01 Mar 1993

An Intelligent Approach To Discrete Sampling Of Parametric Curves

Chaman Sabharwal
Missouri University of Science and Technology, chaman@mst.edu

AN INTELLIGENT APPROACH TO DISCRETE SAMPLING OF PARAMETRIC CURVES

CHAMAN L. SABHARWAL

UMR Engineering Center, University of Missouri-Rolla

Abstract

In graphics and animation applications, two of the problems are: (1) representation of an analytic curve by a discrete set of sampled points and (2) determining the similarity between two parametric curves. It is necessary to measure the accuracy of approximation and to have a metric to calculate the disparity between two parametric curves. Both of these problems have been associated with the reparameterization of the curves with respect to arc length. One of the methods uses Gaussian Quadrature to determine the arc length parameterization [Guenter and Parent 1990], while another interesting technique is a simple approximation method [Fritsch and Nielson 1990]. There are various ways to compute the similarity between two curves. For 2D Cartesian curves, max norm yields a satisfactory distance metric. For parametric curves, Euclidean norm is frequently used. Arc length is reasonable parameterization, but explicit arc length parameterization is not easy to compute for arbitrary parametric curves. We give a new technique for discretizing parametric curves. These sampled points can be used to approximate curves, determine arc length parameterization , and similarity between them. This technique is accurate, robust and simpler to implement. Comparisons of the previous methods with the new one are presented.

Introduction

In graphics and animation applications, analytic curves are approximated by discrete sets of points. It is necessary to measure the accuracy of approximation. Also it is desirable to have a metric to calculate the disparity between two parametric curves. Both of these problems involve the reparameterization of the curves with respect to arc length. The numerical arc length parameterization methods are either *ad hoc* or based on Gaussian Quadrature.

When an analytic curve is approximated by a discrete sampled curve, it is desirable to have a metric to measure the accuracy of approximation. There are several ways compute the distance between two curves. For 2D Cartesian curves, max norm yields a satisfactory distance metric. For parametric curves, Euclidean norm is frequently used. The Euclidean norm of difference curve between two different parameterizations of the same curve may yield non-zero Euclidean norm that is not acceptable. For two Cartesian curves $y = f(x)$ and $y = g(x)$, $a \le x \le b$, in the xy-plane, the distance between two curves is defined by the metric

$$D_1(f,g) = \max_{a\le x\le b} |f(x) - g(x)| \approx \max_{0\le i\le n} |f(x_i) - g(x_i)|$$

For two parametric curves $F(t) = (f_1(t), f_2(t))$, $G(t) = (g_1(t), g_2(t))$, $a \le t \le b$, the distance between two curves may be defined by

$$D_2(F,G) = \max_{a\le t\le b} \|F(t) - G(t)\| \approx \max_{0\le i\le n} \|F(t_i) - G(t_i)\|$$

Since parameterization is not unique, a curve can always be represented with different parameterizations. For example, a semi-circle may be parameterized by $(\frac{2t}{1+t^2}, \frac{1-t^2}{1+t^2})$ or $(\cos\frac{\pi(1-t)}{2}, \sin\frac{\pi(1-t)}{2})$ on the interval $[-1, 1]$. There are various ways to reparameterize a curve, e.g., ν-splines [Nielson 1974], β-splines [Barsky 1981], γ-splines [Boehm 1985], and Wilson-Fowler splines in [Fritsh 1986]. Since parameterization is not unique, the application of straightforward Euclidean metric $D_2(F,G)$ is not satisfactory. It is desirable to have a satisfactory parameterization before the distance metric can be applied. An arc length parameterization is reasonable, but not foolproof. The arc length parameterization is not easy to compute for arbitrary free form parametric curves. If the curves are parameterized with respect to arc length, $s \in [0, \ell]$, the distance metric

$$D_3(F,G) = \max_{0\le s\le \ell} \|F(s) - G(s)\|$$

is relatively but not quite satisfactory. The arc length parameterization involves integration with respect to the curve defining parameter. In general, there is no closed form solution to the calculation of $s = s(t)$ is a non-trivial proposition for an arbitrary curve. A very simple reparameterization to arc length parameterization is given by an example of semi-circle.

