ELSEVIER

Computer Aided Geometric Design 11 (1994) 289–301

COMPUTER AIDED GEOMETRIC DESIGN

Data point selection for piecewise linear curve approximation

Bernd Hamann [a,b,*], Jiann-Liang Chen [c]

[a] *Department of Computer Science, Mississippi State University, P.O. Drawer CS, Mississippi State, MS 39762, USA*

[b] *NSF Engineering Research Center for Computational Field Simulation, Mississippi State University, P.O. Box 6176, Mississippi State, MS 39762, USA*

[c] *Department of Electrical Engineering, Mississippi State University, P.O. Drawer EE, Mississippi State, MS 39762, USA*

Received August 1992

Abstract

A method for selecting data points from a finite set of curve points is discussed. The given curve points originate from a smooth curve and are weighted with respect to a local curvature measure. The most significant points are selected and used to approximate the curve. The selected subset of data points is distributed in such a way that they are uniformly distributed with respect to integrated absolute curvature. The technique is tested for various planar curves and is applied to 2D image compression and volume visualization.

Key words: Approximation; Curvature; Data reduction; Discretization; Parametric curve

1. Introduction

In the context of discretizing a smooth curve, an efficient scheme for approximating the curve by line segments is discussed. Such a scheme should generate just as many line segments as necessary to approximate the given curve with respect to some prescribed error tolerance. It is assumed that the initial curve is already given by a finite set of points from which a smaller subset of points is to be constructed. This allows application of the scheme to analytically defined as well as to discretized curves. The selection of data points is made based on assigning weights to all initial data points. Eventually, only the most significant points are used in the piecewise linear curve approximation.

*Corresponding author.

SSDI 0167-8396(93)E0031-8

An obvious choice to measure the significance of a single data point is the absolute curvature at that point. Therefore, a low-order polynomial approximation scheme is used to locally approximate the given data points and obtain local curvature estimates. Each original data point is weighted according to its associated absolute curvature estimate. Data selection/reduction can be done using one of these alternatives: Either the number of data points to be selected is specified, or a maximal error tolerance is defined implying how many points are required to approximate the curve within this tolerance.

Previous work done in the area of piecewise linear curve approximation can be found in (Cantoni, 1971; Stone, 1961; Tomek, 1974; Williams, 1978). A survey of several schemes for data reduction of piecewise linear curves is provided in (McMaster, 1987). Curvature approximation and data reduction for triangulated surfaces are discussed in (Hamann, 1993, 1994). The data selection technique is applied to 2D image compression and volume visualization (ray casting) algorithms. Two standard ray casting methods for 3D scalar fields are described in (Levoy, 1988) and (Sabella, 1988). The performance of such volume visualization methods can be improved significantly by performing data point selection as a preprocessing step.

First, the method for weighting data points and selecting a proper subset of them is described. Second, the selection technique is applied to curves (graphs of univariate functions and parametric curves), 2D image compression, and ray casting.

2. Local curvature approximation

In the following, it is assumed that a finite set of (ordered) planar data points originating from a smooth curve is given. This point set is denoted by $\mathcal{X} = \{\boldsymbol{x}_i = (x_i, y_i) \mid i = 0, \ldots, n\}$. The curvature at a point $\boldsymbol{x}_i$ is approximated by computing a quadratic polynomial interpolating the three points $\boldsymbol{x}_{i-1}$, $\boldsymbol{x}_i$, and $\boldsymbol{x}_{i+1}$. This polynomial is used to obtain a curvature estimate at $\boldsymbol{x}_i$.

A local coordinate system is constructed for the three data points $\boldsymbol{x}_{i-1}$, $\boldsymbol{x}_i$, and $\boldsymbol{x}_{i+1}$, $i = 1, \ldots, n-1$. Two difference vectors are computed,

$$\boldsymbol{d}_1 = \frac{\boldsymbol{x}_{i-1} - \boldsymbol{x}_i}{\|\boldsymbol{x}_{i-1} - \boldsymbol{x}_i\|} \quad \text{and} \quad \boldsymbol{d}_2 = \frac{\boldsymbol{x}_{i+1} - \boldsymbol{x}_i}{\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|}, \tag{2.1}$$

where $\| \ \|$ denotes the Euclidean norm. Provided that the points $\boldsymbol{x}_{i-1}$, $\boldsymbol{x}_i$, and $\boldsymbol{x}_{i+1}$ are not collinear, a local orthonormal coordinate system with $\boldsymbol{x}_i$ as its origin is defined by the two unit basis vectors

$$\boldsymbol{b}_2 = (b_2^x, b_2^y) = \frac{\boldsymbol{d}_1 + \boldsymbol{d}_2}{\|\boldsymbol{d}_1 + \boldsymbol{d}_2\|} \quad \text{and} \quad \boldsymbol{b}_1 = (b_2^y, -b_2^x). \tag{2.2}$$

