

[Egyptian Informatics Journal (2015) 16, 175–185](http://dx.doi.org/10.1016/j.eij.2015.05.001)

Cairo University

Egyptian Informatics Journal

[www.elsevier.com/locate/eij](http://www.elsevier.com/locate/eij) [www.sciencedirect.com](http://www.sciencedirect.com/science/journal/11108665)

ORIGINAL ARTICLE

Data fitting by G1 rational cubic Be´zier curves using harmony search



Najihah Mohamed [a](#_bookmark0),[\*](#_bookmark2), Ahmad Abd Majid [b](#_bookmark1), Abd Rahni Mt Piah [b](#_bookmark1)

a *Faculty of Industrial Sciences & Technology, Universiti Malaysia Pahang, Lebuhraya Tun Razak, 26300 Kuantan,*

*Pahang, Malaysia*

b *School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Pulau Pinang, Malaysia*

Received 3 December 2014; revised 23 March 2015; accepted 14 May 2015

Available online 6 June 2015

Abstract A metaheuristic algorithm, called Harmony Search (HS) is implemented for data fitting by rational cubic Be´zier curves. HS is a derivative-free real parameter optimization algorithm, and draws an inspiration from the musical improvisation process of searching for a perfect state of har- mony. HS is suitable for multivariate non-linear optimization problem. It is mainly achieved by data fitting using rational cubic Be´zier curves with *G*1 continuity for every joint of segments of the whole data sets. This approach has significant contributions in making the technique auto- mated. HS is used to optimize positions of middle points and values of the shape parameters. Test outline images and comparative experimental analysis are presented to show effectiveness and robustness of the proposed method. Statistical testing between HS and two other different metaheuristic algorithms is used in the analysis on several outline images. All of the algorithms improvised a near optimal solution but the result that is obtained by the HS is better than the results of the other two algorithms.

© 2015 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.](http://creativecommons/)

KEYWORDS

Rational cubic Be´zier; Data approximation; Harmony search

org/licenses/by-nc-nd/4.0/).

1. Introduction

Data fitting is a well-studied area in computer graphics and mathematics which is also a fundamental problem in many fields, such as computer graphics, image processing, shape

\* Corresponding author.

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

**Production and hosting by Elsevier**

modelling and data mining. Depending on applications, differ- ent types of curves such as parametric curves, implicit curves and subdivision curves are used for fitting. Data fitting is nor- mally divided into two types, approximation and interpola- tion. Under an approximation-fitting scheme, a curve must pass reasonably close to the data points but is not required to pass through them [[12]](#_bookmark23).

Rational Be´zier curves are widely used in CAD/CAGD fields, because they have concise and geometrically meaningful presentation and can be deformed easily by moving the control points or modifying weights. Some studies on data fitting using rational Be´zier functions, to determine the best conic approxi- mation of a given curve which is based on Hausdorff distance

<http://dx.doi.org/10.1016/j.eij.2015.05.001>

1110-8665 © 2015 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-nc-nd/4.0/).](http://creativecommons.org/licenses/by-nc-nd/4.0/))

176 N. Mohamed et al.

curves through the concept of *C*(*u*;*v*) – continuity [[1]](#_bookmark20) and as iter- function [[7]](#_bookmark23), approximate rational Be´zier curves by Be´zier ation method for approximation of rational Be´zier curves by

adjusting control points gradually using the scheme of weighted progressive iteration approximations through a global *Lp* – error [[9]](#_bookmark23). Recently, a few researchers such as Huang et al. [[6]](#_bookmark23) whose derived offset by using cubic Be´zier for approximating degree *n* Be´zier with comparing three methods, Hausdorff dis- tance, shifting control and approximation based on *L*2 norm in order to find the better approximation. While Yang et al. [[21]](#_bookmark24) focused on curves on surfaces which present a parabola approximation method based on the cubic rational Be´zier sur- faces. This study also used Hausdorff distance between the approximate curve and the exact curve; the approximation is controlled under the user-specified tolerance. Shen et al. [[17]](#_bookmark24) proposed a certified approximation as an optimization method to select proper weights in the cubic rational Be´zier curve to approximate the given curve. The error of the approximation is controlled by the size of its tetrahedron, which converges to zero by subdividing the curve segments. Stamati and Fudos [[20]](#_bookmark24) presented a fast curve approximation method that approximates raw data with cubic rational Be´zier curves. The

This paper begins with an overview of rational cubic Be´zier, least-squares error and parameterization used based on cen- tripetal method together with some basic concepts on data fit- ting. A gentle overview on the HS is also given. The *G*1 continuity concept between two segments of our proposed data set is presented. Finally, method and its implementation, with some experimental results are presented. This method also had been compared with other two metaheuristic algo- rithms, which are genetic algorithm and particle swarm opti- mization on four different outline images. Statistical analysis also had been carried out in this paper to identify the reliability and effectiveness of this method.

1. Data fitting with rational cubic Be´zier

A rational cubic Be´zier function is defined as:

Let {(*si*; *Qi* ); *i* = 1; 2; ·· · ; *n*} be a given set of data point where *s*1 < *s*2 < ··· < *sn*. The piecewise rational cubic Be´zier function is defined over each interval *I* = [*si*; *si*+1]; *i* = 1; 2; ··· ; *n* — 1.

*P*(*s*)Ξ *P*(*si*)

1

1

approach combines least squares approximation with continu-

ity constraints to ensure *G*1 continuity between neighbouring

(1 — *u*)3*C*

= 0

+ 3*u*(1 — *u*)2*w C*

+ 3*u*2(1 — *u*)*w*2*C*2

+ *u*3*C*3

curves. This study imposed continuity constraints into the least squares optimization process to ensure that the computed con- trol points respect the estimated tangents at the end points.

Meanwhile, a few researchers had used metaheuristic method recently to curve fit outline images or a set of data points such as Sarfraz [[18]](#_bookmark24) that used simulated annealing to curve fit extracting outlines of images with a generalized cubic spline, the simulated annealing is used to optimize the shape parameter and another paper [[19]](#_bookmark24) also used simulated anneal- ing as the mechanism to globally optimizes the shape parame- ters in the description of the conic splines but in the case of

poor approximation, the insertions of intermediate points are

(1 — *u*)3 + 3*u*(1 — *u*)2*w*1 + 3*u*2(1 — *u*)*w*2 + *u*3

(1)

where *u* = *s*—*si* and *hi* = *si*+1 — *si*; *u* ∈ [0; 1].

*hi*

*w*1 and *w*2 are shape parameters and *Ci*; *i* = 0; 1; 2; 3 are control points with *C*0 and *C*1 are fixed.

1. Least-squares error and reparameterization

By using centripetal method, the length of the data polygon can be written as

*n*

X

made as long as the desired approximation is achieved. Yahya [[16]](#_bookmark24) proposed particle swarm optimization to optimize

*L* = |*pi* — *p*

*i*=1

*i*—1

|1/2 (2)

the control points and weight which were then used in conic

equations. While, Ga´lvez and Iglesias [[3]](#_bookmark21) applied PSO to com-

Hence the parameters are

( 1 X*k* ! )

*s*

|1/2

*s*

= 1

(3)

0

pute an appropriate location of knots as the knots were treated

as free variables for B-splines functions.

In this paper, a metaheuristic approach namely, HS is

*k* *i*

*i*=1

= 0 *s*

=

|*p* — *p*

*L*

*i*—1 *n*

implemented as an approximation tool using rational cubic Be´zier curve from given data points. Our algorithm is based on the idea of minimizing least-squares error by Yahya [[14]](#_bookmark24) in order to improve positions of two middle control points,

*C*1; *C*2 and values of weights, *w*1*andw*2 as in Yahya et al. [[15]](#_bookmark24).

We use the adjustments adjust its shape and parametric struc- ture so as to construct curves that pass as closely as possible between the data sets smoothly. We also adjust and control points and values of weights until the error of the least squares is minimized. Therefore, the best approximation with mini- mum least-squares error can be obtained by this technique. The aim of this study was to prove that HS can be used as a method to fit a set of data points via rational cubic Be´zier and also as a best method based on its time consuming and guarantee to nearly reach the global optimal solution and locally optimal solution as it has a stoping criteria with the best solution it has found so far. In order to prove that statement, a statistical analysis had been done.

function between *P*(*ui*) and *Q*(*si*) is For a specified set of control points, the least-squares error

*n*

X

*E* = |*P*(*ui*)— *Q*(*si*)|2 (4)

*i*=1

We are looking the values of *w*1; *w*2; *C*1 and *C*2 for which *E*

is minimum.

