Curve Matching by Using B-spline Curves

Tet Toe and Tang Van To*
Faculty of Engineering, Assumption University
Bangkok, Thailand

Abstract

This paper presents an algorithm for estimating the control points of the B-spline and curve matching which are achieved by using the dissimilarity measure based on the knot associated with the B-spline curves. The B-splines stand as one of the most efficient curve representations and possess very attractive properties such as spatial uniqueness, boundedness and continuity, local shape controllability, and invariance to affine transformations. These properties made them very attractive for curve representation. Consequently, they have been extensively used in computer-aided design and computer graphics. The curve-matching program is shown in detail in this paper. Any input test object curve can be matched with the B-spline sample curve. The control points of sample curve are computed and stored in the program. The test object curve, a bitmap file, is thinned, then converted to B-spline curve and then to match with the sample curve.

Keywords: *B-spline curve, algorithm, dissimilarity measure, spatial uniqueness, boundedness and continuity, local-shape controllability, invariance, transformations, computer-aided design, computer graphics, curve matching program, bitmap file.*

Introduction

There are many techniques for curve matching. The B-spline stands as one of the most efficient curve representation, and possesses very attractive properties such as spatial uniqueness, boundedness and continuity, local shape controllability, and structure preservation under affine transformation (de Boor 1978). Because of these properties they can be used to represent curves. To recognize the handwritten characters we should notice that the characters differ in different persons, different writing ways, different sizes, different directions, different stretching ways, and so on. The recognition of handwritten characters by computer has been studied intensively for many years. Most of them are using Neural Network in this study.

For the recognition of handwritten characters we deal with matching the curves which are modeled as B-splines. Curve matching is achieved by choosing the best B-spline curve with invariance affine transformation. This affine transformation must be estimated. At first the control points of the B-spline are estimated. Then the best order B-spline and the best number of control points are decided. To solve the problem of curve matching the main objectives of this work are as the following:

1. Skeletonization
2. Determination of B-spline control points for the curve
3. Matching the curve using dissimilarity between their control points.

* *Faculty of Science and Technology, Assumption University, Bangkok, Thailand*

Skeletonization

To make a thinner form, the input curve must be skeletonized according to the Thinning Algorithm (Carlo and Cabriella 1985). Thinning is a process that transforms a binary image into a line skeleton of unit thickness. The major functions of thinning in image processing are to:

1. reduce data storage,
2. reduce transmission time,
3. facilitate the extraction of morphological features from digitized pattern.

An input curve can be thinned by using Thinning Algorithm as follows.

The input curve is scanned row-wise from left to right and from top to bottom. A black point, P_1, is flagged if all of the following 6 tests, H_1-H_6, return the value TRUE:

