m = 31$ then for $\varepsilon = 1$, (i.e. one unit error in gray value) $p > \frac{31\times31\times31}{29\times29} \approx 35.42$, i.e., 36. Note that the maximum value of $\delta = \frac{29}{31}$, because $|t_1 - t_2| = \frac{1}{31} - \frac{30}{31}$ $(t_1, t_2 \in (0,1)$. Therefore, for $k_m = 31$ one can choose p to be equal to 36.

On the other hand, if $k_m = 2$ then $m \approx 1.06$, i.e., $p = 2$. $k_m = 2$ means some of the graylevel values in the row are same and is equal to 1. Since in a gray image it is very likely to have the maximum value anywhere in each row, the order may be as high as the maximum gray level in the image. This makes the method ineffective.

4. Proposed approximation technique

An attempt is made in this section to develop an approximation scheme keeping the order of the polynomial equal to two. As a result, the reconstruction time will be less and the coding scheme will be simple. But due to the selection of the polynomial order of two, the amount of error ε, as expected, will be significantly high. In order to circumvent this, a modification of the conventional approximation scheme based on B–B polynomial is proposed. This leads to the formulation of a new scheme by which it is also possible to obtain any degree of accuracy in approximation.

Given n points, the approximation algorithm requires $(n-2)$ unique quadratic B–B polynomials for their representation. Unlike the method described in Section 3, the scheme, proposed here, decomposes a row (column) either into a single gray segment or into a number of segments so as to enable them to be approximated properly. An error bound has been defined which guides the process of segmentation.

4.1. (B–B) polynomial

Eq. (1) which represents a pth degree Bernstein polynomial for approximating a function $f(t)$, $0 \leqslant t \leqslant 1$ can be written as

$$B_{ip}(t) = \phi_{op}(t) f(0) + \phi_{1p}(t)\, f\left(\frac{1}{p}\right) + \phi_{2p}(t) f\left(\frac{2}{p}\right) + \cdots + \phi_{pp}(t) f(1).$$

Let v_i represent a point in a multi-dimensional space and that $v_i = f(i/p)$. Thus $B_{ip}(t)$ becomes

$$B_{ip}(t) = \sum_{i=0}^{p} \phi_{ip}(t)\, v_i. \tag{3}$$

Eq. (3) can be viewed as a vector-valued Bernstein polynomial and it approximates a polygon with vertices v_i and t in $[0,1]$. $B_{ip}(t)$ is thus seen to generate a space curve. For $p = 2$, the quadratic B–B polynomial (dropping the index i in B_{ip}) is

$$B_2(t) = (1-t)^2 v_o + 2t(1-t)v_1 + t^2 v_2. \tag{4}$$

4.2. Algorithm 1: approximation criteria of $f(t)$

In order to develop an approximation technique, let us first of all formulate the key criteria associated with this technique.

Let us assume $(n-2)$ quadratic B–B polynomials for the representation of n data points such that

$$f(t_i) = B_2^i(t_i), \quad i = 1, 2, 3, \ldots, n-2,$$

where $B_2^i(t_i)$ is the value of the ith quadratic B–B polynomial at the point t_i and is given by

$$B_2^i(t_i) = (1-t_i)^2 v_o + 2t_i(1-t_i)v_1^i + t_i^2 v_2. \tag{5}$$

Let

$$B_2^1(0) = B_2^2(0) = \cdots = B_2^{n-2}(0) = v_o$$

and

$$B_2^1(1) = B_2^2(1) = \cdots = B_2^{n-2}(1) = v_2.$$

In other words, at the end supports all the quadratic B–B polynomials are assumed to be identical. The points at end supports are also the vertices of the underlying $(n-2)$ polygons. The second vertex (also called the control point) v_1^i of the $(n-2)$ polynomials are all different. This is shown in Fig. 4.

From Eq. (5), the second control point of the ith polynomial can be computed as

$$v_1^i = \frac{B_2^i(t_i) - (1-t_i)^2 v_o - t_i^2 v_2}{2\, t_i\, (1-t_i)}. \tag{6}$$

Let

$$\overline{v_1} = \frac{1}{n-2} \sum_{i=1}^{n-2} v_1^i$$

be the average value of the second control points for $(n-2)$ polynomials and let the corresponding B–B polynomial with control points v_o, $\bar{v}_1$, and v_2 be $\overline{B_2}(t_i)$. The discrete form of $\overline{B_2}(t_i)$ can be written as

$$\overline{B_2}(t_i) = (1-t_i)^2 v_o + 2t_i(1-t_i)\bar{v}_1 + t_i^2 v_2. \tag{7}$$

From Eqs. (5) and (7)

$$|\bar{B}_2(t_i) - B_2^i(t_i)| = |\bar{v}_1 - v_1^i| \times 2t_i(1-t_i). \tag{8}$$

Therefore,

$$\begin{aligned} |\overline{B_2} - B_2^i|_{max} &= |\overline{v_1} - v_1^i|_{max} \times [2t_i(1-t_i)]_{max} \\ &= |\overline{v_1} - v_1^i|_{max} \times \frac{1}{2}. \end{aligned} \tag{9}$$

