International Journal of Fuzzy Logic and Intelligent Systems, vol. 2, no. 3, December 2002 pp. 237-241

Representation of hand written decimal digits by a sequence of fuzzy sets

Byung Soo Moon and In Koo Hwang

Korea Atomic Energy Research Institute
P.O. Box 105, Taeduk Science Town, Daejeon 305-600, Korea
E-mail:bsmoon@kaeri.re.kr
Fax:+82-42-868-8916

Abstract

In this paper, we describe how to represent hand witten decimal digits by a sequence of one to five fuzzy sets. Each fuzzy set represents an arc segment of the digit and is a Cartesian product of four fuzzy sets; the first is for the arc length of the segment, the second is for the arc direction, the third is for the arc shape, and the fourth is a crisp number indicating whether it has a junction point and if it has an end point of a stroke. We show that an arbitrary pair of these sequences representing two different digits is mutually disjoint. We also show that various forms of a digit written in different styles can be represented by the same sequence of fuzzy sets and hence the deviations due to different writers can be modeled by using these fuzzy sets.

Key Words : Fuzzy Sets; Decimal Digits; Character Strokes, Arc Segments, Thinning, Curve Extraction, Curve Smoothing; Writing Sequence

1. Introduction

We have shown in our earlier work[1] that hand written English alphabets can be modeled by a sequence of one to seven fuzzy sets. In this paper, we apply a modified version of the same method to decimal digits to see if the method can be extended to represent the decimal digits and if it can be improved.

There have been many studies on the recognition of handwritten characters and many reports on applications of optical character recognition systems. However, there are still many problems to be resolved, including how to handle the deviations from one writer to another, how to handle the errors brought in during the thinning process, how to handle the limitations in the capability of the machine to extract features that distinguish different characters.

L. Chen[2] proposes a method which does not utilize a thinning process and use a segment partition algorithm instead. He tried to partition characters into segments, horizontal or vertical, and synthesize them into curved strokes. C. Lee and B. Wu[3] studied stroke extraction for Chinese characters. They determine character segments by finding the boundary points and applying a contour tracking algorithm. J. Zhou et al.[4] studied a verification scheme to correct some of the errors in the neural network based recognizers, where the errors seem to come mostly from neglecting segment features.

In this paper, we describe a method for representing handwriting sequence of the ten decimal digits. The deviations due to different writers and the character features are handled by using a sequence of fuzzy sets. We break down the writing sequence into arc segments by taking the extreme points in the vertical direction and the inflection points of the plane curve as the segment boundary points. The extreme points are where sharp changes in the vertical direction are made during the writing. This is based on C. Remi et al.[5] where they found through experiments that higher level children tend to break up graphs into segments like adults would normally do while drawing graphs.

In the paper by S. Djeziri et al.[6], we also find that handwriting sequence can be broken into segments where the writer makes a small or large breaks during the course of writing. They use a kinematic theory of handwriting generation to analyze the handwriting kinematically and show that there are large or small breaks in the writing sequence. We consider these breaks are at extreme points of the curve in the vertical direction and take them to be the segment boundaries in our segmentation. The inflection points of a stroke are the points where the center of the curvature moves from one side of the moving track to the other side. By taking the inflection points as end points of arc segments, the 'S' type curves are broken into two parts.

We start with a gray scale image of a handwritten decimal digit as shown in fig.1 and convert it to a binary image consisting of 0's and 1's by setting the pixel value to 1 whenever its original value is greater than say 128 and 0 otherwise. Hence, we will assume throughout this paper that the image is in binary form. We will apply a thinning algorithm based on a gradient operator to obtain a skeleton of the image. For each continuous component of the thinned

This work has been carried out under the nuclear research and development program supported by the Ministry of Science and Technology of Korea.

image, we will derive a curve function $z = f(x, y) = C$ so that it expresses the hand writing sequence. Recall that the variable takes only one of the two values 0 and 1, and hence only the nonzero pixels $Z_n = (x_n, y_n)$ can be used to represent the image. We will order the sequence so that the ordering is equivalent to the normal hand writing sequence. We then break the sequence into arc segments whenever its vertical component makes a direction change.