1. Harmony search

Currently many phenomenon-mimicking meta-heuristic algo- rithms, such as genetic algorithm (GA), simulated annealing (SA), tabu search, ant colony optimization, and particle swarm optimization (PSO), have been used in various science and engineering problems. The advantages of these algorithms over calculus-based optimization algorithms include: not requiring complex gradient derivative and initial vector, ability to per- form global search as well as local search, and efficiently

Data fitting by *G*1 rational cubic Be´zier curves 177

handling discrete variables [[5]](#_bookmark23). Harmony Search (HS) is a metaheuristic algorithm which was originally inspired by the improvisation process of Jazz musicians. The analogy between improvisation and optimization can be described as each musi- cian corresponds to each decision variable; musical instru- ment’s pitch range corresponds to decision variable’s value range; musical harmony at certain time corresponds to solu- tion vector at certain iteration; and audience’s aesthetics corre- sponds to objective function [[8]](#_bookmark23). Just like musical harmony is improved time after time, solution vector is improved iteration by iteration. HS imposes fewer mathematical requirements and does not require initial value settings of decisions variables. As

(HMCR); pitch adjusting rate (PAR); and number of improvi- sation (NI), or stopping criterion. The harmony memory (HM) is a memory location where all the solution vectors (sets of decision variables) are stored. At this step, HM is similar to the genetic pool in the GA where all of the data had been stored [[10]](#_bookmark23).

* 1. *Initialize the harmony memory*

HM matrix is filled with as many randomly generated solution vectors as the HMS

the algorithm uses stochastic random searches, derivative information is also unnecessary. The steps in the procedure

1 1

1 2

*u*

*u*

2

6 1 2

*u*

*u*

2

2

··· *u*1 1

··· *u*2 *N*

*N*—1

*N*—1

*u*

*u*

3

2

*N*

7

of harmony search are shown in [Fig. 1](#_bookmark5). The basic steps are

as follows [[10]](#_bookmark23):

*u*

*4.1. Initialize the problem and algorithm parameters*

HM = .

6

*u*

. . .

. . .

6

*u*

*u*

*N*—1

4

HMS—1

1

HMS 1

.

*u*

*N*—1

HMS—1

2

HMS 2

··· *u*HMS—1

··· *u*HMS

. (6)

7

HMS—1

*N*

HMS

*N*

5

.

*u*

Min *f*(*u*) subject to *ui* ∈ *U*; *i* = 1; 2; ... ; *N* (5) Suppose the optimization problem is specified as follows: where *f*(*u*) is an objective function, decision variables *ui*, *N* is

the number of decision variable, *U* is the set of the possible

range of values for each decision variable, *Lui* 6 *U* 6 *Uui* where *Lui* and *Uui* are lower and upper bounds for each deci- sion variable, respectively. Parameters for HS are harmony memory size (HMS); harmony memory considering rate

Figure 2 Geometric view of *G*1 continuity.

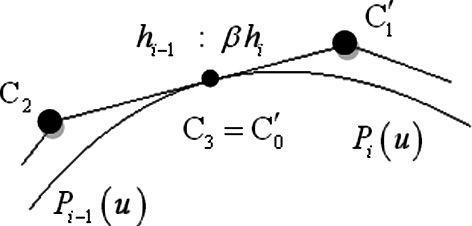
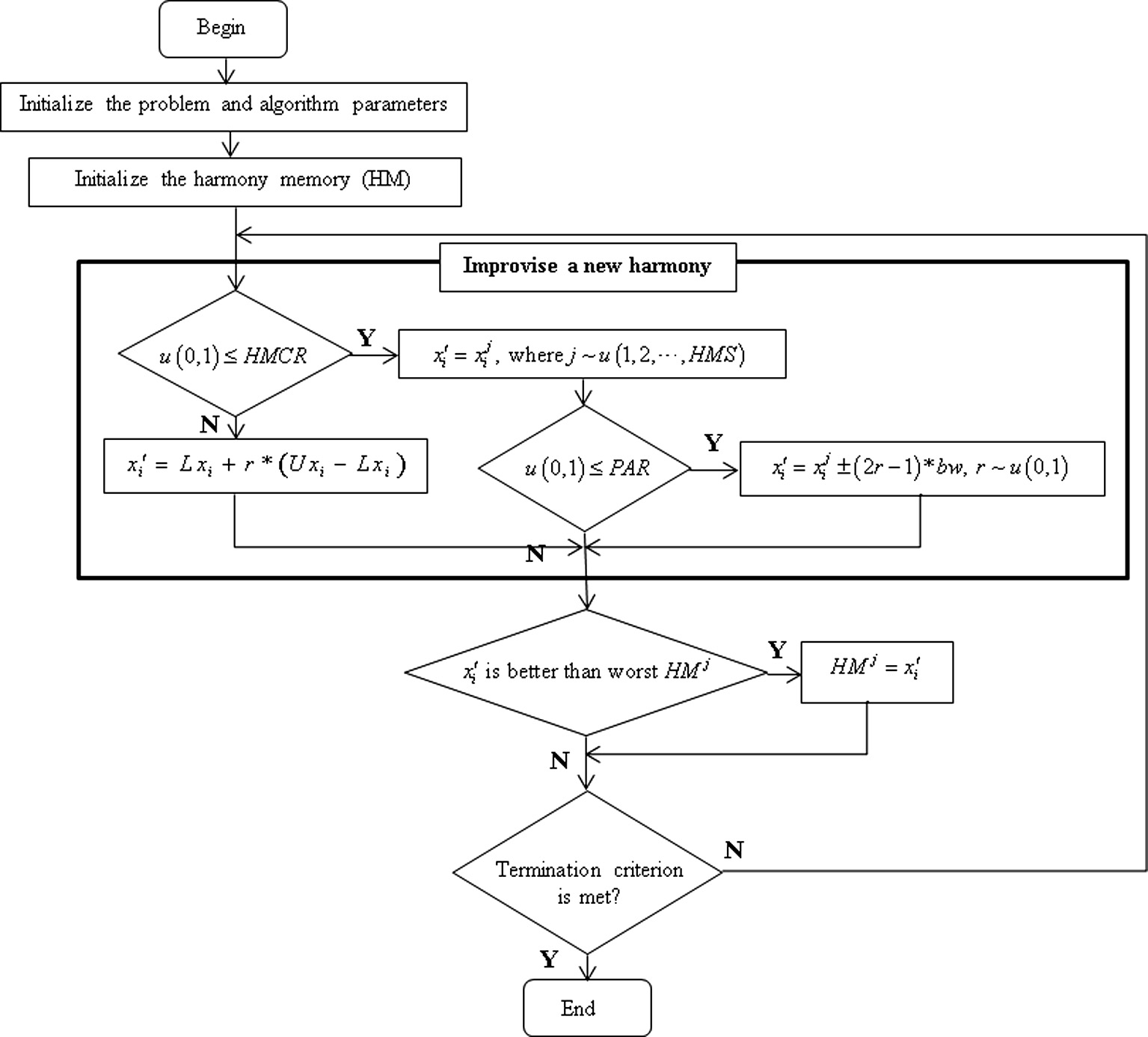
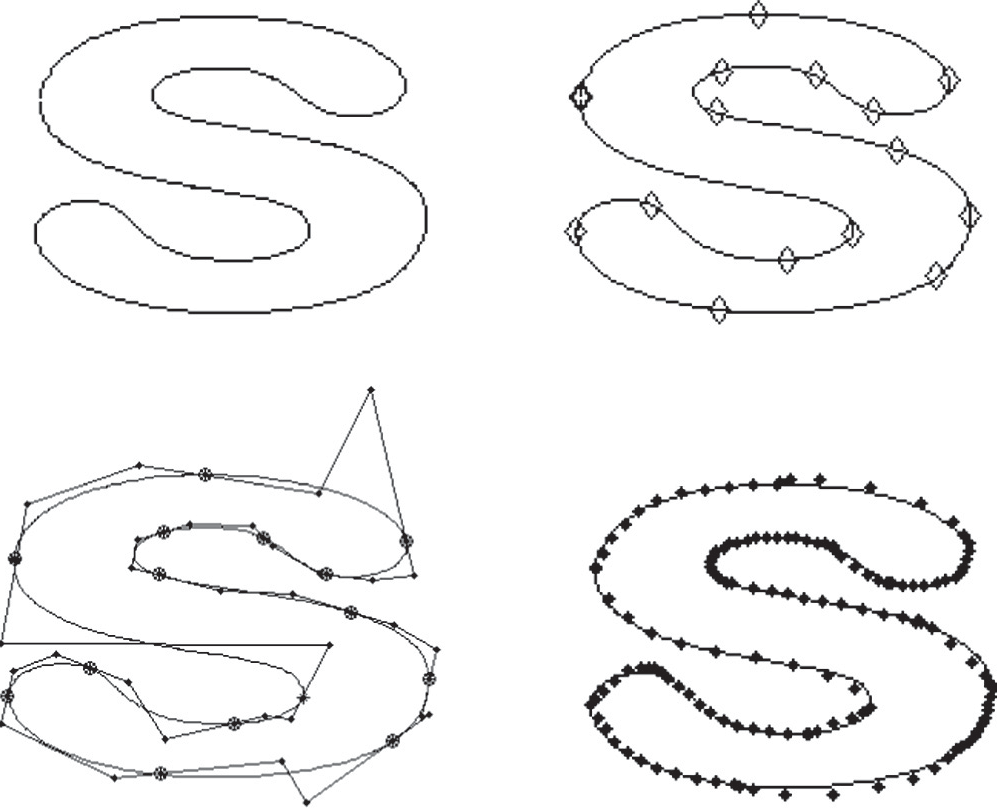


Figure 1 Procedure of harmony search algorithm.