Note that $t_i(1-t_i)$ is always positive. Similarly,

$$|\overline{B_2} - B_2^i|_{min} = |\overline{v_1} - v_2^i|_{min} \times [2t_i(1-t_i)]_{min}. \tag{10}$$

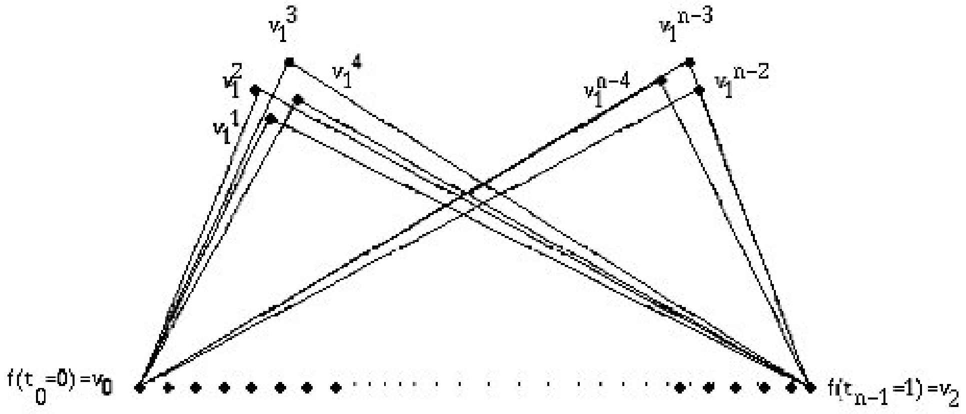

Fig. 4. Second control points due to a sequence of quadratic polynomials.

The expression $t_i(1-t_i)$ has maximum at $t=\frac{1}{2}$ and the value falls symmetrically on either side as t moves away from $\frac{1}{2}$. Since $t_i \in (0,1)$, the expression $2t_i(1-t_i)$ is minimum for the possible minimum/maximum value of t_i. For equally spaced data points, the minimum possible value of t_i is $1/(n-1)$ and the maximum possible value of t_i is $(n-2)/(n-1)$. In either case, $[2t_i(1-t_i)]_{min} = 2(n-2)/(n-1)^2$ With this,

$$|\overline{v_1} - v_1^i|_{min} = \frac{(n-1)^2}{2(n-2)}|\overline{B_2} - B_2^i|_{min}$$

$$= \frac{(n-1)^2}{2(n-2)}\varepsilon_{min} \tag{11}$$

and

$$|\overline{v_1} - v_1^i|_{max} = 2|\overline{B_2} - B_2^i|_{max}$$

$$= 2\varepsilon_{max}, \tag{12}$$

where $|\overline{B_2} - B_2^i|_{min} = \varepsilon_{min}$ and $|\overline{B_2} - B_2^i|_{max} = \varepsilon_{max}$ are, respectively, the minimum and maximum absolute errors in approximating a function $f(t)$ and $t_i(1-t_i)$ is maximum at $t_i = \frac{1}{2}$. It is straightforward to observe from Eqs. (11) and (12) that

$$|\overline{v_1} - v_1^i|_{min} \leqslant |\overline{v_1} - v_1^i| \leqslant |\overline{v_1} - v_1^i|_{max}, \tag{13}$$

or,

$$\frac{2(n-2)}{(n-1)^2} \leqslant |\overline{v_1} - v_1^i| \leqslant 2\varepsilon_{max}. \tag{14}$$

Similarly,

$$\varepsilon_{min} \leqslant |\overline{B_2} - B_2^i| \leqslant \varepsilon_{max}. \tag{15}$$

Therefore, inequality (13) tells that the function $f(t_i) = B_2^i(t_i), i = 1,2,\ldots,n-2$ can be approximated by $\overline{B_2}(t)$ with an error inequality expressed in Eq. (15).

Example. In order to illustrate the method of approximation, let us consider a sequence of 38 data points. The maximum and minimum errors, ε_{max} and ε_{min}, for approximation are chosen 10.0 and 0.000001, respectively. The approximation partitions the data set into three segments. The beginning and end point of each partitioned segment are approximated with zero error whereas all other data points are approximated with errors between ε_{min} and ε_{max}. Note that, the approximation may have much lower error than ε_{max}. Table 1 shows the approximation for only the first partition of that data set to provide an idea.

Table 1
Illustration of approximation techniques

No. of segments	Original data	Approx. values	Error in approximation	$\bar{v}_1$
1	140	140.0000	0.0000	155.8432
	140	143.1117	−3.1117	
	140	146.1098	−6.1098	
	157	148.9941	8.0059	
	157	151.7647	5.2353	
	162	154.4216	7.5784	
	157	156.9647	0.0353	
	157	159.3941	−2.3941	
	157	161.7098	−4.7098	
	166	163.9117	2.0883	
	166	166.0000	0.0000	

4.3. Algorithm 2

Here each row (column) of pixels has been viewed as a space curve and is segmented depending on the homogeneity among the pixels. Each segment is then approximated by the modified approximation scheme. Here, we consider

$$\overline{v_1} = \frac{1}{n}\sum_{i=1}^{n} v_1^i, \tag{16}$$

where v_1^is are computed using Eq. (6).

