International Journal of Computer & Information Technologies (IJOCIT)
www.ijocit.org & www.ijocit.ir
ISSN = 2345-3877

2D Circular Boundary Representation in Computational Geometry

Saeed nejhadhoseini[1], Bahram Sadeghi Bigham[2]
Faculty of Computer and Information Technology Engineering[1], Faculty of Computer Science & Information Technology[2]
Qazvin Branch Islamic Azad University[1], Institute for Advanced Studies in Basic Sciences[2]
Qazvin[1], Zanjan[2]
Iran[1,2]
saeednjd@gmail.com[1], b _sadeghi_b@iasbs.ac.ir[2]

Abstract:- The boundary approximation of plane figures by means of circular arcs possesses many quantitative and qualitative advantages in comparison with direct piecewise linear method. This point may be displayed by means of the current computational techniques including convex hull and triangulation etc. The present essay is intended to emboss application of circular arcs for boundary representation. Hence, period of rendering nonlinear pieces increases in contrast to linear segments. Likewise, this paper is the product of the resultant outcomes from a few studies concerning to boundary representation by means of circular arcs.

Keywords: Convex hull, Approximation, Circular Arcs, Triangulation.

1. Introduction

Most of algorithms in computational geometry have been designed for processing of linear objects such as lines, planes or polygons. This issue is due to this fact that many computational questions are raised in this case. On the other hand, the realistic reason may be in that the linear objects algorithms can be usually further simplified both for extension and in terms of execution. Thus, we work on nonlinear objects, which these objects of piecewise type are usually approximated with relatively appropriate error.

Briefly, approximation of circular arc in figures may be observed in linear time form in appropriate mode. In other words, figures processing algorithms are superior to their opposite side based on line segment in terms of execution time.

At simplest mode, the input object is a unihedron figure called *A* with closed boundary *∂A* in curve form. The activities which should be done on *A* include constructing of convex hull *A* and triangulation *(A)* in primary sections. These tasks are well employed in some modes where figures are polygon.

2. Approximation by Means of Circular Arcs

To display Figure (A) as an appropriate form for geometric computations we use some techniques for approximation of its boundary i.e. *∂A* by circular arcs. We suppose *∂A* is given as polynomial spline curve. Although, it is possible to render spline curves in all order, with respect to its application, we deal with cubic splines in this method [1].

There are many methods to produce circular arc splines [2] which of course we refer only to a simple Bisection Algorithm in this chapter including two steps i.e. "Approximation" and "Error Measurement". So, the primary geometric figure *b* (one or two arcs) are placed on segment *s* from *c(t)* curve and distance interval from *b* to *s* is numerically computed. This algorithm is relatively easy for execution but the only disadvantage of this algorithm is in that quantity of primary geometric figures (volume of data products) is only optimal in asymptotic mode. We define one-way Hausdorff distance from *b* to segment $S \subseteq C(t)$ as follows:

$$\delta(b,s) = \max\ min\ \| p - q \|$$

Where, *b* and *s* are close sets. And now suppose ε denotes error tolerance and it is characterize by algorithm.

Algorithm BISECT(t_0, t_1)

Construct b
Compute δ = δ(b, c[t_0, t_1])
If δ ≤ ε then return {b}
Else return BISECT(t_0 , $\frac{t_0+t_1}{2}$) ∪ BISECT($\frac{t_0+t_1}{2}$,t_1)

In order to move from piecewise parametric spline to other segment of this curve, we can execute various continuity conditions. If we define any part of spline curve with parametric functions as follows:

(1) $x = x(u)$, $y = y(u)$, $z = z(u)$, $u_1 <= u <= u_2$

Thus, one could execute parametric continuity by setting parametric derivatives of this segment in adjacent curves equal in common boundary points. Therefore, zero- order continuity (C^0) means that the calculated x, y, and z- values in u_2 for the first pieces of curve should be equal to the valuated values of x, y, and z in u_1 and first-order parametric continuity (C^1) is also means that first- order parametric derivatives of functions are equal at the point of connection two subsequent points in curve.

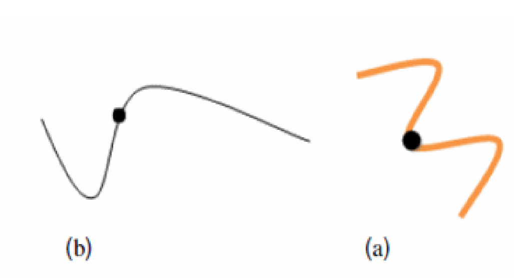

Figure 1: (a) First-order continuity; (b) zero-order continuity

With respect to the given primary *b*, BISECT algorithm produces several splines with different qualities in which circular arc splines (C^0) can be strongly continuous or arc splines (C^1) may be tangentially continuous. Therefore, we can easily select unitary circular arc for *b* with passing through three points $C(t_0)$, $C(\frac{t_0+t_1}{2})$, and $C(t_1)$. To achieve arc splines C^1, term biarc has been used [5].

