

A Robust Approach to Find the Control Points for Wide Variety of 3 rd Order Bézier Curves

Alok Kumar Chowdhury¹, Prithwi Raj Chakraborty² and Md. Ibrahim Khan³

¹ Premier University, Chittagong, Bangladesh.

Received: 28 August 2011 Accepted: 17 September 2011 Published: 27 September 2011

Abstract

This paper represents a new approach that can recover the control points for wide variety of 3rd order Bézier curves. In this regards, the two stage approximation learning algorithm is adopted with some modifications. At 1st stage our key feature is segmentation of the curve which can determine intermediate points of the wide variety of curves. In this respect, an efficient recursive algorithm is used to find out the height of the curve (h) with less iteration. The proposed approach introduced horizontal segmentation rather than vertical segmentation. Different height (H), where the 2nd and 3rd control point are assumed, and also the step-size (?), at which the control points are moved toward the actual direction, are used to find out the exact location of the control points. Experimental results demonstrate that our proposing method can recover control points for wide variety of curves with minimum error level and less iteration. Wide variety of curve shapes are used to test the proposing approach and results are presented to prove its effectiveness

Index terms— Bézier curve, curve fitting, segmentation of curve, learning algorithms.

1 INTRODUCTION

he development of Computer Graphics has made computers easier to interact with, and better for understanding and interpreting many types of data which has put a profound impact on many types of media and have revolutionized animation, movies and the video game industry. Generally, Computer Graphics are the representation and manipulation of image data by a computer with help from specialized software and hardware [1].

In Computer Graphics and related fields, a frequently used parametric curve is Bézier curve. Bézier curves are used to model smooth curves that can be scaled indefinitely. As the curve is completely contained in the convex hull of its control points, the points can be graphically displayed and used to manipulate the curve intuitively. When more complex shapes are needed, A Bézier curve is defined by its order (Linear, Quadratic, Cubic, etc.) and a set of control points. An nth order Bézier curve has n+1 control points (P₁ to P_{n+1}). The first and last control points are always the end points of the curve. These two end points are can be called initial and terminating point of the curve respectively. The intermediate control points generally do not lie on the curve; they define the shape and direction of the curve. A 1 st order (Linear) Bézier curve is simply a straight line between those two given points P₁ and P₂. A 2 nd order (Quadratic) Bézier curve is the path traced by the function B(t), given points P₁, P₂, and P₃. Four points P₁, P₂, P₃ and P₄ in the plane or in three and arrives at P₄ coming from the direction of P₃. P₁ and P₄ are end points [4][5]. Here P₁ and P₄ are end points as well as P₂ and P₃ are intermediate control points. A 3 rd order Bézier curve given by the following equation is shown in fig. 1. Our discussions will be limited to 3 rd order Bézier curve. Though there are several algorithms and approaches to find out control point of 3 rd order Bézier curve, these approaches have limitations like incorrect result, requirement of compatible boundary for control points etc. In our previous work [6], we tried to overcome these limitations and it finds control points of 3 order Bézier curve in an efficient way.

But it is still unable to find out control point of 3 rd order Bézier curves of some certain shapes. In this paper we are going to propose a modified approach of our previous work to erase the existing limitations. Our new approach is capable of finding out the control points of a large variety of 3 order Bézier curve shapes efficiently and accurately with minimum error and less iterations.

Bézier curves are patched together. In Animation applications, such as Adobe Flash and Synfig, Bézier curves are used to outline, for example, movement. Besides, True-type Fonts use Bézier curves. Bézier curve is also a very powerful tool for shape approximation and shape comparison. In identifying of actors drawn in ukiyoe pictures, Bézier curves are widely used [2]. They can be used in face recognition and facial emotion detection [3].