If three consecutive data points are collinear, the basis vectors are defined as

$$\boldsymbol{b}_1 = (b_1^x, b_1^y) = \frac{\boldsymbol{x}_{i+1} - \boldsymbol{x}_i}{\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|} \quad \text{and} \quad \boldsymbol{b}_2 = (-b_1^y, b_1^x). \tag{2.3}$$

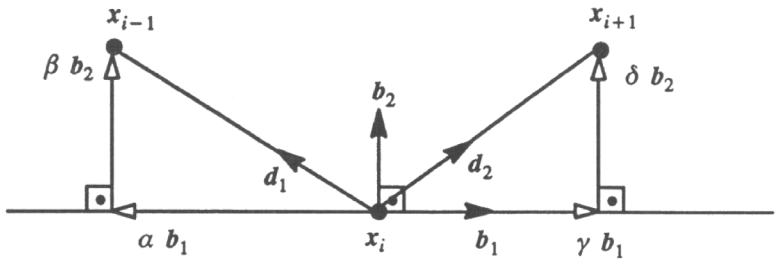

Fig. 1. Construction of local coordinate system.

The choice of the basis vector $\boldsymbol{b}_2$ guarantees that the two points $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_{i+1}$ lie on different sides of the line with direction $\boldsymbol{b}_2$ and passing through $\boldsymbol{x}_i$.

Obviously, the points $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_{i+1}$ can be written with respect to this local coordinate system as

$$\boldsymbol{x}_{i-1} = \boldsymbol{x}_i + \alpha\boldsymbol{b}_1 + \beta\boldsymbol{b}_2 \quad \text{and} \quad \boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \gamma\boldsymbol{b}_1 + \delta\boldsymbol{b}_2. \tag{2.4}$$

Thus, the points $\boldsymbol{x}_{i-1}$, $\boldsymbol{x}_i$, and $\boldsymbol{x}_{i+1}$ have the local coordinates (α, β), $(0,0)$, and (γ, δ). A quadratic polynomial $f(t) = \sum_{i=0}^{2} a_i t^i$ is uniquely defined by the three interpolation conditions

$$\begin{aligned} f(\alpha) &= \sum_{i=0}^{2} a_i \alpha^i = \beta, \\ f(0) &= a_0 = 0, \\ f(\gamma) &= \sum_{i=0}^{2} a_i \gamma^i = \delta. \end{aligned} \tag{2.5}$$

This construction is illustrated in Fig. 1.

The parametric curve $(t, f(t))$ implied by the graph of f has a certain curvature at $\boldsymbol{x}_i$, which is used as a measure of $\boldsymbol{x}_i$'s significance. The (signed) curvature of this local polynomial approximation at $\boldsymbol{x}_i$ is given by

$$\left.\frac{f''(t)}{(1 + f'^2(t))^{3/2}}\right|_{t=0} = \frac{2a_2}{(1 + a_1^2)^{3/2}}. \tag{2.6}$$

The two parabolas through $\boldsymbol{x}_0$, $\boldsymbol{x}_1$, and $\boldsymbol{x}_2$ and $\boldsymbol{x}_{n-2}$, $\boldsymbol{x}_{n-1}$, and $\boldsymbol{x}_n$ are used to obtain curvature estimates at the end points $\boldsymbol{x}_0$ and $\boldsymbol{x}_n$. The curvature estimates at these points are given by

$$\frac{2a_2}{(1 + (a_1 + 2a_2\alpha)^2)^{3/2}} \quad \text{and} \quad \frac{2a_2}{(1 + (a_1 + 2a_2\beta)^2)^{3/2}}. \tag{2.7}$$

The coefficients for the parabolas passing through the first three and the last three data points are considered in this case. If the initial point set originates from a closed curve, i.e., $\boldsymbol{x}_0 = \boldsymbol{x}_n$, a quadratic polynomial must be computed that interpolates the three points $\boldsymbol{x}_{n-1}$, $\boldsymbol{x}_0$, and $\boldsymbol{x}_1$. The absolute value of the curvature estimate at a point $\boldsymbol{x}_i$ is used as its weight. The absolute curvature value is denoted by κ_i.

It must be mentioned that this curvature approximation scheme is sensitive to perturbations of the original data points ("noise"). In general, a different curvature approximation/weighting scheme should be used when dealing with "noisy" data obtained by real-world measurements. The technique presented in this paper is intended primarily for computing a piecewise linear approximation for a given smooth curve. Approximating smooth curves (and surfaces) with a small number of suitably chosen linear segments is an important problem in computer graphics, since the rendering time depends on the number of points used to discretize curves and surfaces.

