MOMENT-PRESERVING PIECEWISE LINEAR APPROXIMATIONS OF SIGNALS AND IMAGES

Thai B. Nguyen and John Oommen

TR-258 DECEMBER 1994

School of Computer Science, Carleton University
Ottawa, Canada, KlS 5B6

Moment-Preserving Piecewise Linear Approximations of Signals and Images[1]

Thai B. Nguyen[2] and B. John Oommen[3]
School of Computer Science
Carleton University
Ottawa, CANADA
K1S 5B6
email: oommen@scs.carleton.ca

Abstract

Approximation techniques are an important aspect of digital signal and image processing. Many lossy signal compression procedures such as the Fourier transform and discrete cosine transform are based on the idea that a signal can be represented by a small number of transformed coefficients which are an approximation of the original.

Existing approximation techniques approach this problem in either a time/spacial domain or transform domain, but not both. This paper reviews various existing approximation techniques. Subsequently, we present a new strategy to obtain an approximation $\hat{f}(x)$ of $f(x)$ in such a way that it is *reasonably* close to the original function in the domain of the variable x, and *exactly* preserves some properties of the transformed domain. In this particular case, the properties of the transformed values that are preserved are geometric moments of the original function. The proposed technique has been applied to one-dimensional functions, two-dimensional planar curves, and two-dimensional images. This paper also provides a reasonably comprehensive survey of the existing time/spacial and transform domain approximation strategies.

Keywords: signal processing, image processing, sampling, approximation, moment-preserving approximations

1 Introduction

Approximation techniques are an important aspect of signal processing. Electronic voice mail messages, photographic images, and X-ray images are often digitized and encoded to decrease their storage and transmission costs. Approximation techniques are also used to remove excess noise or to simplify highly complex signals and reduce analysis time required in the pattern recognition or image understanding phase. The focus of this paper is the approximation of digitized signals and images.

[1]A preliminary version of this paper will appear in the Proceedings of IAS-4, the Fourth International Conference on Intelligent Autonomous Systems, Karlsruhe, Germany, March 1995.

[2]The work of this author was supported by Object Technology International, Inc.

[3]Senior member of the I.E.E.E. The work of this author was partially supported by the Natural Sciences and Engineering Research Council of Canada.

1.1 Terminology and Concepts

Let $f(x)$ represent a one-dimensional function to be approximated, and let it be defined over a domain D. For a continuous function, D describes an interval $[a, b]$, and the function is defined for all values of x between a and b. For a discrete function, D is a set of N distinct values of $x_0, x_1, \ldots, x_{N-1}$. A two-dimensional function is often used to describe a planar curve or a digitized image. A continuous planar curve, $f(x, y)$, describes the position of the curve on the xy-plane for all values of x defined in the interval $[A_x, B_x]$ and all values of y defined in the interval $[A_y, B_y]$. A discrete planar curve is usually defined by a set of points $S = \{p_i = (x_i, y_i) | i = 0, 1, 2, \ldots, N-1\}$.

The main goal of an approximation technique is to find a set of of n points, $\widehat{S}$, such that $\widehat{S} = \{(\hat{x}_0, \hat{f}(\hat{x}_0)), (\hat{x}_1, \hat{f}(\hat{x}_1)), \ldots, (\hat{x}_{n-1}, \hat{f}(\hat{x}_{n-1}))\}$, where $n < N$. If $x_i = \hat{x}_j$ for some $(x_i, f(x_i)) \in S$ and $(\hat{x}_j, \hat{f}(x_j)) \in \widehat{S}$, it is *not* necessarily true that $f(x_i) = \hat{f}(\hat{x}_j)$. The approximated data points in $\hat{S}$ are joined by a polynomial of order at most $m - 1$ with an error norm less than a prespecified quantiy ϵ. By restricting $m = 2$, approximation polynomials will be limited to linear equations.

A one-dimensional function, f (describing an audio signal, a simple curve, or a line segment), is the measured value of the function at a specified time or position variable x. Thus, f is referred to as a time/spacial domain function. A function can, therefore, be viewed as a small machine that converts numbers into other numbers. A *transform* can be viewed as a "meta" machine in an analogous way; it converts functions into other functions. For example, the Fourier transform takes a continuous function of time (like a sound wave) or space (like an image), say $f(x)$, and converts it into a continuous function of frequency, $F(\omega)$. An approximation technique in the transform domain quantizes and truncates the coefficients of $F(\omega)$ to derive $\widehat{F}(\omega)$, and an inverse transformation is performed on $\widehat{F}(\omega)$ to derive $\hat{f}$.

An approximation method can be objectively compared with other approximation methods by its time efficiency, memory efficiency (how much data is kept in order to retain a *good* approximation of the original function), and, the amount of error in the approximated function compared to the original function. Since there are several ways to calculate the error norms introduced by the approximating function, we will use three criteria, namely, the maximum error (ME), mean-squared error (MSE), and, signal-to-noise ratio (SNR). Let e_i be the error value of the i^{th} point approximation. For one-dimensional functions, $e_i = |f(x_i) - \hat{f}(x_i)|$, and for two-dimensional functions, e_i is the Euclidean distance between point $(x_i, f(x_i))$ and the approximating curve evaluated at point $(x_i, \hat{f}(x_i))$. The maximum error, mean-squared error, and signal-to-noise ratio (measured in decibels (dB)) can be defined in terms of e_i's as follows:

$$ME = \max_i \{e_i\}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} e_i^2$$

$$SNR = 10\log_{10}\left(\frac{\frac{1}{N}\sum_{i=1}^{N} f(x_i)^2}{\frac{1}{N}\sum_{i=1}^{N} e_i^2}\right).$$

These three metrics will be used to quantify approximation schemes in this paper.

1.2 Motivation

Existing approximation techniques for one-dimensional functions, planar curves, and digitized images derive an approximation by working with constraints in the time/spacial or transform domain but not both. We introduce a new moment-preserving approximation method. This technique preserves a finite number of the geometric moments[4] which are related to the transform domain parameters of the function. A finite number of these moments can, in fact, be used to reconstruct an approximation of the Fourier coefficients. In our scheme, however, it is applied directly to the time/spacial domain approximation instead of the usual *"Transformation→ Approximation→Inverse Transformation"* process. Our new approximation technique has been successfully applied to one-dimensional functions, planar curves, and digitized images.

1.3 Outline of Paper

The literature in the field of pattern recognition and image processing contains hundreds of references to the problem of approximating curves and images. A detailed list of these and their relative features can be found in [Ng 94]. During the course of our study we were unable to locate a single *comprehensive* survey of all these time/space and transform domain approximation strategies. To render this paper complete so that it can also serve as a useful survey of the field, we have included a brief but extensive catalogue of these references, and a summary of their salient aspects.

Section 2 reviews some of the existing approximation techniques. These related works are grouped in either the time/spacial domain or the transform domain schemes, and in each of these two categories, the work is presented in a chronological order. Section 3 continues the review of

[4] The geometric moment of a function $f(x)$ is defined as $\int_{-\infty}^{\infty} x^k f(x)dx$ for $k = 0, 1, 2, \ldots$.

the previous work with a focus on methods concerned with two-dimensional planar curves and digitized images, and the techniques are grouped according to their operating within time/spacial or transform domain. Our new moment-preserving method is introduced in Section 4. Detailed derivations of this approximation technique and experimental results for one-dimensional and two-dimensional functions are presented in this section. Section 5 concludes the paper with some experimental results and a brief summary of our current work.

2 Related Work on One-dimensional Functions

The first group of approximation techniques, including subsampling, piecewise linear methods, least squares polynomial methods, and polynomial splines, calculates its approximation, $\hat{f}(x)$, by directly using the given values of x and $f(x)$. They are considered to be approximation methods in the time or spacial domain. The second group which includes the Fourier transform, Walsh-Hadamard transform, Karhunen-Loeve transform, and cosine transform, determines the approximation of f by *transforming* it into another function, say F, approximating F, and exploiting an *inverse transform* to derive $\hat{f}(x)$. Clearly, these approximation techniques operate on the *transformed* function F and in the domain of the transformed function. These two groups of approximation techniques will be briefly described in their respective sections below. A more detailed survey is found in [Ng 94].

2.1 Time/Spacial Domain Function Approximation

2.1.1 Subsampling

In many signal processing applications, especially in digital audio, it is necessary to convert an analog signal into a discrete sequence of numbers. The conversion process is commonly done by sampling the continuous analogue signal, $f(x)$ at regular interval Δx to produce a sequence $S = \{f(x_0), f(x_1), f(x_2), \ldots, f(x_{n-1})\}$, where $x_i = x_0 + i\Delta x$ and x_0 is some given starting time. Subsampling, also known as decimation, consists of determining a sequence $\widehat{S}$ which contains exactly n uniformly selected points from S, where $n < N$. Frequently, the value of n is not specified, but a decimation factor, δ, is provided, and in these cases, n can be calculated to as, $n = \lfloor N/\delta \rfloor$. If the given sequence S is uniform, then the decimation by a factor of δ will generate an approximation sequence, $\widehat{S} = \{f(x_j)\}$ where $j = 0, \delta, 2\delta, \ldots, n\delta$. This subsampling technique requires no calculations on x_i or $f(x_i)$.

Prusinkiewicz [PC 85] proposed a progressive subsampling scheme which is intuitively based on the following two propositions:

1. A translated set of sampling points uniformly distributed in a line is a set of uniformly distributed points, and,

2. The union of appropriately translated sets of uniformly distributed points is also a set of uniformly distributed points, with a reduced distance between adjacent points.

If a given string of data S represents a signal of full resolution, it can be divided into substrings $S_0, S_1, \ldots, S_n$. Each substring S_i, representing a lower resolution of S, results from the traversal algorithm (subsampling technique) and the overlapping sampled areas. Let $S =< f_0, f_1, f_2, \ldots >$ be a string of points of the one-dimensional function f, and $S(i,j)$ be a substring of S described by:

$$S(i,j) =< f_{j \cdot 2^i}, f_{j \cdot 2^i + 1}, \ldots, f_{(j+1) \cdot 2^i - 1} >$$

Furthermore, let $S_{\vec{x}}(i,j)$ denote the translation of the substring $S(i,j)$ by the vector $\vec{x}$, and let N denote the size of the original sequence S which represents the sampled values of the function f. The string of sampled points is then defined as follows:

$$\begin{aligned} S(0,0) &= < f_N > \\ S(k,1) &= S_{-N \cdot 2^{-k-1}}(k,0) \\ S(k+1,0) &= S(k,0) \oplus S(k,1) \end{aligned}$$

where $\oplus$ denotes the concatenation of strings, and $k = 0, 1, \ldots$.

These subsampling techniques are easy to apply and are computationally inexpensive due to the absence of any complicated mathematical equations. This lack of computation is also the main downfall of subsampling techniques because it introduces aliasing into the approximated signals which, in turn, causes audio signals to lose their relevant higher frequency components and images to become blurry.

2.1.2 Least squares polynomial

The main concept of the least squares polynomial technique [Hi 87] is to employ a set of $n+1$ orthogonal functions $\phi_0(x), \phi_1(x), \ldots, \phi_k(x)$ to perform the approximation such that,

$$\hat{f}(x) \equiv \sum_{i=0}^{n} a_i \phi_i(x) \approx f(x).$$

The approximating polynomial $\hat{f}(x)$ is of degree n, and the values of $\hat{f}(x)$ on a set of $N+1$ points agree "as well as possible" with known values of $f(x)$ at those points. These normal equations can be obtained by first writing down the $N+1$ equations where the $N+1$ points $x_0, x_1, \ldots, x_N$ exactly equal $f(x)$

$$\begin{array}{ccccccccc}
a_0\phi_0(x_0) & + & a_1\phi_1(x_0) & + & \cdots & + & a_n\phi_n(x_0) & = & f(x_0) \\
a_0\phi_0(x_1) & + & a_1\phi_1(x_1) & + & \cdots & + & a_n\phi_n(x_1) & = & f(x_1) \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
a_0\phi_0(x_n) & + & a_1\phi_1(x_n) & + & \cdots & + & a_n\phi_n(x_N) & = & f(x_N)
\end{array}$$

The k^{th} normal equation can thus be obtained by multiplying each equation above by the coefficient of a_k in that equation and also by the weight associated with that equation, and summing the results.

2.1.3 Piecewise linear

In many cases, the original discrete function f is very complex and cannot be expressed by a simple closed-form equation. The function is thus divided up into smaller segments, and each segment is then approximated by a simple linear equation. The point where one segment joins the next is often referred to as a knot or a joining point. All of the techniques presented in this section are based on the same piecewise linear principle, but they do vary on their predefined conditions such as using a fixed number of knots or variable number of knots, and the question of whether the spacing between the knots is of fixed size or of variable size.

Stone [St 61] presented the problem of trying to obtain a set of best-fit line segments to a curve and to give a closed form solution for the procedure. The problem formulated by Stone consists of the original given function $y = f(x)$ defined between the two endpoints u_0 and u_N, where $u_0 \leq x \leq u_N$. Formally, the problem reduces to determining an approximation, y, where

$$y = \begin{cases} a_1 + b_1 x & u_0 \leq x \leq u_1 \\ \vdots & \\ a_N + b_N x & u_{N-1} \leq x \leq u_N \end{cases}$$

such that

$$F(a_1, a_2, \ldots, a_N, b_1, b_2, \ldots, b_N, u_1, u_2, \ldots, u_{N-1}) = \sum_{j=1}^{N} \int_{u_{j-1}}^{u_j} [f(x) - a_j - b_j x]^2 dx$$

is minimized.

Bellman [Be 61] made the following observation in Stone's derivation. For fixed $a = u_0$ and $b = u_{N+1}$, where $b \geq a$,

$$f_N(b) = \min_{[a_i b_i u_i]} F(a_1, \ldots, a_N; b_1, \ldots, b_N; u_1, \ldots, u_N)$$

then

$$f_1(b) = \min \left[\int_a^{u_1} (g(x) - a_1 - b_1 x)^2 dx + \int_{u_1}^{b} (g(x) - a_2 - b_2 x)^2 dx \right]$$

where the minimum is computed over:

$$-\infty < a_1, a_2, b_1, b_2 < \infty, a \leq u_1 \leq b.$$

This function is readily determined since the minima can be computed over the a_i and b_i and then minimized over u_i by means of a discrete search. For $N \geq 2$,

$$f_N(b) = \min \left[\min_{[a_N b_N]} \int_{u_N}^{b} (g(x) - a_N - b_N x)^2 dx + f_{N-1}(u_N) \right]$$

$$f_N(b) = \min_{a \leq u_N \leq b} [h(u_N, b) + f_{N-1}(u_N)].$$

Gluss [Gl 62] reiterated the problem defined by Stone in simpler terms. If the number of points subdividing the given interval $[\alpha, \beta]$ is given, say n, then the main task is to find the n line segments which will give the best least squares approximation to a function $f(x)$ defined in that interval. Although Gluss' approach is similar to Bellman's, he also offered an equation for u_j which may be determined by an equality sign instead of a minimization. He also showed how dynamic programming produces computationally simple equations for the model in which the ends of the line segments are constrained to lie on the curve $f(x)$.