(1) Fig. ?? : Abstract -This paper represents a new approach that can recover the control points for wide variety of 3 rd order Bézier curves. In this regards, the two stage approximation learning algorithm is adopted with some modifications. At 1st stage our key feature is segmentation of the curve which can determine intermediate points of the wide variety of curves. In this respect, an efficient recursive algorithm is used to find out the height of the curve (h) with less iteration. The proposed approach introduced horizontal segmentation rather than vertical segmentation. Different height (H), where the 2 nd and 3 rd control point are assumed, and also the step-size (?), at which the control points are moved toward the actual direction, are used to find out the exact location of the control points. Experimental results demonstrate that our proposing method can recover control points for wide variety of curves with minimum error level and less iteration. Wide variety of curve shapes are used to test the proposing approach and results are presented to prove its effectiveness. dimensional space define a 3 rd order (Cubic) Bézier curve. The 3 rd order curve starts at P 1 going toward P 2 $B(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3$

2 BACKGROUND

There has been several works in the field of recovering control points of Bézier curves. All the familiar algorithms and characteristics are associated with Bézier curves are generation [7], approximation [8], interpolation [9], subdivision [10][11][12], degree elevation [13][14][15][16], blossoming [17], implication [18] etc. X. Ye represented an approach for directly generating Bézier points of curves and surfaces explicitly from the given compatible arbitrary order boundary information of Hermite curves, Coons-Hermite Cartesian sum patches and Coons-Boolean sum patches [19]. But for Hermite curves, this approach requires the end positions, tangents and higher order derivatives at the end points. Moreover it requires the corner points and the compatible arbitrary order partial derivatives at these points for Coons-Hermite Cartesian sum patches. rational Bézier curve, the interpolation method developed by J. Chou and L. A. Piegl works good for special type of 3 rd order Bézier curves, not all kind of Bézier curves [20]. Besides, this approach relies on the convex hull and on the variation diminishing properties of Bézier curves. And the most important drawback is that if the curve segment is highly curved in one region and relatively flat in another, the approximation of this method is not good.

The previous approach, proposed by us, contains none of these problems. It can successfully find out control points of 3 rd order Bézier curve with minimum iteration and error. Fig. 2(b) and 2(c) are showing the way how the previous approach successfully finds out the control points for the curve of a particular shape indicated in fig. 2(a). The details are described at the previous work [6].

However, for some curves of shapes like fig. 3(a) and 4(a), the previous approach is unable to find out the accurate control points. It is because the segmentation process of the previous approach, which uses vertical segmentation, is different from the current one (described at Methodology section). Fig. 3 and 4 are showing limitation of the previous approach in finding out the control points for some special shaped curves. The previous approach cannot rescue the whole area for these special shaped curves, indicated in fig. 3(c) and 4(c). Thus the control points found are also not correct for these curves. The curves drawn in blue color are the recovered curves. But the new approach, with modified segmentation method, is able to find out the actual control points very successfully with minimum iteration indicated in fig. 2(a), 3(a) and 4(a). It can rescue the whole area of special shaped curves which is described later in Simulation Results section.

3 III. METHODOLOGY

curve contains four control points. The objective is to find out them. In order to do this, the new proposing approach is divided into two stages: modified first stage and second stage. Finding out of the end points: The end points will be lying on the Bézier curve. End points must be those two which have minimum y values. Now it is to be decided which one of these end points is initial point and which one is terminating point. Among these two end points the point which is nearest from Y axis (i.e. x) These two end points are shown in fig. 5.

In determining inner control points of 3 rd order and error for all the 3 rd order Bézier curves of shapes As it is mentioned before, a 3 rd order Bézier value is minimum) is our initial point, P 1 (x 1 , y 1). So, the remaining one is for sure the terminating point, P 4 (x 4 , y 4). ? Now the height, h and the peak intersection point, M of the Bézier curve and the parallel line drawn at distance h from the base line are found by a recursive algorithm which makes the new approach faster than the previous one. This algorithm draws some parallel lines at some assumed distances, d from the base line. If a parallel line intersects the curve at two points, this line is stored and the next parallel line is drawn above the currently stored parallel line at a distance d. If the drawn parallel line does not intersect the curve then that line is not stored and the next parallel line is drawn above at a distance d