The semi-circle

$$R(t) = (\cos\frac{\pi(1-t)}{2}, \sin\frac{\pi(1-t)}{2}) \qquad -1 \le t \le 1$$

can be reparameterized with arc length parameter, s, as

$$R(s) = (-\cos(s), \sin(s)) \qquad 0 \le s \le \pi$$

The problem of determining a metric to measure the disparity between two curves is related to the parameterization of the curves. This leads to a problem of determining a satisfactory reparameterization of the parametric curves. The arc length reparameterization provides a reasonable answer.

ACM-SAC '93/2/93/IN, USA
© 1993 ACM 0-89791-568-2/93/0002/0397...$1.50

Previous Related Work

Here we discuss two of the schemes used to approximated a parametric curve. One of the techniques for calculating the arc length function is based on Gaussian Quadrature [Guenter and Parent 1990]. This is an adaptive technique for integrating a parametric function. The numerical integral of the magnitude of the derivative function, $\| \frac{d\mathbf{R}(t)}{dt} \|$, is calculated by recursively subdividing a parametric interval. The value of numerical Quadrature on the interval and its subdivision are compared. If the difference is acceptable, then the process terminates, otherwise the recursive process continues on each of the subintervals. The technique of integration by recursive and adaptive subdivision is called Gaussian Quadrature. The end points of the subdivision intervals constitute the partition of the original parameter interval. The arc length function is interpolated from this partition of the parameter space: $\{t_i\}$. If $\{s_i\}$ is corresponding sequence of arc lengths, then the arc length function s(t) at the parameter value t is interpolated as: s_{i-1} + non-adaptive quadrature on the interval $[t_{i-1}, t]$ where $t_{i-1} \le t \le t_i$. The optimal number of partition points depends on the criteria for acceptable error.

Here the parameter space and arc length space are both partitioned non-uniformly. The inverse process of determining t from given arc length s is based on Newton-Raphson iteration technique. Usually, this technique may fail for extremely rare pathological cases, but it works here because of the judicious partitioning of the parameter space. So in practice it is complex and robust.

The second numerical technique uses arc-length parameterization. Arc-length parameterization refers to a parameterization where a unit change in parameterizing variable results in unit change in the curve length. The uniform arc length partition of $[0, \ell]$ is used to determine a non-uniform partition of the parameter space [a, b]. This helps selecting curve points which are equidistant, and determining a partition $a = T_0 < T_1 < ... < T_m = b$, of parameter space [a, b] with a finite number of breakpoints. This parameter space partition is used to generate the arc length parameterization: for j =1,2, ..., m;

$$s(t) = S_{j-1} + (S_j - S_{j-1}) \frac{t - T_{j-1}}{T_j - T_{j-1}} \qquad T_{j-1} \le t < T_j$$

where $S_j = j \frac{\ell}{m}$, j = 0,1, ...,m; ℓ is the length of the curve to be calculated later.

One interesting numerical technique for arc length parameterization is presented in Fritsch and Nielson [1990]. Let $\mathbf{R}(t)$, $t \in [a, b]$, be a curve. Let $\{t_i = a + i \frac{(b-a)}{n}$: i = 0,1, ...,n$\}$ be a uniform partition of [a, b]. The approximate arc length partition is defined as $s_0 < s_1 < ... < s_n$ where $s_0 = 0$, $s_i = s_{i-1} + \|\mathbf{R}(t_i) - \mathbf{R}(t_{i-1})\|$, $\ell = s_n$. Then, arc length parameter s is a function of t with s(a) = 0, s(b) = ℓ, and is linearly approximated as

$$s(t) = s_{i-1} + (s_i - s_{i-1}) \frac{t - t_{i-1}}{t_i - t_{i-1}} \quad t_{i-1} \le t < t_i$$

for i =1, 2, ..., n.

