An Application of Bezier curves in the off-line verification of handwritten signatures

Piotr Gawor [1] and Joanna Kołodziej [2]

University of Bielsko - Biała, Department of Mathematics and Computer Science, Bielsko-Biała, Poland
[1] email: pgawor@ath.bielsko.pl, [2] email: jkolodziej@ath.bielsko.pl

Abstract. The task of verifying human signatures as a pattern recognition problem is considered in this paper. We propose two-step vectorization procedure of the binary handwritten signature images with the application of the cubic Bezier curves. We introduce a new contour segments extraction algorithm to generate the discrete sets of signature contour points. A multi-layer neural network trained by the real-coded genetic algorithm was proposed as the signature patterns classifier. Some simple numerical tests results are also reported.

1 Introduction

Handwritten signatures are still the most widely employed form of securing personal identification, especially for cashing cheques and credit card transactions.

The two major methods of handwriting recognition are on-line and off-line systems [9]. An on-line system measures the sequential data such as handwriting and pen pressure with a special device or pen directly attached to it. An off-line system uses an optical scanner to obtain handwriting data already written on a piece of paper.

The task of verifying human signatures can be considered as a difficult pattern recognition problem. The signature samples from the same person are similar but not identical. In addition, a person signature often changes radically during his life. The reason of the ineffectiveness of the signature verification systems is the huge amount of the input data produced by the features extraction procedures.

We propose two-step vectorization procedure to reduce the number of significant signature features and the input data for the classifier system. In the first step of vectorization process we defined a contour segments extraction algorithm to generate the discrete sets of signature contour points. These points were approximated by the cubic Bezier curves ([8], [10]) in the second step of the procedure. We applied a neural network trained by the real-coded genetic algorithm as the signature patterns classifier.

The remainder of the paper is organized as follows. In section 2 we define the signature image preprocessing procedures with application of the cubic Bezier curves. In section 3 we define the classification method. Results of performed numerical experiments are presented in Section 4. The paper ends with some final remarks.

2 Signature image preprocessing with application of the Bezier curves

The first stage of signature verification is an image preprocessing. We usually define the signature images by the scanning of the handwritten signatures. Those scans are modified in *binarization*, *noise reduction* and *skeletonization* processes [12].

We used the *global threshold method* for binarization [12]. The main goal of the noise reduction in binary images is to eliminate single white pixels on black background and single black pixels on white background. We applied the *branch cutting procedure*, which was also used in [1]. We defined a 3×3 mask to the image with a simple decision rule: if we can find only one pixel of the 8-neighbors [1] of a pixel in the center of the mask of the same color as the central one, we reverse the color of the central pixel. After noise reduction we applied the *morphological thinning algorithm* [12] to produce a skeleton of the signature.

2.1 Handwritten signature vectorization with Bezier curves

The obtained signature skeleton was modified in a vectorization process [6]. The pattern vectorization procedure defined in this paper was performed in two main steps:

- **Step1** : curve tracking,
- **Step2** : approximation with cubic Bezier cuves.

In curve tracking step we divided the signature skeleton into several contour segments, which can be defined as the discrete finite sequences of signature contour points. We divide the set S of pixels representing the signature skeleton into two subsets: P_e - the set of all edge points of the signature and $P_a = S \setminus P_e$. The edge points of the signature are defined as the pixels which have only one 8-neighbor[1]. We can apply the 3×3 mask as in the noise reducing procedure to find the elements of the subset P_e.

Let us denote by: $c : \mathbb{N} \to \mathbb{R}^2$ - a finite sequence of contour points (pixels) in a given single contour segment, P_c - a set of c sequences, $N_8(\bar{b})$ - an 8-neighbors of the current point $\bar{p}$.

The general framework of the proposed contour segments extraction algorithm is defined in the following way:

0. Set $i := 1$, $P_c := \emptyset$ and $P_{ci} := \emptyset$ for each $i \in \mathbb{N}$.
1. Define the P_e and P_a sets.
2. If $P_a \neq \emptyset \wedge P_e = \emptyset$, remove one randomly chosen element from P_a and go back to step 1.
3. If $P_a = \emptyset \wedge P_e = \emptyset$, exit.
4. Select randomly one element from P_e and define it as $\bar{p}$.
5. Find all elements of P_a belonging to $N_8(\bar{p})$.
6. If $N_8(\bar{p}) = \emptyset$, remove $\bar{p}$ from P_e and add it to the sequence c, add the sequence c to the set P_c, set $i = i + 1$ and go back to step 2.
7. If $|N_8(\bar{p})| = 1$ then $\bar{q}$ is the unique 8-neighbor of $\bar{p}$. Remove $\bar{q}$ from P_a, add it to the sequence c and go to Step 5.