Since the segments are all homogeneous, approximation for coding depends on the homogeneity parameter and not on any external approximation parameter as required in the case of Algorithm 1. The approximation is faster. Since for each homogeneous segment v_1^is are averaged for $\bar{v}_1$, every approximation has its own ε_{max} that varies from segment to segment.

4.3.1. *Small deformation space curve and the concept of homogeneity*

An image may be considered as an intensity surface with surface contours representing the space curves along the rows and columns of the image. Note that for any curve Γ, the amount of information contained in it can be represented by its curvature vector $\vec{k_v}$ or by any other related quantity. The curvature vector $\vec{k_v}$ is defined as

$$\vec{k_v} = \frac{\mathrm{d}\vec{t}}{\mathrm{d}s},$$

t being the tangent vector and s being the arc length. For a curve Γ, with given end points, its bending energy B_e can be written as

$$B_e = \int_\Gamma k_v^2 \,\mathrm{d}s.$$

Here the deformation of the curve is in the direction normal to the axis of the equilibrium position. Therefore, when the x-axis is along the axis of equilibrium position, the deformation may be represented by $z(x)$ and consequently we have

$$\begin{aligned} B_e &= \int_\Gamma k_v^2 \,\mathrm{d}x \\ &= \int_\Gamma \frac{[z''(x)]^2}{[1+(z'(x))^2]^3} \,\mathrm{d}x. \end{aligned} \tag{17}$$

For small deformation, $z'(x) \approx 0$ and $B_e \approx \int_\Gamma [z''(x)]^2 \,\mathrm{d}x$ Since B_e represents the total energy of the curve, k_v^2 or $(z'')^2$ represents the energy of the curve at an arbitrary point. Therefore, in an image plane k_v^2 will represent the energy of the image space curve at a pixel position.

With the above principle, a curve (a set of pixels along a row or a column) can be considered to be perfectly homogeneous if the bending energy is zero at every pixel position. This is obviously, the most stable state of the curve (i.e., without any deformation). Homogeneity decreases with the increase of deformation. For the purpose of image compression, we are interested in finding the homogeneous segments of pixels in an image because such segments can be approximated with small amount of error and they do not produce significantly any smearing effect. From the space curve analogy, homogeneous segments of pixels are segments with $z'(x) \approx 0$. However, in practice, it is very difficult to obtain long segments of pixels with zero gradient everywhere. In order to circumvent this difficulty, we consider the average of the first-order derivative values for a segment of pixels and compute the variance of these derivative values. Since small value of $z'(x)$ corresponds to small deformation of the image space curve at a pixel position, its average value should correspond to average deformation and hence, the square root of the variance, i.e. the standard error provides a measure for the deformation.

5. Image data compression

Since we are restricted to one-dimensional approximation, we consider both the Hilbert and raster scanned images for compression.

(A) *Coding scheme*: An image on a raster scan can be approximated either row wise or column wise. The one which needs fewer number of segments is selected for coding. In the following section we will be explaining the bit requirement for the proposed methods of coding.

(B) *Bit requirement*: Let us consider an image of size $M \times M$ with L number of gray levels $\{0, 1, 2, \ldots, (L-1)\}$. Since there may be a number of gray segments resulting in the process of approximation, each of them can be coded with their corresponding approximation parameters, namely v_o, v_1, v_2 and the length of the segment, n. Since the positional information of approximation (control parameters of the Bezier curve) parameters is not taken into account for coding, the size of the gray segments plays an important part for regeneration of the image. As the maximum possible size of a segment, on a raster scan, is M, the maximum number of bits required for encoding the size of a segment is $\log_2 M$. In particular, the number of bits required to encode the size of a segment, satisfying the approximation criterion, depends on the maximum value for a segment chosen for approximation. In practice, the size of segments is found to be much less than the length of the raster. The segments, in fact, are found to occur frequently with the same length. As a result, the probability of occurrence for the segments of same size is noticeable. Each of the gray segments is a Bezier arc and is represented by its three parameters namely v_o, $\bar{v}_1$ and v_2. Of them $\bar{v}_1$ may not be an integer. So, instead of $\bar{v}_1$ we consider the integer part of the reconstructed data point d_1 (say) at $t = \frac{1}{2}$ for the segment. We designate this pixel by v_d. Thus, v_o, v_d, v_2 and n completely specifies an approximated data segment, where v_o, v_d and v_2 are the three pixel brightness values on the arc. These brightness values (approximation parameters) in an image are found to occur frequently for different segments. Consequently, Huffman coding for all the parameters provide good results for compression of images. Furthermore, v_o, v_d and v_2 being the brightness values, are found to be indistinguishable from their neighboring values when they differ by small values. This fact can be used to reduce the number of independent brightness values to be encoded. The number of parameters decreases drastically when all the arcs are replaced by horizontal line segments. This increases the compression ratio at the cost of quality of the reconstructed image in terms of

PSNR value. We, therefore, have the following two different situations for compression:

(a) when the segments are all quadratic arc segments,
(b) when the segments are all replaced by horizontal line segments.