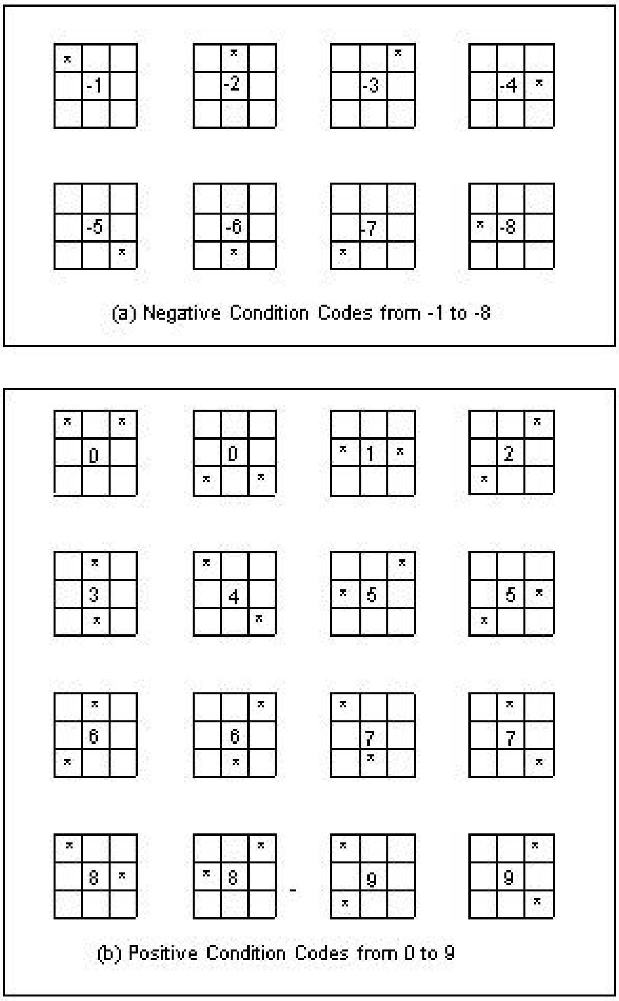
H_1: $P_2+P_4+P_6+P_8 \leq 3$; /* P_1 is an edge-point*/
H_2: $B(P_1) \geq 2$; /*P_1 is not an end-point*/
H_3: $N(P_1) \geq 1$: /* $N(P_1)$ is the number of
unflagged dark 8-neighbor of P_1*/
H_4: $X(P_1) = 1$; /*P_1 is not a break point*/
H_5: Either P_4 is unflagged, or $X_4 = 1$;
/* $X_4(P_1)$ is the crossing-number of P_1 if we
temporarily assume that P_4 is white. */
H_6: Either P6 is unflagged, or $X_6(P_1)=1$;
/* $X_6(P_1)$ is the crossing-number of P_1 if we
temporarily assume that P_6 is white */

At the end of the pass, the flagged points are deleted, the algorithm stopping if there are no flagged points at the end of the pass.

A skeletonized character is represented by a binary matrix. For any pixel P_{ij}, let S be the sum of the eight neighboring entries of P. The condition code C_{ij} of P_{ij} is assigned a value in the set {-9, -8,……,0, 1,…..,9} according to the following rules:

RULE 1: If $S \geq 3$ then P_{ij} is a junction point C_{ij}= -9
RULE 2: If S = 1 then P_{ij} is an end point and C_{ij} is assigned a value in {-9,-8,-7,….,-1} (Fig. 1-a)
RULE 3: If S=2, then C_{ij} is assigned a value in {0,1,….,9} (Fig. 1-b).

The negative condition codes represent end points and junction points. The non-negative condition codes are used to expedite the tracing process when considered in conjunction with the coordinates of the preceding pixel. In this way a curve is traced until a pixel with negative code is encountered, indicating that the end of a section of the skeleton is reached. After a curve has been traced, it is approximated by line segments. The line segments are then grouped into primitives. A scene of scaling and thinning the curve is shown in Fig. 2.



