Generating Algorithm for High Order Curve Based on ACO

Zhi-Jie Li[1], Xiao-Dong Duan[2,] Bo Lu[3], Jian Yun[4]

[1]*School of Computer Science and Engineering, Dalian Minzu University, Dalian, China*
[2]*School of Computer Science and Engineering, Dalian, Minzu University, Dalian, China*
[3]*School of Computer Science and Engineering, Dalian Minzu University, Dalian, China*
[4]*School of Computer Science and Engineering, Dalian Minzu University, Dalian, China*
[1] *lizhijie@dlnu.edu.cn*

Abstract: A generating algorithm for parametric curve based on ant colony optimization is proposed in this paper. In order to produce the maximum step efficiently**,** the ant colony optimization is used to search the optimal step of curve in forward. Based on ant colony optimization, curve generating algorithm can search for the optimal step of parametric curve with any order quickly. Comparison of optimal step value between the proposed algorithm and other methods is also performed. Simulation results show the effectiveness of the proposed method. Since there is no restriction on control point position and curve order, the algorithm can be extended easily to other parametric curves besides Bézier curves.

1. Introduction

Parameter curve1 is the most common one in the research area of computer graphics and CAD. Up to now, raster graphics display monitor is widely used. Raster display can be viewed as a pixel matrix. So, any curve on the raster display is actually a collection of pixels. As the number and size of pixels is limited, the raster curve is only similar to the actual curve. How to make raster curve approximate to the real curve fast and perfectly is a problem to be deeply studied. The non-regularity of curve direction makes the point-to-point drawing for parameter curve difficult. Currently, very effective generating algorithm has not been found. The existing studies about curve generating algorithms can be classified into two different types. One is the traditional algorithms that focus on the interpolation method. Disadvantage of this interpolation approach is that it is difficult to determine the size of the step. Moreover, operations of multiplication and division are used several times, which limits the algorithm efficiency. The other is focus on the derivative method. For example, literature 5 presents an integer algorithm for generating implicit function curve, but for high-order parametric curves, the generation becomes complex. Literature6 contributes a quick generating algorithm for parameter polynomial curve, which determine the maximum step size according to the upper bound of the derivative value. However, the step size produced by the quick algorithm still not achieves the best value. Literature 7 calculates the optimal step by deriving the least upper bound of the derivative value according to the extreme point. Yet, the optimal step size for curve with more than four orders is not considered. Since the parametric curves of high-order have more control points, the derivation of multiple extreme points become very complex.

Based on the above considerations, a simple and very effective search algorithm using ant colony optimization is proposed. The concept of ant colony optimization (ACO), inspired by the behavior of real ants, was initially introduced by Colorni, Dorigo and Maniezzo. The principles of the algorithm can be illustrated by examining the food-searching process of an ant colony. Along their way from the food source to the nest, ants communicate with one another by means of pheromone. As the ants move, a certain amount of pheromone is deposited on their path and the pheromone gradually evaporates. The ants, then, determine their movements by judging the pheromone density on a path. This process can be described as a loop of positive information feedback, in which the more the ants follow a given trail, the more the pheromone is left, and the higher the probability that this trail will be followed by other ants. This selection process is the result of the ant’s self-catalyzing activities. Ant colony optimization is capable to solve many complex problems, which require sophisticated planning. Hence, utilizing ant colony optimization the optimal step size for generating parameter curve with arbitrary order can be solved in a short time. Not only the continuity of the curve is guaranteed, but also unnecessary computation is reduced.

2. Step Solving in Curve Generating

Currently, the basic method for drawing parametric curves effectively is as follows. Because the coordinates of parametric curves is determined by the parametric equation, a forward extent of parameter (i.e. step size *1/n*) is supposed to be determined first. Then, the approximate pixels are calculated and drawn according to the parameter equation of curve and forward step. Many pixels together form a curve. In general, the quick criteria to determine step is that: selecting the largest possible step size under the condition that the forward extend not exceed one pixel. Many algorithms estimate n value from the curve length. The specific method is that let n = max (n_x, n_y), n_x and n_y is the n value corresponding to x-axis and y-axis. The calculation equation is as follows.

$$n_x = \sqrt{2} \cdot m \cdot \max_{0 \le i \le m-1} \left| X_{i+1} - X_i \right| \tag{1}$$

$$n_y = \sqrt{2} \cdot m \cdot \max_{0 \le i \le m-1} \left| Y_{i+1} - Y_i \right| \tag{2}$$