Thus, *b* comprises of two circular arcs, which have tangent singular vector at their connection point. Usually, *b* is described by source *x* with tangent singular vector $\mathbf{V_x}$ and *y* with tangent singular vector$\mathbf{V_y}$ with respect to these data, there will be a family of parametric interpolation bi-arcs.

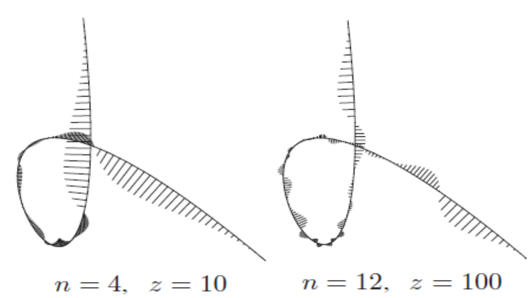

Figure 2: z-magnified error for n biarcs

Any arc is found at time $O(1)$; therefore, the fixed value depends on degree (order) of the polynomial equation that should solved. Fig (2) shows an example from biarc transform. The graduate vertical; curves in this figure indicate error magnitude distribution.

Regarding error measurement, any created circular arc b_i should correspond to it proportional to segment $S = c[t'_0, t'_1]$. In the mode of third- order curvatures, this action may lead to cubic equation. If there are several solutions within intervals among bi-arc $[t_0, t_1]$, the error rate may approach to ∞. Otherwise, we should compute one-way Hausdorff distance $\delta(b_i, s)$ by substitution of the present parameters s with explicit equation *k* (with distance coefficient 1) from b_i. If *r* is radius of *k* and *d* and *D* are minimum and maximum values $(k \circ C)(t)$ for $t \in c[t'_0, t'_1]$ respectively, we will have:

$$\delta(b_i, s) \leq \max\left\{\left|\sqrt{r^2 - d} - r\right|, \left|\sqrt{r^2 + D} - r\right|\right. \quad (2)$$

Thus, this is an acute boundary (steep slope). As a result, in cubic mode, value $\delta(b_i, s)$ can be assessed by solving bivariate polynomial equations within intervals $[t'_0, t'_1]$. Frequently, without solving polynomial equation, the upper boundary can be calculated from $[t'_0, t'_1$ by substitution of the presented *d* and *D* by *Bernstein- Bezier* curve [6]. As length s reduced, this boundary (limit) also becomes convergent toward $\delta(b_i, s)$. As another example, two-way Hausdorff distance among *s*, b_i , and $\max[\delta(b_i, s), \delta(s, b_i)]$ approaches to zero with $\delta(b_i, s)$ since *s* and b_i have fixed order. Therefore through controlling next interval, it can be assured that *s* and b_i are much closer compared to the previous state.

We intend to examine asymptotic behavior of number *n* to reduce allowed error ε. we suppose that the given curve $c(t)$ within range $[t_0, t_1]$ does not include bending (curvature) and angular vertex. Then, we assume that the primary geometric figures have approximate *k*- order. With adjusting analysis for one-way Hausdorff distance δ, we have $\delta = \Theta(h^k)$ while $c(t)$ with pace value of parameter h (with small size) may be approximated and also k is a fixed value [2, 7].

This is the expression for lower boundary. We derive any approximation $c(t)$ by BISECT (t_0, t_1) with using n up to primary geometric figures and with approximate k- order; we acquire the maximum size of pace (step) via $\Delta t \geq \frac{t_1 - t_0}{n}$. Therefore, we have $\delta \leq \varepsilon$ under final conditions of approximation algorithm. We derive n=$\Omega(1/_{\varepsilon^{1/k}})$ from $\delta = \theta((\Delta t^k))$.In other words, the minimum size of pace $\Delta' t$ that is derived from any algorithm within intervals I may apply to expression $\Delta' t \leq \frac{t_1 - t_0}{n}$. Suppose that we stop BISECT (t_0, t_1) with double pace $2\Delta' t$. Therefore, there will be at least one interval. For example, what it is included in I will apply to $\delta > \varepsilon$. As we have $\delta = \theta((2\Delta' t)^k)$ and $n = O(1/_{\varepsilon^{1/k}})$ so we derive:

Lemma 1: Number of n primary geometric figures that is built with tolerance value ε is asymptotically optimal with Bisection algorithm. Similarly, Lemma (1) keeps the general state where $c(t)$ includes bending (curvature) and vertex since number of the derived spirals from $c(t)$ is independent from n. As a result, to achieve tolerance ε, it requires Bisection algorithm to have $n = \theta(1/_{\sqrt[3]{\varepsilon}})$ (third-order) circular arcs while (second-order) line segment $N = \theta(\frac{1}{\sqrt{\varepsilon}})$ is processed by polygonal approximation method.