Let θ_l, θ_{v_o}, θ_{v_d} and θ_{v_2} be the average number of bits/pixel for the length of segments, and the parameters v_o, v_d and v_2, respectively, then the total number of bits N_b, when the segments are all arcs, is given by,

$$(N_b)_A = N_s(\theta_l + \theta_{v_o} + \theta_{v_2} + \theta_{v_d}), \tag{18}$$

where N_s is the number of segments.

When all the segments are lines, the number of bits reduces to

$$(N_b)_L = N_s(\theta_l + \theta_{bl}), \tag{19}$$

where θ_{bl} is the average number of bits/pixel for the pixel values on line segments.

5.1. Discriminating features of the algorithms

Below we provide the discriminating features of the two proposed algorithms.

For Algorithm 1:

- Segmentation of pixels does not need any separate algorithm. The approximation scheme itself selects the specific segments.
- The method of approximation depends on the selection of ε_{max} and ε_{min}. The values of these parameters are the same for all segments in the image. The resulting performance in reconstruction, therefore, is parameter dependent.
- For large ε_{max}, the possibility of long homogeneous segments of pixels for satisfying the approximation criterion increases. This may introduce visual disparity (smearing effect) between the original and the reconstructed segments. This, in turn, may affect the overall picture quality. For a raster scanned image, this effect may become formidable if ε_{max} exceeds a certain value. However, for an Hilbert scanned image this effect is almost negligible even for a very high value ε_{max}.

For Algorithm 2 :

- A separate algorithm selects only those segments which are homogeneous in some sense. For this, an image has been considered as an intensity surface and the homogeneity concept of pixels over segments has been viewed as a small deformation space curve on this intensity surface.
- Length of a homogeneous segment of pixels depends on the standard error of deformation of the segment from its equilibrium position.
- Different homogeneous segments in an image are approximated with different values of ε_{max} which are determined automatically in the process of approximation. The performance of the algorithm, therefore, does not depend on ε_{max} as in algorithm 1 but it depends on the chosen value for the standard error.

6. Regeneration

Reconstruction of the image during decoding is done using quadratic B–B polynomial. We use here the recursive computation algorithm based on Newton's forward difference scheme as described in Refs. [16,17]. Let $y=at^2+bt+c$ be a polynomial representation of Eq. (4) where the constant parameters a, b and c are determined by the three pixels (two end pixels and one mid pixel) of the arc segment. The usual Newton's method for evaluating the polynomial results in multiplications and does not make use of the previously computed values to compute new values.

Assume the parameter t ranges from 0 to 1. Let the incremental value be q. Then the corresponding y values will be c, aq^2+bq+c, $4aq^2+2bq+c$, $9aq^2+3aq+c, \ldots$. It is observed from Refs. [16,17] that

$$\triangle^2 y_j = 2aq^2 \quad \text{and} \quad y_{j+2} - 2y_{j+1} + y_j = 2aq^2, \quad \geqslant 0.$$

This leads to the recurrence formula

$$y_2 = 2y_1 - y_o + 2aq^2 \tag{20}$$

that involves just three additions to get the next value from the two preceding values at hand. Since the gray segment size is known, the increment q can be obtained from

$$q = \frac{1}{\text{segment size} - 1}.$$

The regenerated gray value y_2 can therefore be determined from Eq. (20).

7. Results and discussions

An attempt has been made to demonstrate an application of one-dimensional quadratic B–B polynomial approximation in coding gray tone Hilbert and raster scanned images. Drawbacks in using the conventional way of approximation have been examined and a modification is then introduced in order to make it useful for image data compression. Based on the modified concept, two different algorithms have been formulated. Both the algorithms have been examined to compress 256 × 256 (8 bits) gray tone images following the Hilbert and raster scan. The performance of the algorithms on the Hilbert scanned images is found to be better than that on the raster scanned images. This is due to the neighborhood property of the Hilbert scan. More precisely, the Hilbert curve always passes through the neighborhood pixels, and since the neighborhood pixels are, in general,

Table 2
Performance of Algorithm 1 on raster scanned images

Image	Mode of approx.	ε_{max}	Max length for segment	Compression rate in bpp	MSQ	PSNR in dB
Lena	Line segment	7	128	1.231	211.817	24.871
		10	64	1.084	244.087	24.255
Girl		5	128	1.347	84.673	28.853
		7	64	1.215	99.526	28.151
Lena	Arc segment	20	256	1.767	25.763	34.020
		25	256	1.602	38.745	32.248
		30	256	1.477	56.967	30.574
Girl		20	256	1.839	53.287	30.864
		25	256	1.590	77.102	29.260
		30	256	1.404	101.274	28.075

