International Journal of Engineering & Technology, 7 (4.33) (2018) 93-97

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Optimizing the Shape Parameters of Beta-Spline Using Particle Swarm Optimization

Sharifatul Aniza Suliman, Normi Abdul Hadi*

Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA,40450 Shah Alam, Selangor, Malaysia
**Corresponding author E-mail: normi@tmsk.uitm.edu.my*

Abstract

Beta-spline is an alternative curve for 2D font representation. It is preferred since it has G^2 continuity and two shape parameters, that can be used to control the curve shape. These shape parameters can also be used to optimize the error between fitted curve and original data points. Commonly, most of the researcher use value of shape parameters of beta-spline as $\beta_1 = 1$ and $\beta_2 = 0$ or some of the researcher choose any random value of these two shape parameters that suitable to be used in beta-spline curve fitting. The values of shape parameters are very important since the values affect the total error of the fitted curves. Thus, in this paper, Particle Swarm Optimization (PSO) is employed to determine the optimum value of the two shape parameters that will optimize the approximation error of the fitted curve. The technique is applied on two fonts: ى and δ, and tested using various number of iterations and populations.

Keywords: *Beta-Spline; Curve Fitting; Particle Swarm Optimization; Shape Parameters; 2D Font Image.*

1. Introduction

In this sophisticated computer era, images such as fonts, symbol, and other images can be presented in digitized form using digitization process. Digitization is the process of capturing objects, sound, signals and then converting the information into digital format in terms of points or samples. The sample form for digitization of signal consists of values at a point in time and space and while the image form of 2D arrays of intensity value is the digital representation for objects. Digital images can be done in various processes such as enhancement, segmentation, extraction, and analysis [1].
Digital representation of objects and signals is very useful in this rapidly grow of development of computer [2]. Digitization has been broadly used in various areas such as medical, entertainment, industrial, military, security, scientific, manufacturing, civil, and business in aspects of documentation. From micro cell movement to the structure of galaxies can be studied with the current technology of digitization to get in better visual of objects. In medical technology, for example, which used for diagnosis and treatment, images can be generated in 2D, 3D, and 4D form to carry out the analysis.
Capturing image outlines is one of the steps in the digitization process of an image. Several mathematical and computational stages are involved in the whole process of capturing an image outlines. Curve modelling and designing are the most significant phase in capturing image outlines [3]. Fitting curves to image outlines have three phases which are extracting outlines of images, detecting corner points from detected outlines and curve fitting. Curve fitting is the process of fitting the curve that has the best fit of a series of data points which possibly subject to constraints. Curve fitting can involve either interpolation method or smoothing method. In interpolation, an exact fit curve to the data is required and for smoothing method, smooth function is constructed approximately fit to the data. Development in the area of curves and surfaces are entered into the area of Computer Aided Geometric Design (CAGD) [4].
Spline interpolation is the most common used in CAGD. In CAGD, spline interpolation is commonly used in evaluating the control points of the curve. Control points is used to determine the shape of a spline curve, surface and higher-dimensional object and it is one of the important item used in curve fitting. A good curve fitting is when number of knots and control points is produced minimally, but the accuracy and the continuity is maintained [5].
There are a lot of studies in reconstruction of 2D images especially font and character images. Non-Roman character digitization such as Chinese [6, 7] and Arabic [8, 9] are the common application of 2D images curve fitting that are most researcher prefer to do on their research. According to [10], this type of character known as cursive, complex and needs more attentions.
There are many techniques in performing curve fitting such as Bezier, B-spline, Non-Uniform Rational B-spline, and beta-spline. Most of researchers prefer to choose Bezier curve to be discussed, because it requires less number of control points compared to b-spline and beta-spline. However, beta-spline curve has advantages since it is guaranteed G^2 continuity [11], and consists of two shape parameters that can be manipulated without decreasing the continuity degree.

Beta-spline has shape parameters, β_1 and β_2 to construct the shape curve and to optimize the error of the curve fitting. The study on determining the optimal value of shape parameters are limited. Thus, in this paper will use Particle Swarm Optimization (PSO) as an optimization technique to optimize the values of shape parameters of beta-spline and the sum squared distance of beta-spline curve. The PSO technique used in this study is the improvement of PSO which refer to [12]. After the values of shape parameters are obtained, cubic beta-spline is used to fit the data of 2D font image of ى and δ fonts. The sum squared distance when using values β_1 and β_2 obtained from the proposed method are then compared with the sum squared distance when using values $\beta_1 = 1$ and $\beta_2 = 0$.