/ 2 from the previously stored parallel line. This algorithm contains three methods -draw_parallel(_line, _d), intersections(_curve, _line), and distance(_line1, _line2). The draw_parallel(_line, _d) method takes a line equation and a distance as inputs and returns the parallel line drawn at a distance _d from the line _line. If the equation $ax + by + k = 0$ be the value of input variable _line and d is the value of input variable _d, then draw_parallel(_line, _d) returns a parallel line using the following equation:

$$(3)$$

The next method intersections(_curve, _line) takes a curve equation and a line equation and returns the number that represents at how many points the line intersects the curve. The peak most segment of the Bézier curve may be appeared as a tiny line rather than a curve. Therefore, the parallel line may intersect the curve at several consecutive intersection points. In that case, the middle one from the consecutive points is taken as the desired single intersection point and the number of intersections is considered as one. The last method distance(_line1, _line2) returns the distance between two lines. Now we should look at the recursive algorithm. Segmentation of the Bézier curve: The segmentation process of the new proposing approach follows horizontal approach rather than vertical approach followed by the previous approach. In segmentation, the following steps are followed: The greater value of n is taken, the more accurate segmentation and thus more accurate result is found but in that case computational cost gets higher. This stage is same as our previous work [6]. As proposing method focus on minimization of error, so the 2 nd and 3 rd control point must be in the desired location. In order to find the exact location the approach always calculate the error between the two curves iteratively. If the error becomes minimum, according to algorithm it finds the exact location of 2 nd and 3 rd control points. The value of the error is the summation of the difference between the points obtained from given curve $Q(t)$ and their corresponding points generating from newly found control points in the approximated Bézier curve $Q'(t)$ which is shown in fig. 9. Since Bézier curves can be obtained using a parameter t, where $0 \leq t \leq 1$, the error between two curves $Q(t)$ and $Q'(t)$ can be computed by considering the curves in parametric expressions and finding the corresponding points. For N+1 corresponding points, we are getting locations of corresponding points $Q(t)=(Q_x(t), Q_y(t))$ and $Q'(t)=(Q'_x(t), Q'_y(t))$, $(t=0, 1/N, 2/N, \dots, 1)$. The equation that is used for calculating error as follows: $2011 \text{ October } (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ (a) (b) points. $?? = ? + ? = ? = N k y y x x N k N k Q N k Q N k Q N k Q N k Q N k Q E \text{ Error } 0.2201$ () ({ }) ({ | }) (| ,

The reason why summation is used instead of integration is for computational simplicity. It is applicable when the original Bézier curve is known. In order to recover the exact control points the proposing approach try to move those points at a step ? towards both X and Y coordinates which shown in fig. 10. The value of the control point will be updated if the calculated error is minimized, otherwise it will not update. If the step fails to minimize error according to algorithm it reduces the step by half and recursively performing the same operation as stated above.

Increasing the efficiency of the overall program was the main contribution as well as overcoming the limitation of the step-size. Although the approach able to get the accurate result for some Bézier curve by considering step size $?x=?y=1$ [21], but it takes more iterations compared to our propose step size, which reduce the efficiency of the overall performance. Here proposing modification works well for most of the Bézier curve and get the error level less than 0.000001 which increase the dynamicity of overall program because it need less iteration. With variable step $?x=?y=5, 50, 75$ and 100 in our simulation result we get near to 100% accuracy. The algorithm followed in the second stage is given below.

4 SIMULATION RESULTS

The main contribution of the proposed scheme is to find control points of large number of Bézier curve shapes while keeping the efficiency and accurateness of our existing method. We checked several Bézier curves of different shapes with our computer implemented simulation and got desired results as we expected.

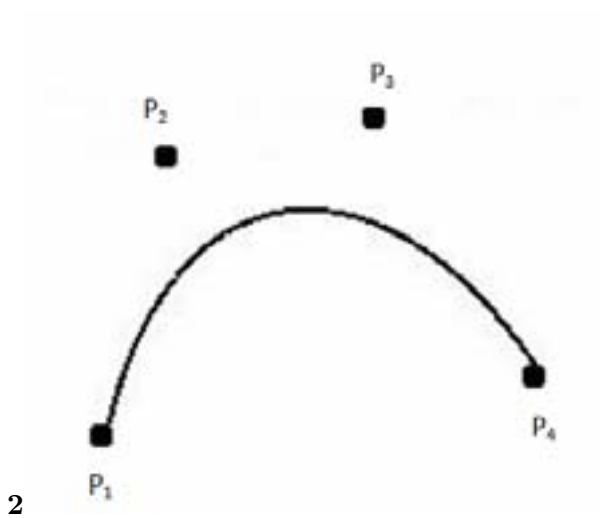
Consider the simulation result shown in fig. ??1 and fig. 12. Figures of simulation results are picked directly from our computer implemented simulation. At first stage we segment given curve to know its internal points and assume some control points at some probable height using our new proposed schema. Then using the algorithm of 2nd stage we successfully find the control points. Previous Method [6] was unable to extract control points of those shaped curves because of its limited segmentation technique.