178 N. Mohamed et al.

Figure 3 *G*1 continuity illustration between (a) 3 segments and (b) 2 segments.



(a)

(b)

- OG

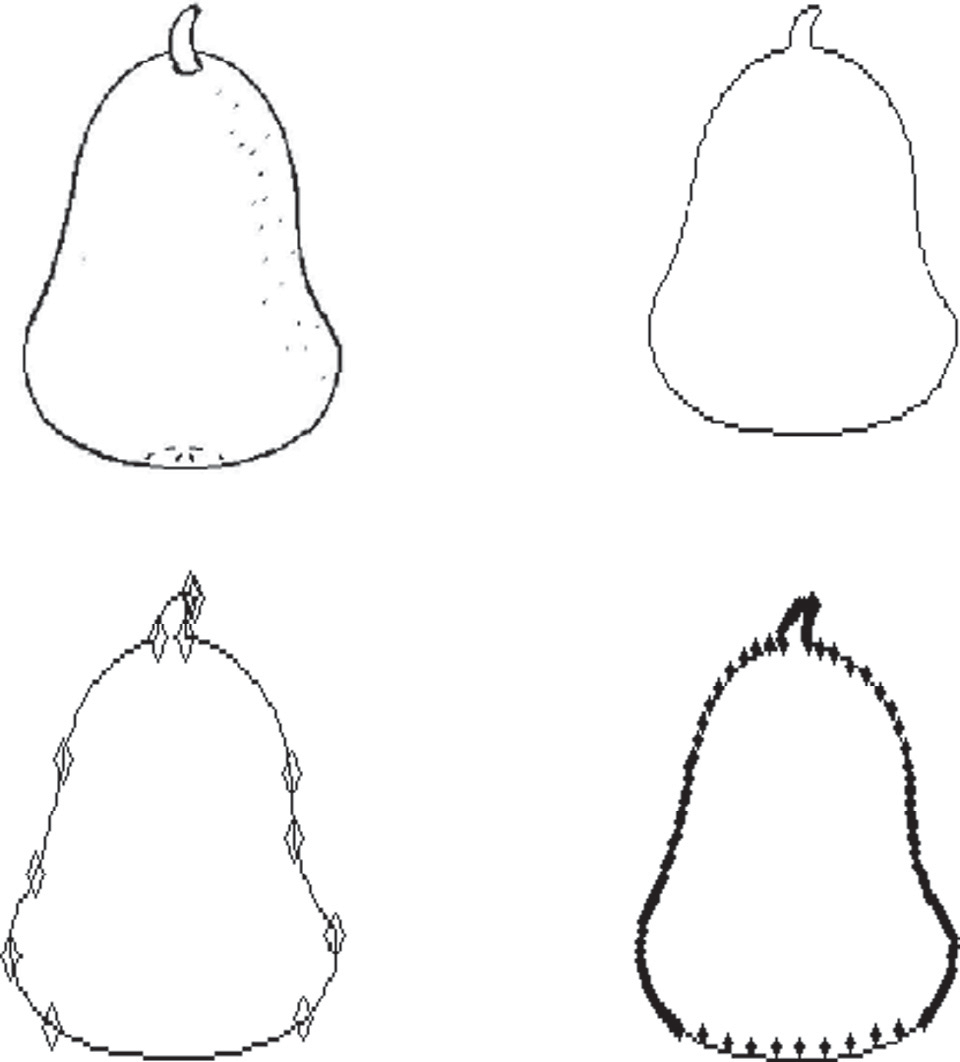
\*\*\*RB

# (c) (d)

Figure 4 (a) Outline of the image, (b) outline with break points,

(c) line connecting C*i*’s in every segment, (d) outline image with correspondent rational Be´zier curve.

* 1. *Improvise a new harmony*



(a)

(b)

- OG

\*\*\*RB

A new harmony vector, *u*' = (*u*' ; *u*' ; ··· ; *u*' ) is generated based on three rules: memory consideration, pitch adjustment and

1 2 *N*

random selection. Generating a new harmony is called ‘impro- visation’. In the memory consideration, the value of the first

# (c) (d)

Figure 5 (a) Bitmapped image, (b) outline of the image, (c) outline with break points, (d) outline image with correspondent rational Be´zier curve.

decision variable, *u*' for the new vector is chosen from any

1

1

of the values in the specified HM range (*u*'1; *u*'2; ··· ; *u*'HMS),

1. G

continuity between two segments

1 1 1

similar process to the other decision variables (*u*' ; *u*' ; ··· ; *u*' ).

2 3 *N*

The detailed process of this steps is illustrated in [Fig. 1](#_bookmark5).

* 1. *Update harmony memory*

If the new harmony vector, *u*' = (*u*' ; *u*' ; ··· ; *u*' ) is better than

Suppose two regular curves B(*s*); *s* ∈ [*s*0; *s*1], and C(*t*); *t* ∈ [*t*0; *t*1], meet at a point P = B(*s*1)= C(*t*0). Then the Marsh [[11]](#_bookmark23) gave a definition of geometric continuity by:

two curves said to meet with *Gk* – *continuity* whenever there

is a Reparameterization b : [*u*0; *u*1]→ [*s*0; *s*1] such that

1 2 *N*

the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

*s*1 = b(*u*1) and

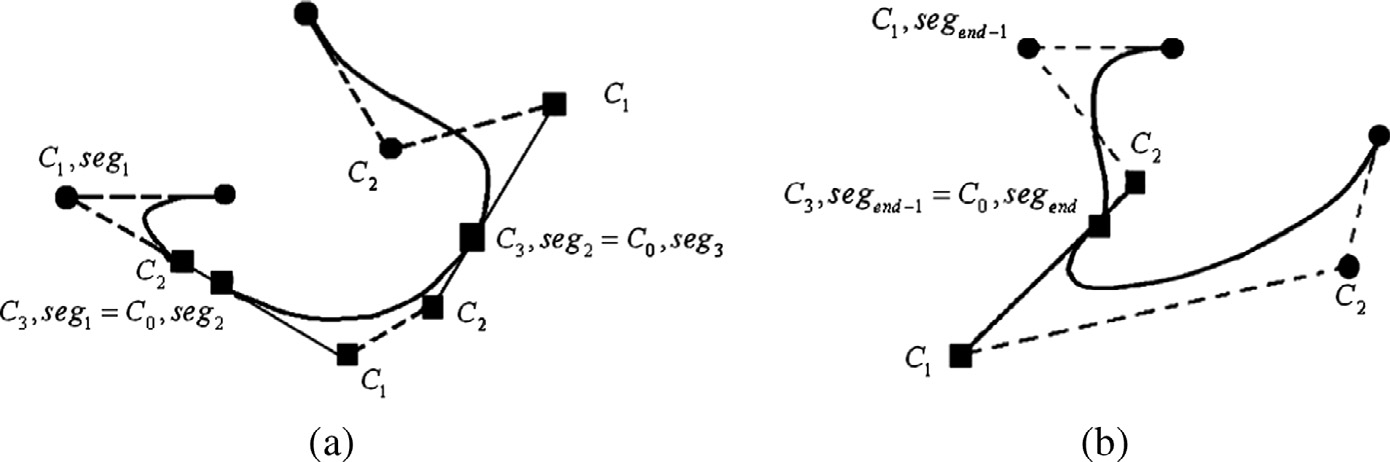
*di*B

*dui* (b(*u*))|*u*=*u*1

*di*C

*dti* (*t*)|*t*=*t*0 (7)

* 1. *Check*



If the stopping criterion is satisfied, computation is terminated. Otherwise, steps 3 and 4 are repeated.

for all *i* = 0; ··· ; *k*. This type of continuity is called *geometric*

*continuity*.

In a geometric view, for a regular curve *P*(*u*), *G*1 at *u* if it is

*G*0 continuous and it possesses continuous unit tangent vector,

Data fitting by *G*1 rational cubic Be´zier curves 179

Figure 6 (a) Outline of the image, (b) outline with break points, (c) outline image with correspondent rational Be´zier curve.

*P*'(*u*—) and *P*'(*u*+) point to the same direction with different values of magnitude as in [Fig. 2](#_bookmark4).