The paper is arranged as follows. In section 2, the proposed PSO method is discussed. Section 3 shows the results obtained and the comparison of performance of shape parameters between using proposed method and conventional method. This paper ends with conclusion in Section 4.

2. Shape Parameters Optimization Using PSO

This section will discuss the process to optimize the beta-spline shape parameters using PSO, and the error calculation.

2.1. Cubic beta-spline curve

Beta-spline is one of the spline interpolation that gives the best curve nearest to data points. Beta-spline was introduced and developed by [13] which generalized from uniform cubic B-spline. Then, in [14] has extended this study by providing the bias parameter, β_1 and tension parameter, β_2 which is used for shape manipulation and also introduced geometric continuity in a way to preserve continuity of the control points. Beta-spline is built on G^2 continuity condition which is it guarantee the smoothness and accuracy of the fitted curve to the data points. Beta-spline curves can be controlled in three ways which are using weights (rational), control points and shape parameters.

The cubic beta-spline equation is given as,

$$F(t) = [T][M][V] \tag{1}$$

where

$[T]$ is the polynomial matrix, $[T] = [t^3 \quad t^2 \quad t \quad 1]$

$[M]$ is the beta-spline basis function matrix,

$$[M] = \frac{1}{\delta}\begin{bmatrix} -2\beta_1^3 & 2(\beta_2+\beta_1^3+\beta_1^2+\beta_1) & -2(\beta_2+\beta_1^2+\beta_1+1) & 2 \\ 6\beta_1^3 & 3(\beta_2+2\beta_1^3+2\beta_1^2) & 3(\beta_2+2\beta_1^2) & 0 \\ -6\beta_1^3 & 6(\beta_1^3-\beta_1) & 6\beta_1 & 0 \\ 2\beta_1^3 & \beta_2+4(\beta_1^2+\beta_1) & 2 & 0 \end{bmatrix}$$

with

and, $[V]$ is the control points matrix, $[V] = [V_1 \quad V_2 \quad V_3 \quad V_4]^T$

This paper will use sum squared distance (SSD) as objective function for PSO technique. To get the SSD value, control points are needed to be evaluated first using least squares method [15] as follows.

$$SSD_i = \left\| F(t_i) - P_i \right\|^2 \tag{2}$$

The distance, SSD_i is calculated between curve points, $F(t_i)$ to its corresponding data point, P_i.

2.2. Particle swarm optimization

Particle Swarm Optimization (PSO) was first introduced by [16]. The idea of the development of PSO comes from the observation of the biological behaviour of bird flock in searching for food. A group of bird randomly searching food in an area, but there is only one piece of food in that area. Every bird does not know where the food is, but they know how far the food is by iterate many times searching in that area. PSO learned from this scenario and used it to solve optimization problems. PSO in this paper are using the improvement PSO based on [12]. Figure 1 shows the pseudocode that will be used in this paper.

For each particle
Initialize position and velocity of particles.
End
For each iteration
For each particle
Step 1. Find average and minimum of fitness value of all particles in current iteration.
Step 2. Calculate inertia weight. If fitness value in current particle and current iteration less than average fitness value in current iteration, the inertia weight is as below,

$$\omega_i^{it} = \omega_{\min} + \frac{(\omega_{\max} - \omega_{\min})(f_i^{it} - f_{\min}^{it})}{f_{avg}^{it} - f_{\min}^{it}};$$

$$\omega_{\min} = 0.4, \omega_{\max} = 0.9$$

and if vice versa, the inertia weight is equal to $\omega_{\max}$.
Step 3. Calculate cognitive coefficient and social coefficient

$$c_1 = 0.5(\omega_i^{it} + 1)^2$$

$$c_2 = \min(4, 2(\omega_{\min} + 1)) - c_1 - 0.000001$$

Step 4. Calculate particle velocity according to equation below

$$v_i^{it+1} = \omega_i^{it} v_i^{it} + c_1 rand()(pbest_i^{it} - x_i^{it})$$
$$+ c_2 rand()(pgbest^{it} - x_i^{it})$$