Where m is order of curve, (X_i, Y_i) is the coordinate of the i-th control points. However, the n value calculated by this traditional method is too large, which lead to the points too dense when drawing. Therefore, literature[6] gives a smaller n value calculation method. It's basic idea is let parametric curve equation as

$$x = f(t), \quad y = g(t), \quad t \in [0,1] \tag{3}$$

The units of x and y is pixels. Then the best value n = max (n_x, n_y), where n_x, n_y is given by

$$n_x = \max_{0 \le t \le 1}(|f'(0)|, |f'(1)|, |f'(t)|), t \in \{t \mid f'(t) = 0\} \tag{4}$$

$$n_y = \max_{0 \le t \le 1}(|g'(0)|, |g'(1)|, |g'(t)|), t \in \{t \mid g'(t) = 0\} \tag{5}$$

Considering the above mentioned method, literature 7 describes the derivation of the optimal value of n based on cubic Bézier curve. Also, the calculation method for the optimal n of quartic curve is given. But for curve more than four orders, there exists no effective method to solve the optimal n. For more than four orders, derivative becomes more complex due to the increased control points. In the range of argument t, more extreme points often appear. The increasing order of curve makes the derivation to obtain extreme become more and more difficult. Especially for arbitrary order Bézier curve, an effective way to solve the extreme of derivative function is needed. Ant Colony Optimization (ACO) is suitable for solving such non-linear and multi-peak optimization of complex issues. The advantage of ACO is that it can deal with some examples that traditional methods can't handle. The following will details how to use ACO for solving the optimal step-size.

3. Optimal Step Based on ACO

3.1 Basic ACO Algorithm

We describe the basic framework of ant colony algorithm taking TSP (Traveling Salesman Problem) as an example. Suppose there are l cities and m ants. d_{ij} (i,j=1,2,…,l) denotes the distance between city i and city j. The city transition is probabilistic. The probability of ant k transferring from city i to city j at time t is given by:

$$P_{ij}^k(t) = \begin{cases} \tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta} \Big/ \sum\limits_{r \in allowed_k} (\tau_{ir}^{\alpha}(t)\eta_{ir}^{\beta}), & j \in allowed_k \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Where τ_{ij}(t) denotes the intensity of pheromone between city i and city j at time t. Pheromone is used to simulated the real ant secretions. In the motion process, ant k determines the next path according to the pheromone on each path. η_{ij}^{β} is the expectation degree and can be set as $1/d_{ij}$. α and β are control parameters. By *allowed*$_k$ we present the set of accessible city for ant k. The amount of information τ_{ij}(t) will gradually decay over time. 1-ρ is used to denote the pheromone evaporation. A cycle of ACO algorithm is completed when every ant has constructed a solution. At the end of each cycle, the intensity of pheromone is updated by a pheromone trail updating rule:

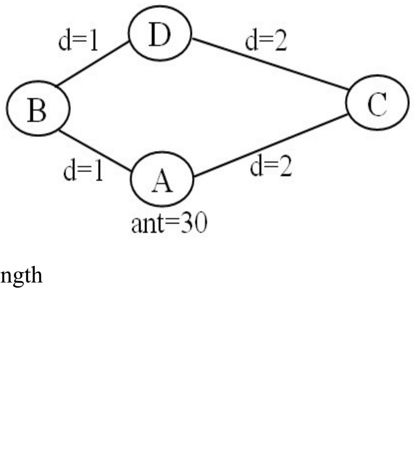
$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k \tag{7}$$

Where $\Delta\tau_{ij}^k$ is the amount of pheromone laid between city i and j by the kth ant and could be given by:

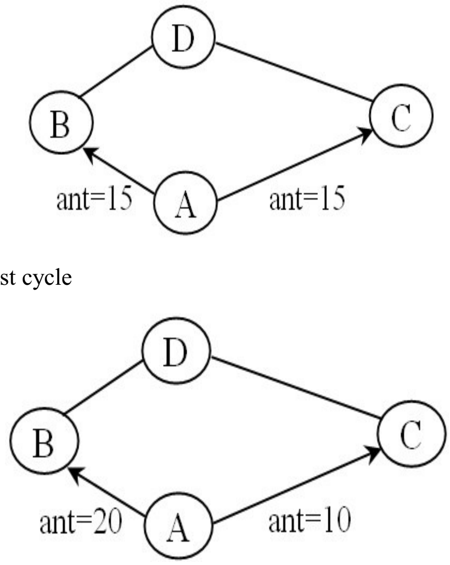
$$\Delta\tau_{ij}^k = \begin{cases} Q / L_k, & \text{if city } i \text{ and } j \text{ is traversed by the} k\text{th ant} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Where L_k is the length of the walking path by the kth ant in this cycle and Q is a constant. The end of the algorithm could be achieved by a predefined number of cycles, or the maximal number of cycles between two improvements of the global best solution.