Those shaped curves, whose control points can be found by existing method, can also be found by the

¹(x₁, y₁) P



Figure 1:



2

Figure 2: Fig. 2 :



3

Figure 3: Fig. 3 :

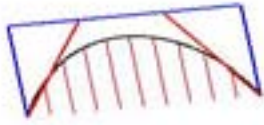
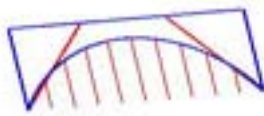


Figure 4:



5

Figure 5: Fig. 5 :



6

Figure 6: Fig. 6 :

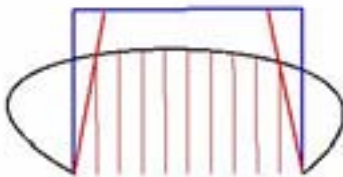


Figure 7:

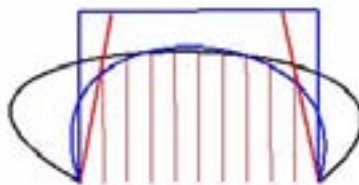


Figure 8: ?

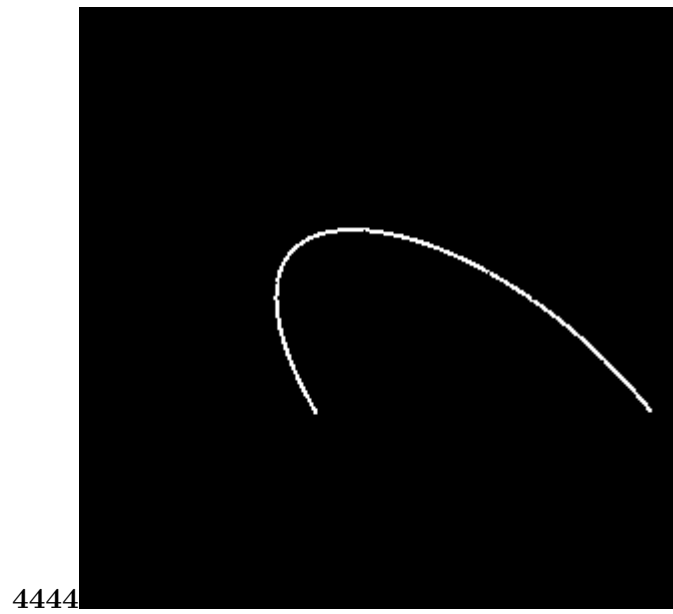
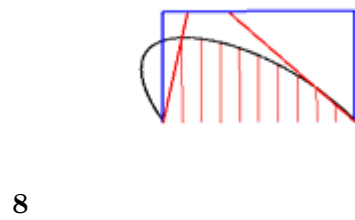
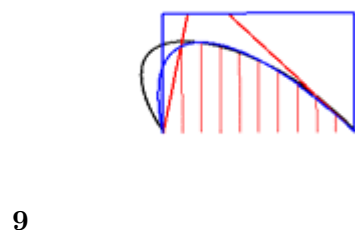


Figure 9: $4 P_4(x_4, y_4)$



8

Figure 10: ?Fig. 8 :



9

Figure 11: Fig. 9 :