3. Selecting a specified number of data points

Having computed absolute curvature estimates, κ_i values are known for each data point $\boldsymbol{x}_i$ and the most important ones can be selected. First, all points with an associated curvature estimate of zero are removed from the point set $\mathcal{X}$. Such points lie in the interior of linear segments on the original curve and are not necessary for a piecewise linear approximation. Only the two end points, $\boldsymbol{x}_0$ and $\boldsymbol{x}_n$, must be kept. The result of removing such points is the point set

$$\overline{\mathcal{X}} = \mathcal{X} \setminus \{\boldsymbol{x}_i \mid \kappa_i = 0\} \cup \{\boldsymbol{x}_0, \boldsymbol{x}_n\}. \tag{3.1}$$

The elements of $\overline{\mathcal{X}}$ are denoted by $\overline{\boldsymbol{x}}_0$, $\overline{\boldsymbol{x}}_1, \ldots,$ and $\overline{\boldsymbol{x}}_{\bar{n}}$.

The next step is the selection of a specified number of points from the set $\overline{\mathcal{X}}$. This in done in such a way that curvature is distributed uniformly with respect to chord length. This step is explained next.

The chord length of the piecewise linear curve implied by $\overline{\mathcal{X}}$ is defined as

$$S = \sum_{i=0}^{\bar{n}-1} s_i = \sum_{i=0}^{\bar{n}-1} \|\boldsymbol{v}_i\|, \tag{3.2}$$

where $\boldsymbol{v}_i = \overline{\boldsymbol{x}}_{i+1} - \overline{\boldsymbol{x}}_i$. A piecewise linear function $k = k(s)$, $s \in [0, S]$, approximating the curvature along the curve is determined. This function interpolates the tuples $\left(\sum_{i=0}^{j} s_i, \kappa_{j+1}\right)$, $j = -1, \ldots, (\bar{n}-1)$, where $\sum_{i=0}^{-1} s_i = 0$. Integrating the function $k(s)$ over $[0, S]$ yields

$$K = \int_0^S k(s)\,\mathrm{d}s = \tfrac{1}{2} \sum_{i=0}^{\bar{n}-1} s_i (\kappa_i + \kappa_{i+1}). \tag{3.3}$$

By rearranging the summation in (3.3) one obtains

$$K = \tfrac{1}{2}\kappa_0 s_0 + \tfrac{1}{2} \sum_{i=1}^{\bar{n}-1} \kappa_i (s_{i-1} + s_i) + \tfrac{1}{2}\kappa_{\bar{n}} s_{\bar{n}-1}. \tag{3.4}$$

Formula (3.4) emphasizes that the integrated absolute curvature estimate κ_i multiplies the lengths of the line segments on both sides, s_{i-1} and s_i. Conse-

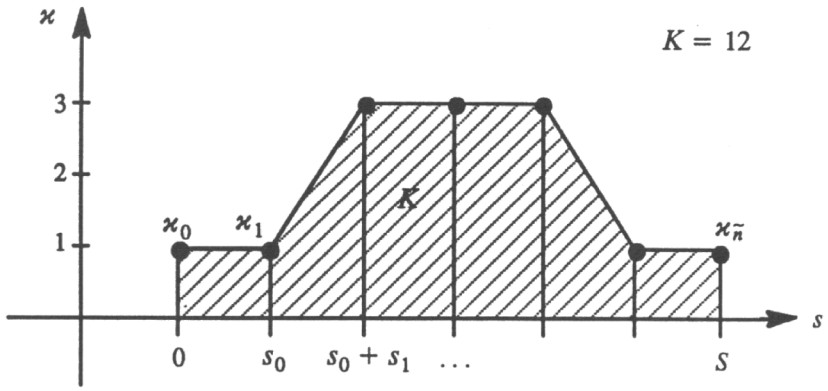

Fig. 2. Integrating absolute curvature estimates.

quently, greatest weight is given to data points with high absolute curvature and long line segments on both sides, while least weight is given to data points with low absolute curvature and short line segments on both sides. This interpretation of (3.3) and (3.4) justifies the use of the integrated absolute curvature estimate K for data point selection. The computation of K is shown in Fig. 2.

Eventually, the set

$$\mathcal{Y} = \{\boldsymbol{y}_0 = \overline{\boldsymbol{x}}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{m-1}, \boldsymbol{y}_m = \overline{\boldsymbol{x}}_{\bar{n}}\} \tag{3.5}$$

is generated. The points in $\mathcal{Y}$ are chosen in a curvature-related manner. The specified number of data points is selected such that the integrated absolute curvature estimate is distributed nearly uniformly with respect to chord length. Assuming that $K > 0$, one computes $m + 1$ values t_j such that

$$\int_0^{t_j} k(s)\,\mathrm{d}s = j\frac{K}{m}, \quad j = 0, \ldots, m. \tag{3.6}$$

Obviously, if $K = 0$ all data points in the initial set $\mathcal{X}$ lie on a line, and the initial point set has already been reduced to the set $\overline{\mathcal{X}} = \{\boldsymbol{x}_0, \boldsymbol{x}_n\}$ according to (3.1). In this case, no further processing is necessary.

