An Iterative Algorithm for Determining the Arc Length of a High Order Flat Bezier Curve

Dmitry Tarasov [1, a)] and Oleg Milder [1, b)]

[1]*Ural Federal University, Yekaterinburg, Russia*

[a)]Corresponding author: datarasov@yandex.ru
[b)]milder@mail.ru

Abstract. Quantifying the spatial characteristics of information stored and disseminated electronically is a complex computational challenge. Flat vector objects such as symbols, tracks, routes, *etc*. are described using the mathematical apparatus of Bezier curves. Finding the perimeters of such objects, especially in the case of curves of order higher than the third, is associated with certain difficulties. Reducing the order of curves by dividing or splitting them into sub-curves of lower orders, accompanied by some decrease in the accuracy of the estimate, is a convenient method for fast calculating the perimeters of plane figures described by Bezier curves. In this work, we propose an iterative algorithm for determining the arc length of a Bezier curve, which compares different criteria for splitting a curve into sub-curves.

Keywords: Flat figure, Bezier curve, perimeter, splitter.

INTRODUCTION

Big Data Analysis is closely related to the calculation and comparison of various metrics of the objects under study [1]. Information stored and disseminated in electronic form may often be represented as sets of flat vector objects (texts, maps, drawings, *etc*.). Certain their metrics might be obtained automatically directly from the files [2]. Calculation of the spatial parameters of such objects is a difficult task both in terms of its formalization and in terms of the required computing power. Describing vector objects using Bezier curves might help one solve this problem. However, when describing the studied objects by curves of the third and higher orders, computational difficulties also arise. Attempts to analytically calculate the arc length of a Bezier curve, the order of which is higher than the second, lead to the irrational integrals, in which, under the radical, there is a polynomial of high even degree that has no roots. Therefore, it is indecomposable into the prime factors on the field of real numbers. The situation may become even more complicated in the case of self-intersecting curves.

An obvious solution to this problem is to reduce the dimension of the curves. The perimeter of a complex vector figure described by curves of the third or higher orders may be represented as the sum of curves of the lower order conjugate at singular points of constant curvature. We replace each individual segment of the "parent" curve of the third order with the second-order Bezier curves that have the property that their arc length may be generally calculated through the coordinates of the control points. The higher order case might be similarly reduced to this one. The selection criteria for the singular points, as well as the algorithm for approximate calculations of the lengths of arcs that make up the object under study, are the subject of this work.

BEZIER CURVES, THEIR ARC LENGTH, AND ORDER REDUCTION

A Bezier curve is a special case of the Bernstein polynomials [3, 4]. It is defined by a set of control points P_0 through P_n, where n is called its order. The first control point P_0 is always the *start* point, and the last one P_n is always the *end* of the curve. However, the intermediate control points (if any) generally do not lie on the curve itself.

For the flat cubic Bezier curve (1), four control points $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, and $P_3(x_3, y_3)$ are used. In them, P_1 and P_2 are the managing points (see Fig.1 (above)).

$$P(t)=(1-t)^3 P_0+3t(1-t)^2 P_1+3t^2(1-t) P_2+t^3 P_3, \Longleftrightarrow$$
$$\Longleftrightarrow \begin{cases} x(t)=(1-t)^3 x_0+3t(1-t)^2 x_1+3t^2(1-t) x_2+t^3 x_3 \\ y(t)=(1-t)^3 y_0+3t(1-t)^2 y_1+3t^2(1-t) y_2+t^3 y_3. \end{cases}, t\subset[0;1]. \qquad ()$$

The arc length l of a smooth parametrized curve is given by the integral (2) where integration and differentiation is carried out with respect to the parameter t. The arc length of the second-order Bezier curve might be easily expressed in terms of the control points coordinates.

$$l=\int_0^1 \sqrt{(x_t')^2+(y_t')^2}\,dt. \qquad ()$$

The general form of the second-order Bezier curve on the Cartesian plane is given by expression (3), where $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_3)$ are the control points, and P_1 is the only managing one. The general form (3) may be transformed to the polynomial with respect to the powers of t (4). Considering the notation (5), the explicit form of the integral (2) takes the form (6). We will not consider the degenerate cases when the integrand (2) is identically equal to zero.

$$P(t)=(1-t)^2 P_0+2(1-t)t P_1+t^2 P_2 \Longleftrightarrow$$
$$\Leftrightarrow \begin{cases} x(t)=(1-t)^2 x_0+2(1-t)t x_1+t^2 x_2, t\subset[0;1]. \\ y(t)=(1-t)^2 y_0+2(1-t)t y_1+t^2 y_2 \end{cases} \qquad ()$$

$$P(t)=t^2(P_0-2P_1+P_2)+t(-2P_0+2P_1)+P_0. \qquad ()$$

$$A=(P_0-2P_1+P_2); B=(-2P_0+2P_1); C=P_0; \Longrightarrow P(t)=At^2+Bt+C; P_t'=2At+B. \qquad ()$$

$$l=\int_0^1 \sqrt{4 ¿¿¿¿}. \qquad ()$$

Note that the square trinomial under the radical in (6) is certainly indecomposable into the simplest factors. Moreover, the discriminant of the resulting square trinomial with respect to the parameter t is strictly negative. Taking into account the introduced designations (7), integral (6) is reduced to the tabular form (8). Thus, we reduce the third order curves to equations of the second order.