(a) Negative Condition Codes from -1 to -8

(b) Positive Condition Codes from 0 to 9

Fig. 1. Condition codes

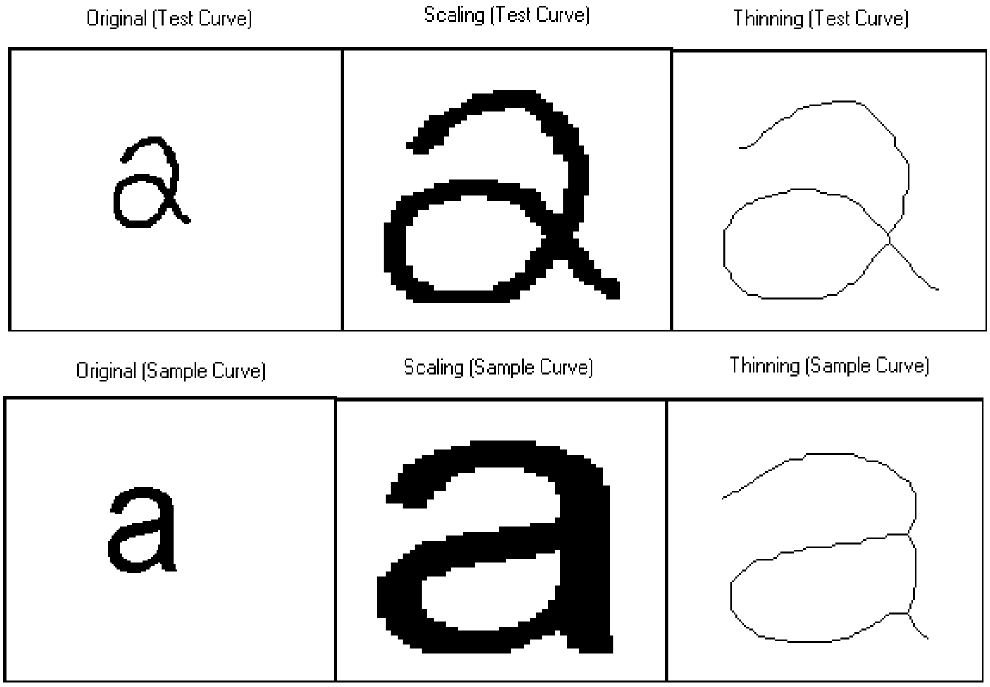


Fig. 2. Scaling and thinning

In Fig. 2, the bitmap is scaled and thinned.

Determination of B-spline control points for the curve

Let $r(t)$ be the position vectors along the curve as a function of the parameter t, a B-spline curve is given by (de Boor 1972):

$$r(t) = \sum_{i=1}^{n+1} C_i Q_{i,k}(t), \quad t_{\min} \le t < t_{\max}, 2 \le k \le n+1 \qquad \text{……......…..(3-1)}$$

where C_i are the position vectors of the $n+1$ control points, and the $Q_{i,k}(t)$ are the normalized B-spline basis functions.

For the i^{th} normalized B-spline basis function of order k, the basis functions $Q_{i,k}(t)$ are defined by the Cox-deBoor recursion formulas. Specifically,

$$Q_{i,0}(t) = \begin{cases} 1 & if \quad t_i \le t \le t_{i+1} \\ 0 & otherwise \end{cases} \qquad \text{……………...(3-2)}$$

$$Q_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} Q_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} Q_{i+1,k-1}(t) \text{...(3-3)}$$

In Equation (3-3), we interpret the terms $\frac{t - t_i}{t_{i+k} - t_i} Q_{i,k-1}(t)$ and $\frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} Q_{i+1,k-1}(t)$ as 0 when their denominators vanish. Equation (3-3) suggests a computational procedure for evaluating a B-spline at some point t. We observe that on an any given interval $[t_i, t_{i+1}]$ there are only $k+1$ B-spline basic functions of degree k that are non-zero. On that interval, $Q_{i,k}(t)$ depends only on $Q_{i,k-1}(t)$ because $Q_{i+1,k-1}(t)$ is zero there while $Q_{i-l,k}(t)$ $(0 < l \le k)$ depends on both $Q_{i-l+1,k-1}(t)$ and $Q_{i-l,k-1}(t)$. The interdependence of the B-splines is shown in Fig. 3.

In order to find the splines of degree k, we must find $k-1$ previous levels in the diagram shown in that figure and at each level we must find the B-splins $Q_{i,j}(t)$ to $Q_{i-l,j}(t)$, where j is the degree at that level and l ranges from 0 to j. This leads to Algorithm 3.1 (Pavlidis 1982).

$$\begin{tikzcd}
Q_{i,0} \arrow[d] \arrow[dr] & & & \\
Q_{i,1} \arrow[d] \arrow[dr] & Q_{i-1,1} \arrow[d] \arrow[dr] & & \\
Q_{i,2} \arrow[d] \arrow[dr] & Q_{i-1,2} \arrow[d] \arrow[dr] & Q_{i-2,2} \arrow[d] \arrow[dr] & \\
Q_{i,,3} & Q_{i-1,3} & Q_{i-2,3} & Q_{i-3,3}
\end{tikzcd}$$

Fig. 3. Interdependence of the values of the B-splines at a point t

In Fig. 3, each term is weighted sum of one or two terms in the line above it. The lines with arrows indicate the flow of the computation. Vertical lines denote multiplication by the first factor in Equation (3-3), and diagonals by the second factor in that equation.