Be´zier points *C*2; *C*3 and *C*' must be collinear but the ratio

1

of *C*2*C*3  and *C*' *C*'  are not fixed, where b is utilized to

0 1

break the chain.

1. Method and implementation

Figure 7 (a) Bitmapped image, (b) outline of the image, (c) outline with break points, (d) outline image with correspondent rational Be´zier curve.

b*P*'(*u*—)= *P*'(*u*+) (8)

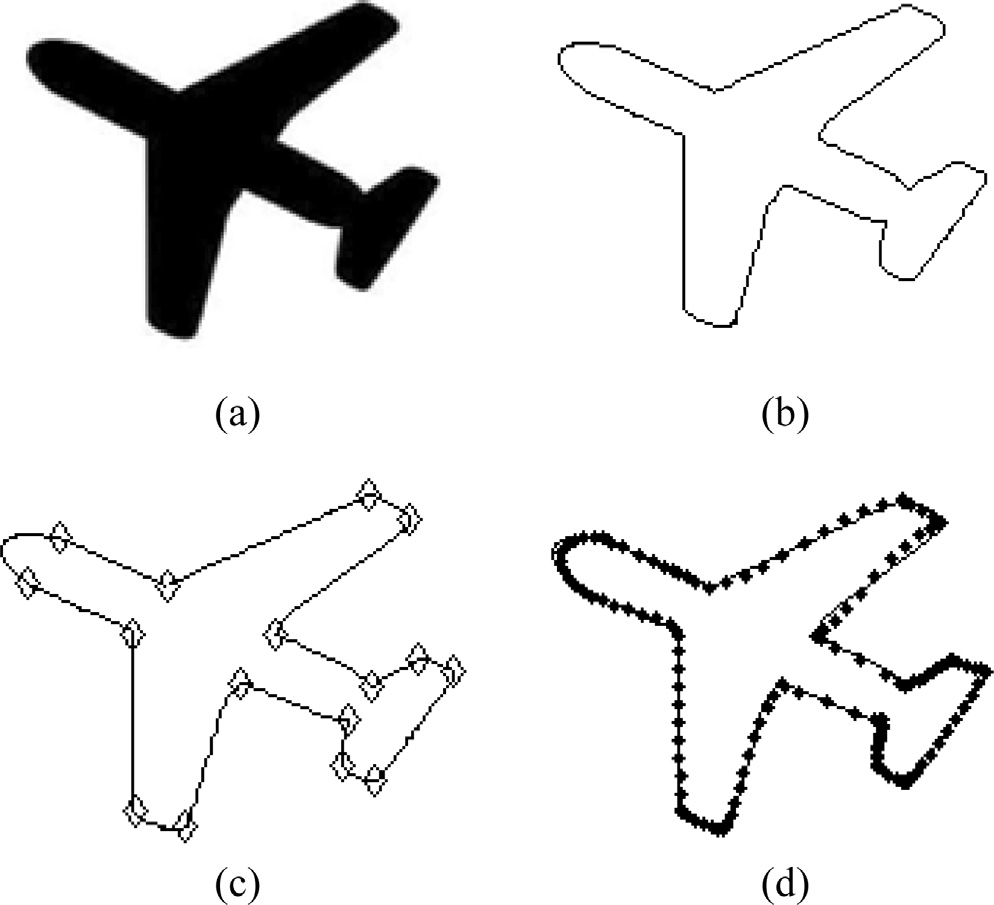
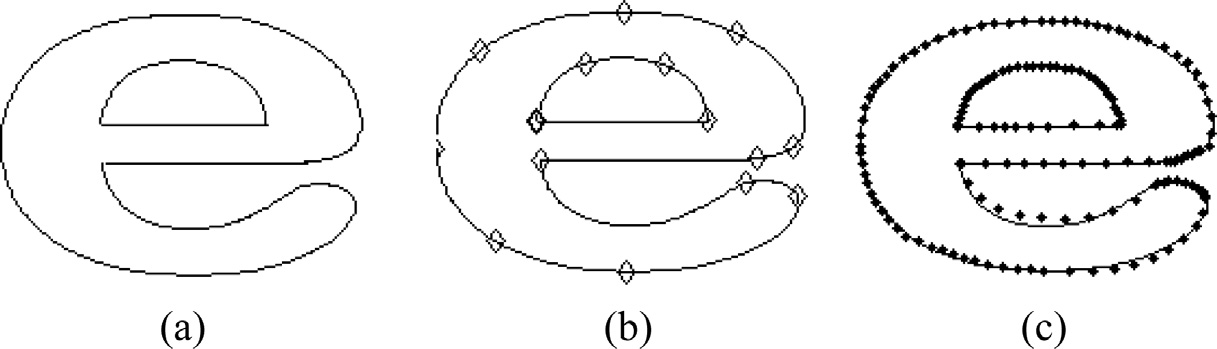
where b = *P*'(*u*+) > 0



*P*'(*u*—)

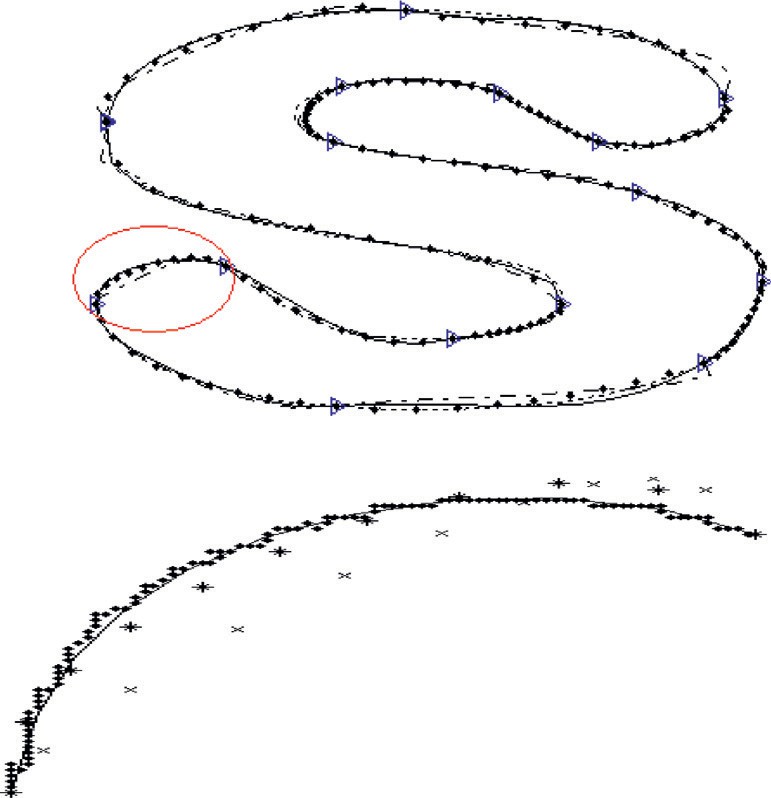
In order to implement HS, Eq. [(4)](#_bookmark3) is used as the objective function. HS parameters in this work are HM = 5, HMCR = 0.9, PAR = 0.3 and bw = 0.03. All these values of parameters are the usual choice in HS community and also supported based on empirical results by [[13]](#_bookmark23). According to results by Omran and Mahdavi [[13]](#_bookmark23), in general, using a small HM seems to be a good and logical choice with the added advantage of reducing space requirements. Actually, since HM resembles the short-term memory of a musician which is known to be small, it is logical to use a small HM as in the paper used the smallest value of HM is 5. As for HMCR, a large value for HMCR (e.g. 0.95) generally improves the performance of the HS. The experiments show that using a relatively small constant value for PAR seems to improve the performance of the HS.

As the beginning, all the data extracted from the outline boundary of the images had been broke into a few curve seg- ments. The completion of the procedures to fit all the data con- sists of 3 sections: original segments, segment between two segments and end segment.