8. If $|N_8(\bar{p})| > 1$, add $\bar{p}$ to the sequence c, add the sequence c to the set P_c , set $i := i + 1$ and go back to step 2.

In the second step of the vectorization process we applied cubic Bezier curves to approximate the set of contour points in a given signature contour segment. Our method is a simple modification of the curve-fitting algorithm applied in [6]. The Bezier curves are widely used in a shape description methods in the fields of pattern recognition and computer vision [3], [7].

The general formula for a cubic Bezier curve is [10]:

$$B(t) = P_0(1-t)^3 + 3P_1(1-t)^2t + 3P_2(1-t)t^2 + P_3t^3 \tag{1}$$

where

$$P_0 = (x_0, y_0),\ P_1 = (x_1, y_1),\ P_2 = (x_2, y_2),\ P_3 = (x_3, y_3)$$

are four control points set for that curve.

Figure 1 shows an example of the cubic Bezier curve.

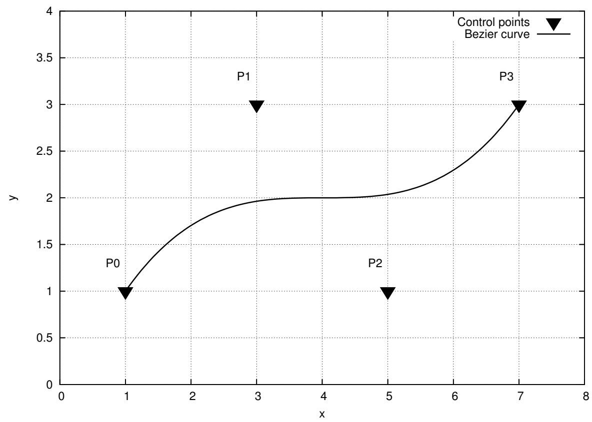

Figure 1. The example of the cubic Bezier curve

We define two edge control points of the cubic Bezier curve (denoted by P_1 and P_4) as the edge points c_0 and c_n in a given contour points sequence $c_0, ...c_n$. These edge points belong to the curve. We wish to move the remaining two control points P_2 and P_3, which are off the curve, to fit the contour segment as well as possible. Figure 2 presents the contour points represented by pixels and the cubic Bezier curve generated for the approximation.

We should minimize the approximation error function E to fit the signature contour points. This error function is defined by the following formula:

$$E(P_1, P_2) = \sum_{i=1}^{n} d^i, \tag{2}$$

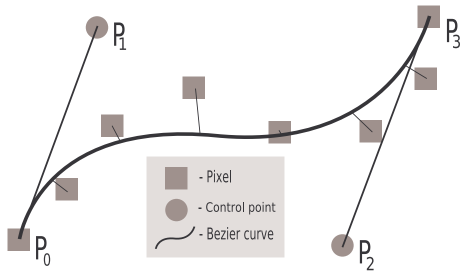

Figure 2. Signature contour points approximation with the cubic Bezier curve

where $c_0, ..., c_n$ is the contour points sequence and d^i denotes the distance of the point c_i from the Bezier curve $B(t)$.

We used in this paper the Fibonacci Search Algorithm [4] as the minimization procedure.

2.2 General features extraction

Signature vectorization process is followed by the signature features extraction procedures. Features extraction is the process of choosing input to the pattern recognition system. Usually several features are required to be able to adequately distinguish inputs that belong to different classes. The key of the features extraction is to reduce the problem data into manageable amount of information without discarding valuable or vital information.

We defined in this paper the following global significant features defined in [1]:

- Height-to width ratio
- The number of local maximums of vertical projections
- Vertical and horizontal center of the signature
- Baseline shift - the measure of the general orientation of the signature calculated as a difference between vertical centers of gravity to left and right half of the pattern
- Global slant angle
- Numbers of the edge and cross points

The vector of features in our case had 8 coordinates. These coordinates were accepted as the input signals for the neural network classifier.