Table 3
Performance of Algorithm 1 on Hilbert scanned images

Image	Mode of approx.	ε_{max}	Max length for segment	Compression rate in bpp	MSQ	PSNR in dB
Lena	Line segment	8	256	1.122	104.865	27.924
		10	256	1.027	110.625	27.692
		15	256	0.846	124.338	27.184
		18	256	0.692	131.624	26.937
Girl		18	128	0.768	99.579	28.149
		20	128	0.720	105.089	27.915
		25	128	0.607	119.794	27.346
Lena	Arc segment	25	256	1.644	48.765	31.249
		30	256	1.443	56.924	30.577
		35	256	1.286	68.758	29.757
Girl		35	256	1.094	81.878	28.999
		38	256	1.007	86.545	28.758
		40	256	0.974	104.883	27.923

strongly correlated, the approximation is done over longer segments. Over such long segments, the variation in pixel intensity is low. As a result, arc approximation is not as economical as the line segment approximation (in terms of approximation parameters). Consequently, lower compression ratio or larger number of bits/pixel is required. But the line segment approximation reduces the PSNR value compared to that for arc segment approximation. On the other hand, for raster scanned images, the quality of the reconstructed images is disturbed when the maximum length of segment exceeds a certain value. Short segments, in general, are found to produce better quality for the reconstructed images. Table 2, shows the results on compression and quality for 256×256 8 bit raster scanned images for Algorithm 1, while Table 3 provides the results for the corresponding Hilbert scanned images. The approximation uses both the line and arc segments. Tables 4 and 5 indicate the performance of Algorithm 2 for the raster and Hilbert scanned images. Finally, the comparison for the algorithm due to Kamata et al. [3] is shown in Table 6.

Note that Algorithm 1 in the raster scan mode may produce smearing for large values of ε_{max}, because with the increase in the value of ε_{max}, the possibility of long homogeneous segments of pixels satisfying the approximation criterion increases. As a result, visual disparity may arise. This fact is also true for Algorithm 2 in the raster mode for larger values of the standard error. Fig. 5 shows this smearing effect for Algorithms 1 and 2 in the raster scan mode. The line segment approximation in the raster mode also affects the reconstructed quality for high values of ε_{max}.

Table 4
Performance of Algorithm 2 on raster scanned images

Image	Mode of approx.	Standard error	Max length for segment	Compression rate in bpp	MSQ	PSNR in dB
Lena	Line segment	4	64	1.897	190.194	25.338
		6	64	1.609	232.550	24.465
Girl		6	64	1.952	111.576	27.655
		7	64	1.785	128.795	27.031
Lena	Arc segment	15	64	1.933	40.275	32.080
		17	64	1.819	50.715	31.079
		19	64	1.713	64.711	30.020
Girl		15	64	1.827	42.645	31.832
		17	64	1.627	54.803	30.742
		18	64	1.539	62.538	30.169

Table 5
Performance of Algorithm 2 on Hilbert scanned images

Image	Mode of approx.	Standard error	Max length for segment	Compression rate in bpp	MSQ	PSNR in dB
Lena	Line segment	16	64	0.788	87.095	28.730
		18	64	0.725	92.981	28.446
		20	64	0.669	101.517	28.065
Girl		16	64	0.763	77.521	29.236
		17	64	0.716	81.201	29.035
		18	64	0.677	85.491	28.811
Lena	Arc segment	17	64	1.444	49.071	31.222
		19	64	1.346	53.818	30.821
Girl		18	64	1.488	101.422	28.069
		20	64	1.336	114.487	27.543

Table 6
Comparison between three different algorithms

Image	Algorithm 1		Algorithm 2		Algorithm 3	
	bits/pixel	PSNR in dB	bits/pixel	PSNR in dB	bits/pixel	PSNR in dB
Lena	1.44	30.577	1.44	31.222	1.45	30.019
	1.28	29.757	1.34	30.821	1.20	29.163
Girl	1.09	28.999	1.07	30.436	1.01	30.361
	0.67	27.692	0.68	28.811	0.68	28.442

For the 8 bit Lena and girl images, compression is found to be higher in the Hilbert scan mode compared to that in the raster scan mode. From Tables 4, and 5, it is seen that Algorithm 2, also behaves in the same way as Algorithm 1. Higher compression is found to occur in the Hilbert scan mode. Fig. 6 shows two different decoded images for Lena and girl images for Algorithm 1, while Fig. 7 shows the results of the decoded images for Algorithm 2 due to Hilbert scan. Comparison with Kamata's algorithm (Figs. 8 and 9) shows that the proposed algorithms perform better for the Lena image, in terms of PSNR value at the same compression rate. At the compression rate of 1.44 bit/pixel, Algorithm 1 provides a PSNR value of 30.57 dB, while the algorithm due to Kamata et al. provides 30.01 dB, and Algorithm