The following specific example demonstrates the working principle of ant colony optimization (shown as Fig. 1).



(a) Path length



(b) First cycle

(c) Second cycle

Fig. 1. Illustration of ant colony optimization

Suppose node A and D are nest and food source, respectively. Path ABD and ACD is different in the path length. At beginning, path AB and AC have the same intensity of pheromone. That means the probability of each ant transferring from A to B and from A to C is the same. So, there are 15 ants select path AB, and 15 ants select path AC. After the first cycle, all ants return to nest A. According to equation (7) and (8), the intensity of pheromone of AB and AC is different, and the ratio of A and B is about 2:1. According to equation (6), in the second cycle, ants will move from A to B with probability 2/3 and from A to C with probability 1/3. Therefore, ants will prefer B with increasing probability than C. Finally, all ants select the short path ABD.

3.2 Continuous ACO for Optimal Step

The technology of ant colony optimization is initially used to solve discrete network routing problem. Since each ant always has limited choices at each stage, ant colony optimization is applicable to the discrete optimization problem, but not to optimization problem of continuous space, such as linear and nonlinear programming. We improve the search strategy of ant colony algorithm based on literatures 11. The solution space is divided into several sub-domains. During each iteration, ant colony algorithm first solves the possible sub-domain that contains optimal solution according to pheromone, then root around within this sub-domain to determine the optimal solution.

In the ant colony optimization, the region of each ant position corresponds to a region of possible optimal solution. For the extremum problem of derivative curve $|f'(t)|$, the representation of ant is relative simple, namely, one-dimensional vector $X_i = (x_i)$, where element x_i denotes an equal division region in the solution interval [0,1]. During each iteration, the region with larger objective function value has higher intensity of pheromone, which attracting more ants moving towards it. After positioning an optimal region x_i, ant colony algorithm will search the parameter t within region x_i for the optimal value $|f'(t)|$.

Suppose there are l regions and m ants. Let S be the region set $\{1,2,...l\}$. For continuous ant colony optimization, the probability of ant k transferring from region i to region j at time t is given by:

$$P_{ij}^k(t) = \begin{cases} \tau_j^\alpha(t)\eta_{ij}^\beta \Big/ \sum\limits_{r\in S, r\neq i}(\tau_r^\alpha(t)\eta_{ir}^\beta), & \eta_{ij} > 0 \\ 0, & \eta_{ij} \le 0 \end{cases} \quad i,j \in S,\ i \neq j \tag{9}$$

Where τ_j(t) denotes the pheromone intensity of region j at time t. η_{ij} is defined as $\eta_{ij} = f_{j\max} - f_{i\max}$, namely, the difference of max objective function value between region i and j. At the end of each cycle, the intensity of pheromone is updated by an updating rule:

$$\tau_j(t+1) = \rho \cdot \tau_j(t) + \sum_{k=1}^{m} \Delta\tau_j^k \tag{10}$$

Where the definition of $\Delta\tau_j^k$ is given by $\Delta\tau_j^k = Qf_{j\max}^k$.

4. Experiments

4.1 Configuration

Suitable algorithm parameters are determined based on experience. α and β control the relative importance of intensity of pheromone versus visibility. Therefore, as a result of high values for α, ants tend to choose regions previously have been chosen by other ants, where low values of α make the algorithm similar to stochastic algorithms. Parameters of our optimal algorithm that will be estimated through experiments are listed in Table I.