If $K > 0$ interior data points must be selected. The interior data points $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{m-1}$ are chosen according to the following rule: If t_j is in the interval $\left[\sum_{i=0}^{l} s_i, \sum_{i=0}^{l+1} s_i\right)$, it is determined whether the condition

$$\left|t_j - \sum_{i=0}^{l} s_i\right| \leqslant \left|t_j - \sum_{i=0}^{l+1} s_i\right|$$

holds for some $l \in \{-1, 0, 1, \ldots, \bar{n} - 2\}$; if this condition holds one chooses $\boldsymbol{y}_j = \overline{\boldsymbol{x}}_{l+1}$; otherwise, one chooses $\boldsymbol{y}_j = \overline{\boldsymbol{x}}_{l+2}$. For the case $t_j = S$ one defines $\boldsymbol{y}_m = \overline{\boldsymbol{x}}_{\bar{n}}$. This procedure ensures that absolute curvature is distributed properly. As a result of this strategy, the selected data points must be clustered

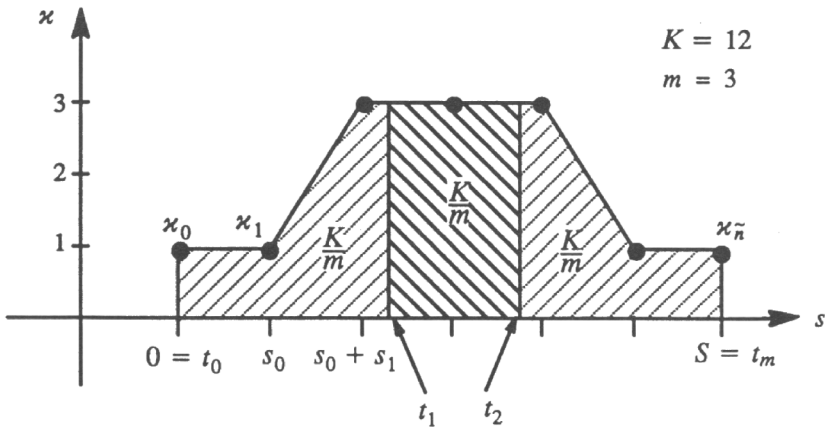

Fig. 3. Uniform curvature distribution.

in regions with high absolute curvature. Fig. 3 illustrates the computation of the values t_j.

The selection of the points $\boldsymbol{y}_j$ allows points to be selected multiple times, i.e., it is possible that $\boldsymbol{y}_j = \boldsymbol{y}_{j+1}$. Such point multiplicities are removed in a postprocessing step. This implies that one might obtain less than $m + 1$ points. The finally selected point set is $\mathcal{Y} = \bigcup_{j=0}^{m} \{\boldsymbol{y}_j\}$.

4. Selecting data points based on an error tolerance

Following the approach described in Section 3, one generates a reduced point set by specifying a number. Alternatively, one can specify an error tolerance to determine how many (and which) points must be selected/kept in order to approximate the initial point sequence considering this maximally allowed error. Choosing the second alternative, one iteratively removes points (the ones with smallest curvature weights first), until a piecewise linear approximation is obtained that differs more than the specified tolerance from the initial piecewise linear curve approximation.

The second alternative for data point selection requires the definition of an appropriate error measure. In order to measure the distance between the initial piecewise linear curve and an intermediate piecewise linear curve the root-mean-square (RMS) error

$$E = \left(\frac{1}{n+1} \sum_{i=0}^{n} d_i^2 \right)^{1/2} \tag{4.1}$$

is used as error measure. Here, the signed, perpendicular Euclidean distance d_i between an initial point $\boldsymbol{x}_i \in \mathcal{X}$ and an intermediate piecewise linear curve is used. The distance is given by

$$d_i = \boldsymbol{n}_j \cdot (\boldsymbol{x}_i - \boldsymbol{y}_j), \tag{4.2}$$

where $\boldsymbol{x}_i$ is a point on the polygon with vertices $\boldsymbol{x}_k = \boldsymbol{y}_j, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_{k+l} =$

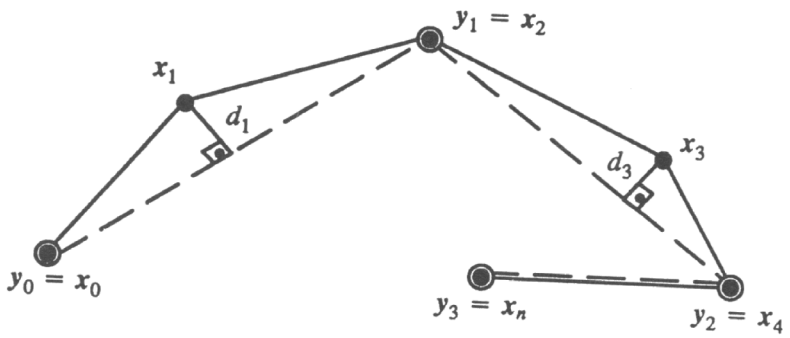

Fig. 4. Distances between initial and intermediate piecewise linear curve.