Phillips [Ph 68] used a different set of premises than Gluss, Bellman, and Stone. His approach was to determine an approximation to a given function $f(x)$ by k line segments with any preassigned accuracy. The approximation is obtained using the minimax norm [4]. If $f''(x) > 0$ on

[4]Minimizing the maximum error norm.

the interval $[\alpha, \beta]$, it is known that the straight line $px + q$ which minimizes the maximum error $|f(x) - (px + q)|$ on $[\alpha, \beta]$ satisfies the following equations,

$$f(\alpha) - (p\alpha + q) = \epsilon$$

$$f(\xi) - (p\xi + q) = -\epsilon$$

$$f(\beta) - (p\beta + q) = \epsilon$$

$$f'(\xi) - p = 0 \Longrightarrow p = f'(\xi)$$

where ξ is an interior point of $[\alpha, \beta]$ and ϵ is the maximum error. Davis [Da 63] showed that for a given α and β, the equations above have unique solutions for p, q, ξ, and ϵ, and thus, for a given α and ϵ, the equations have unique solutions for p, q, ξ, and β.

Pavlidis [Pa 73] also presented a waveform segmentation technique using linear approximation which is based on discrete optimization. Let $\{(x_0, f_0), (x_1, f_1), \ldots, (x_N, f_N)\}$ be the set of points describing the waveform to be analysed in a quantized form. Also, let $u_1^k, u_2^k, \ldots, u_n^k$ be the dividing points at the k^{th} iteration where $u_0 = x_0$ and $u_n = x_N$. Without loss of generality, let u_j^k be last point of the j^{th} segment. At each iteration, a new set of dividing points is selected such that the error norm, e_i, is minimized, where e_i is the error norm of the i^{th} segment. Since at each iteration the error norm becomes smaller and there is a finite number of discrete points describing $f(x)$, the dividing points will eventually converge on the local maxima or minima of the function.

All of these techniques, with the exception of Pavlidis's work, can only be applied to simple one-dimensional, continuous, well-defined functions where the approximated points do not necessarily lie on $f(x)$. Many other piecewise linear approximation methods including those introduced by Ramer[Ram 72], Pavlidis and Horowitz [PH 74], Kurozumi and Davis [KD 82], and Dunham[Du 86], apply to both one-dimensional functions and planar curves; these techniques will be examined in Section 3.

2.1.4 Polynomial splines

A smooth curve passing through the points $\{(x_0, f_0), (x_1, f_1), \ldots, (x_{n-1}, f_{n-1})\}$ is the Lagrange interpolating polynomial:

$$P(x) = \sum_{i=0}^{n-1} \frac{L(x)}{x - x_i} \frac{f_i}{L'(x_i)}$$

where

$$L(x) = \prod_{i=0}^{n-1}(x - x_i)$$

The main problem with the Lagrange polynomial approximation shown above is that as the number of points increases these polynomials tend to oscillate strongly with a direct dependence on the position of the points (x_i, f_i). For example, if the set of points is given as $\{(0,6),(6,6),(8,4),(10,6),(14,3),(16,3),(20,2)\}$, then P takes on functional values greater than 30 in the first interval $[0,6]$.

As a compromise between a linear approximation and a polynomial of higher degree, a combination of low degree polynomials and a function that is often differentiable on the closed interval $[a,b]$ is used. If the set of polynomials of degree $2m+1$ defined in the intervals $[x_i, x_{i+1}]$, where $m \leq n-1$ and $i = 0, 1, \ldots, n-2$ are joined at nodes $x_1, \ldots, x_{n-2}$ in such a way that the resulting function is always $2m$ times differentiable at these nodes, then the function defined in $[a,b]$ is called a spline function of degree $2m+1$. A third order spline or cubic spline consists of $n-1$ third degree polynomials connected at the points $\{(x_1, f_1), (x_2, f_2), \ldots, (x_{n-2}, f_{n-2})\}$ such that they are always twice differentiable at these nodes [Sp 74].

B-splines are piecewise polynomial curves (usually cubic) which are related to a guiding polygon. B-spline approximations not only have good computational properties but also representational ones over linear and polynomial approximations [BB 82]. First, they are *variation diminishing*; the curve is, therefore, assured to lie between the convex hull of groups of $k+1$ consecutive points where k is the degree of the interpolating polynomial. Secondly, the interpolation of the curve is local; if a point of the guiding polygon is shifted, then the effects are restricted to nearby points on the spline and not the entire approximation. A formal derivation of B-spline is given by [BR 74] and de Boor [deB 78].

2.2 Transformed Domain Function Approximation

2.2.1 Karhunen-Loeve Transform

If a sinusoidal signal is transmitted over a medium in a sampled form, with each sampled data point being sent in a sequential manner, then a higher sampling rate will yield a superior reconstruction of the waveform. From a theoretical point of view, the information content of the sampled values is very low due to the fact that the sampled values are highly correlated. On the other hand, it is also well known that the reconstruction of the original signal can be accomplished by knowing its magnitude, phase, frequency, starting time, and the fact that it is sinusoid. The Karhunen-Loeve transform decorrelates the sampled values into an uncorrelated set of values that are equivalent to the five parameters listed above. The Karhunen-Loeve transform is considered to be an optimal transform because it completely decorrelates the signal in the transform domain; it minimizes the MSE in bandwidth reduction or data compression; and it contains the most variance (energy) in the fewest number of transform coefficients. In practice, the Karhunen-Loeve transform is not often applied because it is dependent on the input data; that is, it is difficult to determine the auto-covariance matrix and its diagonalization. Furthermore, the lack of pre-determined basis vectors in the transform domain has made it an ideal but impractical technique.

2.2.2 Fourier Transform

Fourier's theorem states that given any function $f(x)$ defined in the interval $-\infty < x < \infty$, it is possible to express it as a summation of a series of sine and cosine terms of increasing frequency. The discrete Fourier transform is written as

$$F_u = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} f_k e^{-j2\pi uk/n}$$

where $u = 0, 1, \ldots, n-1$, and n is the number of uniformly spaced sampled points along the function $f(x)$. The inverse Fourier transform is similar to the forward transform except for the change in sign in the exponential term,

$$f_k = \frac{1}{\sqrt{n}} \sum_{u=0}^{n-1} F_u e^{j2\pi uk/n}$$

where $k = 0, 1, \ldots, n-1$. The value u represents the number of discrete frequency components added together to construct the sampled value f_k.

2.2.3 Hadamard and Walsh-Hadamard Transforms

The Hadamard and Walsh-Hadamard [RJ 91] transforms are not as efficient in an energy-packing sense as the transforms described above. They are characterized by a symmetric matrix whose elements are ± 1, and their basis vectors are orthonormal. The term *sequency* is defined as one-half the number of changes in sign in one period of a sequence. The sequency of a rectangular basis waveform in the Walsh-Hadamard transform has the same connotation that the frequency of a trigonometric sine and cosine waveform has in the Fourier transform. The Hadamard transformation matrices are only defined for input sequence of sizes 2^p or $4p$ for positive integral values of p. The Walsh transformation matrices are similar to the Hadamard matrices except that they can be of any size and that the ordering of the basis vectors is slightly different. The term Walsh-Hadamard transform is generally used to describe any transform that has basis vectors with elements ± 1 and that has a size, n, which is some power of 2. The Walsh-Hadamard transforms do not diagonalize covariance matrices which are Toeplitz, and thus, they are suboptimal in an energy-packing sense (not even asymptotically). They do possess some decorrelating characteristics, and they are quite popular due to their simplicity.

2.2.4 Discrete Cosine Transform

The discrete cosine transform provides a way of performing frequency analysis, yet it avoids the complex valued results generated from the Fourier transform by exploiting a unique feature of symmetric functions and their Fourier transforms. The discrete cosine transform and its inverse can be defined in a more formal setting than the description above as [ANR 85],

$$F_k = \sqrt{\frac{2}{n}} C(k) \sum_{i=0}^{n-1} f_i \cos \frac{(2i+1)k\pi}{2n}; \qquad k = 0, 1, \ldots, n-1$$

and

$$f_i = \sqrt{\frac{2}{n}} C(k) \sum_{k=0}^{n-1} F_k \cos \frac{(2k+1)i\pi}{2n}; \qquad i = 0, 1, \ldots, n-1$$

where $C(0) = \frac{1}{\sqrt{2}}$ and $C(k) = 1$ for $k \neq 0$. There are several reasons why the discrete cosine transform is used in many signal processing applications. It is almost as optimal as the Karhunen-Loeve transform yet its basis vectors are independent of the input signals. Compared to the Fourier transform, it does not have to handle complex numbers, and it can be computed using a fast discrete Fourier transform technique.

3 Related Work on Two-dimensional Functions

Approximation of planar curves and digitized images is an important part of pattern recognition, image processing, and graphics applications. All of the approximation methods on two-dimensional functions are "philosophically" based on the one-dimensional techniques presented in the previous chapter. These techniques can be divided into those which approximate planar curves, and those which approximate digitized images. Within each one of these categories, the approximation techniques will be grouped as either spacial or transformed domain approximations, and a brief discription of these techniques will be reviewed in this chapter. A more detailed survey is found in [Ng 94].

3.1 Planar Curve Approximation

3.1.1 Time/Spacial Domain Approximation

Montanari [Mo 70] introduces the problem of approximating a digitized contour such that the polygonal approximation has a minimal length. This technique takes into consideration an initial point $P_{0_i} = (x_{0_i}, y_{0_i})$ and a shifted point $P_i = (x_i, y_i)$ such that C_i will be the domain centred in P_{0_i}. If the total number of points in the polygon under examination is N and assuming that $x_{N+1} = x_1$ and $y_{N+1} = y_1$, the function

$$g(\mathbf{z}) = \sum_{k=1}^{N} \left[(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2\right]^{\frac{1}{2}}$$

must be minimized with the constraints

$$P_i \in C_i, \qquad i = 1, 2, \ldots, N.$$

The function $g(\mathbf{z})$ is a "cost" function of $\mathbf{z}$, where $\mathbf{z}$ is assumed to be a vector in an Euclidean space with $2N$ dimensions with N constraints. Montanari's method of computing the smoothed polygon of minimum length thus reduces to a nonlinear programming problem.

Ramer's [Ram 72] objective in approximation of irregular planar curves is to provide an efficient representation of a digitized picture generated in sampling the gray-value function over the domain of an image on a regular square grid. The resulting curves can be interpreted as polygons whose vertices lie on the sampling points or whose edges coincide with the cell boundaries. Ramer's technique is simple, yet it provides a good linear approximation of the given curve. There are a few problems with this algorithm. If the original curve contains undesirable noise or spikes, this technique has a tendency to include them as part of the approximation. Moreover, the number of approximating lines is not guaranteed to be minimal.

Pavlidis and Horowitz [PH 74] present an algorithm which approximates a planar curve by straight line segments. The main goal of their algorithm is to describe boundaries of objects in a compact and meaningful way by filtering undesirable noise while simultaneously retaining the major features of the objects. The split-and-merge algorithm considers two cases. In the first, it attempts to maintain the error norm in each segment to be less than or equal to maximum allowable error such that $E^i \leq E^{max}$ for $i = 1, 2, \ldots, n_k$. In the second case, it is necessary to have the overall error norm of the segments less than or equal to E^{max}. In practice, this method behaves moderately well.

Kurozumi and Davis [KD 82] approach the approximation of the planar curve problem by estimating the error between the given curve and the approximated line segments. Their polygonal approximation algorithm uses the minimax method which obtains a set of line segments which minimize the maximum of the Euclidean distances between the lines and the given points. Furthermore, in order to minimize the number of line segments, an allowable gap is used. Although, two minimax lines are allowed to intersect at some point not on the curve, the point of intersection must be within a gap that is contained in a circle centred by one of the points on the curve with radius ϵ.

Dunham's [Du 86] approach to the approximation of two-dimensional digital curves is based on Bellman's [Be 61] work. Dunham uses the uniform error norm instead of the mean square error norm used by Bellman. Dunham's approximations are continuous, and attempt to minimize the number of approximating line segments by fixing the allowable error. Dunham categorizes his method as an optimal solution based on the criterion that his polygonal approximation contains the least number of edges, and which simultaneously satisfies the uniform error restriction. There are several problems associated with this technique. If the planar curve is closed, the derived approximation is not necessarily optimum (compared to the open planar curves); it tackles the problem by invoking a costly exhaustive search using the open planar curve algorithm. Furthermore, in certain cases, the time required to derive an approximation increases inversely with the maximum error limit.

Lowe [Lo 87] introduces a piecewise linear approximation technique which uses the ratio of the length of the line segment and the maximum deviation of any point from the line. The algorithm is based on the assumption that a digitized curve is derived from a bitmap, and as such, the maximum deviation is always at least two pixels in size to account for limitations on measurement accuracy. This error estimation, thus, provides a scale-independent measure of significance which places no prior expectations on the allowable deviations. This algorithm is similar to that of Ramer's, and it does a reasonable job of detecting the perceptually most significant straight line groupings of the original data points. Whereas other traditional methods require setting of a prior threshold for the amount of "noise" to be removed from a curve, this estimation technique does not require such a threshold value. It tends to find the same structures regardless of the size at which an object appears in an image, and it avoids breaking up long lines if shorter constituents do not possess stronger perceptual elements. The major shortcoming of Lowe's ratio of length-to-deviation is that if the deviation is very small, it causes overflow errors. Rosin and West [RW 89] [RW 91] have generalized Lowe's technique by inverting the ratio to avoid this problem.

3.1.2 Transformed Domain Approximation

Granlund [Gr 72] uses the coefficients of Fourier transformations to represent features which are significant for a pattern. One of Granlund's requirements was that the processed patterns are described in an optimal way. Moreover, in the particular case of character recognition, his aim was to determine features which are invariant for different types of style, size, and slant of character. The main advantage of Granlund's transformed representation is that the inverse transform over a subset of the Fourier descriptors always produces closed curves. Among the disadvantages, this technique requires all N descriptors for a complete description. Moreover, the transform coefficients are not independent, and therefore, there is some redundancy in the description.