Algorithm 3.1

Procedure BSPLINE(i,t,k):
Evaluation of all the B-splines at a point *t* belonging to the interval [t_i , t_{i+1}].

Notation: *k* is the degree of the spline. *t* is the point where the splines are evaluated. The array *Q(I,J)* contains the values *$Q_{i,j}(t)$. a* and *b* auxiliary variables.
Initialize *Q(i,0)* to *1*.

```
For j=1 to k do:
  Begin.
  For l=0 to j do:
    Begin.
    {Compute Q(i-l,j).}
    $a=((t-t_{i-l})/(t_{i-l+j}-t_{i-l}))\times Q(i-l,j-1)$.
    $b=((t_{i-l+j+1}-t)/(t_{i-l+j+1}-t_{i-l+1}))\times Q(i-l+1,j-1)$.
    $Q(i-l,j)=a+b$.
    End.
  End.
End of Algorithm.
```

The authors have given a set of data points $r_1, r_2, ..., r_j$. We have to determine a set of control points that generates a B-spline curve for a set of given data points. If a data point lies on the B-spline curve, then it must satisfy Equation (3-1) (Rogers and Adams 1990). Writing Equation (3-1) for each of j data points yields:

$$r_1(t_1) = Q_{1,k}(t_1)C_1 + Q_{2,k}(t_1)C_2 + \;.\;.\;.\; + Q_{i,k}(t_1)C_i$$
$$r_2(t_2) = Q_{1,k}(t_2)C_1 + Q_{2,k}(t_2)C_2 + \;.\;.\;.\; + Q_{i,k}(t_2)C_i$$
$$.$$
$$.$$
$$.$$
$$r_j(t_j) = Q_{1,k}(t_j)C_1 + Q_{2,k}(t_j)C_2 + \;.\;.\;.\; + Q_{i,k}(t_j)C_i$$

where $2 \le k \le n+1 \le j.$ This system of equations is more compactly written in matrix form as

$$[r] = [Q][C] \qquad \text{……………. (3-4)}$$

where

$$[r]^T = [r_1(t_1)\ r_2(t_2)\ .\ .\ .\ r_j(t_j)]$$
$$[C]^T = [C_1 \quad C_2 \quad .\ .\ . \quad C_i]$$
$$[Q] = \begin{bmatrix} Q_{1,k}(t_1) & Q_{2,k}(t_1) & .\ .\ . & Q_{i,k}(t_1) \\ Q_{1,k}(t_2) & Q_{2,k}(t_2) & .\ .\ . & .Q_{i,k}(t_2) \\ . & . & .\ .\ . & . \\ . & . & .\ .\ . & . \\ . & . & .\ .\ . & . \\ Q_{1,k}(t_j) & Q_{2,k}(t_j) & .\ .\ . & .Q_{i,k}(t_j) \end{bmatrix}$$

If $\mathrm{i} = \mathrm{j}$, then the matrix $[Q]$ is square and the set of control points is directly obtained by matrix inversion, i.e.,

$$[C] = [Q]^{-1}[r], \;\; 2 \le k \le n+1 = j \qquad \text{……...(3-5)}$$

If $\mathrm{i} < \mathrm{j}$, $[Q]$ is no longer square, this problem can be solved according to the rules of matrix. Recalling that a matrix times its transpose is always square. Thus, multiplying both sides of Equation (3-4) by $[Q]^T$ yields

$$[Q]^T[r] = [Q]^T[Q][C]$$

and

$$[C] = [[Q]^T [Q]]^{-1} [Q]^T [r] \quad \text{.........} \; (3\text{-}6)$$

In both cases, resulting B-spline curve passes through each data point, i.e., a curve fit is obtained. This technique assumes that the matrix $[Q]$ is known. Provided that the order of the B-spline basis k, the number of control points $n+1$, and the parameter value along the curve are known, then the basis functions $Q_{i,k}(t_j)$ and hence the matrix $[Q]$ can be obtained.