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 1 *G*1 continuity analysis. | | | | | | | | | | | |
| *n*th joint | [Fig. 5](#_bookmark8)  b*x* | b*y* |  | [Fig. 6](#_bookmark9)  b*x* | b*y* |  | [Fig. 7](#_bookmark10)  b*x* | b*y* |  | [Fig. 8](#_bookmark13)  b*x* | b*y* |
| 1 | 4.26051098 | 4.26051098 |  | 0.01893534 | 0.01893534 |  | 10.43118780 | 10.43118780 |  | 0.59910652 | 0.59910652 |
| 2 | 0.16440373 | 0.16440373 |  | 0.04707052 | 0.04707052 |  | 0.38101031 | 0.38101039 |  | 1.72965880 | 1.72965880 |
| 3 | —2.77452417 | —2.77452417 |  | 3.30416527 | 3.30416527 |  | 4.32293758 | 4.32293758 |  | 5.19218689 | 5.19218689 |
| 4 | —0.34379848 | —0.34379848 |  | 0.41811991 | 0.41811991 |  | 2.05956026 | 2.05956026 |  | 1.32544346 | 1.32544346 |
| 5 | 1.01815645 | 1.01815645 |  | 1.53058122 | 1.53058122 |  | 1.81849452 | 1.81849452 |  | 0.92138456 | 0.92138456 |
| 6 | 0.01585093 | 0.01585093 |  | 0.37198273 | 0.37198273 |  | 0.88708711 | 0.88708711 |  | 0.65161496 | 0.65161496 |
| 7 | 2.55695847 | 2.55695847 |  | 4.12417886 | 4.12417886 |  | 1.17072419 | 1.17072419 |  | 1.08933812 | 1.08933812 |
| 8 | 0.49683132 | 0.49683132 |  | —2.63191694 | 0.65945804 |  | 0.42307881 | 0.42307881 |  | 1.06743386 | 1.06743386 |
| 9 | —0.34326318 | —0.91623243 |  | 0.43726459 | 0.43726459 |  | 7.93291619 | 7.93291619 |  | 0.39554224 | 0.39554224 |
| 10 | 0.30432804 | 0.30432804 |  | 0.409990409 | 0.40999041 |  | 0.20215543 | 0.20215543 |  | 0.84602743 | 0.84602743 |
| 11 | 3.74608222 | 3.74608222 |  | —3.87751453 | 6.62429858 |  | 1.23469281 | 1.23469281 |  | 1.66866911 | 1.66866911 |
| 12 | —1.50717641 | 0.18755597 |  | – | – |  | 1.78335501 | 1.78335501 |  | 0.87416563 | 0.87416563 |
| 13 | —2.00241487 | 4.31008604 |  | – | – |  | – | – |  | 3.24574994 | 3.24574994 |
| 14 | 1.04244158 | 1.04244158 |  | – | – |  | – | – |  | 0.35296832 | 0.35296832 |
| 15 | 1.73650617 | 1.73650617 |  | – | – |  | – | – |  | – | – |
|  |  |  |  |  |  |  |  |  |  |  |  |

* 1. *Original segments*



Outline image HS

\* GA

x PSO

In this section, only the odd segments (1,3,.. .) will be considered and all the data in the segment will be approxi- mated without any constraints. The size of *w*1 and *w*2 is in

[0; 2]. The search space for *C*1 and *C*2 are estimated as

follows:

Let a segment consists a set of points (*d*1; *d*2; ·· · ; *dend*)∈ R2

while *C*\* and *C*\* are extremum points in the segments. The size

1 2

of each *C*\* and *C*\* is determined by:

1 2

size of C\* = |*d*max — *d*min|; *i* = 1; 2 (9)

i

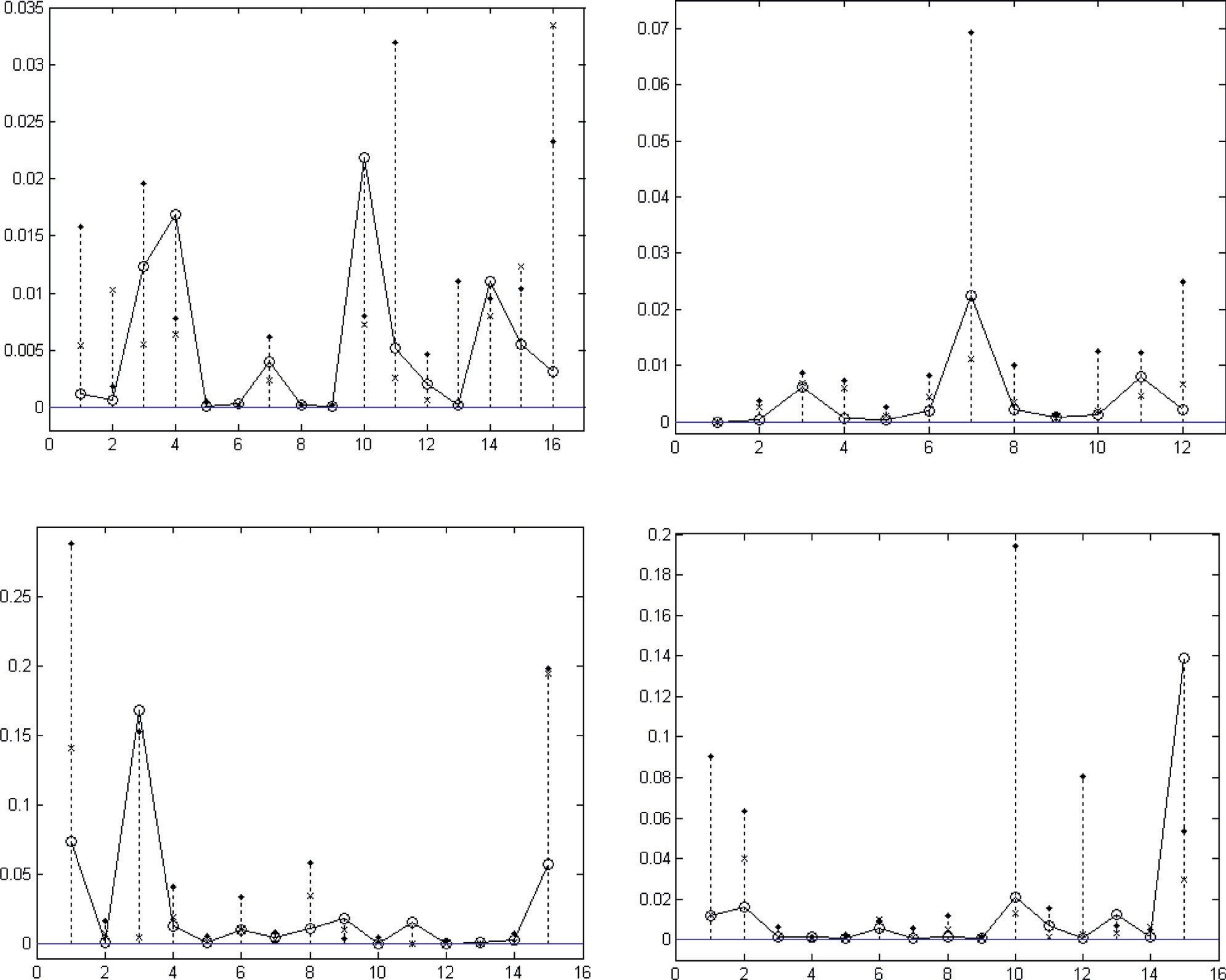
Figure 8 Data fitting for outline for alphabet ‘S’ with 2593 data points. The highlight segment consists of 113 data points, HS error = 0.0008, GA error = 0.0025 and PSO error = 0.0803.

* 1. *Segment between two segments*

For this section, the even segments (2,4,.. .) will be considered and all the data in the segment will be approximated with cer- tain constraints, which are as follows:

Suppose a set of data has three segments, *seg*1; *seg*2 and *seg*3

as in [Fig. 3](#_bookmark6)(a), each segment of rational cubic Be´zier curve



(a) Aeroplane (1875 data points)

(b) Pear (1433 data points)

-o HS

x GA

\* PSO

- o HS

x GA

\* PSO

- o HS

x GA

\* PSO

- o HS

x GA

\* PSO

(c) Letter ‘e’ (2783 data points) (d) Letter ‘s’ (2593 data points)