The problem of extracting a finite set of numerical features from a closed planar curve was examined by Zahn and Roskies [ZR 72]. Their main focus is to employ a set of Fourier descriptors to characterize features in order to separate the shapes of different classes relative to the intraclass dispersion. The idea presented by Zahn and Roskies is based on previous work done by Cosgriff and others [Co 60]. However, Zahn and Roskies use a different normalizing function than the ones used by their predecessors. One of the major advantages of using this representation is that there is no redundant information present in the set of Fourier descriptors compared to the representation provided by Granlund [Gr 72]. There are two major disadvantages of using this technique. First, some sequences of Fourier descriptors describe non-closed curves, and secondly, the reconstruction of the curve requires numerical integration.

3.2 Digitized Image Approximations

3.2.1 Time/Spacial Domain Techniques

Tanimoto [Ta 79] describes a hierchical data structure for picture processing called the *pyramid data structure*. If an image is defined as $I(m,n)$, where $0 \leq m,n \leq N = 2^L$, then lower resolution approximations of the original $N \times N$ images can be defined as

$$\hat{I}_{k-1}(i,j) = \frac{1}{4}[I_k(2i,2j) + I_k(2i+1,2j) + I_k(2i+1,2j+1) + I_k(2i,2j+1)]$$

where $0 \leq k \leq L$ and $0 \leq i,j \leq 2^{k-1}$. A sequence of reduced resolution images from the original is formed by averaging 2×2 blocks of pixels. There are $L+1$ "levels" of the pyramid, the most reduced of which is a 1×1 image (a single pixel) whose value is the overall average of the original image. The levels of the pyramid are transmitted top-down; thus, the average of the whole image is sent first, followed by the 2×2 image, 4×4 image, etc., until the $2^L \times 2^L$ original is sent. The main purpose of this technique is to allow interactions between a user and the application handling a large amount of visual information. If images are stored in this manner, their approximations can be transmitted to allow interactions between the application and the users and to reduce communication cost. The main problem with the pyramid method is that the construction of the pyramid approximations require a large amount of memory. In the worst case scenario, the communication cost is doubled because the original image and all of its approximations need to be sent.

Prusinkiewicz's [PC 85] one-dimensional subsampling technique can be easily extended to two-dimensional images. If S is the original signal at full resolution, it can be divided into substrings $S_0, S_1, \ldots, S_N$ where each substring, S_i, represents a reduced resolution of S. The main difference between this method and the binary tree, quadtree, or the pyramid structure [Ta 79] is that any substring, S_k, can be used to reconstruct the original signal S at the particular resolution at level k. As opposed to this, with the hierarchical structure representation only various initial substrings can be used to reconstruct the approximation of the entire string S - their non-initial substrings are meaningless. Moreover, the cost of the overhead of node structures which are inherent in hierarchical representations is not incurred by Prusinkiewicz's algorithm.

The sampling method proposed by Prusinkiewicz starts at the top right hand corner of an image and traverses to the bottom-left corner from one level to the next. One of the problems with this traversal algorithm is that if important information is contained in the lower left-hand corner of an image, then the first several substrings will be devoid of this information. Nevertheless, because the width and height double at each successive stage, the sampling area is quadrupled, and therefore, converges very quickly. The reconstruction method proposed by Prusinkiewicz consists of linearly interpolating the unknown cells using the sampled values, and by doing so, the staircase effects caused by undersampling are minimized.

3.2.2 Transform Domain Techniques

The discrete Fourier transform of a two-dimensional function $f(i,k)$ is an extension of the one-dimensional transform, and is given as:

$$F(u,v) = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{k=0}^{N-1} f(i,k)e^{-\frac{j2\pi(ui+vk)}{N}}$$

where $0 \leq u, v < N$. Its inverse is defined as

$$f(i,k) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{\frac{j2\pi(ui+vk)}{N}}.$$

In both equations above, the value of j is defined as $j = \sqrt{-1}$. The discrete Fourier transform coefficients are complex values consisting of real and imaginary components or magnitude and phase components. Storage and manipulation of these complex values can be cumbersome. Another disadvantage of the discrete Fourier transform is that spurious spectral components are generated due to the implicit periodicity of the image blocks. When encoded at low bit rates, these spurious components may give rise to severe block artifacts leading to the so-called "staircase effect" [RJ 91].

The discrete cosine transform for a two-dimensional function $f(i,k)$ can be described as

$$F(u,v) = \frac{4C(u)C(v)}{N^2}\sum_{i=0}^{N-1}\sum_{i=0}^{N-1} f(i,k)\cos\left[\frac{(2i+1)u\pi}{2N}\right]\cos\left[\frac{(2k+1)v\pi}{2N}\right]$$

and the inverse two-dimensional discrete cosine transform is defined as

$$f(i,k) = \sum\sum C(u)C(v)F(u,v)\cos\left[\frac{(2i+1)u\pi}{2N}\right]\cos\left[\frac{(2k+1)v\pi}{2N}\right]$$

where

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } w = 0 \\ 1 & \text{for } w = 1, 2, \ldots, N-1. \end{cases}$$

The discrete cosine transform does not generate spurious spectral components like the discrete Fourier transform, and therefore, it is very efficient in coding by compacting a large amount of information into a few coefficients. The discrete cosine transform has become the most widely used transform for audio and image compression, and has been adapted by the International Standards Organization (ISO) Joint Photographic Experts Group (JPEG) still-image coding scheme.

4 Moment-Preserving Method

Most of the approximation techniques that have been proposed, try to approximate $f(x)$ by some function $\hat{f}(x)$ in such a way that the two functions are *reasonably* close to each other in the domain of the variable x. This requirement will be extended so as to obtain an approximated function which is reasonably close in both the domain of variable x and its transform domain, where the transformation is the geometric moment function. Since the constraints to be satisfied are much more stringent, they will hopefully lead to a more interesting and powerful strategy.

4.1 Introduction to Moments

In classical mechanics, the concept of moments is used extensively. For example, an object of mass m located at position x on the x-axis is said to have moment xm about the point $x = 0$, or more generally, moment $(x - x_0)m$ about the point x_0. For a one-dimensional distribution of mass with continuously variable line density $\rho(x)$ lying along an interval $[0, a]$ of the x-axis, an element of length dx at position x contains mass $dm = \rho(x)dx$, has moment $dM = xdm = x\rho(x)dx$ about the origin. Therefore, the total moment about the origin is defined as

$$M = \int_0^a x\rho(x)dx.$$

In probability theory of random variables, moments play an important role. If a random variable can take on values x_1 with probability p_1, x_2 with probability $p_2, \ldots, x_N$ with probability p_N where $\sum_i^N p_i = 1$, then the mean μ or expectation $E[X]$ of that random variable is given by

$$\mu = E[X] = \sum_{i=1}^{N} x_i p_i.$$

For a continuous random variable X with probability density function $\rho(x)$ [5] in the interval $[a, b]$, the analogous definition of the mean or the expectation of X is

$$\mu = E[X] = \int_a^b x\rho(x)dx.$$

More generally, the expectation of any function $g(X)$ of the random variable X is given as

$$E[g(X)] = \int_a^b g(x)\rho(x)dx$$

and the particular case of the mean of a random variable X, $g(x) = x$. The variance σ^2 of a random variable X with density $\rho(x)$ in the interval $[a, b]$ is the expectation of the square of the distance from X to the mean μ, and hence

$$\begin{aligned} \sigma^2 &= E[(X - \mu)^2] \\ &= E[X^2] - E[X]^2. \end{aligned}$$

[5] If $\rho(x)$ were a mass density such as the case in the classical mechanics example, then μ would be the moment of the mass about 0, and since the total mass would be $\int_a^b \rho(x)dx = 1$, μ would in fact be the centre of mass.

Typically, the first three moments of a function are very useful in describing some of the characteristics of a random variable X with density function $\rho(x)$. The zeroth moment is the area under the curve $\rho(x)$, which can be interpreted as the mass of an object or the total probability. The first moment describes the centre of mass of an object or equivalently the mean or average of the random variable X. The second moment in combination with the mean describes the moment of inertia for that object or the variance of the random variable X.

4.2 Relationship between Moments and the Characteristic Equations

The characteristic function of a random variable X is the Fourier transform of its density function $\rho(x)$ (with a reversal in sign) [Pap 65], and is defined as

$$\begin{aligned}\phi(\omega) &= E[e^{j\omega X}] \\ &= \int_{-\infty}^{\infty} e^{j\omega x}\rho(x)dx.\end{aligned}$$

This function describes the expected value of the complex function $e^{j\omega X} = \cos\omega X + j\sin\omega X$. If X is of discrete type, taking values x_k, then

$$\phi(\omega) = \sum_k e^{j\omega x_k} P\{X = x_k\}$$

where $P\{X = x_k\}$ is the probability that the random variable X is equal to some value x_k. If m_k represents the k^{th} geometric moment of a function $\rho(x)$ and is given as

$$m_k = E[X^k] = \int_{-\infty}^{\infty} x^k \rho(x)dx$$

where $\rho(x)$ is the density function of a random variable X, then the derivatives of the characteristic function of a random variable X are related to its moments by

$$\frac{d^k\phi(0)}{d\omega^k} = j^k m_k.$$

4.3 Moment-Preserving Approximation

If $f(x)$ and $g(x)$ are two functions defined between the interval $[a,b]$, then from the theory of moments all the k^{th} order moments $\int x^k f(x)dx$ and $\int x^k g(x)dx$ will be exactly equal for all values of k iff $f(x) \triangleq g(x)$. However, $\hat{f}(x)$ will be considered to be a *reasonably* good approximation of $f(x)$ if their respective moments match to some finite order, k. Another requirement in this approximation technique is that it must also be *reasonably* good in the transform domain.

To initiate this strategy, it is assumed that the $f(x)$ is continous, integrable, and that all of its moments exist. The above assumptions are basically identical to assuming that the appropriate transform of the function exists. In a practical situation, however, the explicit functional form of $f(x)$ will be unknown, and it is often specified by a set S of N discrete sampling points as

$$S = \{p_i = (x_i, y_i) | i = 0, 1, \ldots, N\}.$$

An approximation $\hat{f}(x)$ will be derived from S such that $\hat{f}(x)$ is specified a set $\widehat{S}$ of n unique points where $n << N$,

$$\widehat{S} = \{\hat{p}_k = (\hat{x}_k, \hat{y}_k) | k = 0, 1, \ldots, n\}.$$

$\hat{f}(x)$ is a piecewise polynomial joining the points in the set $\widehat{S}$, and in this particular case the polynomial is restricted to be linear functions.

The main premise of this approximation technique is based on the idea that two functions will be approximately the same if a finite number of their moments are equal. Since the function of $f(x)$ is assumed to be known, the moments of the function, m_k, are also assumed to be known. The main task is to approximate $f(x)$ by piecewise linear functions. As indicated in the previous sections, there are many different ways of approaching this problem. However, our constraint here will be to determine an approximation function $\hat{f}(x)$ such that a finite number of its theoretical moments will be **exactly identical** to the moments of the original function $f(x)$.

Definition 1 *Let $f(x)$ and $g(x)$ be two integrable functions between interval $[a,b]$. If $\int x^i f(x)dx = \int x^i g(x)dx$ for $a \leq x \leq b$ for all $0 \geq i \geq k$, then $f(x)$ and $g(x)$ are said to be k-order equivalent.*

Consider a known function f and some function g which is used to approximate f. If we make f and g to be $k-$order equivalent, it implies that the moments, up to order k, of f and g are exactly identical which, in turn, implies that in the time/spacial domain and in the transform domain they are approximations. In particular, we note that for $k = 0$ the moment definition for m_0, the zeroth moment, is the area of under the curve $f(x)$. In the analogous case of a discrete function f defined by the set of points $\{(x_0, f_0), (x_1, f_1), \ldots, (x_N, f_N)\}$, the k^{th} moment is

$$m_k = \sum_{i=0}^{N-1} x_i^k f_i \Delta x_{i+1}$$

where $\Delta x_{i+1} = x_{i+1} - x_i$.

Derivations for the moment-preserving approximation method will be done for three cases. The first case approximates a given function f by a single line segment; the second case approximates it by two line segments; the third case is the general solution where the number of line segments is n.

4.3.1 Single Straight Line (2-point) Approximation

Let $f(x)$ be a one-dimensional function to be approximated by two points $(x_{i_0}, \hat{f}_{i_0})$ and $(x_{i_1}, \hat{f}_{i_1})$ where x_{i_0} and x_{i_1} are the knot values defined in the domain of $f(x)$ and are uniformly (or non-uniformly) selected. The values $\hat{f}_{i_0}$ and $\hat{f}_{i_1}$ must be calculated such that a finite number of moments of the line segment passing through the approximated points $(x_{i_0}, \hat{f}_{i_0})$ and $(x_{i_1}, \hat{f}_{i_1})$ are exactly identical to that of the given function f. Since there are two unknowns, a system of at least two analytic equations involving the moments and the two unknowns $\hat{f}_{i_0}$ and $\hat{f}_{i_1}$ is required. This leads us to the following results.