Inference 1: In contrast to approximation of $c(t)$ curve with multi-linear equation, when circular arc splines are employed, the size of data is reduced from N to $n = \theta\ (N^{2/3})$.

It should be noticed that when $c(t)$ approximation is done in a point sample (as it usually is done for calculations of middle axis [8]), size of data increases compared to n circular arc to $\theta(n^3)$.

International Journal of Computer & Information Technologies (IJOCIT)
Corresponding Author: *Saeed nejhadhoseini*
May, 2014 Volume 2, Issue 2

3. Convex Hull

Suppose boundary of figure *A* is represented by an arc. More specifically, ∂A is approximated by a close and simple curve *b*, which is composed of *n* circular arcs. It is clear that if *b* is converged to ∂A then convex hull *b* is also converged to convex hull *A* as well. Moreover, Hausdorff distance of two convex hulls may be bounded with *b* and ∂A within Hausdorff distance. In this chapter, we will indicate that in Melkman technique, multi-linear convex hull algorithm can be also augmented by circular arc curves rather than remaining at time of execution of *O(n)* [5].

Suppose $b_1 \ldots . b_n$ as a simple circular arc curve. The second terminal point of every arc b_i is called target b_i. Some of arc may be line segment and the curve can be periodic (cycle). Initially, suppose there is a curve C^1 that identifies convex hull CH and consider $CH(b_1, \ldots b_i)$ briefly as CH_i (Fig 3).

3.1 Hull algorithm

At first, $CH_2 = CH(b_1 b_2)$ is built. Suppose *v* is the last point along with chain $b_1 b_2$ that is placed on CH_2. For $i = 3, \ldots . n$, arc b_i is processed as follows:

Look for the first arc, we move *(a)* with CH_{i-1} at clockwise direction from *v* with non-zero length to$CH(a, b_i)$ so that this hull and CH_{i-1} are placed at one side of a. Similarly, look for first arc *c* and rotate *v* counterclockwise with the same characteristics (possibly a=c). Arcs *a* and *c* have already prepared the needed data to construct CH_i properly.

State 1: Arc *a* (and equally arc *c*) does not exist. This means that $CH_{i-1} \subset CH(b_i)$. Then put $CH_i = CH(b_i)$ and allocate it to *v*, which is targetb_i.

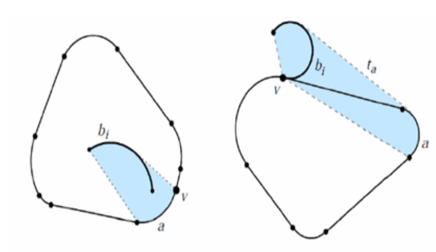

Figure 3: states 2.1 (left figure), and 2.4 (right figure)

State 2: There are arcs *a* and *c*. Check tangent line t_a in which $CH(a, b_i)$ is displayed and it is clockwise tangent on CH_{i-1}. Check figure 4,5 (right side) as well as tangent line t_c that is shown on $CH(c, b_i)$ and they move counterclockwise in CH_{i-1}.

State 2-1: There are not both tangents t_a and t_c. This means $b_i \in CH_{i-1}$. Therefore it is $CH_i = CH_{i-1}$.

State 2-2: There is t_a but not t_c. Suppose $t_a = x_a y_a$ so that x_a is a tangent point on CH_{i-1}. To acquire CH_i, omit clockwise rotary part among *v* and delete x_a from CH_{i-1}. And add t_a and that piece of arc b_i among *v* and y_a. Update *v* as the last point along with b_i on CH_i (or y_a or target b_i).

State 2-3: There is (singular) t_c but t_a does not exist. Consider $t_c = x_c y_c$ in which x_c point is tangent on CH_{i-1}. To acquire CH_i, erase counterclockwise rotary segment among *v* and x_c from CH_{i-1}; and add t_c and a piece of arc b_i among y_c and *v*. Update *v* (state 2,2) (or y_c or target b_i).