3 Signature patterns classification with neural network

Classification assigns input data into one or more classes based on the extraction of significant features or attributes. We applied a multi-layer feed-forward neural networks for the signature patterns classification.

For experimentation purposes, the following issues need to be considered:

- The organization of the training and test sets
- Design of network structure
- Selection of a learning algorithm

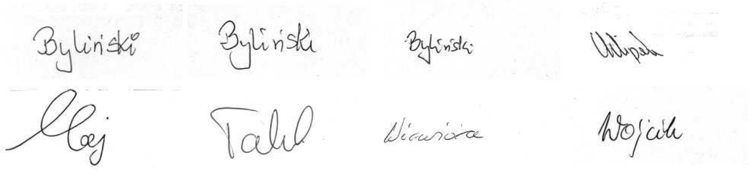

Figure 3. Example of training set

- Network testing and performance evaluation

These points are discussed in the following sections.

3.1 The organization of the training and test sets

In our experiments we applied a signature database defined in [2]. The database contains 157 signature patterns taken from 6 persons. Each person was asked to sign 7 or 8 times and to try to forge the other people signatures.

As the training set we defined the set of several genuine signatures of the given person and one or two genuine signatures of the other people as the examples of forgeries. All remaining genuine signatures and all forgeries define the test set.

The simple example of the training set applied in our experiments is presented on Figure 3.

3.2 Network architecture and learning algorithms

The classification of the signature as a genuine or forgery is performed by using an artificial neural network. We construct the 3-layer feed-forward network with 8 inputs due to the number of extracted pattern features. The implemented network contains two hidden layers. We defined eight neurons with sigmoid activation function

$$f(x) = \frac{1}{1+e^{-\beta x}} \tag{3}$$

with the scaling parameter $\beta = 0,1$ in the input layer and in each hidden layer. The output layer contains only one neuron. The output signal is the number from the interval $[0, 1]$. It is assumed that the output signal close to 1 represents a genuine signature, and the signal close to 0 means that the signature is a forgery.

We train the neural network using standard backpropagation method [11], *real-coded genetic algorithm* (RCGA) [5] and hybrid RCGA algorithm. The hybrid genetic algorithm is a two-steps procedure. In the first step we applied the RCGA algorithm to modify the weight coordinates. The fitness function for that algorithm is a mean square error function:

$$E = \frac{1}{M}\sum_{i=1}^{M}(y_i - d_i)^2 \tag{4}$$

where M is the number of elements in a training set, y_i is the current network output, d_i is the theoretical network output for an i-th element in the training set. We used BLX-α

Table 1. The number of signatures used in tests

Number of	Test1	Test2	Test3
genuine signatures	3	5	5
forgeries	5	10	10

crossover operator [5] with $\alpha = 0,5$ and non-uniform mutation defined in the following way: a mutated value of a gene c_i in a chromosome $C = (c_1, ..., c_i, ..., c_n)$ is calculated by the formula

$$c_i' = \begin{cases} c_i + \Delta(t, b_i - c_i) & \text{, for } \tau = 0 \\ c_i - \Delta(t, c_i - a_i) & \text{, for } \tau = 1 \end{cases}$$

where τ is a random number from the interval $\{0, 1\}$. The parameter Δ is calculated in the following way:

$$\Delta(t, y) = y \left(1 - r^{\left(1 - \frac{t}{g_{max}}\right)^b}\right),$$

where r randomly chosen number from the unit interval, t is the current generation, g_{max} is the maximum number of generation and b is the parameter that determines the degree of dependence on the generation number. The RCGA algorithm was stopped after the assumed maximal number of the executed generations.

After stopping the RCGA algorithm we activated the local search method which was the Fibonacci Search Algorithm [4]. The local search procedure started in the neighborhood of the best adapted individual found by RCGA algorithm in the first step of the network training procedure.

3.3 Network testing and evaluation

Trained networks were tested using the test sets described in the section 3.1 with the purpose of testing the memorization and generalization abilities of each network. Tests were also aimed at analyzing the ability of each net to classify the different types of forgeries correctly.

The experimental results were reported in terms of false rejection rate and false acceptance rate. *False rejection rate* (FRR) is the percentage of genuine signatures rejected as false. *False acceptance rate* (FAR) is the percentage of false signatures accepted as genuine.