Theorem 4.1 *Let m_0 and m_1 be the true zeroth and first moments of the function $f(x)$. Then the first order equivalent function $\hat{f}(x)$ approximating $f(x)$ can be solved as the line $\overline{(x_{i_0}, \hat{f}_{i_0}), (x_{i_1}, \hat{f}_{i_1})}$, where x_{i_0} and x_{i_1} are the knot values defined in the domain of $f(x)$ and $\hat{f}_{i_0}$, and $\hat{f}_{i_1}$ obey:*

$$\begin{bmatrix} \hat{f}_{i_0} \\ \hat{f}_{i_1} \end{bmatrix} = \begin{bmatrix} \frac{x_{i_1}-x_{i_0}}{2} & \frac{x_{i_1}-x_{i_0}}{2} \\ \frac{x_{i_1}^3-3x_{i_1}x_{i_0}^2+2x_{i_0}^3}{6(x_{i_1}-x_{i_0})} & \frac{-x_{i_1}^3+3x_{i_1}x_{i_0}^2-2x_{i_0}^3}{6(x_{i_1}-x_{i_0})} + \frac{x_{i_1}^2-x_{i_0}^2}{2} \end{bmatrix}^{-1} \cdot \begin{bmatrix} m_0 \\ m_1 \end{bmatrix}$$

<u>Proof</u>:

The first two moments of the approximated function $\hat{f}(x)$ are:

$$\widehat{m}_0 = \int_{x_{i_0}}^{x_{i_1}} \hat{f}(x)dx \tag{1}$$

$$\widehat{m}_1 = \int_{x_{i_0}}^{x_{i_1}} x\hat{f}(x)dx, \tag{2}$$

and the equation of the straight line segment passing through the points $(x_{i_0}, \hat{f}_{i_0})$, and $(x_{i_1}, \hat{f}_{i_1})$ is given by:

$$\hat{f}(x) = \frac{\hat{f}_{i_1} - \hat{f}_{i_0}}{x_{i_1} - x_{i_0}}(x - x_{i_1}) + \hat{f}_{i_1}.$$

Substituting $\hat{f}(x)$ into the moment equations (1) and (2) above and proceeding with the integration, the following equations are derived,

$$\widehat{m}_0 = \int_{x_{i_0}}^{x_{i_1}} \left[\frac{\hat{f}_{i_1} - \hat{f}_{i_0}}{x_{i_1} - x_{i_0}} \cdot (x - x_{i_1}) + \hat{f}_{i_1} \right] dx$$

which can be simplified as:

$$\widehat{m}_0 = \hat{f}_{i_0} \cdot \left[\frac{x_{i_1} - x_{i_0}}{2}\right] + \hat{f}_{i_1} \cdot \left[\frac{x_{i_1} - x_{i_0}}{2}\right]. \tag{3}$$

$$\tag{4}$$

Similarly, $\widehat{m}_1$ can be expanded as

$$\widehat{m}_1 = \int_{x_{i_0}}^{x_{i_1}} x \left[\frac{\hat{f}_{i_1} - \hat{f}_{i_0}}{x_{i_1} - x_{i_0}} \cdot (x - x_{i_1}) + \hat{f}_{i_1}\right] dx,$$

which has the form:

$$= \hat{f}_{i_0} \cdot \left[\frac{x_{i_1}^3 - 3x_{i_1}x_{i_0}^2 + 2x_{i_0}^3}{6(x_{i_1} - x_{i_0})}\right] + \hat{f}_{i_1} \cdot \left[\frac{-x_{i_1}^3 + 3x_{i_1}x_{i_0}^2 - 2x_{i_0}^3}{6(x_{i_1} - x_{i_0})} + \frac{x_{i_1}^2 - x_{i_0}^2}{2}\right]. \tag{5}$$

To get a first-order equivalent approximation we encounter the particular case of a single straight line segment approximation. We therefore require that the first two moments of the original function be preserved, and we equate $\widehat{m}_0 = m_0$ and $\widehat{m}_1 = m_1$. By simplifying the equations (3) and (5) from above, the following system of equations of two unknowns $\hat{f}_{i_0}$ and $\hat{f}_{i_1}$ are formed:

$$\begin{aligned}
\left[\frac{x_{i_1} - x_{i_0}}{2}\right] \cdot \hat{f}_{i_0} + \left[\frac{x_{i_1} - x_{i_0}}{2}\right] \cdot \hat{f}_{i_1} &= m_0 \\
\left[\frac{x_{i_1}^3 - 3x_{i_1}x_{i_0}^2 + 2x_{i_0}^3}{6(x_{i_1} - x_{i_0})}\right] \cdot \hat{f}_{i_0} + \left[\frac{-x_{i_1}^3 + 3x_{i_1}x_{i_0}^2 - 2x_{i_0}^3}{6(x_{i_1} - x_{i_0})} + \frac{x_{i_1}^2 - x_{i_0}^2}{2}\right] \cdot \hat{f}_{i_1} &= m_1.
\end{aligned}$$

They can be expressed in a matrix notation as

$$\begin{bmatrix} \frac{x_{i_1} - x_{i_0}}{2} & \frac{x_{i_1} - x_{i_0}}{2} \\ \frac{x_{i_1}^3 - 3x_{i_1}x_{i_0}^2 + 2x_{i_0}^3}{6(x_{i_1} - x_{i_0})} & \frac{-x_{i_1}^3 + 3x_{i_1}x_{i_0}^2 - 2x_{i_0}^3}{6(x_{i_1} - x_{i_0})} + \frac{x_{i_1}^2 - x_{i_0}^2}{2} \end{bmatrix} \cdot \begin{bmatrix} \hat{f}_{i_0} \\ \hat{f}_{i_1} \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \end{bmatrix} \tag{6}$$

Whence the theorem is proved. □

A detailed example of how the approximation works for a polynomial is found in [Ng 94]. We now proceed to approximating $f(x)$ using an additional central knot value.

4.3.2 Two Piecewise-Linear (3-point) Approximation

The same approach can be applied when the number of approximated points is increased from two to three. The only difference between this case and the single line approximation is the location of the middle point which is required to join the two line segments together. In order to approximate $f(x)$ with two line segments passing through the points $(x_{i_0}, \hat{f}_{i_0})$, $(x_{i_1}, \hat{f}_{i_1})$, and $(x_{i_2}, \hat{f}_{i_2})$ where $\hat{f}_{i_k}$'s are the approximated values, there must be a third linearly independent equation in the new system of equations involving the three unknowns. By using the first three moments of the approximated and actual functions, a linear system of three equations involving the three unknowns $\hat{f}_{i0}$, $\hat{f}_{i1}$, and $\hat{f}_{i2}$ can be set up. This motivates the following rule necessary for this moment-preserving technique:

Rule 1 *Let $f(x)$ be the original function and $\hat{f}(x)$ be its approximation. If $\hat{f}(x)$ approximates $f(x)$ by $n-1$ piecewise linear segments, it is necessary that the first n or more moments of $f(x)$ be preserved. Equivalently, this implies that at least the first n moments of $f(x)$ and $\hat{f}(x)$ must be identically equal.*

The above rule leads us to the following three-knot theorem:

Theorem 4.2 *If m_0, m_1, and m_2 are the true zeroth, first, and second moments of the function $f(x)$, respectively, the second-order equivalent function $\hat{f}(x)$ approximating $f(x)$ can be solved as the lines $\overline{(x_{i_0}, \hat{f}_{i0}), (x_{i_1}, \hat{f}_{i1})}$ and $\overline{(x_{i_1}, \hat{f}_{i1}), (x_{i_2}, \hat{f}_{i2})}$ where x_{i_0}, x_{i_1}, and x_{i_2} are the knot values defined in the domain of $f(x)$, and $\hat{f}_{i0}$, $\hat{f}_{i1}$, and $\hat{f}_{i2}$ obey:*

$$\begin{bmatrix} \hat{f}_{i0} \\ \hat{f}_{i1} \\ \hat{f}_{i2} \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} m_0 \\ m_1 \\ m_2 \end{bmatrix}$$

where

$$A = \begin{bmatrix} \frac{x_{i_1}-x_{i_0}}{2} & \frac{x_{i_1}-x_{i_0}}{2} + \frac{x_{i_2}-x_{i_1}}{2} & \frac{x_{i_2}-x_{i_1}}{2} \\ \frac{x_{i_1}^3-3x_{i_1}x_{i_0}^2+2x_{i_0}^3}{6(x_{i_1}-x_{i_0})} & \frac{-x_{i_1}^3+3x_{i_1}x_{i_0}^2-2x_{i_0}^3}{6(x_{i_1}-x_{i_0})} + \frac{x_{i_1}^2-x_{i_0}^2}{2} + \frac{x_{i_2}^3-3x_{i_2}x_{i_1}^2+2x_{i_1}^3}{6(x_{i_2}-x_{i_1})} & \frac{-x_{i_2}^3+3x_{i_2}x_{i_1}^2-2x_{i_1}^3}{6(x_{i_2}-x_{i_1})} + \frac{x_{i_2}^2-x_{i_1}^2}{2} \\ \frac{x_{i_1}^4-4x_{i_1}x_{i_0}^3+3x_{i_0}^4}{12(x_{i_1}-x_{i_0})} & \frac{-x_{i_1}^4+4x_{i_1}x_{i_0}^3-3x_{i_0}^4}{12(x_{i_1}-x_{i_0})} + \frac{x_{i_1}^3-x_{i_0}^3}{3} + \frac{x_{i_2}^4-4x_{i_2}x_{i_1}^3+3x_{i_1}^4}{12(x_{i_2}-x_{i_1})} & \frac{-x_{i_2}^4+4x_{i_2}x_{i_1}^3-3x_{i_1}^4}{12(x_{i_2}-x_{i_1})} + \frac{x_{i_2}^2-x_{i_1}^2}{2} \end{bmatrix}.$$

Proof: The first three moments of $\hat{f}(x)$ are defined as follows,

$$\widehat{m}_0 = \int_{x_{i_0}}^{x_{i_1}} \hat{f}(x)dx + \int_{x_{i_1}}^{x_{i_2}} \hat{f}(x)dx$$

which can be expanded to:

$$\begin{aligned} \widehat{m}_0 &= \int_{x_{i_0}}^{x_{i_1}} \left[\frac{\hat{f}_{i1} - \hat{f}_{i0}}{x_{i_1} - x_{i_0}} \cdot (x - x_{i_1}) + \hat{f}_{i1} \right] dx + \int_{x_{i_1}}^{x_{i_2}} \left[\frac{\hat{f}_{i2} - \hat{f}_{i1}}{x_{i_2} - x_{i_1}} \cdot (x - x_{i_2}) + \hat{f}_{i2} \right] dx \\ &= \frac{\hat{f}_{i1} - \hat{f}_{i0}}{2(x_{i_1} - x_{i_0})} \cdot (x_{i_1}^2 - x_{i_0}^2) - x_{i_1} \cdot \frac{\hat{f}_{i1} - \hat{f}_{i0}}{x_{i_1} - x_{i_0}} \cdot (x_{i_1} - x_{i_0}) + \hat{f}_{i1} \cdot (x_{i_1} - x_{i_0}) \\ &\quad + \frac{\hat{f}_{i2} - \hat{f}_{i1}}{2(x_{i_2} - x_{i_1})} \cdot (x_{i_2}^2 - x_{i_1}^2) - x_{i_2} \cdot \frac{\hat{f}_{i2} - \hat{f}_{i1}}{x_{i_2} - x_{i_1}} \cdot (x_{i_2} - x_{i_1}) + \hat{f}_{i2} \cdot (x_{i_2} - x_{i_1}). \end{aligned}$$

Similarly, $\widehat{m}_1$ has the form:

$$\begin{aligned}
\widehat{m}_1 &= \int_{x_{i_0}}^{x_{i_1}} x\hat{f}(x)dx + \int_{x_{i_1}}^{x_{i_2}} x\hat{f}(x)dx \\
&= \frac{\hat{f}_{i_1}) - \hat{f}_{i_0}}{3(x_{i_1} - x_{i_0})} \cdot (x_{i_1}^3 - x_{i_0}^3) - x_{i_1} \cdot \frac{\hat{f}_{i_1} - \hat{f}_{i_0}}{2(x_{i_1} - x_{i_0})} \cdot (x_{i_1}^2 - x_{i_0}^2) + \frac{\hat{f}_{i_1}}{2} \cdot (x_{i_1}^2 - x_{i_0}^2) \\
&\quad + \frac{\hat{f}_{i_2} - \hat{f}_{i_1}}{3(x_{i_2} - x_{i_1})} \cdot (x_{i_2}^3 - x_{i_1}^3) - x_{i_2} \cdot \frac{\hat{f}_{i_2} - \hat{f}_{i_1}}{2(x_{i_2} - x_{i_1})} \cdot (x_{i_2}^2 - x_{i_1}^2) + \frac{\hat{f}_{i_2}}{2} \cdot (x_{i_2}^2 - x_{i_1}^2).
\end{aligned}$$

Finally,

$$\begin{aligned}
\widehat{m}_2 &= \int_{x_{i_0}}^{x_{i_1}} x^2\hat{f}(x)dx + \int_{x_{i_1}}^{x_{i_2}} x^2\hat{f}(x)dx \\
&= \frac{\hat{f}_{i_1} - \hat{f}_{i_0}}{4(x_{i_1} - x_{i_0})} \cdot (x_{i_1}^4 - x_{i_0}^4) - x_{i_1} \cdot \frac{\hat{f}_{i_1} - \hat{f}_{i_0}}{3(x_{i_1} - x_{i_0})} \cdot (x_{i_1}^3 - x_{i_0}^3) + \frac{\hat{f}_{i_1})}{3} \cdot (x_{i_1}^3 - x_{i_0}^3) \\
&\quad + \frac{\hat{f}_{i_2} - \hat{f}_{i_1}}{4(x_{i_2} - x_{i_1})} \cdot (x_{i_2}^4 - x_{i_1}^4) - x_{i_2} \cdot \frac{\hat{f}_{i_2} - \hat{f}_{i_1}}{3(x_{i_2} - x_{i_1})} \cdot (x_{i_2}^3 - x_{i_1}^3) + \frac{\hat{f}_{i_2}}{3} \cdot (x_{i_2}^3 - x_{i_1}^3).
\end{aligned}$$

By simplifying and equating these respective quantities with the three corresponding moments of the given function $f(x)$, a system of equations is formed and can be expressed using the matrix notation as

$$A \cdot \begin{bmatrix} \hat{f}_{i_0} \\ \hat{f}_{i_1} \\ \hat{f}_{i_2} \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \\ m_2 \end{bmatrix}$$

where

$$A = \begin{bmatrix} \frac{x_{i_1}-x_{i_0}}{2} & \frac{x_{i_1}-x_{i_0}}{2} + \frac{x_{i_2}-x_{i_1}}{2} & \frac{x_{i_2}-x_{i_1}}{2} \\ \frac{x_{i_1}^3-3x_{i_1}x_{i_0}^2+2x_{i_0}^3}{6(x_{i_1}-x_{i_0})} & \frac{-x_{i_1}^3+3x_{i_1}x_{i_0}^2-2x_{i_0}^3}{6(x_{i_1}-x_{i_0})} + \frac{x_{i_1}^2-x_{i_0}^2}{2} + \frac{x_{i_2}^3-3x_{i_2}x_{i_1}^2+2x_{i_1}^3}{6(x_{i_2}-x_{i_1})} & \frac{-x_{i_2}^3+3x_{i_2}x_{i_1}^2-2x_{i_1}^3}{6(x_{i_2}-x_{i_1})} + \frac{x_{i_2}^2-x_{i_1}^2}{2} \\ \frac{x_{i_1}^4-4x_{i_1}x_{i_0}^3+3x_{i_0}^4}{12(x_{i_1}-x_{i_0})} & \frac{-x_{i_1}^4+4x_{i_1}x_{i_0}^3-3x_{i_0}^4}{12(x_{i_1}-x_{i_0})} + \frac{x_{i_1}^3-x_{i_0}^3}{3} + \frac{x_{i_2}^4-4x_{i_2}x_{i_1}^3+3x_{i_1}^4}{12(x_{i_2}-x_{i_1})} & \frac{-x_{i_2}^4+4x_{i_2}x_{i_1}^3-3x_{i_1}^4}{12(x_{i_2}-x_{i_1})} + \frac{x_{i_2}^2-x_{i_1}^2}{2} \end{bmatrix}.$$

Thus, the theorem follows. $\square$

4.3.3 Approximation using n Piecewise Linear Segments

The general case in which the number of piecewise linear segments contained in the approximation function is n can be derived in the same manner as the cases for a single line and two-line approximations. Rather than stating the result as a theorem, we shall directly proceed with the derivation of obtaining an $n-$order equivalent function for $f(x)$.