Figure 9 Minimum least-squares error for each segment of test outline images.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |  |
|  |
| Letter ‘e’\_GA | 0.172 | 30 | 0.024 | 0.932 | 30 | 0.056 |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Aeroplane\_GA 0.102 30 0.200 0.962 30 0.345  Aeroplane\_PSO 0.112 30 0.200 0.953 30 0.204  Pear\_HS 0.172 30 0.024 0.938 30 0.079  Pear\_GA 0.102 30 0.200 0.973 30 0.621  Pear\_PSO 0.088 30 0.200 0.948 30 0.153 | | | | | | | The end segment, *seg*end will be considered if total data points are even. All the data in the segment will be approximated with certain constraints, which are: |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |  |
|  |
|  |
|  |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 2 Descriptive data of four outline images. | | | | | | |
|  | Best | Worst | Mean | Median | Variance | Standard deviation |
| *Letter ‘s’*  HS | 0.0500 | 0.1400 | 0.1087 | 0.1063 | 0.00099 | 0.03153 |
| GA | 0.0900 | 0.1700 | 0.1599 | 0.1533 | 0.00172 | 0.04151 |
| PSO | 0.2300 | 0.7300 | 0.5610 | 0.5377 | 0.03745 | 0.19353 |
| *Letter ‘e’*  HS | 0.1151 | 0.2567 | 0.1648 | 0.1633 | 0.00129 | 0.03597 |
| GA | 0.1837 | 0.9036 | 0.6007 | 0.5963 | 0.03636 | 0.19068 |
| PSO | 0.7127 | 1.1592 | 0.9362 | 0.9595 | 0.01072 | 0.10354 |
| *Aeroplane*  HS | 0.0437 | 0.0890 | 0.068247 | 0.06665 | 0.00013 | 0.01161 |
| GA | 0.0290 | 0.1393 | 0.074257 | 0.0712 | 0.00047 | 0.02162 |
| PSO | 0.1082 | 0.3716 | 0.22588 | 0.22215 | 0.00241 | 0.04910 |
| *Pear*  HS | 0.0123 | 0.0222 | 0.01695 | 0.0168 | 0.00000 | 0.00274 |
| GA | 0.0235 | 0.0426 | 0.03226 | 0.0321 | 0.00003 | 0.00512 |
| PSO | 0.1107 | 0.2482 | 0.174553 | 0.1728 | 0.00179 | 0.04227 |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 3 Tests of normality.  Kolmogorov–Smirnov Shapiro–Wilk Statistic df Sig. Statistic df Sig. | | | | | | | should consist of (*C*0; *C*1; *C*2; *C*3), the control points. All these  three segments are *G*1. Therefore, data fitting in *seg* must fulfil  2  certain constraints involving *seg*1 and *seg*3.  The size of *w*1 and *w*2 is in [0; 2.5]. While search space for *C*\*  1 |
| Letter ‘s’\_HS | 0.087 | 30 | 0.200 | 0.960 | 30 | 0.308 | and *C*\* are similar to the previous section but one of the vari-  2 |
| Letter ‘s’\_GA Letter ‘s’\_PSO Letter ‘e’\_HS | 0.108  0.169  0.101 | 30  30  30 | 0.200  0.029  0.200 | 0.971  0.943  0.947 | 30  30  30 | 0.572  0.109  0.138 | ables, *Ci* = (*xi*; *yi* ) depends on the other one, for which  (*C*2; *segi* ); (*C*3; *segi* = *C*0; *segi*+1); (*C*1; *segi*+1) , *i* = 1; 2 must  be collinear. For this section, the number of decision variables |
| Letter ‘e’\_PSO 0.183 30 0.012 0.935 30 0.067  Aeroplane\_HS 0.073 30 0.200 0.980 30 0.825 | | | | | | | for HS is reduced.  *6.3. End segments* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 4 Independent group ANOVA. | | | | | | |
| ANOVA |  |  |  |  |  |  |
| Source of variation | SS | df | MS | *F* | *P*-value | *F* crit |
| *(a) Letter ‘s’* |  |  |  |  |  |  |
| Between Groups | 3.681533865 | 2 | 1.840767 | 137.4745378 | 1.16929E—27 | 3.101296 |
| Within Groups | 1.16491916 | 87 | 0.01339 |  |  |  |
| Total | 4.846453025 | 89 |  |  |  |  |
| *(b) Letter ‘s’* |  |  |  |  |  |  |
| Between Groups | 8.974284755 | 2 | 4.487142 | 278.2765 | 1.56918E—38 | 3.101295757 |
| Within Groups | 1.402854501 | 87 | 0.016125 |  |  |  |
| Total | 10.37713926 | 89 |  |  |  |  |
| *(c) Aeroplane* |  |  |  |  |  |  |
| Between Groups | 0.47874 | 2 | 0.23937 | 238.294 | 5.03791E—36 | 3.101295757 |
| Within Groups | 0.087393 | 87 | 0.001005 |  |  |  |
| Total | 0.566133 | 89 |  |  |  |  |
| *(d) Pear* |  |  |  |  |  |  |
| Between Groups | 0.453205995 | 2 | 0.226603 | 373.463 | 1.99546E—43 | 3.101296 |
| Within Groups | 0.052788262 | 87 | 0.000607 |  |  |  |
| Total | 0.505994257 | 89 |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 5 *F*-test of two samples for variances. | | | | | | | | | | |
| *F*-test two-sample | for variances |  |  | *F*-test two-sample | for variances |  |  | *F*-test two-sample | for variances |  |
|  | *GA* | *HS* |  |  | *PSO* | *HS* |  |  | *PSO* | *GA* |
| *(a) Letter ‘s’* |  |  |  |  |  |  |  |  |  |  |
| Mean | 0.159926667 | 0.10868 |  | Mean | 0.561043333 | 0.10868 |  | Mean | 0.561043333 | 0.1599267 |
| Variance | 0.001723099 | 0.000994 |  | Variance | 0.037452536 | 0.000994 |  | Variance | 0.037452536 | 0.0017231 |
| Observations | 30 | 30 |  | Observations | 30 | 30 |  | Observations | 30 | 30 |
| df | 29 | 29 |  | df | 29 | 29 |  | Df | 29 | 29 |
| *F* | 1.733516623 |  |  | *F* | 37.67896352 |  |  | *F* | 21.73556516 |  |
| *P*(*F*<=*f*) one-  tail | 0.072201294 |  |  | *P*(*F*<=*f*) one-  tail | 2.71646E—16 |  |  | *P*(*F*<=*f*) one-  tail | 4.75936E—13 |  |
| *F* critical one- | 1.619899621 |  |  | *F* critical one- | 1.860811435 |  |  | *F* critical one- | 1.860811435 |  |
| tail |  |  |  | tail |  |  |  | tail |  |  |
| *F*-test two-sample | for variances |  |  | *F*-test two-sample | for variances |  |  | *F*-test two-sample | for variances |  |
|  | *HS* | *GA* |  |  | *HS* | *PSO* |  |  | *GA* | *PSO* |
| *(b) Letter ‘e’* |  |  |  |  |  |  |  |  |  |  |
| Mean | 0.164833333 | 0.6007 |  | Mean | 0.164833333 | 0.936147 |  | Mean | 0.6007 | 0.936147 |
| Variance | 0.001293492 | 0.03636 |  | Variance | 0.001293492 | 0.010721 |  | Variance | 0.036359512 | 0.010721 |
| Observations | 30 | 30 |  | Observations | 30 | 30 |  | Observations | 30 | 30 |
| df | 29 | 29 |  | df | 29 | 29 |  | df | 29 | 29 |
| *F* | 0.035575064 |  |  | *F* | 0.120647051 |  |  | *F* | 3.391337565 |  |
| *P*(*F*<=*f*) one-  tail | 1.4988E—14 |  |  | *P*(*F*<=*f*) one-  tail | 8.71309E—08 |  |  | *P*(*F*<=*f*) one-  tail | 0.000769416 |  |
| *F* critical one- | 0.537399965 |  |  | *F* critical one- | 0.537399965 |  |  | *F* critical one- | 1.860811435 |  |
| tail |  |  |  | tail |  |  |  | tail |  |  |
| *F*-test two-sample | for variances |  |  | *F*-test two-sample | for variances |  |  | *F*-test two-sample | for variances |  |
|  | *HS* | *GA* |  |  | *HS* | *PSO* |  |  | *GA* | *PSO* |
| *(c) Aeroplane* |  |  |  |  |  |  |  |  |  |  |
| Mean | 0.068247 | 0.074257 |  | Mean | 0.068246667 | 0.22588 |  | Mean | 0.074256667 | 0.22588 |
| Variance | 0.000135 | 0.000468 |  | Variance | 0.000134765 | 0.002411209 |  | Variance | 0.000467574 | 0.002411 |
| Observations | 30 | 30 |  | Observations | 30 | 30 |  | Observations | 30 | 30 |
| df | 29 | 29 |  | df | 29 | 29 |  | Df | 29 | 29 |
| F | 0.288223 |  |  | F | 0.055891197 |  |  | F | 0.193916687 |  |
| *P*(*F*<=*f*) one-  tail | 0.000634 |  |  | *P*(*F*<=*f*) one-  tail | 6.20171E—12 |  |  | *P*(*F*<=*f*) one-  tail | 1.54586E—05 |  |
| *F* critical one- | 0.412637 |  |  | *F* critical one- | 0.412636754 |  |  | *F* critical one- | 0.412636754 |  |
| tail |  |  |  | tail |  |  |  | tail |  |  |
| *F*-test two-sample | for variances |  |  | *F*-test two-sample | for variances |  |  | *F*-test two-sample | for variances |  |
|  | *HS* | *GA* |  |  | *HS* | *PSO* |  |  | *GA* | *PSO* |
| *(d) Pear* |  |  |  |  |  |  |  |  |  |  |
| Mean | 0.01695 | 0.03226 |  | Mean | 0.01695 | 0.174553 |  | Mean | 0.03226 | 0.174553333 |
| Variance | 7.51776E—06 | 2.62E—05 |  | Variance | 7.51776E—06 | 0.001787 |  | Variance | 2.62E—05 | 0.001786553 |
| Observations | 30 | 30 |  | Observations | 30 | 30 |  | Observations | 30 | 30 |
| df | 29 | 29 |  | df | 29 | 29 |  | Df | 29 | 29 |
| *F* | 0.286781845 |  |  | *F* | 0.004207969 |  |  | *F* | 0.014673 |  |
| *P*(*F*<=*f*) one- | 0.000607464 |  |  | *P*(*F*<=*f*) one- | 0 |  |  | *P*(*F*<=*f*) one- | 0 |  |
| tail |  |  |  | tail |  |  |  | tail |  |  |
| *F* critical one- | 0.412636754 |  |  | *F* critical one- | 0.412636754 |  |  | *F* critical one- | 0.412637 |  |
| tail |  |  |  | tail |  |  |  | tail |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

Suppose a set of data has two last segments, *seg*end—1 ; *seg*end

as in [Fig. 3](#_bookmark6)(b). All these two segments should have smooth

joints with *G*1 continuity. Therefore, data fitting in *seg* must fulfil certain constraints involving *seg*end—1 .

end

The size of *w*1 and *w*2 is in [0; 2.5]. While search space for *C*1

able, *Ci* = (*xi*; *yi*) depends on the other one at the last joint, i.e., [(*C*2; *seg*end—1); (*C*3; *seg*end—1 = *C*0; *seg*end); (*C*1; *seg*end)] and *C*2 are similar to the previous section but one of the vari-

must be collinear.