$\boldsymbol{y}_{j+1}$, and $\boldsymbol{n}_j$ is the unit normal of the line passing through $\boldsymbol{y}_j$ and $\boldsymbol{y}_{j+1}$, i.e.,

$$\boldsymbol{n}_j = \frac{\boldsymbol{y}_{j+1} - \boldsymbol{y}_j}{\|\boldsymbol{y}_{j+1} - \boldsymbol{y}_j\|} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \tag{4.3}$$

This measure allows the computation of RMS errors for graphs of univariate functions and for parametric curves. Clearly, $E = 0$ if the initial data points are collinear. The error measure E can be computed in $O(n)$. Fig. 4 illustrates the computation of distances between the initial and an intermediate piecewise linear curve.

5. Test results and applications

The data point selection technique has been tested for univariate functions, i.e., curves of the form $\{(t_i, f(t_i)) \mid t_i \in [a,b],\ t_i = a + i(b-a)/n,\ i = 0,\ldots,n\}$, and parametric curves, i.e., curves of the form $\{(x(t_i), y(t_i)) \mid t_i \in [a,b],\ t_i = a + i(b-a)/n,\ i = 0,\ldots,n\}$. Table 1 lists the test results for several planar curves. The initial point sets consist of 101 data points and are reduced by 50% ($m = 49$), 75% ($m = 24$), and 90% ($m = 9$). The RMS errors are computed based on (4.1).

Fig. 5 shows the data point selection results for a parabola (curve 1 in Table 1) and Fig. 6 for a trigonometric function (curve 3 in Table 1). The

Table 1
RMS errors ($\times 10^{-3}$) for different curves and different degrees of reduction, $\mathcal{X} = \{\boldsymbol{x}_i = (x(t_i), y(t_i)) \mid t_i \in [a,b],\ t_i = a + i(b-a)/100,\ i = 0,\ldots,100\}$.

Curve		Number of selected points		
		$m = 49$	$m = 24$	$m = 9$
1. Parabola:	(t_i, t_i^2), $t_i \in [-1,1]$	0.49	1.91	11.71
2. Exponential function:	(t_i, e^{t_i}), $t_i \in [0,2]$	0.34	1.32	9.20
3. Trigonometric function:	$(t_i, \sin(\pi t_i))$, $t_i \in [0,2]$	3.38	12.35	26.66
4. Circular arc:	$(\cos(\pi t_i), \sin(\pi t_i))$, $t_i \in [0,1]$	0.39	1.63	11.08
5. Elliptic arc:	$(2\cos(\pi t_i), \sin(\pi t_i))$, $t_i \in [0,1]$	0.95	3.88	25.90