With this arc length function, the values s_i are not necessarily equally spaced. Now the inverse function of this approximate s(t), is defined as τ such that $\tau s(t) = t$ where $s(t) = s_i$ if and only $t = t_i$. If $\tau(0) = a$, $\tau(\ell) = b$, and

$$\tau(s) = t_{i-1} + (t_i - t_{i-1}) \frac{s - s_{i-1}}{s_i - s_{i-1}} \quad s_{i-1} \le s < s_i,$$

for i = 1, 2, ..., n

then $\tau(s)$ defines a unique, inverse, parameter function of arc length s.

Let $\{S_j : 0 \le j \le m\}$ be a uniform partition of $[0, \ell]$ defined by $S_j = j \frac{\ell}{m}$, $0 \le j \le m$. Let the function $\tau(s)$ define $\{T_j : 0 \le j \le m\}$, the new partition of t parameter space,

$$T_j = \tau(S_j) = t_{i-1} + (t_i - t_{i-1}) \frac{S_j - s_{i-1}}{s_i - s_{i-1}}$$

for $s_{i-1} \le S_j < s_i$; i = 1, 2, ..., n; j = 0, 1, ..., m.

The sequence $\{T_j : 0 \le j \le m\}$ is a non-uniform interpolated partition of [a, b]. The distance metric based on the partition $\{T_j : 0 \le j \le m\}$ is more reliable than the metric based on the partition $\{t_i : 0 \le i \le n\}$. This is an *ad hoc* procedure, in fact, an optimal value of n and m will depend on the form of the curve.

Using these m+n+2 values of t for partition of parameter space, we get an arc length parameterization

$$s(t) = S_{j-1} \frac{T_j - t}{T_j - T_{j-1}} + S_j \frac{t - T_{j-1}}{T_j - T_{j-1}}$$

$$T_{j-1} \le t < T_j \quad \text{for } j = 1, 2, ..., m$$

and

$$t(s) = T_{j-1} \frac{S_j - s}{S_j - S_{j-1}} + T_j \left(1 - \frac{S_j - s}{S_j - S_{j-1}}\right)$$

$$S_{j-1} \le s < S_j \quad \text{for } j = 1, 2, ..., m.$$

This is a technique for partitioning the parameter space so that the parameter values correspond to the uniform partitioning of length parameter space. The contribution of this interpolated partition $\{T_j : 0 \le j \le m\}$ to the accuracy of distance metric can be improved by replacing linear interpolation with non-linear interpolation. Such non-linear interpolant may also be defined by the formula

$$T_j = \tau(S_j) = t_{i-1} + (t_i - t_{i-1}) \varphi\left(\frac{S_j - s_{i-1}}{s_i - s_{i-1}}\right)$$

for $s_{i-1} \le S_j < s_i$; i = 1, 2, ..., n; j = 0, 1, ..., m.

where φ may be a non-linear function satisfying the conditions:

$$\varphi(0) = 0, \varphi(1) = 1$$

such as $\varphi(s) = s^k$ for k = 1,2,3

or $\varphi(s) = s^2(3-2s)$

There are two advantages of this method. First, it yields a satisfactory arc length parameterization. It yields a method for measuring the accuracy of approximation. Secondly, it yields a metric for measuring the disparity between two parametric curves. However, this non-linear function calculation is more expensive than the linear approximation. Finally, there exist examples [Fritsch and Nielson 1990] where the curve using arc-wise equally spaced points give incorrect results. This happens in the case of perturbed quarter circle.

New Approach

Various questions arise immediately. How is arc length parameterization determined? How are m & n determined and what are the optimal values of m & n? Optimal values of m & n improve the accuracy of approximation. The above procedure is an *ad hoc* method based on experimentation. It yields different metric values for different choices of m and n. Similar questions arise about the adaptive Gaussian Quadrature technique. None of these techniques take into account the curvature of the curve.

A better approach is to base the partition on the curvature of the curve. The purpose of this paper is to determine the parameter space partitioning which can be used to determine (1) approximate arc length function, and (2) the accurate distance between two parametric curves. We present a new approach to this problem which uses only linear approximation and improves the accuracy of previous arc length functions. The new technique yields better numerical results. We present experimental results comparing the previous methods and the new technique.