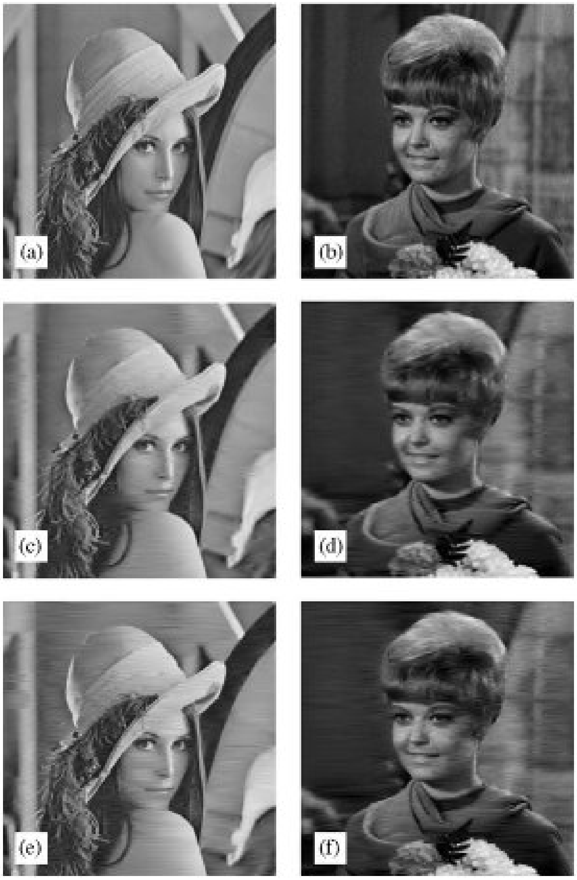

Fig. 5. Results for Algorithm 1 ((c), (d)) and Algorithm 2 ((e), (f)) due to raster scan, (a) input Lena image, (b) input girl image, (c) bpp = 1.47, PSNR = 30.574, (d) bpp = 1.40, PSNR = 28.075, (e) bpp = 1.71, PSNR = 30.020, (f) bpp = 1.539, PSNR = 30.169.

2 provides 31.22 dB. At the compression rate of approximately 1.28 bpp, the PSNR due to Algorithm 1 is 29.75 dB; the PSNR due to Kamata's algorithm is 29.16 dB while Algorithm 2 provides a PSNR of 30.82 dB at a slightly higher compression rate of 1.34 bpp. For the girl image, Algorithm 2 provides a PSNR value of 28.81 dB at the compression rate of 0.68 bpp compared to 28.44 dB as provided by the algorithm due to Kamata et al.

The proposed approximation technique described is different from the conventional least-square method of approximation. Instead of minimizing the global squared sum of errors, it controls an absolute maximum error for each data point. It should be noticed in this context that if the pixels of a segment have low-intensity variation, then the techniques based on conventional quadratic least-square and the quadratic B–B polynomial approximation will produce the same result. Since the proposed method of approximation controls an absolute local error instead of global sum of errors, it is expected that even for moderate variation of intensity within data points, the proposed method will produce better results. Also, given an error term, the conventional least-square technique does not ensure that all the data points will satisfy the error criterion, whereas in the proposed method this is not the case. Furthermore, it

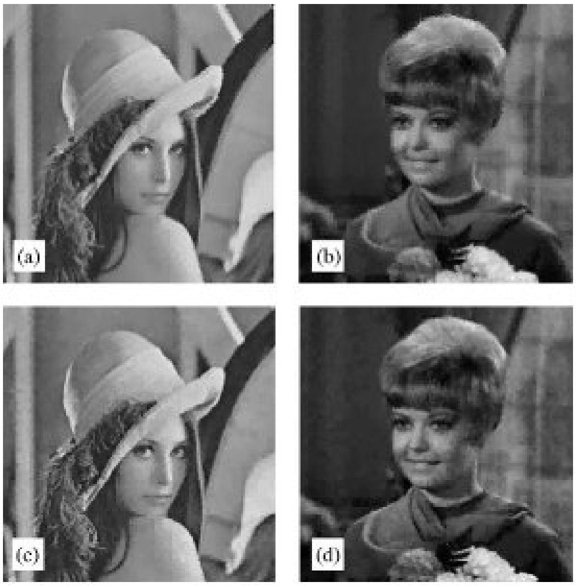

Fig. 6. Results for Algorithm 1 due to Hilbert scan, (a) bpp = 0.69, PSNR = 26.937, (b) bpp = 0.60, PSNR = 27.346, (c) bpp = 1.28, PSNR = 29.757, (d) bpp = 0.97, PSNR = 27.923.

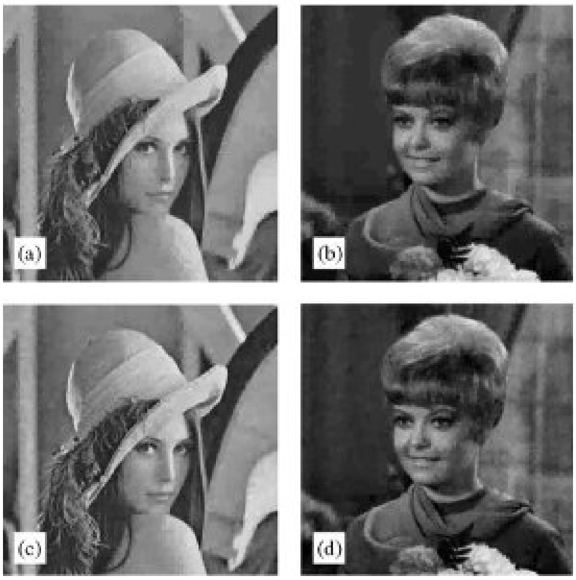

Fig. 7. Results for Algorithm 2 due to Hilbert scan, (a) bpp = 0.72, PSNR = 28.446, (b) bpp = 0.67, PSNR = 28.811, (c) bpp = 1.34, PSNR = 30.821, (d) bpp = 1.33, PSNR = 27.543.