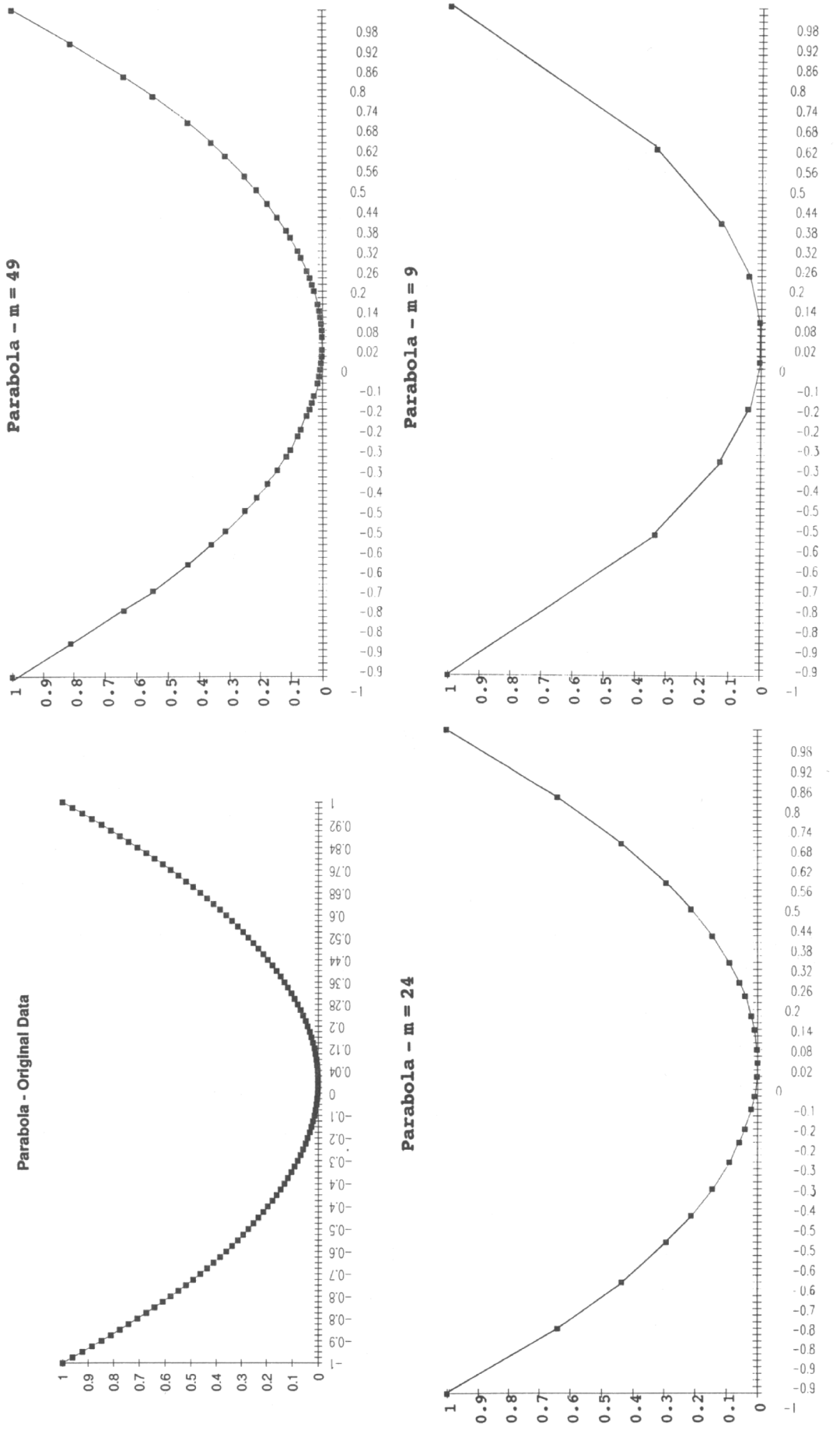

Fig. 5. Data point selection for parabola.