The amount of partition can be controlled by the curvature of the curve and pre specified upper bound on the number of partition points. This algorithm is also suitable for parallelization. This algorithm partitions the curve into chunks of decreasing-curvature size. This method is based on recursive and adaptive subdivision of the parameter space.

Main idea is to determine values t_i, $a \le t_i \le b$, for the given curve

$$\mathbf{R}(t) = (x(t), y(t), z(t)), \quad a \le t \le b,$$

such that they are distributed in such a manner that the linear approximation is as close to the curve as desired. An analytic technique for curvature, which requires C^2 continuity to compute upper bound on the overall curvature, is found in Emery [1986]. The new technique does not impose any continuity constraints. The new formulation is based on approximate curvature of the curve, not the Gaussian curvature. That is, the approximate curvature of the curve between t_i and t_{i+1} is the same as that between t_j and t_{j+1}, $i \neq j$. It will be shown that the new partition technique is a better than the previous two methods.

The key to the new technique is to use adaptive subdivision, and use position values only, not the derivative data. One starts with parametrically uniformly distributed n+1 points, $t_i = a + i\,\frac{(b-a)}{n}$, i = 0, 1, ..., n. Usually, n = 4 is sufficient for all practical purposes. In all the examples, n = 4 is used. The next step is to calculate the maximum of the angles between the vectors $\mathbf{R}(t_i) - \mathbf{R}(t_{i-1})$, $\mathbf{R}(t_{j+1}) - \mathbf{R}(t_j)$ with double nested loop for j = i, ..., n-1, for i = 1, ..., n. If the calculated maximum angle is greater than the tolerance angle, the number of subdivision points is less than a pre specified upper bound, and the length of the vector $\mathbf{R}(t_{start}) - \mathbf{R}(t_{end})$ on a subinterval is greater than the pre specified length-tolerance, the parameter interval is subdivided into two equal subintervals. The subdivision point, also called break point, is retained for future use. In fact, for implementation consideration, there is no need to calculate the angles. The dot products between the unit vectors along $\mathbf{R}(t_i) - \mathbf{R}(t_{i-1})$ and $\mathbf{R}(t_{j+1}) - \mathbf{R}(t_j)$ are sufficient for computation purposes. The above procedure continues on each of the subintervals. At any stage, if the length of the vector $\mathbf{R}(t_{start}) - \mathbf{R}(t_{end})$ on the subinterval is less than the prespecified length-tolerance, or the calculated maximum angle is less than the tolerance angle, or the number of subdivision points exceeds a pre specified upper bound, then the subdivision process terminates. Let the sequence of break points be: t_j for j = 0, 1, 2, ..., m. The arc length parameter s is defined by the linear equation

$$s(t) = s_{j-1}\frac{t_j - t}{t_j - t_{j-1}} + s_j\frac{t - t_{j-1}}{t_j - t_{j-1}} \qquad t_{j-1} \le t < t_j$$

for j = 1, 2, ..., m.

The inverse relation t is defined in terms of s as

$$t(s) = t_{j-1}\frac{s_j - s}{s_j - s_{j-1}} + t_j\frac{s - s_{j-1}}{s_j - s_{j-1}} \qquad s_{j-1} \le s < s_j$$

for j = 1, 2, ..., m.

For comparing two parametric curves, the "correspondence" problem arises. If $\tau = g(t)$ is correspondence function, then $\tau_i = g(t_i)$ are corresponding points. The Euclidean metric may be used to find the distance between them

$$D_4(\mathbf{F},\mathbf{G}) = \max_{a\le t\le b} \|\mathbf{F}(t) - \mathbf{G}(t)\| \approx \max_{0\le i\le m} \|\mathbf{F}(t_i) - \mathbf{G}(\tau_i)\|$$

This value of this metric is zero for different parameterizations of the same curve, the zero metric value indicates identical curves. The smaller the value of the metric the stronger the similarity between the two curves. But the correspondence function g(t) may not be known. In this case new values of τ_i may be generated as follows. To determine the distance between two parametric curves, curvature method is used to get points on the curves: t_{1i}, i = 0, 1, ..., n_1; t_{2i}, i = 0, 1, ..., n_2. Using these break points, two sequences of the same length N are generated : t_i, i = 0, 1, ..., N; τ_i, i = 0, 1, ..., N, then the distance metric becomes:

$$D_5(\mathbf{F},\mathbf{G}) \approx \max_{0\le i\le N} \|\mathbf{F}(t_i) - \mathbf{G}(\tau_i)\|$$

Although pathological examples can be created to frustrate any numerical technique, such cases have not appeared in the test cases considered so far. It should be noted that no derivative calculations are required in this method, nor is any experimentation with m and n required to arrive at an optimal solution.