Step 5. Update particle position according to equation below

$$x_i^{it+1} = x_i^{it} + v_i^{it+1}$$

Step 6. Evaluate the fitness value of every particle after velocity and position updating formula.
Step 7. If the fitness value is better than best fitness value in history, set current value as new best fitness value. Choose the particle with the best fitness value of all particles as the global best value.
End
End

Fig. 1: Pseudocode of PSO algorithm based on [12]

where

ω_i^{it} is the inertia weight of particle i at time it which controls the momentum of the particle,

$\omega_{\min}$ is the minimum value for the inertia weight range,

$\omega_{\max}$ is the maximum value for the inertia weight range,

f_i^{it} is the fitness function of particle i at time it,

$f_{\min}^{it}$ is the minimum fitness value of all particles at time it,

f_{avg}^{it} is the fitness value average of all particle at time it,

c_1 is the cognition factor that help in local distance searching,

c_2 is the social factor that help in global distance searching,

v_i^{it} is the velocity vector of particle i at time it,

x_i^{it} is the position vector of particle i at time it,

$rand()$ are the random numbers in the range of 0 to 1,

$pbest_i^{it}$ is the personal best position of particle i at time it, and

$pgbest^{it}$ is the global best position at time it.

it is the number of iteration

i is the number of particle

The process in Figure 1 is embedded into the main loop for beta-spline shape parameter optimization using PSO

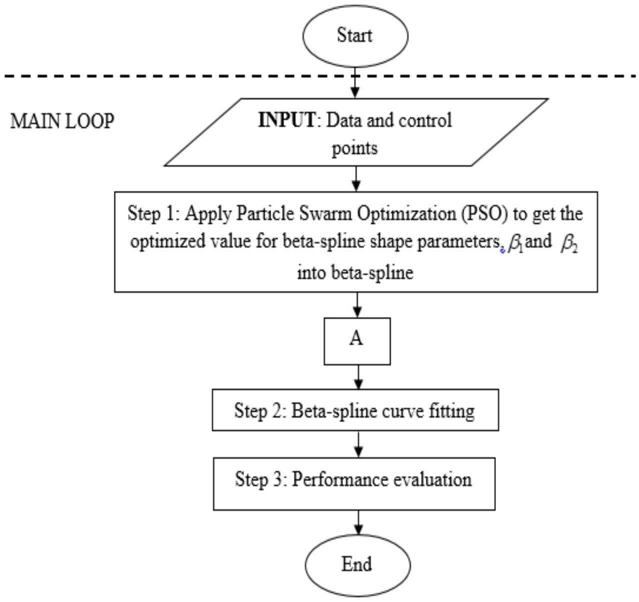

Fig. 2: Flowchart of beta-spline curve fitting with using PSO

In the flowchart, process 'A' is the PSO process as stated in Figure 1. Once the optimum values of shape parameters are obtained, beta-spline curve fitting process is done as shown in Step 2. Finally, the performance of the fitted curve is measured based on approximation error as stated in Step 3 where the approximation error when $\beta_1 = 1$ and $\beta_2 = 0$ is compared with the approximation error when shape parameters of beta-spline, β_1 and β_2 is optimized by PSO.

3. Results and Discussion

The developed method is applied on two fronts: ى and δ. These fonts are chosen due to its curvy edges. The range for β_1 and β_2 chose are $0.5 \le \beta_1 \le 100$ and $-2.8 \le \beta_2 \le 100$ respectively which satisfy convex hull properties. These range are manipulated later to analyse its effect on the optimized shape parameters values. The stopping criteria for this experiment is when the maximum iteration is achieved. For comparison purpose, the approximation errors using this method are compared with the approximation errors using the basic values of shape parameters, $\beta_1 = 1$ and $\beta_2 = 0$.

3.1. Font ى

The calculated shape parameters, and the corresponding SSD in given in Table 1.

Table 1: PSO Iterations for ى

Iterations	Best SSD	β_1	β_2
1	1052.49647062	2.26037661	32.59690583
20	201.01303034	0.99545086	0.61122668
40	200.96401596	0.99565410	0.55331729
49	200.96401589	0.99565529	0.55326597
⋮	⋮	⋮	⋮
100	200.96401589	0.99565529	0.55326597

Table 1 shows that the best SSD after 100 iterations is 200.96401590 accurate to eight decimal places. In the first iteration, the best SSD is too far from the final SSD. The best SSD values start to converge from 49th iteration, and so the shape parameters. Therefore, the optimized values of shape parameters for this case are $\beta_1 = 0.99565529$ and $\beta_2 = 0.55326597$. Basic shape parameters values give error 205.73029191. Thus, the optimized shape parameters values have improved the error by 2.32%. the fitted curves on font ى is shown in Figure 3.