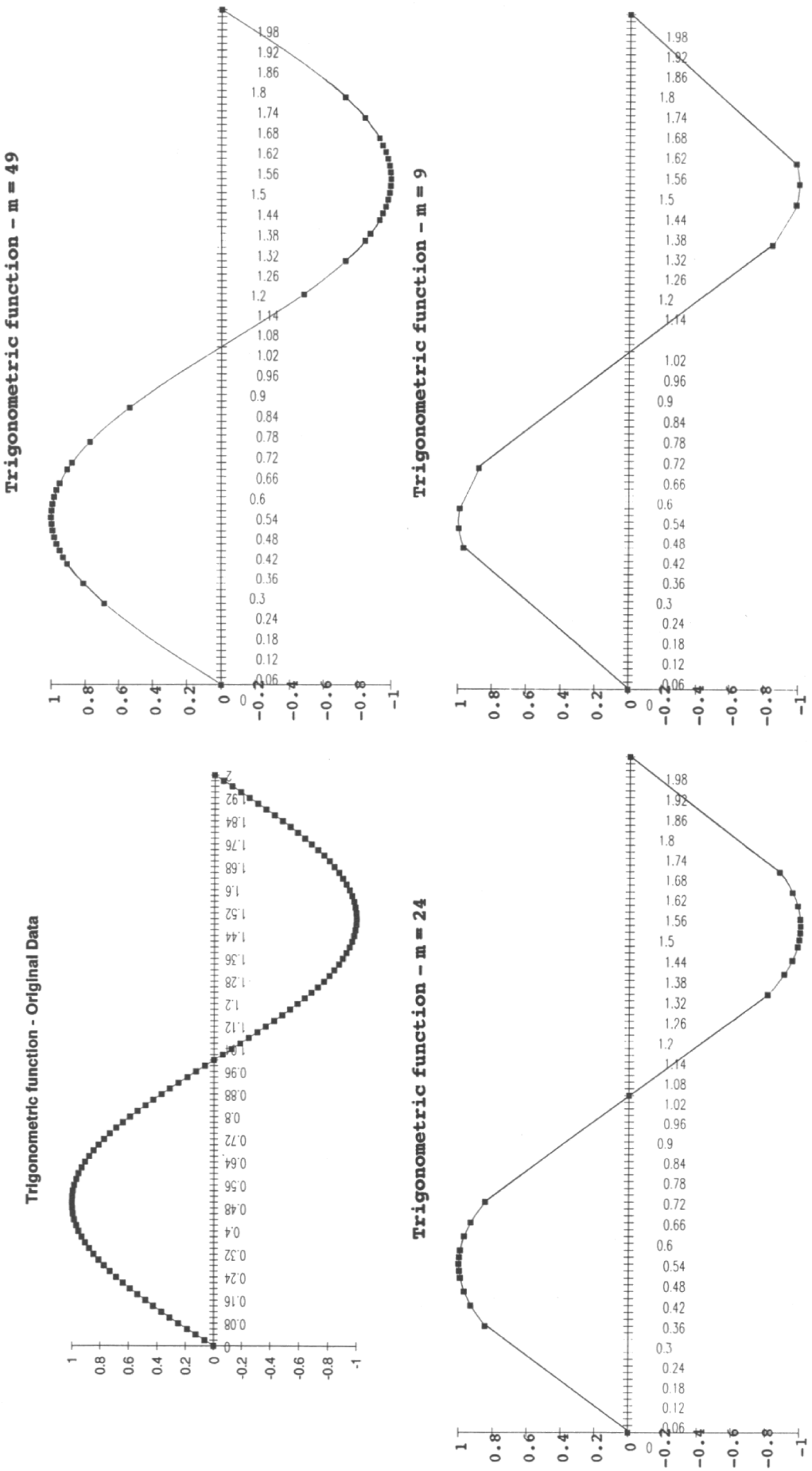

Fig. 6. Data point selection for trigonometric function.

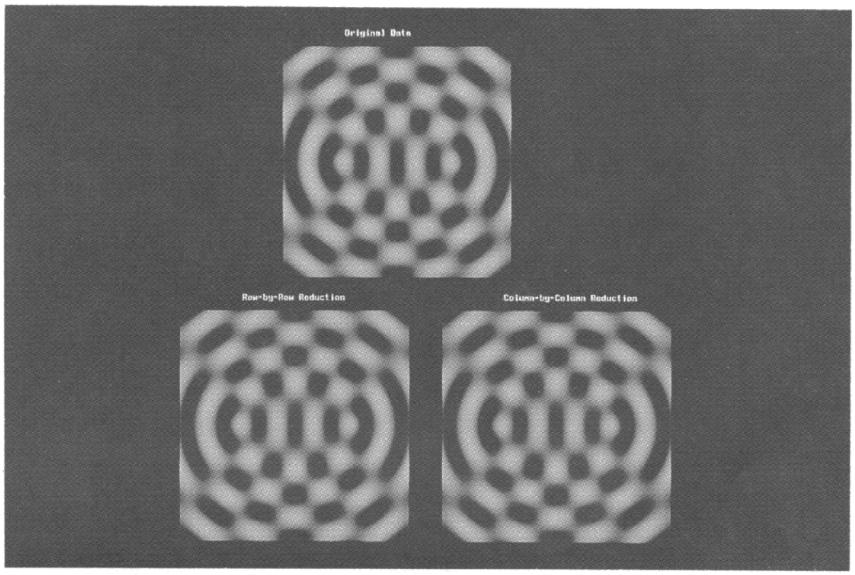

Fig. 7. Data selection for 2D image compression.

upper-left corner in both figures shows the initial point set $\mathcal{X}$, the upper-right corner shows the result for $m = 49$, the lower-left corner the result for $m = 24$, and the lower-right corner the result for $m = 9$.

Generally, a different data selection/reduction method should be applied to "noisy" data. The technique can also be used for data selection for discrete bivariate and trivariate data sets. A bivariate example is shown in Fig. 7. The initial data set for the top image in Fig. 7 consists of 401×401 function values $f_{i,j}$ (mapped to corresponding colors) given as a 2D array. The initial values are obtained by computing the finite set of trigonometric function values

$$f(i,j) = \cos\sqrt{\left(\frac{i}{10} - 10\right)^2 + \left(\frac{j}{10} - 20\right)^2} + \cos\sqrt{\left(\frac{i}{10} - 30\right)^2 + \left(\frac{j}{10} - 20\right)^2},$$

where $i, j = 0, \ldots, 400$.

The data point selection algorithm is iteratively applied to each row or, alternatively, to each column in this 2D image. The lower-left image in Fig. 7 shows the result after reducing the number of data in each row by 53%, and the lower-right image shows the result after reducing the number of data in each column by 52%.

In the area of volume visualization, the basic principle used to generate transparent images of 3D volumetric data is the integration of a univariate "density" function along rays penetrating the given volume. For each ray, the univariate function to be integrated is determined by approximating "density" values along the ray. Each ray is uniformly discretized. As a result of this approach, data lying "deeper" in the volume have a smaller influence on the

resulting image than data data being close to the 2D viewing plane. Two standard methods are described in (Levoy, 1988) and (Sabella, 1988).

Normalizing the length of each ray penetrating the 3D volume, one class of ray casting techniques can be viewed as integration of some "density" function $D(x)$ along each ray (see (Sabella, 1988)). The resulting intensity I perceived in the viewing plane is given by

$$I = \int_0^1 D(x) \exp\left(-\lambda \int_{y=0}^{x} D(y)\,\mathrm{d}y \right) \mathrm{d}x, \quad \lambda \geqslant 0. \tag{5.1}$$

Usually, this integral is approximated by its discrete version

$$I_{\text{Sabella}} = \sum_{i=0}^{n} D_i \exp\left(-\lambda \sum_{j=0}^{i-1} D_j \right), \quad \lambda \geqslant 0, \tag{5.2}$$

using equidistant spacing between the data points along each ray.