The parameter value t_j for each data point is a measure of the data point's distance along the B-spline curve. We can use the Chord length method (Cohen and Wang 1994) for the estimation of t_j. For this method the chord length l of the curve is first computed according to

$$l = \sum_{j=1}^{n} \left\| r_j - r_{j-1} \right\| \qquad \text{.............} \; (3\text{-}7)$$

t_j associated with the point r_j is computed as

$$t_j = t_{j-1} + \left\| r_j - r_{j-1} \right\| \frac{t_{\max}}{l} \;, \quad \text{.......} \; (3\text{-}8)$$

$j = 1,2,3,...,n+1$ where $t_1 = 0$ and $t_{\max} = n+1$.

When the sampled curve points on a B-spline are dense, the B-spline curve between two consecutive sampled curve points is well approximated by a line segment, and the distance between two points is well approximated by the chord length, which is a linear approximation to the B-spline between points.

Matching the Curve Using Dissimilarity Between Their Control Points

We are presented with a set of sample curve data points from a transformed object curve, and are asked to recognize the test curve from the B-spline best fitted to the sample curve data points. Given two B-splines $r(t)$ and $r'(t)$, let S and S' be their total arc length respectively. Assume that their centers of mass coincide with the origin of the coordinate system. The control points assignment of curve, $r'(t)$, (Cohen *et al.* 1995) is as follows.

Let $f = (r_1, r_2, ..., r_{n+1})$ be the control points set of curve $r(t)$ and let $s_1, s_2, ..., s_{n+1}$ be the distances in arc length traveled counterclockwise from the point of intersection of the curve $r(t)$ with the positive x-axis (from the starting point if the curve is open) to $r_1, r_2, ..., r_{n+1}$ respectively. Denote $f' = (r_1', r_2', ..., r_{n+1}')$ as the assigned control points set for the curve $r'(t)$, and let $s_1', s_2', ..., s_{n+1}'$ be the distances in arc length traveled counterclockwise from the point of intersection of the curve $r'(t)$ with the positive x-axis (from the starting point if the curve is open) to $r_1', r_2', ..., r_{n+1}'$ respectively. The f' set is determined such that:

$$\frac{s_i'}{S'} = \frac{s_i}{S} \;, \quad i = 0,1,2,...,n \quad \text{..........} \; (4\text{-}1)$$

This control points assignment guarantees that the ratio of the distances in arc length between the control points of $r'(t)$ relative to its total length is the same as that for $r(t)$.

The error (dissimilarity) E between $r(t)$ and $r'(t)$ is then defined as:

$$E = \frac{\sum_{i-0}^{i=n} \sqrt{\left\| r_i - r_i' \right\|^2}}{n} \qquad \text{...............} \; (4\text{-}2)$$

Equation (4-2) should theoretically be zero when $r(t)$ and $r'(t)$ are the same, $r'(t)$ is recognized as sample curve for which the error (dissimilarity) E is the smallest. According to the curve matching algorithm the error (dissimilarity) E between the test object curve and sample curve is calculated. If the value of E is equal to zero we can say that the test curve and sample curve coincide. Otherwise the minimum value of E which is less than the threshold value is chosen as the best match.