1. Demonstration and experimental results

Proposed data fitting method has been implemented practi- cally in [Figs. 4–7(a)](#_bookmark7). Each data segments are evaluated at uni-

collection of 201 data points on the interval of [0; 1]. formly distributed values of *u* in its domain to generate a

[Figs. 4](#_bookmark7)(d), [5](#_bookmark8)(d), [6](#_bookmark9)(c) and [7](#_bookmark10)(d) are the best fitting curves,

where OG is the original graph and, RB is the corresponding rational Be´zier curve. [Fig. 4](#_bookmark7)(c) shows lines which connects

Table 6 *T*-test of two samples assuming unequal variances.

*t*-test: two-sample assuming unequal variances *t*-test: two-sample assuming unequal

variances

*t*-test: two-sample assuming unequal variances

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *HS* | *GA* |  |  | *HS* | *PSO* |  |  | *GA* | *PSO* |
| *(a) Letter ‘s’* |  |  |  |  |  |  |  |  |  |  |
| Mean | 0.10868 | 0.159927 |  | Mean | 0.10868 | 0.561043 |  | Mean | 0.159926667 | 0.5610433 |
| Variance | 0.000993991 | 0.001723 |  | Variance | 0.000993991 | 0.037453 |  | Variance | 0.001723099 | 0.0374525 |
| Observations | 30 | 30 |  | Observations | 30 | 30 |  | Observations | 30 | 30 |
| Hypothesized | 0 |  |  | Hypothesized | 0 |  |  | Hypothesized | 0 |  |
| mean difference |  |  |  | mean difference |  |  |  | mean difference |  |  |
| df | 54 |  |  | df | 31 |  |  | df | 32 |  |
| *t* stat | —5.38485789 |  |  | *t* stat | —12.6362865 |  |  | *t* stat | —11.1000084 |  |
| *P*(*T*<=*t*) one-  tail | 8.08374E—07 |  |  | *P*(*T*<=*t*) one-  tail | 4.56312E—14 |  |  | *P*(*T*<=*t*) one-  tail | 8.30798E—13 |  |
| *t* critical one-tail | 2.397409645 |  |  | *t* critical one-tail | 2.452824193 |  |  | *t* critical one-tail | 2.448677634 |  |
| *P*(*T*<=*t*) two-  tail | 1.61675E—06 |  |  | *P*(*T*<=*t*) two-  tail | 9.12624E—14 |  |  | *P*(*T*<=*t*) two-  tail | 1.6616E—12 |  |
| *t* critical two-tail | 2.669984796 |  |  | *t* critical two-tail | 2.744041919 |  |  | *t* critical two-tail | 2.738481482 |  |

*t*-test: two-sample assuming unequal variances *t*-test: two-sample assuming unequal

variances

*t*-test: two-sample assuming unequal variances

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *HS* | *GA* |  |  | *HS* | *PSO* |  |  | *GA* | *PSO* |
| *(b) Letter ‘e’* |  |  |  |  |  |  |  |  |  |  |
| Mean | 0.164833333 | 0.6007 |  | Mean | 0.164833333 | 0.936147 |  | Mean | 0.6007 | 0.936147 |
| Variance | 0.001293492 | 0.03636 |  | Variance | 0.001293492 | 0.010721 |  | Variance | 0.036359512 | 0.010721 |
| Observations | 30 | 30 |  | Observations | 30 | 30 |  | Observations | 30 | 30 |
| Hypothesized | 0 |  |  | Hypothesized | 0 |  |  | Hypothesized | 0 |  |
| mean difference |  |  |  | mean difference |  |  |  | mean difference |  |  |
| Df | 31 |  |  | df | 36 |  |  | df | 45 |  |
| *t* stat | —12.3030977 |  |  | *t* stat | —38.5419363 |  |  | *t* stat | —8.4676361 |  |
| *P*(*T*<=*t*) one-  tail | 9.14105E—14 |  |  | *P*(*T*<=*t*) one-  tail | 3.6084E—31 |  |  | *P*(*T*<=*t*) one-  tail | 3.63208E—11 |  |
| *t* critical one-tail | 1.695518783 |  |  | *t* critical one-tail | 1.688297714 |  |  | *t* critical one-tail | 1.679427393 |  |
| *P*(*T*<=*t*) two-  tail | 1.82821E—13 |  |  | *P*(*T*<=*t*) two-  tail | 7.2168E—31 |  |  | *P*(*T*<=*t*) two-  tail | 7.26417E—11 |  |
| *t* critical two-tail | 2.039513446 |  |  | *t* critical two-tail | 2.028094001 |  |  | *t* critical two-tail | 2.014103389 |  |

*t*-test: two-sample assuming unequal variances *t*-test: two-sample assuming unequal

variances

*t*-test: two-sample assuming unequal variances

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *HS* | *GA* |  |  | *HS* | *PSO* |  |  | *GA* | *PSO* |
| *(c) Aeroplane* |  |  |  |  |  |  |  |  |  |  |
| Mean | 0.068247 | 0.074257 |  | Mean | 0.068246667 | 0.22588 |  | Mean | 0.074256667 | 0.22588 |
| Variance | 0.000135 | 0.000468 |  | Variance | 0.000134765 | 0.002411 |  | Variance | 0.000467574 | 0.002411 |
| Observations | 30 | 30 |  | Observations | 30 | 30 |  | Observations | 30 | 30 |
| Hypothesized | 0 |  |  | Hypothesized | 0 |  |  | Hypothesized | 0 |  |
| mean difference |  |  |  | mean difference |  |  |  | mean difference |  |  |
| df | 44 |  |  | df | 32 |  |  | df | 40 |  |
| *t* stat | —1.34127 |  |  | *t* stat | —17.1112491 |  |  | *t* stat | —15.4782647 |  |
| *P*(*T*<=*t*) one-  tail | 0.093358 |  |  | *P*(*T*<=*t*) one-  tail | 5.79073E—18 |  |  | *P*(*T*<=*t*) one-  tail | 8.71294E—19 |  |
| *t* critical one-tail | 1.30109 |  |  | *t* critical one-tail | 1.308572793 |  |  | *t* critical one-tail | 1.303077053 |  |
| *P*(*T*<=*t*) two-  tail | 0.186717 |  |  | *P*(*T*<=*t*) two-  tail | 1.15815E—17 |  |  | *P*(*T*<=*t*) two-  tail | 1.74259E—18 |  |
| *t* critical two-tail | 1.68023 |  |  | *t* critical two-tail | 1.693888748 |  |  | *t* critical two-tail | 1.683851013 |  |

*t*-test: two-sample assuming unequal variances *t*-test: two-sample assuming unequal

variances

*t*-test: two-sample assuming unequal variances

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *HS* | *GA* |  |  | *HS* | *PSO* |  |  | *GA* | *PSO* |
| *(d) Pear* |  |  |  |  |  |  |  |  |  |  |
| Mean | 0.01695 | 0.03226 |  | Mean | 0.01695 | 0.174553 |  | Mean | 0.03226 | 0.174553 |
| Variance | 7.51776E—06 | 2.62E—05 |  | Variance | 7.51776E—06 | 0.001787 |  | Variance | 2.62142E—05 | 0.001787 |
| Observations | 30 | 30 |  | Observations | 30 | 30 |  | Observations | 30 | 30 |
| Hypothesized | 0 |  |  | Hypothesized | 0 |  |  | Hypothesized | 0 |  |

(*continued on next page*)

Table 6 (*continued*)

*t*-test: two-sample assuming unequal variances *t*-test: two-sample assuming unequal

variances

*t*-test: two-sample assuming unequal variances

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *HS* | *GA* |  |  | *HS* | *PSO* |  |  | *GA PSO* |
| mean difference |  |  |  | mean difference |  |  |  | mean difference |  |
| df | 44 |  |  | df | 29 |  |  | df | 30 |
| *t* stat | —14.4382645 |  |  | *t* stat | —20.3800972 |  |  | *t* stat | —18.3051871 |
| *P*(*T*<=*t*) one-  tail | 1.33021E—18 |  |  | *P*(*T*<=*t*) one-  tail | 4.92526E—19 |  |  | *P*(*T*<=*t*) one-  tail | 3.95498E—18 |
| *t* critical one-tail | 2.414134368 |  |  | *t* critical one-tail | 2.46202136 |  |  | *t* critical one-tail | 2.457261542 |
| *P*(*T*<=*t*) two-  tail | 2.66042E—18 |  |  | *P*(*T*<=*t*) two-  tail | 9.85052E—19 |  |  | *P*(*T*<=*t*) two-  tail | 7.90995E—18 |
| *t* critical two-tail | 2.692278266 |  |  | *t* critical two-tail | 2.756385904 |  |  | *t* critical two-tail | 2.749995654 |

all the C*i*‘s in every segment which applies *G*1 continuity between two segments.