Table1. Experimental parameters for ACO based algorithm

Number	Parameter	Value
1	m	40
2	l	10
3	α	1
4	β	1
5	ρ	0.7
6	Q	1

We use the Bézier curve to verify the effectiveness of proposed ACO algorithm for solving the optimal curve step. Bézier curve is a kind of parametric polynomial curves. It uses Bernstein polynomial as basis function, which makes it easy to represented and possess many beautiful properties. The definition of Bézier curve is as follows:

Let P_0, P_1, P_2,, P_n be the control points of curve with n-order, then the Bézier curve is defined as

$$P(t)=\sum_{i=0}^{n}P_i\cdot B_{i,n}(t),\quad t\in[0,1] \tag{11}$$

Where $B_{i,n}(t)=C_n^i\cdot(1-t)^{n-i}\cdot t^i$ is the Bernstein basis function. Bézier curve is exactly the linear combination of Bernstein basis and control points. The form of Bézier curve is simple and intuitive, and has geometric invariance, symmetry and many other beautiful natures. Equation (11) is called Bernstein-Bézier curve defined by control polynomial P_0, P_1, P_n. It can also be referred to as Bézier curve. For the two-dimensional Bézier function, the parametric function is usually given by

$$\begin{cases} x=f(t)=\sum_{i=0}^{n}X_i\cdot C_n^i\cdot(1-t)^{n-i}\cdot t^i \\ y=g(t)=\sum_{i=0}^{n}Y_i\cdot C_n^i\cdot(1-t)^{n-i}\cdot t^i \end{cases}\quad t\in[0,1] \tag{12}$$

Where (X_0,Y_0), (X_1,Y_1),…, (X_n,Y_n) is the coordinates the of control points. For example, the Bézier curve in Fig. 2 is drawn by a polynomial with 12 control points. Any change of control points will result the shape change of the curve.

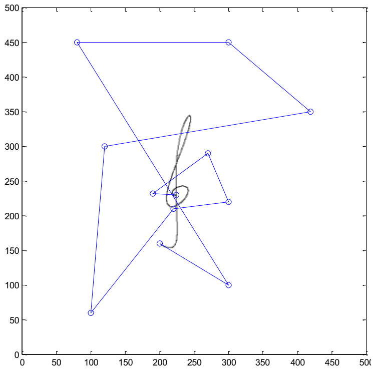


Fig. 2. Bézier curve with 11-order and its control points

In this paper, function of Bézier curve with order from 2 to 10 will be discussed, as well as derivatives of x-axis and y-axis. Also, the comparison results of optimal step between ACO and existing algorithms are summarized. Since the parameter function of x-axis and y-axis is the same except the coefficient, only the parameter functions of x-axis are listed below.

4.2 Results

In this section, the effectiveness of the suggested algorithm is validated using experimental data. The Bézier curve with order from 2 to 10 are generated on the basis of the continuous ACO algorithm. The first raw shows the Bézier curve using ACO algorithm. Second raw shows the x-axis derivative curve $|f'(t)|$ .Third raw shows the y-axis derivative curve $|g'(t)|$. The maximum of absolute derivative is solved as value n based on the derivatives of x-axis and y-axis. Then, the corresponding Bézier curve is drawn point-to-point taking the reciprocal of n as the step size. From Table II, we can see that Bézier curve shows a more complex graphics with the increase of order. Especially for more than four order, curve shape tends to be more irregular due to the increase of control points. In order to generate curve with least pixels and without breakpoints, the minimize upper bound of absolute curve derivative ($|f'(t)|$ or $|g'(t)|$) needs to be found. Therefore, the way that only using deduction method to obtain the extreme derivative becomes increasingly difficult with the rise of curve order. This paper utilizes ACO algorithm to search the extreme point of curve derivative. In this way, Bézier curve can be drawn without breakpoint, and achieve the maximize step size compared with other algorithms.

Table2. Bézier and its derivative curve with order from 8 to 10

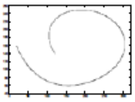
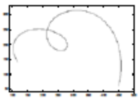
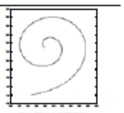
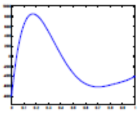
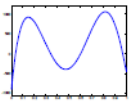
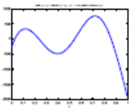
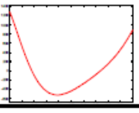
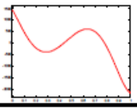
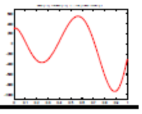
	8-order	9-order	10-order
Bézier			
x-axis derivative			
y-axis derivative			

Fig. 3 compares the values of n between the ACO algorithm and other algorithms based on the Bézier curves with order from 2 to 10. From the comparison results, we can see that the value of n in different algorithm presents a rising tendency with the increase of curve order. Except quadratic Bézier curves, the value of n using ACO algorithm is the smallest of the three methods. It is known that the value of n determines the forward step size that affects the cycle times. Therefore, the smaller the n value, the faster the algorithm. For a Bézier curve with determined order, just one search for n is needed using ACO algorithm. Therefore, ACO will not affect the speed of curve generating.