State 2-4: There are both t_a and t_c. Here, we derive CH_i by erasing arcs from CH_{i-1} at modes of 2,2 and 2,3 then adding piecewise values t_a and t_c from b_i among y_a and y_c. Then we put *v* as a point among y_a and y_c that is closer to target b_i. The accuracy and precision of HULL algorithm have been derived with observing that t_a and t_c are real tangent lines from arc b_i in convex hull CH_{i-1}. As a result, as it shown in this algorithm, input curve is C^1. And this guarantees that boundary CH_{i-1} is as precise as C^1 (probably except in target b_{i-1}) so that arcs *(a)* and *(c)* were created accurately. The partial corrections in selection of criteria for these arcs may cause algorithm to operate without constraint [5].

With searching for arcs *(a)* and *(c)* where number of contacts of biarc hull subsets is proportional to total number of the built and or omitted arcs, execution period is controlled. This quantity is *O(n)* because arc *O(1)* is only built for any ring i. The rest of them may be done per any arc b_i at time *O(1)* if CH_i is saved as double linking list and or they are done in general time $O(n)$ and CH_i is placed in the same queue.

4. Triangulation

In this section, triangulation algorithm is suggested for circular arc figures. We consider an arc triangle as a figure that is maximally bounded with three circular arcs or line segments.

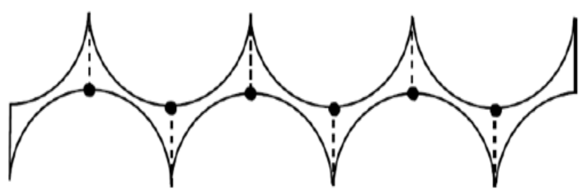

Figure 4: Many Steiner Points.

Circular arc in figure *A* should not be divided into arc triangles when using Steiner points is not allowed. These conditions will not vary if *n* arcs, which describe ∂A, are in uniform *x*- piecewise segments. Hence, these openings should maximally in semicircular form so we have assumed them in the following. In fact there are some examples in which it necessitates 2n-7 Steiner points at least. Look at Fig 4. There is circular arc connecting inside *A* for none of the pairs of shown vertices. Therefore, no segment of *A* can be divided by means of circular arc among two vertices. Note that any segment of singular point is sufficient when circular arcs are used for dividing of segments instead of line segments.

5. Conclusion

We indicated for the aforesaid problems above that there are simple and efficient algorithms, which can be also employed for circular arc data. In other words, these algorithms can be rendered by means of circular arcs. Thus, approximation of circular arcs of figures may be seen under the condition that is proportional to linear time form. Compared to their opposite side based on line segment, figures processing algorithms are superior in terms of rendering time. As a result, it is useful to utilize circular arcs for boundary transform and at the same time results of them are advantageous in terms of geometric approximation and computational geometry.

6. References

[1] G. Farin, “Curves and Surfaces for Computer Aided Geometric Design”.Academic press,san Diego، 1997, pp. 9-10,.

[2] D.S.Meek and D.J.Walton, “Approximation of a planar cubic Bezier spiral by circular arcs,” J.Comput .Appl.Math. 1996, pp. 47-56,.

[3] M.A.Sabin, “The use of circular arcs to form curves interpolated through data points” Rep. VTO/ MS/ 164, British Aircraft Corporation, 1976.

[4] A.Melkman, “On-Line construction of the convex hull of a simple polygon”,inform.Process.Lett, 1987, pp. 11-12 ,.

[5] O.Aichholzer and F. Aurenhammer and T. Hackl,” Computational and Structural Advantages of Circular Boundary Representation” International Journal of Computational Geometry & Applications,Vol. 21, No. 1, 2011, pp. 47-69..

[6] Farin. G, Hoschek. J and Kim. M.-S., " Handbook of Computer Aided Geometric Design" . Elsevier. 2002.

[7] Sir. Z, Feichtinger. R And Juttler. B. "Approximation curves and their offsets using biarcs and Pythagorean hodograph quintics". Comput. Aided Des. 38. 2006, pp 608-618..

[8] Attali. D, Boissonnat. J. D and Edelsbrunner. H., "Stability and computation of medial axes – a state-of-the-art report". Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration. 2008.

Authors profile:

Saeed Nejhadhoseini

He is currently pursuing his master of Computer Engineering, Software Engineering, at Faculty of Electrical and Computer Engineering in Qazvin Islamic Azad University. His research interests are Algorithms, Computational Geometry. He now works as project manager software is a network issue.