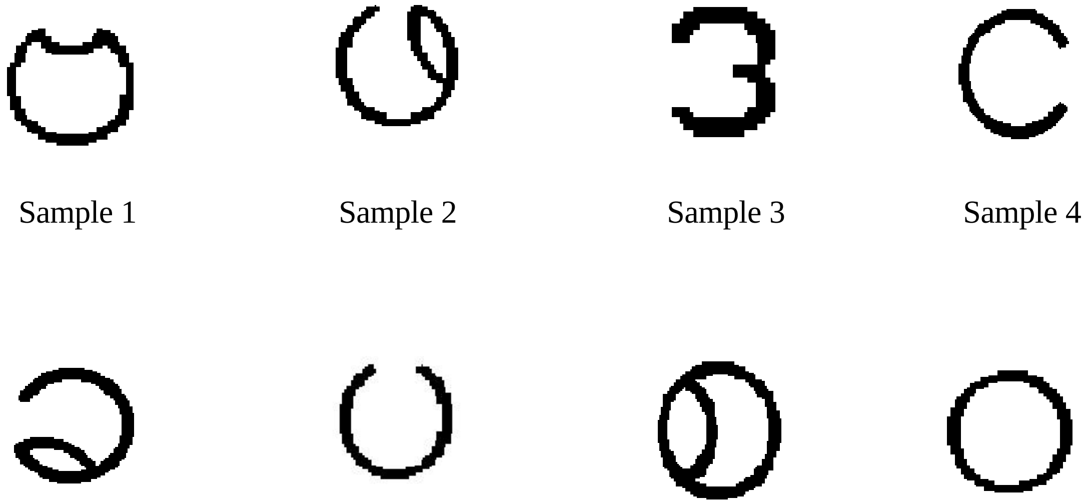
Results and Discussion

There are two main parts for the curve-matching program. The first part is devoted to the B-spline sample curves. Sample curves are saved as the bitmap files (Rimmer 1993).

In order to help the classification process more dependent on the difference in shape rather than size, the sample curves are normalized to have the same size, position, and orientation by scaling. For each sample curve or test object curve, after scaling, the bitmap file is smoothed and skeletonized by using thinning algorithm. The skeleton is then traced and the coordinates of the curve points are determined (Murray and van Ryper 1994). Using these curve points to determine the control points of its B-spline. The same process is done for the test object curves.

The second part is matching the test object curve with the sample curves. We make many experiments and the results of one experiment are shown in this paper. In this experiment, a set of eight Myanmese characters are used as the sample curves which are represented as 20 segmented B-spline of degree 3. Given a test object curve, the B-spline for this test object curve is determined, then the dissimilarity between this test object curve and the sample curves are computed.

The test object curve is recognized as a sample object, which has the dissimilarity between them is minimal and less than the threshold value. The sample curves, test object curves, and the result of this experiment are shown as below. The test objects differ from the sample patterns in font, size and own some errors such as some parts of its print longer or shorter. The last test object is totally not closed to any sample pattern.