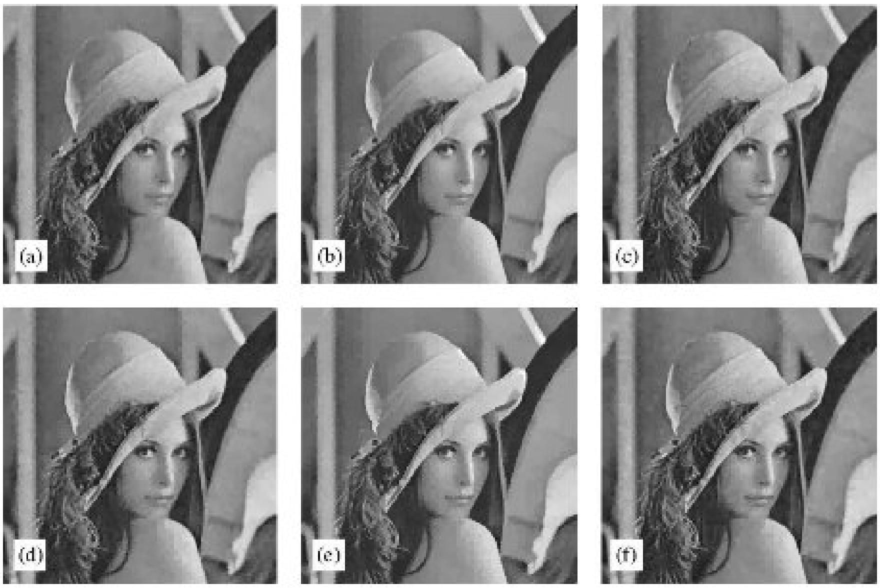

Fig. 8. Comparison of Lena image, (a) Algorithm 1 : bpp = 1.44, PSNR = 30.577, (b) Kamata : bpp = 1.45, PSNR = 30.019, (c) Algorithm 2 : bpp = 1.44, PSNR = 31.222, (d) Algorithm 1 : bpp = 1.28, PSNR = 29.757, (e) Kamata : bpp = 1.20, PSNR = 29.163, (f) Algorithm 2 : bpp = 1.34, PSNR = 30.821.

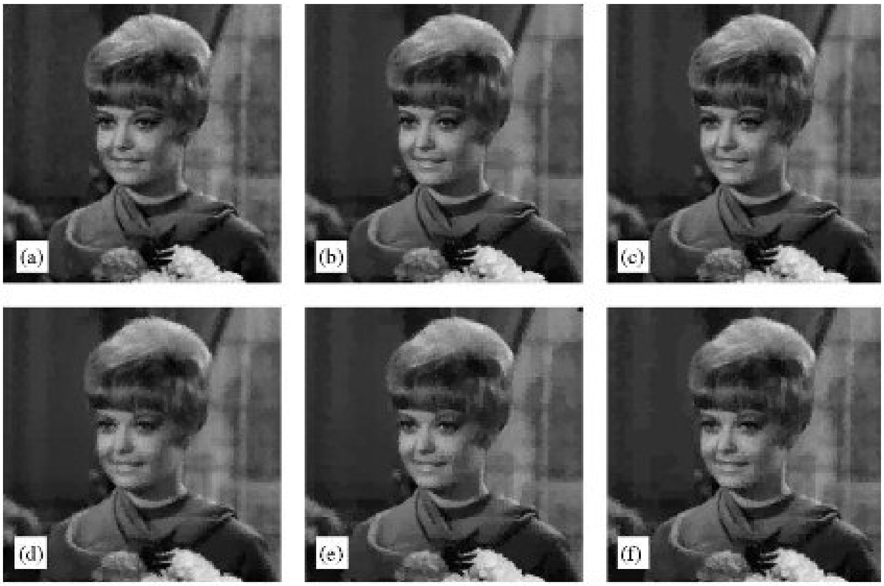

Fig. 9. Comparison of girl image, (a) Algorithm 1 : bpp = 1.09, PSNR = 28.999, (b) Kamata : bpp = 1.01, PSNR = 30.361, (c) Algorithm 2 : bpp = 1.07, PSNR = 30.436, (d) Algorithm 1 : bpp = 0.60, PSNR = 27.346, (e) Kamata : bpp = 0.68, PSNR = 28.442, (f) Algorithm 2 : bpp = 0.68, PSNR = 28.811.

is not needed to compute any functional distance to justify the goodness of approximation because the error term itself quantifies this.

Note further that our intention here is to demonstrate, through an application, the effectiveness of one-dimensional B–B polynomial in image data compression for both the raster and Hilbert scanned images. The algorithms are efficient for the Hilbert scanned images because of strong correlation between pixels over long segments. Both the schemes are fast and simple in hardware implementation. However, it is needless to mention that the two-dimensional approximation always provides a better compression ratio than the corresponding one-dimensional approximation.

8. Summary

Bermstein approximation, though found good in many areas, is not suitable for image data approximation due to a number of difficulties. This paper, using Bernstein basis, proposes a new approximation technique with emphasis on the local control of data points (pixels) instead of minimizing the global squared error. An absolute error criterion has been developed to keep the absolute error within a bound. And for the sake of data compression, we have chosen the second-order polynomial.