This discussion can be summarized in the following algorithm. In the algorithm, all user defined procedures and data elements begin with upper case letters. All types, counters and system procedures are in lowercase. Also the text is used as *padding* to facilitate reading the algorithm is in lowercase .

Conceptual Algorithm {Discrete Sampling of Points}

Initialize(a Tree with Curve, a linked TempList with Curve and an empty linked List);
Initialize(the curvature tolerance with Curv_Tol, number of break points upper bound with Num_Tol, segment length tolerance with Seg_Tol);
repeat

{ Remove(the Curve from the TempList);
Determine NeedToSubdivide = Subdivide(the Curve);
//This criteria is based on *subdivide* if (Calculated_curvature_angle > Curve_Tol and Count < Numb_Tol and Seg_Len > Seg_Tol)
if (NeedToSubdivide)
{ Partition(the Curve into Curve1 and Curve2 at the parametric mid point);
Make(Curve1 and Curve2 as the Child nodes in the Tree);
Insert(the Curve1 into the TempList);
Insert(the Curve2 into the TempList);
}
else
Insert(the Curve into the List);
}
until (TempList is empty)

Create_List(of discrete sample points from partition Points in the Tree);
Interpolate(arc length S using the partition Points)
End of the Conceptual Algorithm

To determine the similarity/disparity between two curves, an additional step is required. Since the two curves may yield different numbers of sample points on the two curves, the following steps may used to determine the equal number of partition points.

Conceptual Algorithm {for Similarity Metric}

(1) Use the above algorithm to determine partitions of the two curves $\mathbf{R_1}(t)$ and $\mathbf{R_2}(T)$ based on curvature

t_i: $i = 0,1,2,\dots, n_t$.

T_j: $j = 0,1,2,\dots, N_T$.

(2) Based on these partitions, determine approximate arc-length functions

$s_0 = 0;\ S_0 = 0$

$s_i = s_{i-1} + \|\mathbf{R_1}(t_i) - \mathbf{R_1}(t_{i-1})\| \quad i = 1,2,\dots,n_t$.

$S_j = S_{j-1} + \|\mathbf{R_1}(T_j) - \mathbf{R_2}(T_{j-1})\| \quad j = 1,2,\dots,N_T$.

(3) Normalize s_i and S_j to a range of unit length

$s_i = s_i / s_{n_t}$

$S_j = S_j / S_{N_T}$

(4) Merge the two sequnces to

m_k: $k = 0,1,2,\dots,N$.

(5) Generate the sequences of parameter values for $k = 0,1,2,\dots,N$

$t_0 = 0$

$t_k = t_{i-1} + \dfrac{m_k - s_{i-1}}{s_i - s_{i-1}}(t_i - t_{i-1})$ where $s_{i-1} \le m_k < s_i$ for some i: $i = 1,2,\dots,n_t$

and

$T_0 = 0$

$T_k = T_{j-1} + \dfrac{m_k - S_{j-1}}{S_j - S_{j-1}}(T_j - T_{j-1})$ where $S_{j-1} \le m_k < S_j$ for some j: $j = 1,2,\dots,N_T$

(6) Now define the distance metric as

$$D_5(\mathbf{R_1},\mathbf{R_2}) = \max_{0\le k\le N} \|\mathbf{R_1}(t_k) - \mathbf{R_2}(T_k)\|$$

Experimental Results

The experimental results suggest that the curvature method is preferable to the other two methods. Three examples are used for illustration. The first example deals with two straight line segments connected at (0, 1/3). This example is used due to its extensive use in literature [Fritsch and Nielson 1990]. The error is calculated as the deviation between the parametric mid point on the curve and straight line segment connecting the end points of each curve segment. The second example is used due to the non-linear nature of the curve. The third example is used due to the non-linear behavior at one end and approximate linear behavior at the end of the curve. For testing purposes, thirty four points are used to approximate the curves. The curvature tolerance of 2 degrees is used.