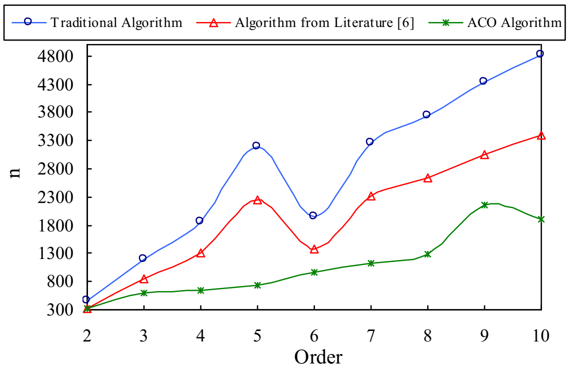


Fig. 3. Comparison of n using three algorithms

5. Conclusion

This paper presents a common algorithm for generating parametric curves on graphical display device. Ant colony optimization is utilized to search the extreme point of curve derivation. The maximum value of absolute derivative is used as the reciprocal of step to generate parametric curve. Searching the optimal step based on the proposed algorithm is simple and effective, without adding too much computation. As the Bézier curve is relatively stable and easily calculated, the algorithm is tested based on Bézier curve from low order to high order. Experimental results show that the algorithm could generate curve without break, and achieve the maximum step size in comparison with other methods. In addition, the algorithm is not limited by the location of control point and order of curve, and hence achieves good generality.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No.61370146, No.61040054, No.61602085, the Fundamental Research Funds for the Central Universities under Grant (No.DC201502030203, No.DC201502030403, No.DC201501030401), and the National Social Science Foundation of China under Grant No. 16BMZ010.

References

[1] F.H. Guo. Optimal Parameterization of Parametric Curves [J]. Journal of Computer-Aided Design & Computer Graphics, 2007, 19(4): 464-467.

[2] S.L. Lien, M. Shantz, V. Pratt. Adaptive forward differencing for rendering curves and surfaces [J]. Computer Graphics, 1987, 21(4): 111-118.

[3] S.L. Chang, M. Shantz, R. Rocchetti. Rendering cubic curves and surfaces with integer adaptive forward differencing [J]. Computer Graphics, 1989, 23(3): 157-166.

[4] R.V. Klassen. Integer forward differencing of cubic polynomials: Analysis and algorithms [J]. ACM Transactions on Graphics, 1991, 10(2): 152-181.

[5] Y.K. Liu, J.Y. Shi. An All-Integer Arithmetic Algorithm for Drawing Curves [J]. Chinese Journal of Computers, 1998, 21(3): 270-280.

[6] Y.D. Huang, G.Q. Zhu. A Fast Point-by-Point Generating Algorithm for Polynomial Parametric Curve [J]. Chinese Journal of Computers, 2000, 23(4): 393-397.

[7] Y.K. Liu, X.M. Zhou. Pixel-Level Double-Step Algorithm for Rasterizing Parametric Curve [J]. Journal of Computer-Aided Design & Computer Graphics. 2002, 14(7): 619-623.

[8] D.W. Crowder, A.R. Horowitz, P.J.D. Barro, et al. Mating behaviour, life history and adaptation to insecticides determine species exclusion between whiteflies. Journal of Animal Ecology 2010, 79(3): 563-570.

[9] D. Merkle, M. iddendorf, H. Schmeck. Ant Colony Optimization for Resource-Constrained Project Scheduling. IEEE Transactions on Evolutionary Computation. 2002, 6(4): 333-345.

[10] M. Dorigo, C. Blum. Ant colony optimization theory: A survey. Theoretical Computer Science. 2005, 344(2-3): 243-278.

[11] L. Cheng, J. Shen, L. Qin. A Method for Solving Optimization Problem in Continuous Space by Using Ant Colony Algorithm. Journal of Software, 2002, 13(12): 2317-2323.

[12] X.Q. Wu, X.L. Han. Extension of Cubic Bézier Curve [J]. Journal of Engineering Graphics, 2005, (6): 98-102.

[13] J. Chen, G.J. Wang. Optimal Parameterizations of the Degree 2 Rational Bézier Curves [J]. Journal of Computer Research and Development, 2008, 45(9): 1601-1604.