Let the k^{th} moment of an $n-$piecewise linear approximation function $\hat{f}(x)$ be defined as the sum of the moments of the individual segments

$$\widehat{m}_k = S_{k,1} + S_{k,2} + \cdots + S_{k,n}.$$

Then the k^{th} moment of the individual segments can be obtained as follows,

$$\begin{aligned}
S_{k,1} &= \int_{x_{i_0}}^{x_{i_1}} x^k \left[\frac{\hat{f}_{i_1}-\hat{f}_{i_0}}{x_{i_1}-x_{i_0}}\cdot(x-x_{i_1})+\hat{f}_{i_1}\right]dx \\
&= \int_{x_{i_0}}^{x_{i_1}} \left[\frac{\hat{f}_{i_1}-\hat{f}_{i_0}}{x_{i_1}-x_{i_0}}\cdot x^{k+1}-\frac{x_{i_1}(\hat{f}_{i_1}-\hat{f}_{i_0})}{x_{i_1}-x_{i_0}}\cdot x^k+\hat{f}_{i_1}\cdot x^k\right]dx \\
&= \left[\frac{\hat{f}_{i_1}-\hat{f}_{i_0}}{(k+2)(x_{i_1}-x_{i_0})}\cdot x^{k+2}-\frac{x_{i_1}(\hat{f}_{i_1}-\hat{f}_{i_0})}{(k+1)(x_{i_1}-x_{i_0})}\cdot x^{k+1}+\hat{f}_{i_1}\cdot\frac{x^{k+1}}{k+1}\right]_{x_{i_0}}^{x_{i_1}} \\
&= \frac{\hat{f}_{i_1}-\hat{f}_{i_0}}{(k+2)(x_{i_1}-x_{i_0})}\cdot(x_{i_1}^{k+2}-x_{i_0}^{k+2}) \\
&\quad -\frac{x_{i_1}[\hat{f}_{i_1}-\hat{f}_{i_0}]}{(k+1)(x_{i_1}-x_{i_0})}\cdot(x_{i_1}^{k+1}-x_{i_0}^{k+1})+\hat{f}_{i_1}\cdot\frac{x_{i_1}^{k+1}-x_{i_0}^{k+1}}{k+1} \\
&= \hat{f}_{i_0}\cdot\left[-\frac{x_{i_1}^{k+2}-x_{i_0}^{k+2}}{(k+2)(x_{i_1}-x_{i_0})}+\frac{x_{i_1}(x_{i_1}^{k+1}-x_{i_0}^{k+1})}{(k+1)(x_{i_1}-x_{i_0})}\right] \\
&\quad +\hat{f}_{i_1}\cdot\left[\frac{x_{i_1}^{k+2}-x_{i_0}^{k+2}}{(k+2)(x_{i_1}-x_{i_0})}-\frac{x_{i_1}(x_{i_1}^{k+1}-x_{i_0}^{k+1})}{(k+1)(x_{i_1}-x_{i_0})}+\frac{x_{i_1}^{k+1}-x_{i_0}^{k+1}}{k+1}\right] \\
&= \hat{f}_{i_0}\cdot\left[\frac{x_{i_1}^{k+2}-(k+2)x_{i_1}x_{i_0}^{k+1}+(k+1)x_{i_0}^{k+2}}{(k+1)(k+2)(x_{i_1}-x_{i_0})}\right] \\
&\quad +\hat{f}_{i_1}\cdot\left[\frac{-x_{i_1}^{k+2}+(k+2)x_{i_1}x_{i_0}^{k+1}-(k+1)x_{i_0}^{k+2}}{(k+1)(k+2)(x_{i_1}-x_{i_0})}+\frac{x_{i_1}^{k+1}-x_{i_0}^{k+1}}{k+1}\right] \\
S_{k,2} &= \int_{x_{i_1}}^{x_{i_2}} x^k \left[\frac{\hat{f}_{i_2}-\hat{f}_{i_1}}{x_{i_2}-x_{i_1}}\cdot(x-x_{i_2})+\hat{f}_{i_2}\right]dx \\
&= \hat{f}_{i_1}\cdot\left[\frac{x_{i_2}^{k+2}-(k+2)x_{i_2}x_{i_1}^{k+1}+(k+1)x_{i_1}^{k+2}}{(k+1)(k+2)(x_{i_2}-x_{i_1})}\right] \\
&\quad +\hat{f}_{i_2}\cdot\left[\frac{-x_{i_2}^{k+2}+(k+2)x_{i_2}x_{i_1}^{k+1}-(k+1)x_{i_1}^{k+2}}{(k+1)(k+2)(x_{i_2}-x_{i_1})}+\frac{x_{i_2}^{k+1}-x_{i_1}^{k+1}}{k+1}\right] \\
&\vdots \\
S_{k,n-1} &= \int_{x_{i_{n-2}}}^{x_{i_{n-1}}} x^k \left[\frac{\hat{f}_{i_{n-1}}-\hat{f}_{i_{n-2}}}{x_{i_{n-1}}-x_{i_{n-2}}}\cdot(x-x_{i_{n-1}})+\hat{f}_{i_{n-1}}\right]dx \\
&= \hat{f}_{i_{n-2}}\cdot\left[\frac{x_{i_{n-1}}^{k+2}-(k+2)x_{i_{n-1}}x_{i_{n-2}}^{k+1}+(k+1)x_{i_{n-2}}^{k+2}}{(k+1)(k+2)(x_{i_{n-1}}-x_{i_{n-2}})}\right] \\
&\quad +\hat{f}_{i_{n-1}}\cdot\left[\frac{-x_{i_{n-1}}^{k+2}+(k+2)x_{i_{n-1}}x_{i_{n-2}}^{k+1}-(k+1)x_{i_{n-2}}^{k+2}}{(k+1)(k+2)(x_{i_{n-1}}-x_{i_{n-2}})}+\frac{x_{i_{n-1}}^{k+1}-x_{i_{n-2}}^{k+1}}{k+1}\right] \\
S_{k,n} &= \int_{x_{i_{n-1}}}^{x_{i_n}} x^k \left[\frac{\hat{f}_{i_n}-\hat{f}_{i_{n-1}}}{x_{i_n}-x_{i_{n-1}}}\cdot(x-x_{i_n})+\hat{f}_{i_n}\right]dx \\
&= \hat{f}_{i_{n-1}}\cdot\left[\frac{x_{i_n}^{k+2}-(k+2)x_{i_n}x_{i_{n-1}}^{k+1}+(k+1)x_{i_{n-1}}^{k+2}}{(k+1)(k+2)(x_{i_n}-x_{i_{n-1}})}\right]
\end{aligned}$$

$$+ \hat{f}_{i_n} \cdot \left[\frac{-x_{i_n}^{k+2} + (k+2)x_{i_n}x_{i_{n-1}}^{k+1} - (k+1)x_{i_{n-1}}^{k+2}}{(k+1)(k+2)(x_{i_n} - x_{i_{n-1}})} + \frac{x_{i_n}^{k+1} - x_{i_{n-1}}^{k+1}}{k+1} \right]$$

Therefore, $\widehat{m}_k$ can be rewritten as

$$\begin{aligned}
\widehat{m}_k &= \hat{f}_{i_0} \cdot \left[\frac{x_{i_1}^{k+2} - (k+2)x_{i_1}x_{i_0}^{k+1} + (k+1)x_{i_0}^{k+2}}{(k+1)(k+2)(x_{i_1} - x_{i_0})} \right] \\
&\quad + \hat{f}_{i_1} \cdot \left[\frac{-x_{i_1}^{k+2} + (k+2)x_{i_1}x_{i_0}^{k+1} - (k+1)x_{i_0}^{k+2}}{(k+1)(k+2)(x_{i_1} - x_{i_0})} + \frac{x_{i_1}^{k+1} - x_{i_0}^{k+1}}{k+1} \right. \\
&\qquad \left. + \frac{x_{i_2}^{k+2} - (k+2)x_{i_2}x_{i_1}^{k+1} + (k+1)x_{i_1}^{k+2}}{(k+1)(k+2)(x_{i_2} - x_{i_1})} \right] \\
&\quad + \hat{f}_{i_2} \cdot \left[\frac{-x_{i_2}^{k+2} + (k+2)x_{i_2}x_{i_1}^{k+1} - (k+1)x_{i_1}^{k+2}}{(k+1)(k+2)(x_{i_2} - x_{i_1})} + \frac{x_{i_2}^{k+1} - x_{i_1}^{k+1}}{k+1} \right. \\
&\qquad \left. + \frac{x_{i_3}^{k+2} - (k+2)x_{i_3}x_{i_2}^{k+1} + (k+1)x_{i_2}^{k+2}}{(k+1)(k+2)(x_{i_3} - x_{i_2})} \right] \\
&\quad + \\
&\quad \vdots \\
&\quad + \hat{f}_{i_{n-1}} \cdot \left[\frac{-x_{i_{n-1}}^{k+2} + (k+2)x_{i_{n-1}}x_{i_{n-2}}^{k+1} - (k+1)x_{i_{n-2}}^{k+2}}{(k+1)(k+2)(x_{i_{n-1}} - x_{i_{n-2}})} + \frac{x_{i_{n-1}}^{k+1} - x_{i_{n-2}}^{k+1}}{k+1} \right. \\
&\qquad \left. + \frac{x_{i_n}^{k+2} - (k+2)x_{i_n}x_{i_{n-1}}^{k+1} + (k+1)x_{i_{n-1}}^{k+2}}{(k+1)(k+2)(x_{i_n} - x_{i_{n-1}})} \right] \\
&\quad + \hat{f}_{i_n} \cdot \left[\frac{-x_{i_n}^{k+2} + (k+2)x_{i_n}x_{i_{n-1}}^{k+1} - (k+1)x_{i_{n-1}}^{k+2}}{(k+1)(k+2)(x_{i_n} - x_{i_{n-1}})} + \frac{x_{i_n}^{k+1} - x_{i_{n-1}}^{k+1}}{k+1} \right]
\end{aligned}$$

Let

$$\begin{aligned}
C_{k,j} &= \frac{x_{j+1}^{k+2} - (k+2)x_{j+1}x_j^{k+1} + (k+1)x_j^{k+2}}{(k+1)(k+2)(x_{j+1} - x_j)} \\
D_{k,j} &= \frac{x_{j+1}^{k+1} - x_j^{k+1}}{k+1}
\end{aligned}$$

where $0 \le j \le (n-1)$ and $0 \le k \le n$, then

$$\begin{aligned}
\widehat{m}_k &= \hat{f}_{i_0} \cdot C_{k,0} \\
&\quad + \hat{f}_{i_1} \cdot (-C_{k,0} + D_{k,0} + C_{k,1}) \\
&\quad + \hat{f}_{i_2} \cdot (-C_{k,1} + D_{k,1} + C_{k,2}) \\
&\quad + \\
&\quad \vdots
\end{aligned}$$

$$+ \hat{f}_{i_{n-1}} \cdot (-C_{k,n-2} + D_{k,n-2} + C_{k,n-1})$$
$$+ \hat{f}_{i_n} \cdot (-C_{k,n-1} + D_{k,n-1})$$

If $\mathbf{E_j}$ is a column vector of n elements defined by

$$\mathbf{E_j} = \begin{cases} \begin{bmatrix} C_{0,j} \\ C_{1,j} \\ \vdots \\ C_{n,j} \end{bmatrix} & j = 0 \\ \begin{bmatrix} -C_{0,j-1} + D_{0,j-1} + C_{0,j} \\ -C_{1,j-1} + D_{1,j-1} + C_{1,j} \\ \vdots \\ -C_{n,j-1} + D_{n,j-1} + C_{n,j} \end{bmatrix} & 1 \le j \le n-1 \\ \begin{bmatrix} -C_{0,j-1} + D_{0,j-1} \\ -C_{1,j-1} + D_{1,j-1} \\ \vdots \\ -C_{n,j-1} + D_{n,j-1} \end{bmatrix} & j = n \end{cases}$$

then the system involving the n equations and n unknowns $f_{i_0}, f_{i_1}, \ldots, f_{i_n}$ can be summarized in a matrix notation as

$$\begin{bmatrix} \mathbf{E_0} & \mathbf{E_1} & \mathbf{E_2} & \cdots & \mathbf{E_{n-1}} & \mathbf{E_n} \end{bmatrix} \cdot \begin{bmatrix} \hat{f}_{i_0} \\ \hat{f}_{i_1} \\ \hat{f}_{i_2} \\ \vdots \\ \hat{f}_{i_{n-1}} \\ \hat{f}_{i_n} \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{n-1} \\ m_n \end{bmatrix}$$

4.3.4 Continuous versus Discrete Functions $f(x)$

All of the derivations described previously are based on the assumption that the given function $f(x)$ is continuous. However, in most practical applications, the function is never given in its closed form. In these situations, if $f(x)$ is described by a sequence of discrete points, the same formula from above can be applied to the original function $f(x)$. Let $f(x)$ be given as a sequence of $N+1$ points:

$$\{(x_i, f_i) | i = 0, 1, 2, \ldots, N\}.$$

Then the k^{th} moment of the original function $f(x)$ is given by:

$$m_k = f_0 \cdot C_{k,0}$$

$$
\begin{aligned}
&+ f_1 \cdot (-C_{k,0} + D_{k,0} + C_{k,1}) \\
&+ f_2 \cdot (-C_{k,1} + D_{k,1} + C_{k,2}) \\
&+ \\
&\vdots \\
&+ f_{N-1} \cdot (-C_{k,N-2} + D_{k,N-2} + C_{k,N-1}) \\
&+ f_N \cdot (-C_{k,N-1} + D_{k,N-1})
\end{aligned}
$$

where

$$
\begin{aligned}
C_{k,j} &= \frac{x_{j+1}^{k+2} - (k+2)x_{j+1}x_j^{k+1} + (k+1)x_j^{k+2}}{(k+1)(k+2)(x_{j+1} - x_j)} \\
D_{k,j} &= \frac{x_{j+1}^{k+1} - x_j^{k+1}}{k+1}
\end{aligned}
$$

for $0 \leq j \leq (N-1)$ and $0 \leq k \leq N$. Using this, the exact k^{th} moment of $f(x)$ and $\hat{f}(x)$ can be equated to yield the piecewise linear approximation.