Method	Parametric	ArcLength	Curvature
MAX	7.926971e-04	2.266252e-02	1.490116e-08
AVG	2.734580e-05	7.814803e-04	9.099892e-09
STD	1.472886e-04	4.210912e-03	1.225224e-08

Table 1. $\mathbf{R}(t) = (0,1/3) + (1-t)^2(-1,\ 2/3)$ $0\le t\le 1$; $\mathbf{R}(t) = (0,1/3 + (t-1)^2(1,\ 2/3)$ $1\le t\le 2$. The maximum, average, standard deviation of errors using the three methods: Uniform parametric, uniform arc length, and uniform curvature method.

Method	Parametric	ArcLength	Curvature
MAX	9.518231e-04	9.518162e-04	1.204573e-04
AVG	9.517809e-04	9.517723e-04	1.003801e-04
STD	5.328747e-08	5.328650e-08	2.086170e-08

Table 2. $\mathbf{R}(t) = (\ t,\ |t|^2)$ $-1\le t\le 1$. The maximum, average, standard deviation of errors using the three methods: Uniform parametric, uniform arc length, and uniform curvature method.

Method	Parametric	ArcLength	Curvature
MAX	2.512092e-03	1.554835e-03	2.088427e-04
AVG	6.499185e-04	6.048873e-04	3.594497e-04
STD	7.533531e-04	5.813080e-04	5.685419e-04

Table 3. $\mathbf{R}(t) = (\ t,\ 1 - e^{-8t})$ $0\le t\le 1$. The maximum, average, standard deviation of errors using the three methods: Uniform parametric, uniform arc length, and uniform curvature method.

It is clear from these experiments that the maximum error, average error, and standard deviation in error are consistently less for the curvature method as compared to the uniform parametric and uniform arc length methods. The similarity metric for the distance between two parametric curves yields similar results.

Similar experiments are performed for the similarity metric between pairs of curves. The error is calculated as the deviation between the corresponding parametric points on both the curves. For testing purposes, thirty four points are used to approximate the curves. The curvature tolerance of 2 degrees is used. In this case also, three examples are used for illustration. The first example deals with two straight line segments connected at (0, 1/3) with *different* parameterizations. This example is used due to its extensive use in literature [Fritsch and Nielson 1990]. The second example is used due to the different non-linear behavior of the two curves. The third example is used due to the non-linear behavior at one end and approximate linear behavior at the end of both the curves.

Method	Parametric	ArcLength	Curvature
MAX	0.30018669	0.00674229	0.00000000
AVG	0.19423853	0.00350765	0.00000000
STD	0.09018283	0.00176407	0.00000000

Table 4. (1) $\mathbf{R}(t) = (0,1/3) + (1-t)(-1, 2/3)$ $0 \le t \le 1$; $\mathbf{R}(t) = (0,1/3 + (t-1)(1, 2/3)$ $1 \le t \le 2$. (2) $\mathbf{R}(t) = (0,1/3) + (1-t)^2(-1, 2/3)$ $0 \le t \le 1$; $\mathbf{R}(t) = (0,1/3 + (t-1)^2(1, 2/3)$ $1 \le t \le 2$. The maximum, average, standard deviation of errors using the three methods: Uniform parametric, uniform arc length, and uniform curvature method.

Method	Parametric	ArcLength	Curvature
MAX	0.24977043	0.17863822	0.17774135
AVG	0.16161619	0.11449315	0.11197755
STD	0.07503668	0.05339361	0.06019229

Table 5. (1) $\mathbf{R}(t) = (t, |t|)$ $-1 \le t \le 1$. (2) $\mathbf{R}(t) = (t, |t|^2)$ $-1 \le t \le 1$. The maximum, average, standard deviation of errors using the three methods: Uniform parametric, uniform arc length, and uniform curvature method.