10

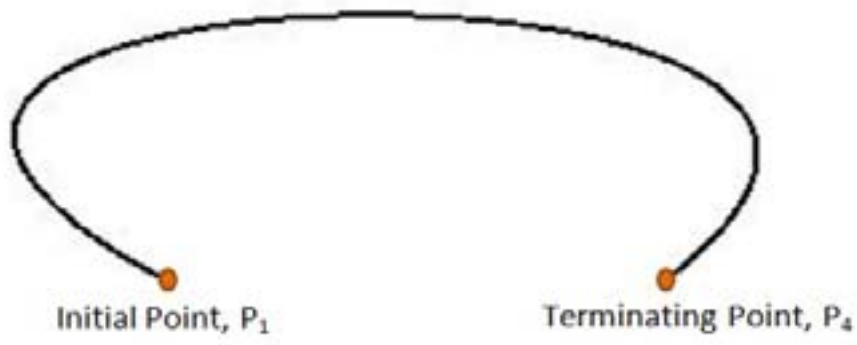


Figure 12: Fig. 10 :

222

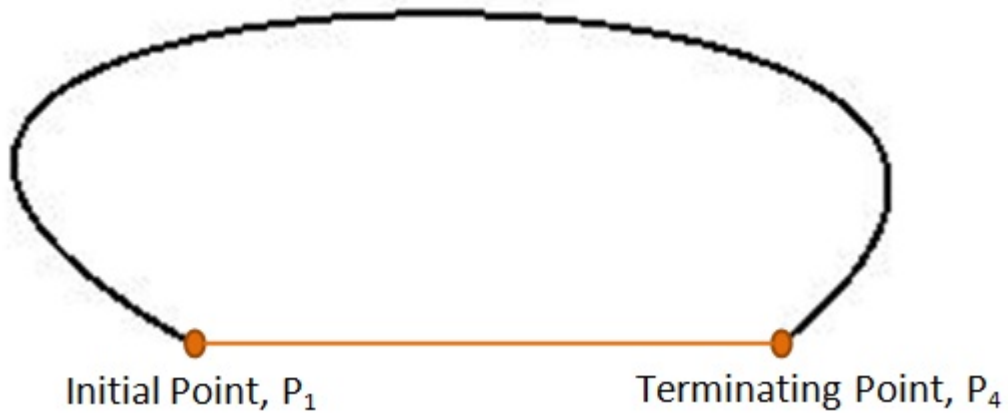


Figure 13: [2 [2 [2 [

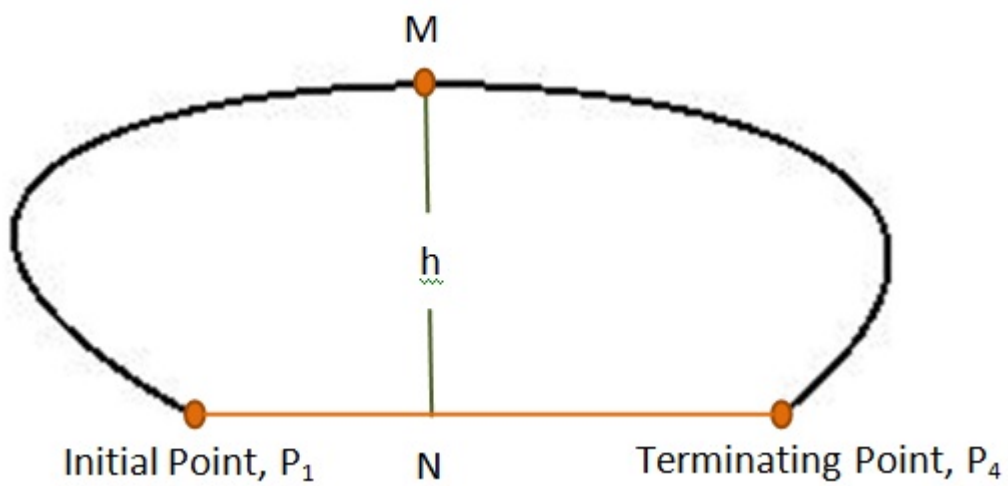


Figure 14: Global

Figure 15: Table . 1

(4)

(5)

Data from Table ?? shows that minimum error is found at different combination of step-size and height. In this case when height=1.5h and step-size= 5, or height =1.5h and step-size=75 the error is minimum that is $2E-3$. But when height=1.5h and step-size=75, our simulation takes less iteration. That's why we consider (282, 338), (775.37674, 220.0396) The new modified approach can recover the control points of 3 rd order Bézier curves more accurately and efficiently than those methods which was proposed earlier. In addition, the emphasis of this paper is on the implementation of a new method that can recover the more different shaped Bézier curve which is not possible according to earlier methods. While conducting the experiment different shapes of 3 rd order Bézier curves are taken for simulation and method is proved by the simulation result.