4 Test results

We performed three series of numerical tests to examine the effectiveness of the proposed method. In each test we set the different values of the parameters of the RCGA algorithm and we applied the different numbers of signatures to train the network. The genetic algorithm parameters applied to the network training and the numbers of the signature patterns used in each group of experiments are presented in tables 1 and 2.

Each experiment was repeated 30 times in case of evolutionary algorithm applied as a training method. The results obtained for three implemented method of network training

Table 2. RCGA algorithm parameters

Parameter	Test1	Test2	Test3
crossover probability	0,7	0,7	0,5
mutation probability	0,05	0,05	0,05
population size	50	50	30
maximal error rate	0,000001	0,000001	0,000001
maximal numbers of generations	2000	2000	2000

Table 3. Test results

Training method		Test1	Test2	Test3
RCGA	FAR	21 %	20 %	34%
	FRR	22 %	19 %	33 %
hybrid RCGA	FAR	33 %	28 %	28 %
	FRR	13 %	6 %	3 %
Backpropagation	FAR	100 %	100 %	100 %
	FRR	100 %	100 %	100 %

Table 4. Test results obtained in [2]

	Test1	Test2	Test3
FAR	32,4 %	28,5 %	31,7 %
FRR	20,5 %	19,4 %	23,6 %

are reported in Table 3. The results for genetic approaches are the average rates obtained in 30 runs.

False acceptation rates in both cases of genetic training methods are significant and similar. False rejection rates were reduced very fast (2 times in Test1, 3 times in Test2 and 10 times in Test 3) after hybridization of the RCGA algorithm.

In case of backpropagation training method the network answers were incorrect in each run, which surprised us. The error function values were greater than 0.5 .

The obtained results were compared with the results of similar tests performed in [2]. We repeated those experiments. The results are presented in Table 4. In that case only RCGA training method was used and the input vector of the signature pattern features had 16 coordinates.

The results for the same training method are comparable, but the input data structure is twice less complex after vectorization procedure proposed in this paper. The average execution time was also reduced from 34 to 10 seconds.

5 Conclusions

- In this paper we introduced a contour segments extraction algorithm to generate the discrete sets of signature contour points and we applied the cubic Bezier curves to approximate those sets. A multi-layer neural network trained by the real-coded genetic algorithm was proposed as the signature patterns classifier.
- This method was tested on the set of 157 signatures.
- The experimental comparison analysis proves that proposed vectorization technique can reduce the set of the signature significant features with no negative effect on the results.
- We want to improve the approximation procedure applying the derivations of the Bezier curve.

Bibliography

[1] H. Baltzakis and N. Papamarkos. A new signature verification technique based on a two-stage neural network classifier. *Engineering Applications of Artificial Intelligence*, 2001(14):95–103, 2000.

[2] B. Bylińska. Off-line signature verification using artificial neural network. In *Proc. of VIII KAEiOG, Korbielów, 30.05-1.06.2005*, pages 31–36, 2005.

[3] H.H. Chang and H. Yan. Vectorization of hand-drawn image using piecewise cubic bezier curves fitting. *Pattern Recognition*, 31(11):1747–1755, 1998.

[4] D.E.Knuth. *The Art of Computer Programming, vol 1,2,3.* A-Wesley, 1998.

[5] J.L. Verdegay F. Herrera, M. Lozano. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. In *Artificial Intelligence Review*, 1998.

[6] J.-J. Lu H.-M. Yang and H.-J. Lee. A bézier curve-based approach to shape description for chinese calligraphy characters. In *Working Report*, 2001.

[7] S.Levialdi L.Cinque and A.Malizia. Shape description using cubic polynomial bezier curves. *Pattern Recognition*, 19(9):821–828, 1998.

[8] L.Shao and H. Zhou. Curve fitting with bezier cubics. *Graphical Models and Image Prcessing*, 58(3):223–228, 1996.

[9] S. N. Srihari R. Plamondon. On-line and off-line handwriting recognition: A comprehensive survey. *1EEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

[10] T.W. Sederberg and G.J.Wang. A simple verification of the implicitization formulae for bezier curves. *Computational Aided Geom. Design*, 11(2):222–228, 1994.

[11] R. Tadeusiewicz. *Artificial Neural Networks (in Polish).* Akademicka Oficyna Wydaw. RM, 1993.

[12] R. Tadeusiewicz. *Computer Analysis and Image Processing (in Polish).* Wydawnictwo Fundacji Postpu Telekomunikacji, 1997.