Method	Parametric	ArcLength	Curvature
MAX	0.53465772	0.22260903	0.22039755
AVG	0.18221708	0.12150877	0.11505167
STD	0.19532605	0.07824913	0.07034623

Table 6. (1) $\mathbf{R}(t) = (t, 1 - e^{-8t})$ $0 \le t \le 1$. (2) $\mathbf{R}(t) = (t, 1 - e^{-8t^2})$ $0 \le t \le 1$. The maximum, average, standard deviation of errors using the three methods: Uniform parametric, uniform arc length, and uniform curvature method.

The similarity metric for the distance between two parametric curves yields similar results.. It is clear from these experiments [see tables 4-6] that the maximum error, average error, and standard deviation in error are consistently less for the curvature method as compared to the uniform parametric and uniform arc length methods.

Conclusion

This paper presents a new technique for solving the two problems. First, this paper addressed the problem of representation of analytic curves by a discrete set of sampled points. Previous techniques used Gaussian Quadrature and *ad hoc* methods to determine the arc length parameterization to approximate sampled points. This paper presents a curvature based technique which is independent of the Gaussian Quadrature and *ad hoc* experimentation. The experimental results show that new technique is preferable to the other methods. It is clear from these experiments that the maximum error, average error, and standard deviation in error are consistently less for the curvature method as compared to the uniform parametric and uniform arc length methods. Secondly, it is shown that the curvature method naturally lends itself to determine a similarity metric between two parametric curves. This method is easier to understand and simpler to implement.

The author may be reached at UMR Engineering Center, University of Missouri-Rolla, 8001 Natural Bridge Road, St. Louis, MO 63121,E-mail: Chaman@umrvmb.umr.edu

References

Barsky, B. [1981], *The B-Spline: A Local Representation Based On Shape Parameters And On Fundamental Geometric Measures*, **Dissertation, University Of Utah, Salt Lake City.**

Barsky, B. and DeRose T. [1989], *Geometric Continuity fo Parametric Curves: Part I*, **IEEE Computer Graphics Applications**, November 1989, Pp. 60-68.

Barsky, B. and DeRose T. [1991], *Geometric Continuity fo Parametric Curves: Part II*, **IEEE Computer Graphics Applications**, January 1991, Pp. 60-68.

Boehm, W. [1985], *Curvature Continuous Curves And Surfaces*, **Computer Aided Geometric Design** 2, 1985, Pp.313-323.

Emery, James D. [1986], *The Definition And Computation Of A Metric On Plane Curves*, **Computer Aided Geometric Design**, 18(1986), Pp.25-28.

Fritsch, F. N. And Nielson, G. M. [1990], *On The Problem Of Determining The Distance Between Parametric Curves*, **Lawrence Livermore National Laboratory**, UCRL-JC-105408, 1990, Pp.1-21.

Fritsch, F. N. [1986], The *wilson-Fowler Spline Is A v-Spline*, **Computer Aided Geometric Design**, 2, 1986, **Pp.** 325-328.

Fritsch, F. N. [1989], *Representation Of Parametric Cubic Splines*, **Computer Aided Geometric Design**, 6(1989), **Pp.** 79-82.

Guenter, Brian & Parent, Richard [1990], *Computing The Arc Length Of Parametric Curves*, **IEEE Computer Graphics And Applications**, 1990, Pp. 72-78.

Nielson, G.M. [1974], *Some Piecewise Polynomial Alternatives To Splines Under Tension*, In:R.E. Barnhill And R.F. Riesenfeld, Eds., **Computer Aided Geometric Design**, Academic Press, New York, 1974, Pp.209-235.

Sabharwal, C. L., Melson, T. G. And Fraser, M. D. [1991], *Curve Representation In CAD/CAM: An Intelligent Approach*, **The 4th International Conference On Industrial And Engineering Applications Of Artificial Intelligence And Expert Systems**, Hawaii, Jun 2-5, 1991, Pp. 498-503.