[E (1974)] *A subdivision algorithm for computer display of curved surfaces*, E . December, 1974. University of Utah (Report UTEC-CSc-74-133)

[Gravesen ()] 'Adaptive subdivision and the length and energy of Be'zier curves'. J Gravesen . *Comput. Geometry* 1997. 8 (1) p. .

[Piegl and Tiller ()] 'Algorithm for degree reduction of B-spline curves'. L Piegl , W Tiller . *Comput. Aided Design* 1995. 27 (2) p. .

[Khan et al. ()] *An Efficient Algorithm for Finding the Control Point of 3rd Order Bézier Curve*. *International Forum on Strategic Technology*, M I Khan , S Chowdhury , A K Chowdhury , K Deb . 2010. Ulsan. p. .

[Sederberg et al. ()] 'Approximate implicitization using monoid curves and surfaces'. T W Sederberg , J Zheng , K Klimaszewski , T Dokken . *Graph. Models Image Process* 1999. 61 (4) p. .

[Sederberg and Farouki ()] 'Approximation by interval Be'zier curves'. T W Sederberg , R T Farouki . *IEEE Comput. Graphics Appl* 1992. 12 (5) p. .

[Ramshaw ()] *Blossoming: A connect-the-dots approach to splines*, L Ramshaw . 1987. (Digital Systems Research Center Report No. 19, CA)

[Farin ()] *Curves and surfaces for computer aided geometric design: a practical*, G Farin . 1997. Academic Press.

[Yamaguchi ()] *Curves and Surfaces in Computer Aided Geometric Design*, F Yamaguchi . 1988. Springer Verlag.

[Chou and Piegl ()] 'Data reduction using cubic rational B-splines'. J J Chou , L A Piegl . *IEEE Computer Graphics Applications* 1992. 12 (3) p. .

[Bogacki et al. ()] 'Degree reduction of Be'zier curves by uniform approximation with endpoint interpolation'. S E Bogacki , X Weinstein , Ye . *Comput. Aided Design* 1995. 27 (9) p. .

[Mohammad Ibrahim Khan and Al-Amin Bhuiyan (2009)] 'Facial Expression Recognition for Human-Robot Interface'. Md Mohammad Ibrahim Khan , Al-Amin Bhuiyan . *IJCSNS International Journal of Computer Science and Network Security* April 2009. 9 (4) p. .

[Hama and Okamoto ()] 'Fast generation method of Be'zier curves and anti-aliasing'. H Hama , T Okamoto . *J. TV Engineers Japan* 1993. 47 (12) p. .

[Hu et al. ()] 'Generalized subdivision of Be'zier surfaces'. S Hu , G Wang , T Jin . *Graph. Models Image Process* 1996. 58 (3) p. .

[Ye ()] 'Generating Be'zier points for curves and surfaces from boundary information'. X Ye . *Comput. Aided Design* 1995. 27 (12) p. .

[Md et al. ()] 'Identification of Actors drawn in Ukiyoe picture'. Md , Hiromitsu Al-Amin Bhuiyan , Hama . *Pattern Recognition* 2002. 35 (1) p. .

[Barry and Goldman ()] *Interpolation and approximation of curves and surfaces using po'lya polynomials*. *Graph. Models Image Process*, J Barry , R N Goldman . 1991. 53 p. .

[Eck ()] 'Least squares degree reduction of Be'zier curves'. M Eck . *Comput. Aided Design* 1995. 27 (11) p. .

[Md et al. ()] 'On recovering the control points of Bézier curves for line image indexing'. Md , Hiromitsu Al-Amin Bhuiyan , Hama . *Global Journals Inc* 2002. 2011. US. 11 (2) p. 177. (J. Electron Imaging)