4.4 Approximating Two-dimensional Planar Curves Using Moments

The moment-preserving approximation technique can be extended and applied to planar curve approximations. However, in order to approximate two-dimensional planar curves using the moment-preserving method described above, the planar curve must be normalized to be a one-dimensional function. For any arbitrary planar curve described by a discrete set of $N+1$ points, $S = \{p_i = (x_i, y_i) | 0 \leq i \leq N\}$, the $x-$ and $y-$coordinates of the centre of the curve, denoted by $p_c = (x_c, y_c)$, are defined as

$$
\begin{aligned}
x_c &= \frac{1}{N} \sum_{i=0}^{N} x_i \\
y_c &= \frac{1}{N} \sum_{i=0}^{N} y_i
\end{aligned}
$$

The normalizing function, $\varphi(\theta)$, of a planar curve is defined as the Euclidean distance of a point on the curve from its centre as a function of the normalized angle between 0 and 2π. Formally, it can be summarized as

$$
\varphi(\theta_i) = | < p_i, p_c > |
$$

where

$$\theta_i = \frac{2\pi i}{N} \qquad 0 \leq i \leq N.$$

The moment-preserving approximation technique is applied to the one-dimensional normalized function. The approximation of the normalized function is "reversed" to derive an approximation to the original planar curve.

There are a few reasons why this particular normalizing function is used instead of the one used by Zahn and Roskies. First of all, since the moments of the curve are of primary interest, the centre of mass (closely related to the zeroth moment) is the most logical position to be used as a reference point describing all of the other points on the curve. A circle, for example, when described by this normalizing function becomes a straight horizontal line. Secondly, by using the Euclidean distance, the normalized function is always positive. Finally, the normalizing function is invariant with respect to certain affine transformations.

4.5 Approximating Digitized Images Using Moments

A digitized image $f(x,y)$ is often considered as a two-dimensional matrix of size $N \times N$, whose elements are the quantized brightness values at that point. Thus, $f(x,y)$ is defined as

$$f(x_i, y_k) = \widehat{I}(x_0 + i\Delta x, y_0 + k\Delta y) \qquad \text{for } i, k = 0, 1, 2, \ldots, N-1$$

where $\widehat{I}$ is the quantized brightness of the analogue image I, and Δx and Δy are the sampling intervals along the $x-$ and $y-$axis, respectively.

An often used strategy of subdividing an image is to break it up into 8×8 (or 16×16) blocks. By subdividing the image into such smaller blocks, the correlation between adjacent pixels can be maintained and can be used to assist in the approximation of other pixels. A conversion technique called the zigzag scan [RJ 91] is used to convert a square block of values into a linear form. The moment-preserving approximation is applied to the linear form of the block of pixels of an image. Once an approximation of the zigzag pixel values is found, the zigzag operation is reversed to restore the square block structure of the approximated image.

5 Experimental Results and Discussions

The moment-preserving approximation method introduced here has been implemented and compared with other piecewise linear approximation methods including Ramer's linear search-and-split, Pavlidis split-and-merge, Prusinkiewicz's hologram-like subsampling, and Kurozumi's minimax error scheme. It has also been compared with two well-known transform domain approximation techniques: discrete Fourier transform, and discrete cosine transform. A more detailed comparison of it to other techniques, as well as when applied on other functions, planar curves, and images can be found in [Ng 94].

All the previous approximation techniques and the new moment-preserving method were fully implemented using O.T.I. Smalltalk[6], and all the time and error statistics were gathered from a 50MHz-486-PC. They have been applied to one-dimensional functions, two-dimensional planar curves, and digitized images. In the implementation of the moment-preserving technique, there is no maximum error limit requirement, ϵ, that is typically used by other time/spacial domain approximation techniques such as the linear split, recursive split, split-and-merge, and minimax methods. As opposed to them, for each moment-preserving approximation, the number of knot values must be specified. Since there is no restriction on the location of the knot values, they can be uniformly or non-uniformly selected by the the user. However, in order to make the current implementation easy to understand and maintain, the knot values are assumed to be uniformly distributed along the $x-$axis.

The results of the approximation techniques applied to functions $f_1(x) = \frac{20}{\sqrt{2\pi}}e^{-2x^2}$ defined in interval $[-6, 6]$ and $f_2(x) = 5\sin 4x + 4\cos 3x + 2\cos 200x$ defined in interval $[0, 10]$ appear in Figures 1 and 2, and Tables 1 and 2. The moment-preserving method requires only 5 points to approximate the normal bell curve with a SNR value of 14.7 dB which is better than Ramer's search-and-split, Kurozumi's minimax, and Pavlidis' split-and-merge techniques. The noisy sine/cosine waves can be approximated by the moment-preserving method using 25 points with a SNR value of -12.4134 dB which is comparable or better than any other transform or time/spacial domain technique.

Similarly, the results of the approximation techniques applied to an open elliptical curve, a noisy (closed) planar curve, and a digitized image are found in Figures 3, 4, and 5, and Tables 3, 4, and 5. The moment-preserving method approximates the elliptical curve using only 11 points with an SNR of 24.4992 dB and requiring only 177 milliseconds, whereas Ramer's search-and-split method using the same number of points has an SNR of 23.1234 dB and requiring 784 milliseconds. For the noisy planar curve, the moment-preserving method approximates it in 320 milliseconds

[6]Each approximation technique is a class in the Smalltalk hierarchy, and each is a subclass of the class named "ApproximationTechnique". The current implementation is extensible, and is available upon request.

with the ME of 1.649, which is comparable with the discrete cosine technique which requires 369 milliseconds with the ME of 4.598.

The moment-preserving method is better than all existing spacial domain techniques which use uniform knot values such as Prusinkiewicz's hologram-like approximation techniques. Although it requires more computational time than these other techniques, the amount of error between the approximation and the actual function is significantly smaller. This property is clearly demonstrated in the approximation examples of one-dimensional functions and planar curves.

Of the three approximation techniques which use non-uniform knots (linear search-and-split, split-and-merge, and minimax), only the linear search-and-split methods consistently perform better than the moment-preserving technique. This conclusion is consistent with an observation made by others [Ri 69] that the key to the successful use of splines is to have the location of the knots as variables. However, the moment-preserving method consistently yields superior approximation results compared to the minimax and the split-and-merge methods.

Only two transform domain approximation techniques are used in the comparison, namely the Fourier transform and discrete cosine transform (see Figures 1 (f) and (g) to 4 (f) and (g) and Figures 5 (e) and (f), and Tables 1 to 5). The Fourier coefficients contain too many spurious frequencies, and as such when a small number of truncated coefficients are used to approximate the original function, they introduce a lot of noise into the approximation. The moment-preserving method outperforms the Fourier transform on approximating one-dimensional and two-dimensional functions. Results from the planar curve and digitized image approximations show that the moment-preserving method performs as well as or better than the discrete cosine transform. However, for one-dimensional functions, the discrete cosine transform is more effective than the moment-preserving technique.

6 Conclusion

Approximation techniques are an important aspect of digital signal and image processing. Existing approximation techniques approach this problem in either a time/spacial domain or transform domain, but not both. In this paper we have presented a new strategy to obtain an approximation $\hat{f}(x)$ of $f(x)$ in such a way that it is *reasonably* close to the original function in the domain of the variable x, but it also *exactly* preserves some properties of the transformed domain. In this particular case, the properties of the transformed values that are preserved are geometric moments of the original function. The proposed technique has been applied to one-dimensional functions, two-dimensional planar curves, and two-dimensional images and a comparative survey has been experimentally justified.

Apart from describing our new strategy the paper also provides a reasonably comprehensive survey of the existing time/spacial and transform domain approximation strategies. Experimental evidence obtained from the results of using this technique indicate that it is a viable option, and that it could well be a powerful approximation strategy for the future.

Acknowlegments

We would like to thank the Natural Sciences and Engineering Research Council of Canada for its support. Also, without the generous resources from Object Technology International, Inc. and the encouragements provided by Dave and Margaret Thomas, and other staff members at O.T.I. we would not have been able to complete this study - and this paper.

References

[ANR 85] Ahmed, H., Natarajan, T., and Rao, K. R., *Discrete Cosine Transform*, Discrete Transforms and Their Applications, Van Nostrand Reinhold Company Inc., pp 9-12, 1985.

[BB 82] Ballard, D. H., and Brown, C. M., Computer Vision, Prentice-Hall, Inc., 1982.

[BR 74] Barnhill, R. E., and Riesenfeld, R. F., Computer Aided Geometric Design, Academic Press, New York, 1974.

[Be 61] Bellman, R., *On the approximation of curves by line segments using dynamic programming*, Comm. ACM, Vol. 4, No. 6, pg. 284, June 1961.

[Co 60] Cosgriff, R. L., *Identification of Shape*, Ohio State University Research Foundation, Columbus, Ohio, Report 820-11, ASTIA AD 254 792, December 1960.

[Da 63] Davis, P. J., Interpolation and Approximation, Blaisdell, pg.151, 1963.

[deB 78] de Boor, C., A Practical Guide to Splines, Springer-Verlag, New York, 1978.

[Du 86] Dunham, J. G., *Optimum Uniform Piecewise Linear Approximation of Planar Curves*, I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, pp. 67-75, 1986.

[Gl 62] Gluss, B., *A line segment curve-fitting algorithm related to optimal encoding of information*, Information and Control, Vol. 5, pp261-267, 1962.

[Gr 72] Granlund, G. H., *Fourier Preprocessing for Hand Print Character Recognition*, I.E.E.E. Transactions on Computers, Vol. C-21, pp. 195-201, 1972.

[Hi 87] Hildebrand, F. B., *Introduction to Numerical Analysis*, 2nd Edition, Dover Publications, 1987.

[KD 82] Kurozumi, Y., and Davis, W. A., *Polygonal Approximation by the Minimax Method*, Computer Graphics and Image Processing, Vol. 19, pp. 248-264, 1982.

[Lo 87] Lowe, D. G., *Three-dimensional Object Recognition from Single Two-Dimensional Images*, Artificial Intelligence, Vol. 31, pp. 355-395, 1987.

[Mo 70] Montanari, U., *A Note on Minimal Length Polygonal Approximation to a Digitized Contour*, Communications of the A.C.M., Vol. 13, No. 1, pp.41-47, January 1970.

[Ng 94] Nguyen, T. B., *Moment-Preserving Piecewise Linear Approximations of Signals and Images - Master's Thesis*, Carleton University, Ottawa, Ontario. 1994.

[Pap 65] Papoulis, A., Probability, Random Variables, and Stochastic Processes, McGraw-Hill Book, Inc., 1965.

[Pa 73] Pavlidis, T., *Waveform Segmentation Through Functional Approximation*, I.E.E.E. Transactions on Computers, Vol. C-22, No. 7, pp. 689-697, July 1973.

[PH 74] Pavlidis, T., and Horowitz, S. L. *Segmentation of Plane Curves*, I.E.E.E. Transactions on Computers, Vol. C-23, No. 8, pp. 860-870, August 1974.

[Ph 68] Phillips, G. M., *Algorithm for piecewise straight line approximations*, Computer Journal, Vol. 11, pp 211-212, 1968.

[PC 85] Prusinkiewicz, P., and Christopher, M., *Hologram-like transmission of pictures*, The Visual Computer, Springer Verlag, pp. 185-193, 1985.

[Ram 72] Ramer, U., *An Iterative Procedure for the Polygonal Approximation of Plane Curves*, Computer Graphics and Image Processing, Vol. 1, pp. 244-256, 1972.

[RJ 91] Rabbani, M., and Jones, P. W., Digital Image Compression Techniques, SPIE - The International Society for Optical Engineering, Washington, 1991.

[Ri 69] Rice, J. R., The Approximation of Functions, Vol. 1, 2, Reading, Mass., Addison-Wesley, 1964, 1969.

[RW 89] Rosin, P. L., and West, G. A. W., *Segmentation of Edges into Lines and Arcs*, Image and Vision Computing, Vol. 7, No. 2, pp. 109-114, 1989.

[RW 91] Rosin, P. L., and West, G. A. W., *Techniques for Segmenting Image Curves into Meaningful Descriptions*, Pattern Recognition, Vol. 24, No. 7, pp. 643-652, 1991.

[Sp 74] Späth, H., Spline Algorithms for Curves and Surfaces, Translated from German by Hoskins, W. D., and Sager, H. W., Utilitas Mathematica Publishing Incorporated, Winnipeg, 1974.

[St 61] Stone, H., *Approximation of curves by line segments*, Mathematics of Computation, Vol. 15, pp 40-47, 1961.

[Ta 79] Tanimoto, S. L., *Image Transmission with Gross Information First*, Computer Graphics and Image Processing, Vol. 9, pp. 72-76, 1979.

[ZR 72] Zahn, C. T., Roskies, R. Z., *A Fourier Description for Plane Close Curves*, I.E.E.E. Transactions on Computers, Vol. C-21, pp. 269-281, 1972.