Sample 1

Sample 2

Sample 3

Sample 4

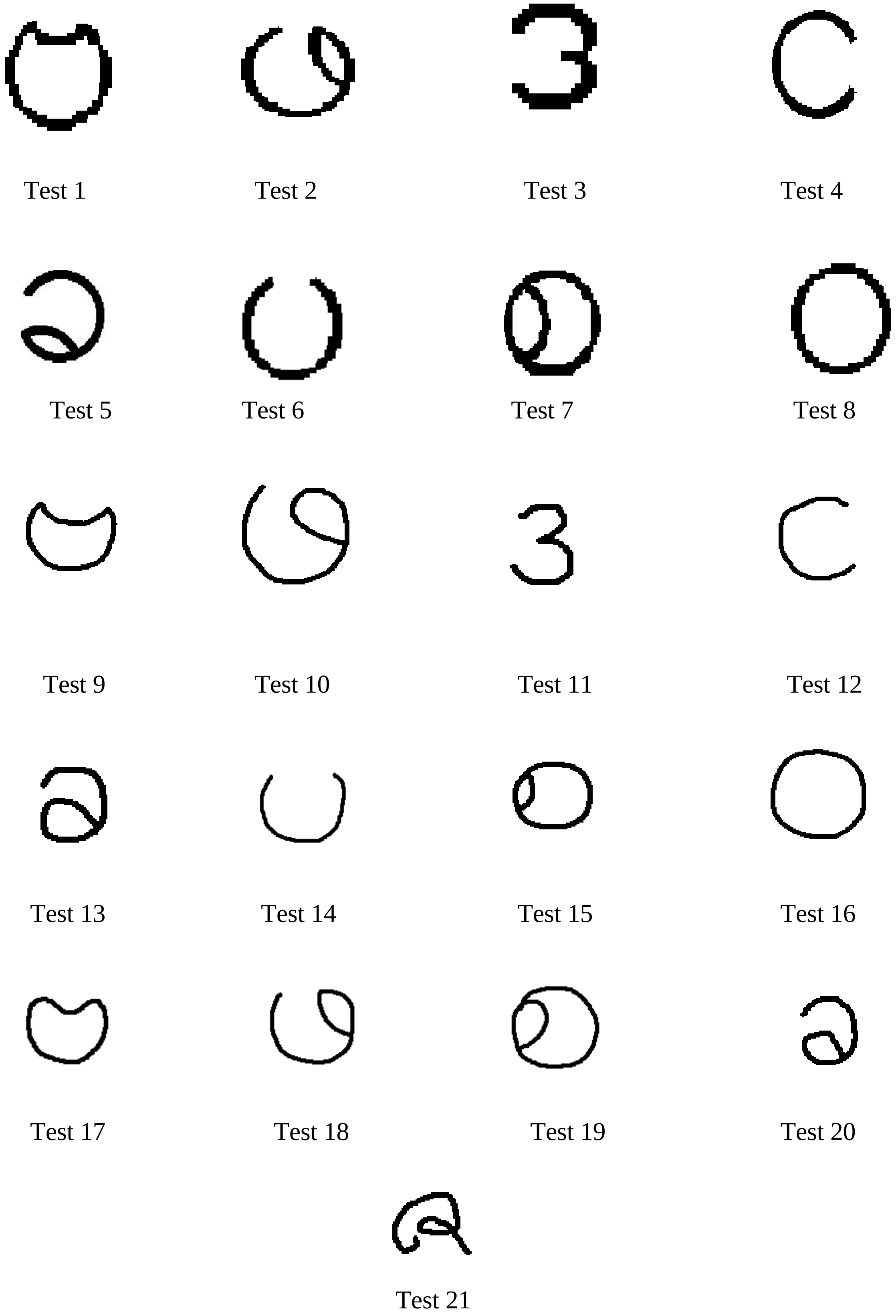
Sample 5

Sample 6

Sample 7

Sample 8

Fig. 4. Sample curves of the experiment



Test 1

Test 2

Test 3

Test 4

Test 5

Test 6

Test 7

Test 8

Test 9

Test 10

Test 11

Test 12

Test 13

Test 14

Test 15

Test 16

Test 17

Test 18

Test 19

Test 20

Test 21

Fig. 5. Test object curves of the experiment

Table 1. Result of the experiment

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8
Test 1	**1.96**	80.19	50.84	77.42	29.24	37.86	52.78	19.65
Test 2	79.19	**5.98**	72.37	88.22	72.47	85.58	73.09	81.31
Test 3	51.12	72.46	**0.78**	72.37	33.27	82.28	72.77	66.69
Test 4	76.23	94.35	72.55	**10.9**	64.64	74.91	79.11	76.03
Test 5	30.67	72.39	32.34	70.65	**1.52**	61.36	63.33	44.94
Test 6	38.84	87.26	83.12	75.92	59.52	**1.63**	37.73	16.86
Test 7	54.68	75.15	73.46	81.1	63.33	34.43	**4.07**	42.39
Test 8	30.7	81.34	74.11	76.15	51.66	13.85	36.3	**11.77**
Test 9	**13.5**	80.56	44.9	81.66	30.81	48.83	61.86	31.08
Test 10	81.16	**17.93**	74.86	98.73	72.52	88.34	73.96	83.85
Test 11	51.48	72.3	**15.96**	67.25	38.16	80.34	70.24	65.33
Test 12	78.37	91.62	73.31	**9.91**	65.95	78.36	81.19	77.72
Test 13	41.21	78.67	43.35	60.05	**27.53**	62.75	61.7	49.2
Test 14	40.79	92.48	85.91	78.21	63.91	**9.38**	41.35	24.02
Test 15	42.86	80.37	72.62	77.74	58.28	28.87	**23.5**	27.26
Test 16	31.68	84.54	75.87	77.99	53.83	14.79	39.65	**12.35**
Test 17	**7.25**	83.9	49.06	80.39	30.54	43.65	59.43	24.97
Test 18	82.85	**16.79**	76.22	102.62	75	90.19	77.82	85.82
Test 19	56.4	77.85	76.51	79.23	65.37	34.69	**13.43**	42.92
Test 20	34.6	72.69	37.59	70.49	**13.02**	61.56	61.48	46.55
Test 21	85.84	60.51	66.2	71.12	69.3	99.91	93.64	94.15