$$a=4(A_x ¿¿ 2+A_y{}^2); b=4(A_x B_x+A_y B_y); c=B_x{}^2+B_y{}^2 ¿. \qquad ()$$

$$l=\int_0^1 \sqrt{at^2+bt+c}\,dt \equiv \int \sqrt{X}\,dt,$$
$$\int \sqrt{X}\,dt=\frac{(2at+b)\sqrt{X}}{4a}+\frac{1}{2K\sqrt{a}} Arsh\left(\frac{2at+b}{\sqrt{D}}\right)+Const, \qquad ()$$
$$X=at^2+bt+c; D=4ac-b^2; K=\frac{4a}{D}; Arsh\,Z=\ln\left(Z+\sqrt{Z^2+1}\right).$$

SINGULAR POINTS AND THE ITERATIVE ALGORITHM

Bezier curves, the order of which is higher than the second (see Fig.1 (above)), in relation to the curves of the second order (see Fig.1(b, below)), may have some geometric features. They may intersect, may be self-closed (see Fig.1(a, right)), and may have a sharp peak point (the "cusp" or "return", see Fig.1(a, left)). The presence of such features complicates the application of numerical methods to calculate the arc length of a curve.

On the other hand, according to *de Casteljau*'s algorithm, any “parent” Bezier curve might be split into “child” fragments with respect to the curve parameter. At the same time, the set of “child” curves of the same order as the “parent” one is exactly corresponds to the “parent” curve (see Fig.1(below)).

Since the nature of the partition might be set completely arbitrarily within the range of the curve parameter variation, it seems reasonable to carry out the primary partition by "special" or “singular” points. Points of zero curvature should be taken as such singular points, because they are not tied to the coordinate system, in contrast to the seemingly obvious maxima or minima. Points of zero curvature are geometrically defined as points of intersection of a curve with its own tangent, and, analytically, these are real solutions of the equation (9) within the range of variation of the curve parameter *t*.

$$x_t^{'} y_{tt}^{''} - x_{tt}^{''} y_t^{'} = 0, t \in [0;1]. \qquad ()$$

In the case of a third-order Bezier curve, equation (9) is reduced to a quadratic equation for the curve parameter. The number of real roots within the range of variation of the parameter will determine the geometric features of the curve, namely: there are no roots, which means that the curve has either an arched or self-intersecting form; the presence of one non-multiple root indicates an S-shaped curve, *i.e.*, the second root is outside the range of the parameter; two different roots determine the presence of a smooth peak; one multiple root indicates the presence of a sharp peak (cusp).

After splitting the curve by the roots of equation (9), each “child” curve is considered independent. However, each of the “child” curves is now devoid of geometric features, with the possible exception of self-intersection. Further considerations concern precisely such “child” curves, separately.

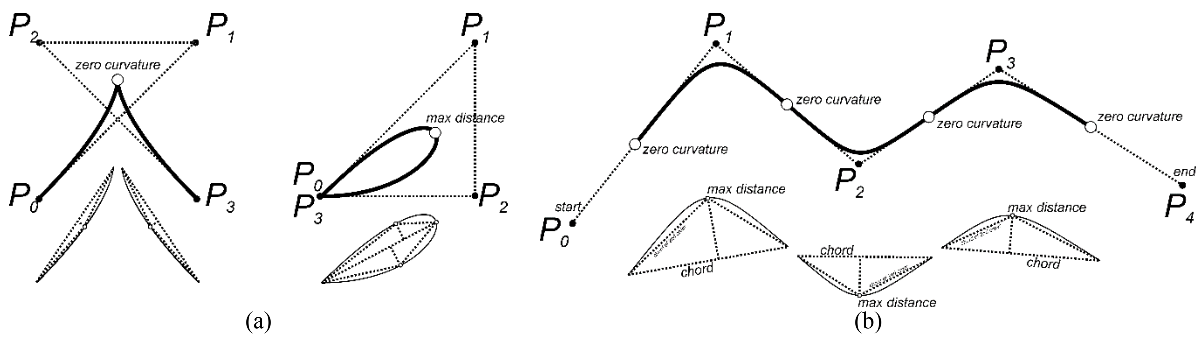

FIGURE 1. (a) Third-order Bezier curves special cases “cusp” (right) and “egg” (left), dotted lines (above) are auxiliary elements constructed at the Bezier control points; their division on arcs and sub-arcs with corresponding chords (below); (b) S-shape case of the fourth-order Bezier curve (above) and its division by the singular points on the second-order arcs and sub-arcs with corresponding chords (below).

Thus, the calculation of the third-order Bezier curve arc length is based on two principles. Firstly, the curve is splitted into two parts, and, secondly, the arc segment of the third order curve is replaced according to a certain rule by the second order arc segment. Obviously, to improve the accuracy of calculations, the procedure for splitting the curve into parts might be repeated. This forms an iterative procedure suitable for determining the arc length of a curve of any order.