[Table 1](#_bookmark12) summarizes the *G*1 continuity analysis for above test outline images. Values of b for each test function are cal- culated based on Eq. [(8)](#_bookmark11).

* 1. *Comparison with other methods and analysis*

Our *G*1 HS approach performs well for the above test outline images. To support this claim, a comparison with recent alter- native curve fitting based on soft computing techniques has been carried out. There were a few of soft computing method used in this data fitting problem, such as curve fitting by B- splines using GA by Ga´lvez et al. [[4]](#_bookmark23), Sarfraz [[19]](#_bookmark24) used cubic spline by SA and Yahya [[15–16]](#_bookmark24) proposed an approach of curve fitting by PSO. Here a comparison between HS, GA and PSO that used on the similar data points. The procedure of PSO was taken from [[2]](#_bookmark22). [Fig. 8](#_bookmark13) highlights on a segment of letter ‘s’ and shows that HS approach better the data points within the same time range as others. [Fig. 9](#_bookmark14) summarizes the least-squares errors of each segment for test outline images and in the graphs; the output for HS in segments 4 and 10 for the first image, segment 3 in third image and also segment 15 in fourth image are larger than others as there are cusps before the segments. However, HS approach each outlines bet- ter by looking at the total of least-squares error in the data sets.

* 1. *Statistical analysis*

All evolutionary algorithms, including HS, GA and PSO are stochastic population based search methods. Accordingly, there is no guarantee that the optimal solution will be reached consistently. Therefore, in order to deny that there is a guaran- tee that HS can be used to have better approximation to the global optimal solution, a comparison on optimization prob- lem using such algorithm where a statistical analysis had been carried out. 30 sample data of total least-squares error of four outlines for methods HS, GA and PSO were being used with the time taken for all the data which were fixed.

From [Table 2](#_bookmark15), it is clearly shows the prominent method for all images based on descriptive data is HS. All the sample data had been assessed their normality by the Shapiro-Wilks statis- tics in order to verify their significance different between their variance. The results of normality were shown in [Table 3](#_bookmark16).

According to [Table 4](#_bookmark17), there is sufficient evidence that the HS has the smallest variation compared to GA and PSO at

1–10% significance level (*p*-value: 0.0722, 2.7164 · 10—16;

1.4988 · 10—14, 8.7131 · 10—8; 0.0006, 6.2017 · 10—12; 0.0006,

0.0000). While [Table 5](#_bookmark18) shows that the mean of HS is the small-

(*p*-value: 8.0837 · 10—7, 4.5621 · 10—14; 9.1411 · 10—14, est value compared to GA and PSO at 1–10% significance level

3.6084 · 10—31; 0.0934, 5.7907 · 10—18; 1.3302 · 10—18,

4.9253 · 10—19). While, [Table 6](#_bookmark19) also supported the same con- clusion of HS compared to other two methods. These leads

to say that, in this study, HS was found to give the best fit for data fitting using rational cubic Bezier for each segment of all tested outlines images. These results give strong indica- tion that the HS method is more stable and accurate compared to GA and PSO.

1. Conclusions

A derivative-free real parameter optimization technique, based on HS, is implemented for data fitting. This technique opti- mizes the control points and shape parameters of rational cubic Be´zier curves in order to approximate the data sets. The technique data fitting by *G*1 continuity for every joint of segments for the whole data set, the rational Be´zier ultimately produces optimal results in approximating the data. It pro- vides an optimal fit with an efficient computation cost. A com- parison between HS, GA and PSO were done on four different outline images and a few of statistical testing also had been carried out over a 30 sample data set each. Based on the statis- tical analysis carried out on the sample data, there are suffi- cient evidences to say that HS gave a smaller values of error compared to other two methods.

References

1. [Cai HJ, Wang GJ. Constrained approximation of rational Be´zier](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0005) [curves based on a matrix expression of its end points continuity](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0005) [condition. Comput Aided Des 2010;42(6):495–504](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0005).
2. [Das S, Abraham A, Konar A. Particle swarm optimization and](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0010) [differential evolution algorithms: technical analysis, applications](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0010) [and hybridization perspectives. Stud Comput Intel (SCI)](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0010) [2008;116:1–38](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0010).
3. [Ga´lvez A, Iglesias A. Efficient particle swarm optimization](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0015) [approach for data fitting with free knot –splines. Comput Aided](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0015) [Des 2011;43(12):1683–92](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0015).
4. [Ga´lvez A, Iglesias A, Puig-Pey J. Iterative two-step genetic-](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0020) [algorithm-based method for efficient polynomial B-spline surface](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0020) [reconstruction. Inf Sci 2012;182:56–76](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0020).
5. [Geem ZW, Sim K-B. Parameter-setting-free harmony search](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0025) [algorithm. Appl Math Comput 2010;217:3881–9](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0025).
6. [Huang W-X, Jin C-J, Wang G-J. Approximate Be´zier curves by](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0030) [cubic LN curves. Appl Math Comput 2011;218:3083–92](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0030).
7. [Hur S, Kim T. Finding the best conic approximation to the](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0035) [convolution curve of two compatible conics based on Hausdorff](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0035) [distance. Comput Aided Des 2009;41(7):513–24](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0035).
8. [Lee KS, Geem ZW. A new meta-heuristic algorithm for contin-](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0040) [uous engineering optimization: harmony search theory and](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0040) [practice. Comput Methods Appl Mech Eng 2005;194:3902–33](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0040).
9. [Lu L. Sample-based polynomial approximation of rational Be´zier](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0045) [curves. J Comput Appl Math 2011;235(6):1557–63](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0045).
10. [Mahdavi M, Fesanghary M, Damangir E. An improved harmony](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0050) [search algorithm for solving optimization problems. Appl Math](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0050) [Comput 2007;188(2):1567–79](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0050).
11. [Marsh D. Applied geometry for computer graphics and CAD. 2nd](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0055) [ed. London: Springer-Verlag; 2005](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0055).
12. [Mortenson ME. Geometric modeling. Canada: John Wiley &](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0060) [Sons; 1997](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0060).
13. [Omran MGH, Mahdavi M. Global-best harmony search. Appl](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0065) [Math Comput 2008;198:643–56](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0065).
14. Yahya F, Ali JM, Majid AA, Ibrahim A. An automatic generation of G1 curve fitting of arabic characters. In: International conference on computer graphics imaging and visualisation CGIV06; 2006. p. 1–6.
15. Yahya ZR, M Piah AR, Majid AA. G1 continuity conics for curve fitting using particle swarm optimization. In: 2011 15th international conference on information visualisation; 2011. p. 497–501.
16. [Yahya ZR, Piah ARM, Majid AA. Curve fitting conic by](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0080) [evolutionary computing. J Math Stat 2012;8(1):107–10](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0080).
17. [Shen L-Y, Yuan C-M, Gao X-S. Certified approximate of](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0085) [parametric space curves with cubic B-spline curves. Comput](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0085) [Aided Des 2012;29:643–63](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0085).
18. [Sarfraz M. Vectorizing outlines of generics shapes by cubic spline](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0090) [using simulated annealing. Int J Comp Math 2010;87(8):1736–51](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0090).
19. [Sarfraz M. Intelligent approaches for vectorizing image outlines. J](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0095) [Software Eng Appl 2012;5:78–83](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0095).
20. [Stamati V, Fudos I. On reconstructing 3D feature boundaries.](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0100) [Comput-Aid Des Appl 2008;5(1–4):316–24](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0100).
21. Yang Y-J, Zeng W, Yang C-L, Meng X-X, Yong J-H, Deng B. G[1](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0105) [continuous approximate curves on NURBS surfaces. Comput](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0105) [Aided Des 2012;44:824–34](http://refhub.elsevier.com/S1110-8665(15)00019-5/h0105).