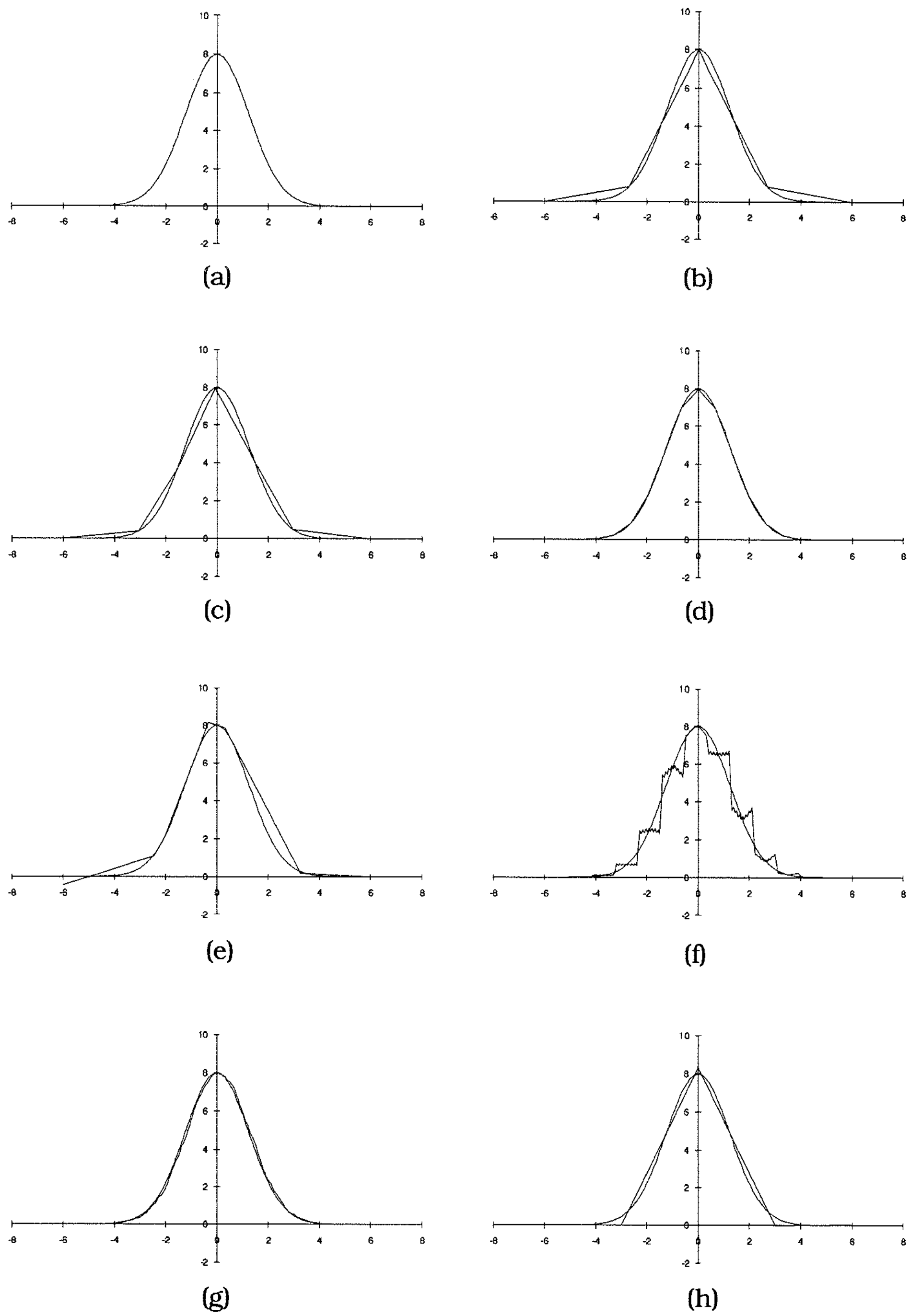

Figure 1: (a) Original normal bell curve. The results obtained using the following approximation methods: (b) Ramer's search-and-split, (c) Pavlidis' split-and-merge, (d) Prusinkiewicz's hologram-like subsampling, (e) Kurozumi's minimax, (f) Discrete Fourier transform. (g) Discrete cosine transform, (h) Moment-preserving using 7 points.

Approximation Method	*# of Approx. Points*	*Time Required*	*ME (E_∞)*	*MSE (E_2/N)*	*SNR (dB)*
Linear Split (Ramer)	5	129	0.73505	0.1346	11.9091
Split/Merge (Pavlidis)	12	692	0.8403	0.133	11.9599
Hologram-like (Prusinkiewicz)	18	55	0.2442	0.0055	25.7988
Minimax (Kurozumi)	7	5901	1.4142	0.2721	8.8518
DFT	N/A	243	1.6673	0.2059	10.0619
DCT	N/A	198	0.1275	0.002	30.2411
Moment-Preserving	5	247	0.5522	0.0697	14.7607

Table 1: Comparison table of various approximation techniques on normal bell curve. The **Time Required** is in milliseconds.

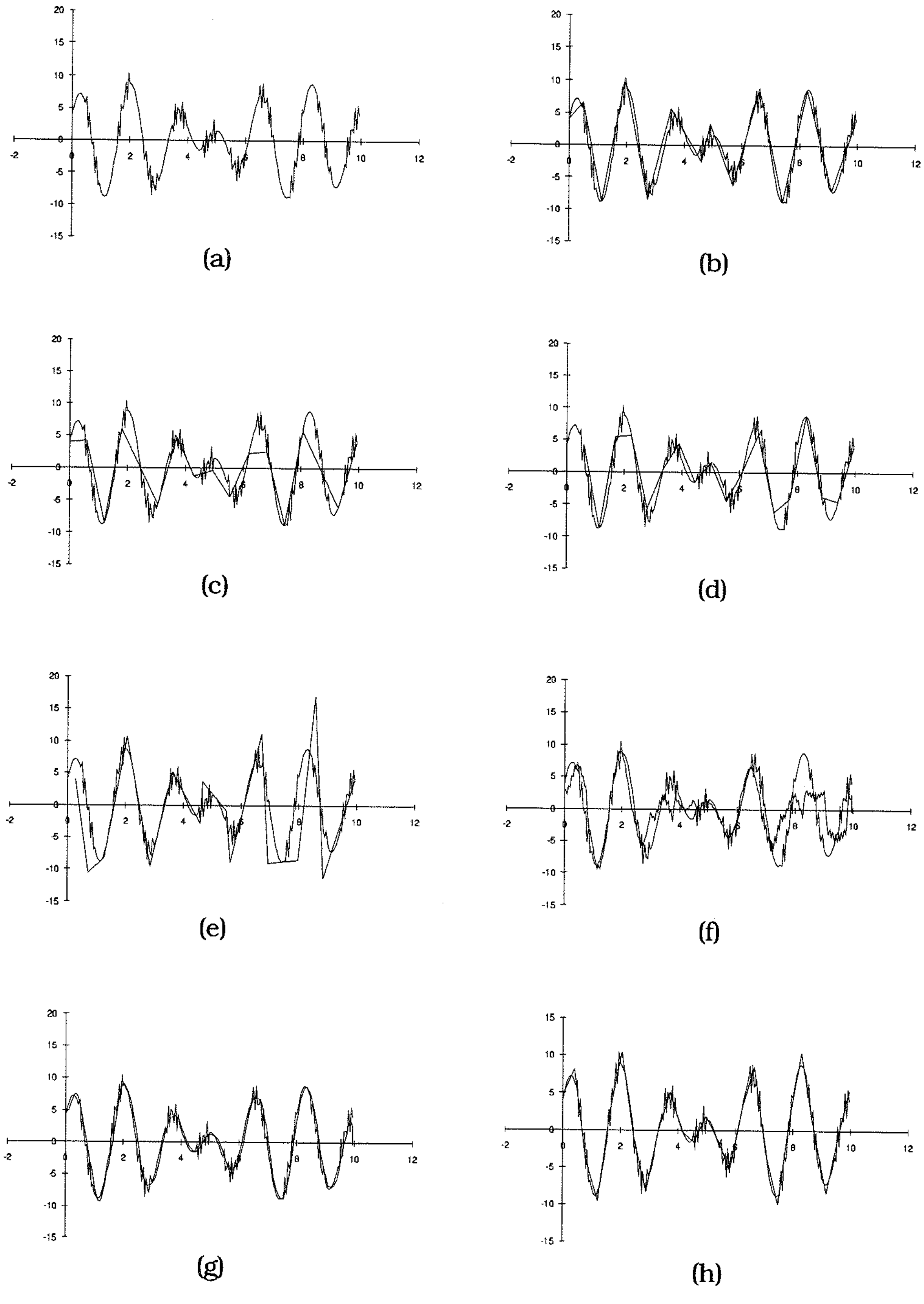

Figure 2: (a) Original function of a simulated noisy sine/cosine waves. The results obtained using the following approximation methods: (b) Ramer's search-and-split, (c) Pavlidis' split-and-merge, (d) Prusinkiewicz's hologram-like subsampling, (e) Kurozumi's minimax, (f) Discrete Fourier transform, (g) Discrete cosine transform, (h) Moment-preserving using 25 points.

Approximation Method	*# of Approx. Points*	*Time Required*	*ME (E_∞)*	*MSE (E_2/N)*	*SNR (dB)*
Linear Split (Ramer)	14	252	4.6696	2.7531	-15.525
Split/Merge (Pavlidis)	28	873	6.3583	6.218	-19.0632
Hologram-like (Prusinkiewicz)	18	45	4.8693	3.9821	-17.1278
Minimax (Kurozumi)	16	1491	42.5632	115.6809	-31.7593
DFT	N/A	237	9.5968	10.5518	-21.36
DCT	N/A	195	2.8431	1.0729	-11.4325
Moment-Preserving	25	330	2.8278	1.3448	-12.4134

Table 2: Comparison table of various approximation techniques on simulated noisy sine/cosine waves. The **Time Required** is in milliseconds.

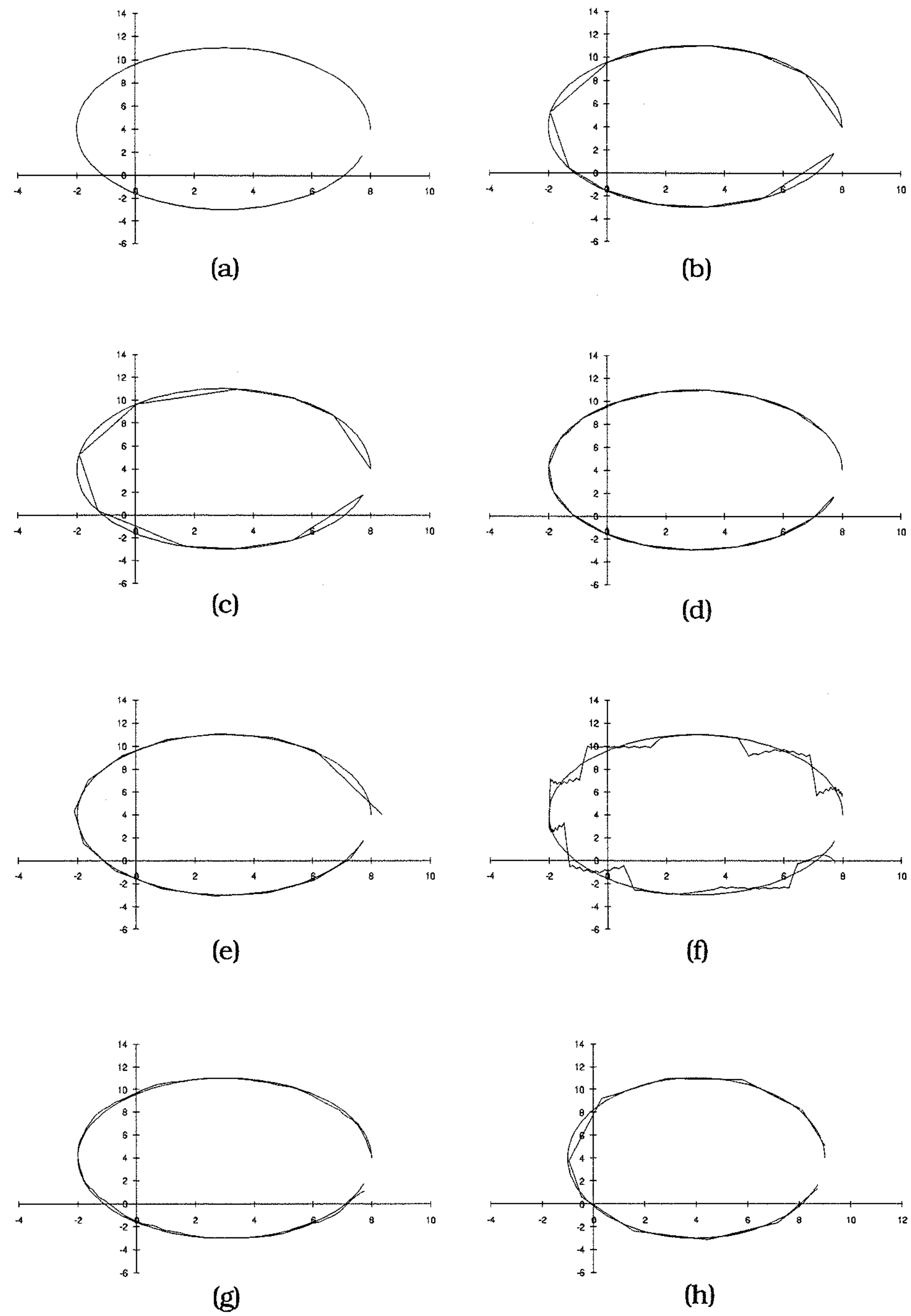

Figure 3: (a) Original function of an open elliptical curve. The results obtained using the following approximation methods: (b) Ramer's search-and-split, (c) Pavlidis' split-and-merge, (d) Prusinkiewicz's hologram-like subsampling, (e) Kurozumi's minimax, (f) Discrete Fourier transform, (g) Discrete cosine transform, (h) Moment-preserving using 11 points.

Approximation Method	*# of Approx. Points*	*Time Required*	*ME (E_∞)*	*MSE (E_2/N)*	*SNR (dB)*
Linear Split (Ramer)	11	784	0.4035	0.0197	23.1234
Split/Merge (Pavlidis)	30	1004	4.9756	3.9396	-17.1851
Hologram-like (Prusinkiewicz)	18	72	0.0888	1.47E-03	34.4056
Minimax (Kurozumi)	16	2079	0.6281	0.1146	15.4858
DFT	N/A	210	2.1044	0.5862	8.397
DCT	N/A	173	0.6462	0.0103	25.9476
Moment-Preserving	11	177	0.303	0.0144	24.4992

Table 3: Comparison table of various approximation techniques on open elliptical curve. The **Time Required** is in milliseconds.