Let us consider how, from the point of view of the problem of finding the arc length, the division of the curve into two parts might be carried out in an optimal way. The division should be carried out at such a value of the curve parameter *t*, which corresponds to the situation when the tangent to the curve is parallel to the contracting chord. This is the point where the curve is farthest from the chord (see Fig.1 (b, below), “max distance” points). This is the point of extreme curvature because it is the maximum with respect to the contracting chord. The curve parameter corresponding to the condition of parallelism of the tangent to the curve and the contracting chord is determined from condition (10), which, in order to avoid ambiguity, should be written as (11), where $E = P_3 - P_0$.

$$\frac{y_t^{'}}{x_t^{'}} = \frac{y_3 - y_0}{x_3 - x_0} \qquad ()$$

$$y_t^{'} \cdot E_x = x_t^{'} \cdot E_y \qquad ()$$

Expression (11) leads to a quadratic equation for the parameter of the curve with coefficients at powers of t (12). It necessarily has only one real root in the range of variation of the curve parameter, since the point of maximum distance of the arc from the contracting chord, provided that the oriented curvature of the arc does not change sign, is unique. The found root is used to split the third-order curve into two parts.

$$\begin{array}{ll} t^2: & 3\left(E_x A_y - A_x E_y\right) \\ t^1: & 2\left(E_x B_y - B_x E_y\right). \\ t^0: & E_x C_y - C_x E_y \end{array} \qquad ()$$

As a computational assessment of the proposed method, we engage data from [5] where the perimeters of the “egg” and “cusp”, as they are shown in Fig.1(a), have been calculated by the elliptic integrals. We calculate the total arc lengths of the same objects and compare the results for different number of splits N (see Table 1). We also include the chords approximations. As it is seen, from five to seven splits is enough to meet the same results that the far more complicated method of the elliptic integrals gives. It corresponds to from 32 to 128 split segments on each curve. We also see that the more straight “cusp” sample is well approximated even by chords when split on 64 sub-arcs. At the same time, the more orbed “egg” does not applicable to the chord approximation, at least with satisfactory accuracy. Thus, we may deduce that the proposed iterative algorithm works perfectly. The accuracy of the method might be tuned depending on the situation and the purposes of the calculations.

Table 1. Perimeter (total arc length) for the model objects, relative units; coincide values are in bold.

No of Splits (N)	No. of segments	“Egg” chord approx.	“Egg” elliptic [5]	“Egg” parabolic	“Cusp” chord approx.	“Cusp” elliptic [5]	“Cusp” parabolic
1	2	16,7705	18.3557	18,5426	18,2025	18.2843	18,3796
2	4	17,6606	18.3557	18,3669	18,2604	18.2843	18,3066
3	8	18,1540	18.3557	18,3549	18,2776	18.2843	18,2895
4	16	18,3072	18.3557	18,3556	18,2824	18.2843	18,2855
5	32	18,3435	18.3557	**18,3557**	18,2838	18.2843	18,2846
6	64	18,3526	18.3557	18,3557	18,2842	18.2843	18,2844
7	128	18,3549	18.3557	18,3557	18,2842	18.2843	**18,2843**

CONCLUSION

The perimeter of an arbitrary flat figure described by Bezier curves of order higher than the second order in some cases cannot be determined analytically. Therefore, a quantitative assessment of the spatial metrics of a significant part of information stored in the form of electronic documents is impossible, since, for example, fonts are described by third-order Bezier curves. Since the analytical and computational methods for quadratic Bezier curves are well developed, we propose to solve the problem described above by splitting the high-order curves into arcs and sub-arcs, which are well described by second-order curves, followed by summation. The proposed iterative algorithm solves almost all possible combinations of curves found in electronic documents. The only possible exception is a self-intersecting curve. The method allows to increase the accuracy of calculations by increasing the number of partitions. In the limit, the length of a sub-arc can be approximated by its chord. In particular, it is reasonable to use this when evaluating objects that are indistinguishable by the human eye (about 100 µm, [6]).

REFERENCES

1. G. Amir, H. Murtaza, “Big data concepts, methods and analytics”. International Journal of Information Management, 2015, 35, p.140.
2. M. Sarfraz, M.F.A. Razzak “An algorithm for automatic capturing of the font outlines”, Computers and Graphics, 2002, Vol.26, Iss.5, 795–804.
3. G. Farin “Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide”, Academic Press, 1997, 473p.
4. B. Casselman “From Bézier to Bernstein”, 2006, feature column from American Mathematical Society http://www.ams.org/publicoutreach/feature-column/fcarc-bezier#2
5. H. Henkel “Calculating the Cubic Bezier Arc Length by Elliptic Integrals”, 2014, derived from http://www.circuitwizard.de/metapost/arclength.pdf

6. D. Tarasov “Vision and reading”, Ekaterinburg: UrFU, 2015. 76p. (in Russian).