In Table 1, the dissimilarity between test object curves and sample curves are computed. The dissimilarity between the test object curve and its corresponding sample curves (bold values) is very small compared to the nearest dissimilarity values. This means that our method works well in recognizing hand written characters.

Table 2. The comparison between the smallest similarity and the second smallest similarity for experiment

Test Object Curve	Smallest Similarity	Second Smallest Similarity
Test 1	1.96	19.65
Test 2	5.98	72.37
Test 3	0.78	33.27
Test 4	10.9	64.64
Test 5	1.52	30.67
Test 6	1.63	16.86
Test 7	4.07	34.43
Test 8	11.77	13.85
Test 9	13.5	30.81
Test 10	17.93	72.52
Test 11	15.96	38.16
Test 12	9.91	65.95
Test 13	27.53	41.21
Test 14	9.38	24.02
Test 15	23.5	27.26
Test 16	12.35	14.79
Test 17	7.25	24.97
Test 18	16.79	75
Test 19	13.43	34.69
Test 20	13.02	34.6

Conclusion and Recommendation

This curve-matching program can be used to recognize the curves and character recognization. In this study the experiment is made to test the curve-matching program. According to the result of the experiment, this curve-matching program can provide recognition of the handwritten characters which differ in different persons, writing ways, sizes, directions and stretching ways. In this paper, the curve-matching program covers only the curve-matching method involving uniform B-spline curves of degree 3 and limited control points.

The uniform B-spline with equal chord length does not work well with the curve with the curvature change very fast, or in another words, the curve has more details. This can be done by using non-uniform B-spline with inverse chord lengths.

If the test object curve and sample curve are not in the same orientation, or the test object curve is stretching in some direction, the program developed also fails to recognize. This can be improved by determining the best affinity between the test object curve and sample curve. The number of segments as well as degree of B-spline depends significantly to the (sample/test object) curves. However, this study has not covered the best degree, the best number of segments, or how to divide a curve to segments, are very interesting.

References

Carlo, A.; and Cabriella, S. 1985. A width-independent fast thinning algorithm. IEEE Trans. Patt. Anal. Machine Intell. PAMI-7: 463-74.

Cohen, F.S.; and Wang, J.Y. 1994. Modeling image curves using invariant 3-D object curve model. IEEE Trans.16: 1-12.

Cohen, F.S.; Huang, Z.; and Yang, Z. 1995. Invariant matching and identification of curves using B-splines curve representation. IEEE Trans. on Image Process. 4: 1-10.

de Boor, C. 1972. On calculation with B-splines, J. Approx. Theory 6: 50-62.

de Boor, C. 1978. A Pratical Guide to Splines, Springer-Verlag, New York.

Murray, J.D.; and van Ryper, W. 1994. Encyclopedia of Graphics File Formats. O'Reilly & Associates, Sebastopol, CA, USA.

Pavlidis, T. 1982. Algorithms for Graphics and Image Processing. Computer Science Press, Rockville, MD, USA.

Rimmer, S. 1993. Windows Bitmapped Graphics, McGraw-Hill, New York.

Rogers, F.D.; and Adams, J.A. 1990. Mathematical Elements for Computer Graphics. McGraw-Hill, New York.