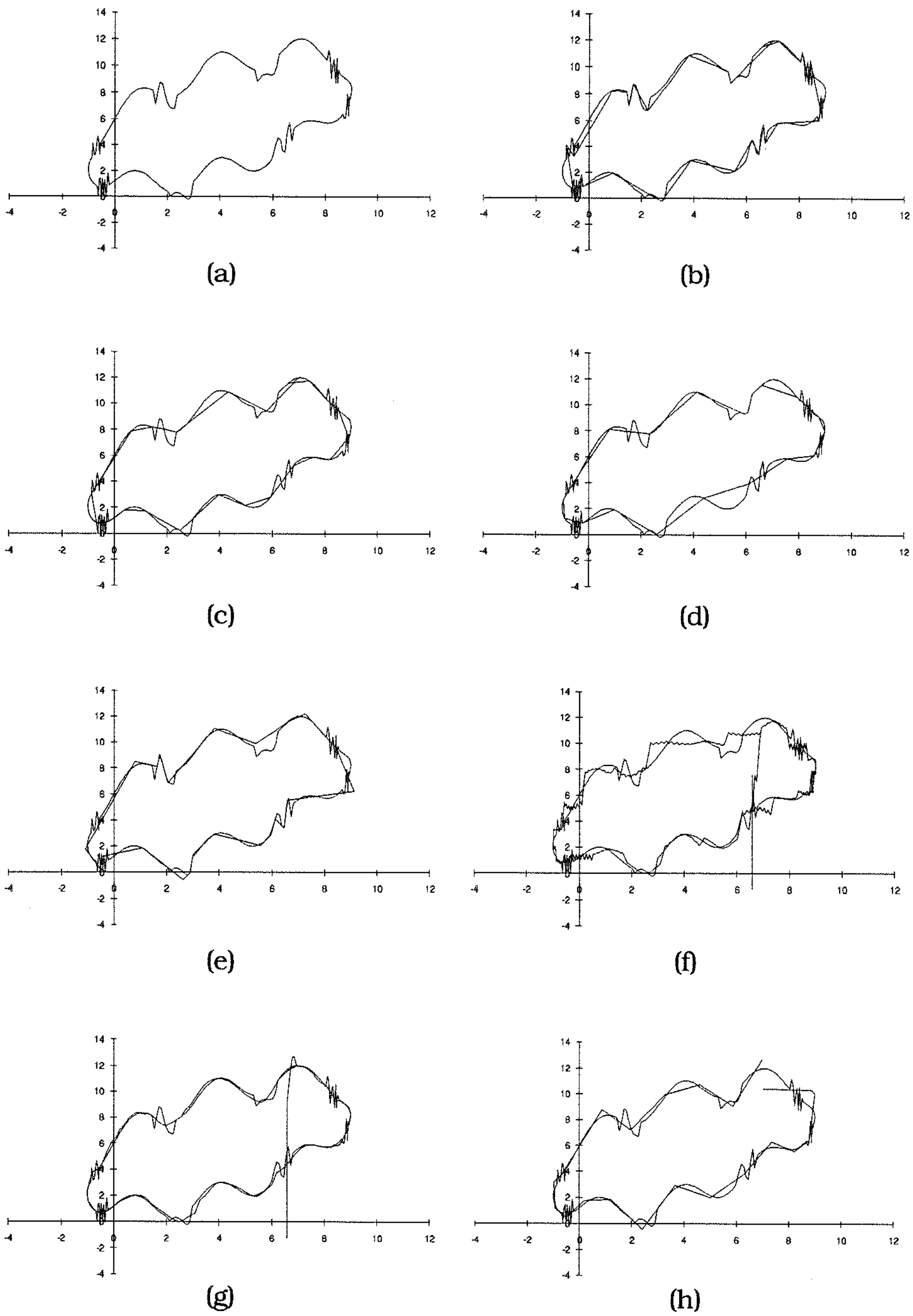

Figure 4: (a) Original function of a closed complex curve. The results obtained using the following approximation methods: (b) Ramer's search-and-split, (c) Pavlidis' split-and-merge, (d) Prusinkiewicz's hologram-like subsampling, (e) Kurozumi's minimax, (f) Discrete Fourier transform, (g) Discrete cosine transform, (h) Moment-preserving using 11 points.

Approximation Method	*# of Approx. Points*	*Time Required*	*ME (E_∞)*	*MSE (E_2/N)*	*SNR (dB)*
Linear Split (Ramer)	32	558	0.3718	0.0151	25.8644
Split/Merge (Pavlidis)	24	2252	1.0721	0.0578	20.0493
Hologram-like (Prusinkiewicz)	18	140	1.2565	0.1183	16.9391
Minimax (Kurozumi)	24	3479	0.9114	0.0676	19.3705
DFT	N/A	471	7.1492	0.7526	8.9026
DCT	N/A	369	4.598	0.1729	15.2897
Moment-Preserving	23	320	1.649	0.1833	14.981

Table 4: Comparison table of various approximation techniques on closed complex curve. The **Time Required** is in milliseconds.

Approximation Method	*Compression Ratio*	*Time Required*	*ME (E_∞)*	*MSE (E_2/N)*	*SNR (dB)*
Hologram-like	16:1	118 sec.	198	545.0332	-6.5354
Nearest-neighbour	16:1	242 sec.	128	262.3665	-3.3603
DFT	8:1	710 sec.	255	21665.0011	-22.5288
DCT	16:1	660 sec.	69	220.1519	-2.5984
Moment-preserving	16:1	943 sec.	154	289.9788	-3.7949

Table 5: Comparison table of various approximation techniques on digitized image.

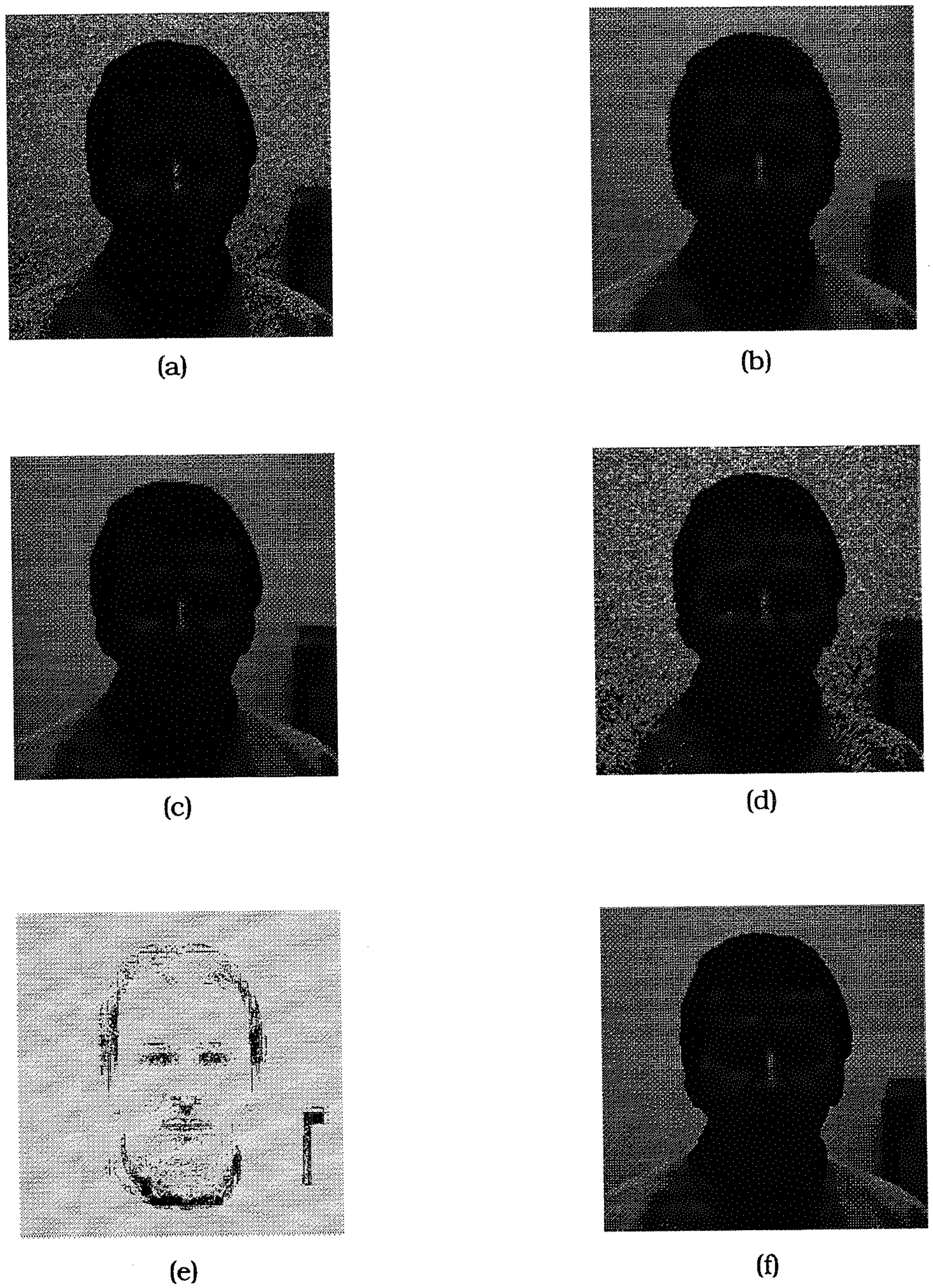

Figure 5: (a) Original digitized image of a human face (512×512) with 256 greyscale values. The results obtained using the following approximation methods: (b) Moment-preserving using 32×32 points, (c) Prusinkiewicz's hologram-like subsampling, (d) Tanimoto's nearest-neighbour, (e) Discrete Fourier transform, (f) Discrete cosine transform.

School of Computer Science, Carleton University
Recent Technical Reports

TR-239 **Killing Two Birds with One Stone**
Evangelos Kranakis, Danny Krizanc, Anil Maheshwari, Jörg-Rüdiger Sack, Jorge Urrutia, April 1994

TR-240 **Some Computational Problems on Central Simple Algebras over Q**
Vincenzo Acciaro, April 1994 (Not available)

TR-241 **Extending Cryptographic Logics of Belief to Key Agreement Protocols**
Paul C. van Oorschot, May 1994

TR-242 **Modern Key Agreement Techniques**
Rainer A. Rueppel and Paul C. van Oorschot, May 1994

TR-243 **On Unifying Some Cryptographic Protocol Logics**
Paul F. Syverson and Paul C. van Oorschot, May 1994

TR-244 **Efficient DES Key Search**
Michael J. Wiener, May 1994

TR-245 **Optimal Fault-Tolerant Leader Election in Chordal Rings**
Bernard Mans and Nicola Santoro, May 1994

TR-246 **Information Privacy in Canada (Legislation in the Face of Changing Technologies)**
Václav Matyás, Jr., June 1994

TR-247 **Randomized Encryption with Insertions and Deletions Based on Probability Distributions**
B. John Oommen and P.C. van Oorschot, June 1994

TR-248 **On the Number of Directions in Visibility Representations of Graphs**
Evangelos Kranakis, Danny Krizanc and Jorge Urrutia, July 1994

TR-249 **Parallel Collision Search with Application to Hash Functions and Discrete Logarithms**
Paul C. van Oorschot and Michael J. Wiener, July 1994

TR-250 **Graph Partitioning Using Learning Automata**
B. John Oommen and Edward V. de St. Croix, July 1994

TR-251 **Cellular Automata in Fuzzy Backgrounds**
G. Cattaneo, P. Flocchini, G. Mauri, N. Santoro, C. Quaranta Vogliotti, August 1994

TR-252 **CATS 94**
September 1994

TR-253 **Bubbles: Adaptive Routing Scheme for High-Speed Dynamic Networks**
Shlomi Dolev, Evangelos Kranakis and Danny Krizanc, September 1994

TR-254 **Object-Oriented Methodologies: A Matter of Perception**
John Pugh and Amir Zeid, October 1994

TR-255 **VC-Dimensions for Graphs**
Evangelos Kranakis, Danny Krizanc, Berthold Ruf, Jorge Urrutia, Gerhard J. Woeginger, October 1994

TR-256 **String Recognition on Anonymous Rings**
Evangelos Kranakis, Danny Krizanc and Flaminia L. Luccio, November 1994

TR-257 **Isomorphic Triangulations with Minimal Number of Steiner Points**
Evangelos Kranakis and Jorge Urrutia, November 1994

TR-258 **Monument-Preserving Piecewise Linear Approximations of Signals and Images**
Thai B. Nguyen and John Oommen, December 1994

School of Computer Science, Carleton University
Recent Technical Reports

TR-238 **Finding the Extrema of a Distributed Multiset**
Paola Alimonti, Paola Flocchini and Nicola Santoro, March 1994

TR-239 **Killing Two Birds with One Stone**
Evangelos Kranakis, Danny Krizanc, Anil Maheshwari, Jörg-Rüdiger Sack, Jorge Urrutia, April 1994

TR-240 **Some Computational Problems on Central Simple Algebras over Q**
Vincenzo Acciaro, April 1994 (Not available)

TR-241 **Extending Cryptographic Logics of Belief to Key Agreement Protocols**
Paul C. van Oorschot, May 1994

TR-242 **Modern Key Agreement Techniques**
Rainer A. Rueppel and Paul C. van Oorschot, May 1994

TR-243 **On Unifying Some Cryptographic Protocol Logics**
Paul F. Syverson and Paul C. van Oorschot, May 1994

TR-244 **Efficient DES Key Search**
Michael J. Wiener, May 1994

TR-245 **Optimal Fault-Tolerant Leader Election in Chordal Rings**
Bernard Mans and Nicola Santoro, May 1994

TR-246 **Information Privacy in Canada (Legislation in the Face of Changing Technologies)**
Václav Matyás, Jr., June 1994

TR-247 **Randomized Encryption with Insertions and Deletions Based on Probability Distributions**
B. John Oommen and P.C. van Oorschot, June 1994

TR-248 **On the Number of Directions in Visibility Representations of Graphs**
Evangelos Kranakis, Danny Krizanc and Jorge Urrutia, July 1994

TR-249 **Parallel Collision Search with Application to Hash Functions and Discrete Logarithms**
Paul C. van Oorschot and Michael J. Wiener, July 1994

TR-250 **Graph Partitioning Using Learning Automata**
B. John Oommen and Edward V. de St. Croix, July 1994

TR-251 **Cellular Automata in Fuzzy Backgrounds**
G. Cattaneo, P. Flocchini, G. Mauri, N. Santoro, C. Quaranta Vogliotti, August 1994

TR-252 **CATS 94**
September 1994

TR-253 **Bubbles: Adaptive Routing Scheme for High-Speed Dynamic Networks**
Shlomi Dolev, Evangelos Kranakis and Danny Krizanc, September 1994

TR-254 **Object-Oriented Methodologies: A Matter of Perception**
John Pugh and Amir Zeid, October 1994

TR-255 **VC-Dimensions for Graphs**
Evangelos Kranakis, Danny Krizanc, Berthold Ruf, Jorge Urrutia, Gerhard J. Woeginger, October 1994

TR-256 **String Recognition on Anonymous Rings**
Evangelos Kranakis, Danny Krizanc and Flaminia L. Luccio, November 1994

TR-257 **Isomorphic Triangulations with Minimal Number of Steiner Points**
Evangelos Kranakis and Jorge Urrutia, November 1994