Another way to generate transparent images is based on the following principle: The effect of a data point with density value D_i on the resulting intensity in the image plane depends on the number of data points "in front of it" (see (Levoy, 1988)). This model computes the resulting intensity for each ray as

$$I_{\text{Levoy}} = \left\{ \sum_{i=0}^{n-1} D_i (1 - t_i) \prod_{j=0}^{i-1} t_j \right\} + D_n \prod_{j=0}^{n-1} t_j, \tag{5.3}$$

where $\prod_{j=0}^{-1} t_j = 1$, and $t_i \in [0, 1]$ is a transparency parameter associated with density value D_i. Again, equidistant spacing is used between data along a ray.

Formulas (5.1), (5.2), and (5.3) are simplifications, but still convey the essential principles used in ray casting. Algorithms based on such methodologies can be improved significantly by first reducing the number of data points along each ray before performing the integration step. The computation of I_{Sabella} in (5.2) is of complexity $O(n^2)$! Data point selection can be performed using the described selection technique. In the example shown in Fig. 8, Sabella's algorithm is applied to a 3D medical data set (computerized axial tomography data). Considering all data points along each ray yields the left image; using approximately 3% of the data points along each ray generates the right image. Both images are generated by using 680×640 rays. The integration step for the right image is performed based on linear interpolation of the selected data points.

6. Conclusions

A new method for selecting points for piecewise linear curve approximation has been presented. The technique chooses data points with respect to local absolute curvature estimates. The degree of reduction is controlled either by a number of points to be selected or by an error tolerance. The test results

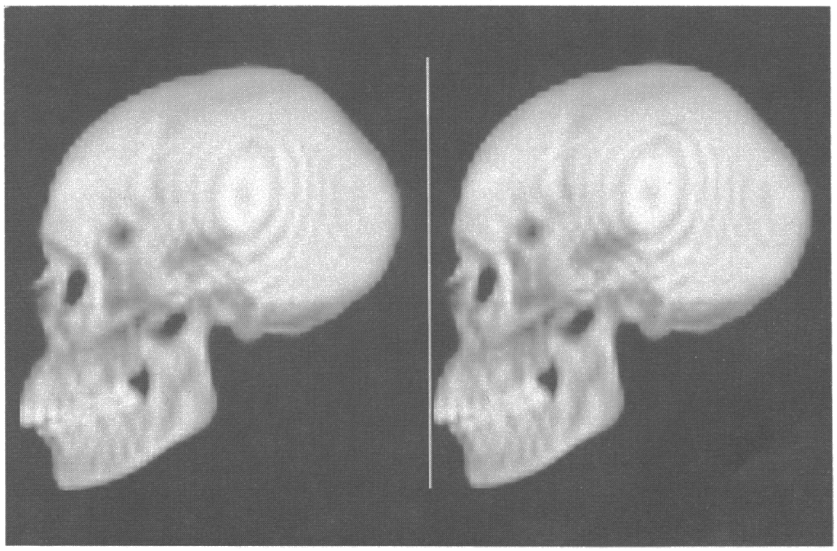

Fig. 8. Using entire and reduced data set for ray casting.

confirm the quality of the approach for planar parametric curves. The new method can be viewed as an alternative to more "classical" approaches based on minimizing the average mean-squared error or the maximum absolute (uniform) error. While those approaches have originated from approximation theory, the approach introduced in this paper uses simple geometrical concepts.

It is planned to extend the technique to parametric surfaces and graphs of trivariate functions. Furthermore, the method will be compared with classical minimization approaches.

Acknowledgements

This research was supported by Mississippi State University through its 1992 Research Initiation Program (Research Initiation Award). The reviewers' comments are gratefully acknowledged.

References

Cantoni, A. (1971), Optimal curve fitting with piecewise linear functions, IEEE Trans. Comput. C-20, 59–67.

Hamann, B. (1993), Curvature approximation for triangulated surfaces, in: Farin, G., Hagen, H. and Noltemeier, H., eds., *Geometric Modelling*, Springer, New York, 139–153.

Hamann, B. (1994), A data reduction scheme for triangulated surfaces, Computer Aided Geometric Design 11, 197–214.

Levoy, M. (1988), Display of surfaces from volume data, IEEE Comput. Graphics Appl. 8, 29–37.

McMaster, R.B. (1987), Automated line generalization, Cartographica 24, 74–111.

Sabella, P. (1988), A rendering algorithm for visualizing 3D scalar fields, Comput. Graphics 22, 51–55.

Stone, H. (1961), Approximation of curves by line segments, Math. Comput. 15, 40–47.

Tomek, I. (1974), Two algorithms for piecewise-linear continuous approximation of functions of one variable, IEEE Trans. Comput., 445–448.

Williams, C.M. (1978), Note – An efficient algorithm for the piecewise linear approximation of planar curves, Comput. Graphics Image Processing 8, 286–293.