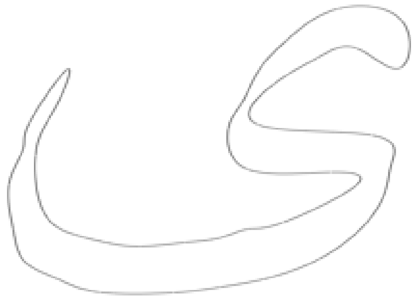

Fig. 3: Fitted curves on ى

3.2. Font δ

As done in Section 3.1, 100 iterations are applied in optimizing the shape parameters values for δ. The calculated best SSD and shape parameters are shown in Table 2.

Table 2: PSO Iterations for δ

Iterations	Best SSD	β_1	β_2
1	4419.91960698	3.92146606	59.52839866
20	354.41959172	1.00025457	-0.14897484
40	354.37501387	0.99998095	-0.16980013
44	354.37501260	0.99998171	-0.16968763
⋮	⋮	⋮	⋮
100	354.37501260	0.99998171	-0.16968763

The approximation error when $\beta_1 = 1$ and $\beta_2 = 0$ is 357.46727520. From Table 2, the best error is 354.37501261 which started to converge from 44th iteration. Thus, the error has been improved by 0.87%. The optimized shape parameters values are $\beta_1 = 0.99998172$ and $\beta_2 = -0.16968764$. The fitted curves on δ is shown in Figure 4.

Fig. 4: Fitted curves on δ

3.3. Manipulation of Shape Parameter Range

As mentioned before, the range of shape parameters can be manipulated if it satisfies the convex hull properties. Some of the tested range are shown in Table 3 and Table 4. The number of iterations and particles are 100 and 30 respectively.

Table 3: Different range of β_1 and β_2 for ى

Range of β_1 and β_2	Iter	Best SSD	β_1	β_2
$0.001 \le \beta_1 \le 100$	1	674.51792872	1.85396824	50.78017934
$-0.004 \le \beta_2 \le 100$	100	200.96401589	0.99565528	0.55326589
$0.5 \le \beta_1 \le 100$	1	1052.49647062	2.26037661	32.59690583
$-2.8 \le \beta_2 \le 100$	100	200.96401589	0.99565528	0.55326597
$1 \le \beta_1 \le 100$	1	3297.94999171	10.61604634	80.93344545
$-8 \le \beta_2 \le 100$	100	201.21977946	1.00000000	0.56638686
$5 \le \beta_1 \le 100$	1	2867.09577925	7.05208725	75.04043988
$-93.5 \le \beta_2 \le 100$	100	2173.06945438	5.00000000	100.0000000

Table 4: Difference range of β_1 and β_2 for δ

Range of β_1 and β_2	Iter	Best SSD	β_1	β_2
$0.001 \le \beta_1 \le 100$	1	2203.69081074	1.25004079	67.00774058
$-0.004 \le \beta_2 \le 100$	100	357.30257349	1.00034755	-0.00400000
$0.5 \le \beta_1 \le 100$	1	4419.91960698	3.92146606	59.52839866
$-2.8 \le \beta_2 \le 100$	100	354.37501260	0.99998171	-0.16968763
$1 \le \beta_1 \le 100$	1	3616.44168100	3.04605397	3.89368384
$-8 \le \beta_2 \le 100$	100	354.37503371	1.00000000	-0.16964698
$5 \le \beta_1 \le 100$	1	6492.79305427	6.47786641	69.09894408
$-93.5 \le \beta_2 \le 100$	100	5053.07054065	5.00000000	100.0000000

From Table 3, the first two range gives the best SSD for ى. However, the first range does not suitable for δ as shown in Table 4. It explains the reason of choosing $0.5 \le \beta_1 \le 100$ and $-2.8 \le \beta_2 \le 100$ in this paper. For the last range where $5 \le \beta_1 \le 100$ and $-93.5 \le \beta_2 \le 100$, it is not suitable for both fonts. This is because β_1 is the bias shape parameter. The increment of β_1 value will attract the curve to the other side of its control polygon. If this range is chosen, more number of iterations might be needed.