Based on the new concept of approximation, two different algorithms are proposed for selecting segments from an image for compression. The first algorithm uses an error bound to partition the pixel data set into different segments through approximation. The partition can be made into either arc or line segments depending on the choice of approximation for both the raster and Hilbert scanned images. These segments are then coded in the subsequent stage. The second algorithm, on the other hand, considers, for a raster scanned image, a row (or column) of pixels as a space curve on an intensity surface and separates out the small deflection curve segments on the basis of a homogeneity criterion. A Hilbert scanned image, however, for a raster scanned square graylevel image can be considered as a space curve with length equal to the square of the length or width of the raster scanned image. Note that the size of the image provides the resolution of the Hilbert curve. Due to the neighborhood property of the Hilbert scan, long homogeneous segments are found to be approximated; resulting in less number of segments for encoding than that for the raster scanned image. Consequently, the compression ratio, is found to be higher.

The performance of the algorithms is tested on a set of input images. The discriminating features of the proposed two algorithms are also discussed. Finally, the results are compared with those of a different existing algorithm.

References

[1] P.E. Bezier, Mathematical and practical possibilities of unisurf, in: E.B.R, R.E. Barnhill and R.F. Risenfeld (Eds.), Computer Aided Geometric Design, Academic Press, New York, 1974.

[2] N. Macon, Numerical Analysis, Wiley, New York, 1963.

[3] S. Kamata, N. Niimi, E. Kawaguchi, A gray image compression using Hilbert scan, in: Proc. ICPR, Vienna, Austria, 1996, pp. 905–909.

[4] J. Quinqueton, M. Berthod, A locally adaptive Peano scanning algorithm, IEEE Trans. Pattern Anal. Mach. Intelligence 3 (1981) 403–412.

[5] R.J. Stevens, A.F. Lehar, F.H. Preston, Manipulation and presentation of multidimensional image data using peano scan, IEEE Trans. Pattern Anal. Mach. Intelligence 5 (1983) 520–526.

[6] A.C. Ansari, I. Gertner, Y.Y. Zeevi, Image compression: wavelet type transform along generalized scan, in: Proceedings of the SPIE Conference on Synthetic Aperture Rader, Orlando, Vol. 1630, 1992, pp. 99–107.

[7] A.C. Ansari, I. Gertner, Y.Y. Zeevi, Combined wavelets DCT image compression, in: Proceedings of the SPIE International Society Optical Engineering, Orlando, Vol. 1699, 1992, pp. 308–317.

[8] S. Kamata, R.O. Eason, E. Kawaguchi, An implementation of Hilbert scanning algorithm and its application to data compression, IEICE Trans. Inform. Syst. 76 (1993) 420–428.

[9] S. Kamata, R.O. Eason, E. Kawaguchi, An efficient Hilbert scanning algorithm and its application to data compression, in: Proceedings of the Scandinavian Conference on Image Analysis Troms, Norway, 1993, pp. 1333–1340.

[10] S. Kamata, N. Niimi, E. Kawaguchi, Interactive analysis of multi-spectral images using a Hilbert curve, in: Proceedings of the IAPR, Jeruselam, Israel, 1994, pp. 93–97.

[11] A.J. Cole, Compaction technique for raster scan graphics using space filling curves, Comput. J. 30 (1987) 87–92.

[12] N. Sorek, Y.Y. Zeevi, On-line visual data compression along a one dimensional scan, in: Proceedings of the SPIE conference on Visual Communication and Image Processing, Cambridge, Mass., Vol. 1001, 1988, pp. 764–770.

[13] W. Skarbek, T. Agui, M. Nikajima, Compression of dithered binary images using Hilbert scan, Trans. IEICE 72 (1989) 1235–1242.

[14] B. Moghaddam, K.J. Hintz, C.V. Stewart, Space filling curves for image compression, in: Proceedings of the SPIE conference on Automatic Object Recognition, Bern, Switzerland, Vol. 1471, 1991, pp. 414–421.

[15] H.O. Peitjen, H. Jurjens, D. Saupe, Chaos and Fractals, Springer, New York, 1992.

[16] S. Biswas, S.K. Pal, D.D. Majumder, Binary contour coding using Bezier approximation, Pattern Recognition Lett. 8 (1988) 237–249.

[17] S. Biswas, S.K. Pal, Approximate coding of digital contours, IEEE Trans. Syst. Man Cybernet. 18 (1988) 1056–1066.

About the Author—SAMBHUNATH BISWAS obtained the M.Sc. degree in Physics from the University of Calcutta in 1973 and Ph.D. (Tech.) in Radio-Physics and Electronics in 2001. He was in electrical industries in the beginning as a Graduate Engineering Trainee and then as a Design and Development Engineer. He was a UNDP fellow at MIT, USA to study Machine Vision in 1988–89. He visited the Australian National University at Canberra in 1995 and Zhejiang University at Hangzhou in China in 2001. He is now a System Analyst at the Machine Intelligence Unit in Indian Statistical Institute, Calcutta. His research interests include image processing, machine vision, computer graphics, and pattern recognition, neural networks and genetic algorithms.