4. Conclusion

PSO technique is implemented to automatically calculate the optimum sum squared distance and shape parameters of beta-spline, β_1 and β_2. Previously, the values of these parameters are user-defined, and no specific guideline is given. These values are very significant to be determined since it affects the accuracy of the fitted curve in terms of SSD. Therefore, minimizing SSD is decided as the objective of this PSO. The range of β_1 and β_2 is decided based on to the convex hull properties of beta-spline curve. Several ranges of β_1 and β_2 are tested to get the most suitable one. From the result, it is shown that the calculated shape parameters can improve the approximation error (SSD) of the fitted curves. From the β_1 values, it can be concluded that the calculated values do not vary too much, although the maximum value in the range is 100. This is because the increment of β_1 value will attract the curve away from its original position and may reduce the accuracy. Based on analysis, the suggested range of shape parameters is $0.5 \le \beta_1 \le 100$ and $-2.8 \le \beta_2 \le 100$ with 30 number of particles, and 100 iterations. Thus, the optimum value of shape parameters of beta-spline, β_1 and β_2 for ى image is $\beta_1 = 0.99565528$ and $\beta_2 = 0.55326597$; for δ image is $\beta_1 = 0.99998171$ and $\beta_2 = -0.16968763$. The accuracy for both fonts are improved by 2.32% and 0.87% respectively. This improvement can be increased if higher number of iterations and populations are considered.

Acknowledgement

This study is supported by Universiti Teknologi MARA (UiTM) under LESTARI grant.

References

[1] Hadi NA (2015), G² parametric curve and surface fitting using beta-spline. Universiti Teknologi MARA.

[2] Koundinya GG, Jaikumar G, Akash, NR & Subramanian MSV (2012), Survey on digital image processing in sports. *Research Journal of Applied Sciences, Engineering and Technology* 4(24), 5552-5556.

[3] Sarfraz M, Irshad M & Hussain MZ (2012), Vectorization of image outlines using rational spline and genetic algorithm. *International Proceedings of Computer Science and Information Technology* 50, 16-20.

[4] Farin G (2014), *Curves and surfaces for computer-aided geometric design: A practical guide*. Elsevier.

[5] Park H & Lee JH (2007), B-spline curve fitting based on adaptive curve refinement using dominant points. *Computer-Aided Design* 39, 439-451.

[6] Liao CW & Huang JS (1991), Font generation by beta-spline curve. *Computers and Graphics* 15, 527-534.

[7] Yang HM, Lu JJ & Lee HJ (2001), A Bezier curve-based approach to shape description for Chinese calligraphy characters. *Proceedings of the International Conference* on *Document Analysis and Recognition*, pp. 276-280.

[8] Masood A & Sarfraz M (2008), An efficient technique for capturing 2D objects. *Computers and Graphics* 32, 93-104.

[9] Yahya F, Ali JM, Majid AA & Ibrahim A (2018), Automatic G 1 surface reconstruction from serial cross-sectional images. *Proceedings of the International Conference on Advances in Visual Information Systems* pp. 96-99.

[10] Sarfraz M & Khan M (2004), An automatic algorithm for approximating boundary of bitmap characters. *Future Generation Computer Systems* 20, 1327-1336.

[11] Hadi NA, Ibrahim A, Yahya F & Ali JM (2013), A comparative study on cubic bezier and beta-spline curves. *Matematika* 29, 55-64.

[12] Yang Q, Tian J & Si W (2017), An improved particle swarm optimization based on difference equation analysis. *Journal of Difference Equations and Applications* 23, 135-152.

[13] Barsky BA (1981), The beta-spline: A local representation based on shape parameters and fundamental geometric measures. Phd thesis, University of Utah.

[14] Barsky BA and Beatty JC (1983), Local control of bias and tension in beta-splines. *Proceedings of the ACM SIGGRAPH Computer Graphics* pp. 193-218.

[15] Mazlan NT (2017), Cubic beta-spline curve fitting using fuzzy interpolation. .

[16] Eberhart R & Kennedy J (1995), A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp.* 39-43.