We will represent each arc segment of a digit by the Cartesian product of four fuzzy sets; the first of which is for the arc length, the second is for the arc direction, the third is for the arc shape, and the fourth is to indicate whether it has a junction point and/or whether it has an end point of the stroke. We will show that these sequences of fuzzy sets are mutually disjoint so that the representation is valid.

2. Extracting the curve function and its arc segments from an image

In this section, we describe how the gray scale image of a decimal digit is thinned and how the curve $Z_n = (x_n, y_n)$ is extracted from the thinned image. First, we consider how the skeleton of a image shown in fig.3 can be obtained. We apply a gradient operator[7] shown in Table 1 to the image in fig.1 so that the boundary of the image shown in fig.2 is obtained. We delete the boundary pixel from the image whenever the deletion does not leave the image disconnected; this check is done by using its Hilditch's crossing number. The above process is repeated as many times as necessary until we obtain a thinned skeleton, examples of which are shown in fig.3.

Fig. 1 Gray scale images of '3'

Table 1. A Gradient Operator

	J=1-1	J=1	J=1+1
i=k-1	(1,1)	(4,0)	(1,-1)
I=k	(0,4)	(0,0)	(0,-4)
I=k+1	(-1,1)	(-4,0)	(-1,-1)

Next, we consider how the curve function is derived from the thinned image. The starting point (x_1, y_1) is selected so that it is the same as the starting point of the normal writing sequence. To define the next point (x_2, y_2) in the curve, we search for nonzero neighbor pixels. Note that there is just one nonzero neighbor pixel for the digits '1', '2', '3', '6', '7' and the nonzero pixel is taken as (x_2, y_2). For the other digits, '4', '5', '8', '9' where either the starting point matches with the end point of the stroke or the starting point of a stroke is the same as the starting point of another stroke, we choose (x_2, y_2) to be the next pixel in the normal writing sequence. To determine (x_{n+1}, y_{n+1}) from (x_n, y_n), note that there are at most two nonzero neighbor pixels unless it is a junction point. If it is a junction point, we choose the pixel in the direction which is closer to the previous moving direction and mark the pixel previous to the junction point deleted so that the curve will not pass through the same path on the way back passing through the junction. Next, we apply a smoothing algorithm described in [9] to obtain the smoother curves shown in fig.4.

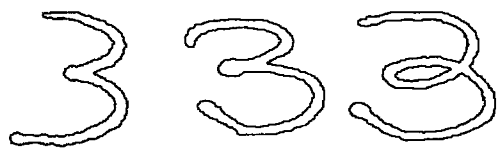


Fig. 2 Boundary of the image in fig.1

Fig. 3 Thinned images of letter '3' in fig.1

Finally, we consider how the smoothed curve $\{(x_n, y_n) \mid n = 1, 2, \Lambda N\}$ can be broken into arc segments. First, we normalize the curve so that we have min $\{x_n\}$=min$\{y_n\}$=0 and max $\{x_n\}$=min$\{y_n\}$=1, for all digits except for '1'. For the case of '1', we simply normalize the y-coordinates and use the same scale to rescale the x-coordinates. Next, consider the sequence of vectors $\chi_j = (x_{j+1} - x_j, y_{j+1} - y_j)$, that are the tangential vectors to the curve even though they are not normalized. One can take the derivative of the unit tangential vector to obtain the curvature vector and determine the inflection points as the points where the curvature vector changes its direction from left hand side of the curve to the right hand side, and vice versa. To determine the extreme points of the y-coordinates for the curve $z_n = (x_n, y_n)$, we locate points where the increment $\triangle y_n = y_{n+1} - y_n$ changes its sign. Fig.4 shows three examples of arc segments for the digit '3' where the curves are drawn after the smoothing algorithm is applied.

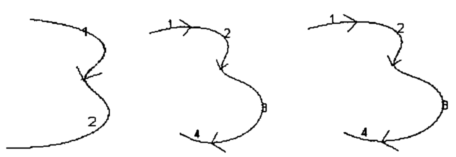


Fig. 4 Arc segments of a thinned image of digit '3' obtained by smoothing 2nd image of fig.3

3. Representation of an arc segment by fuzzy sets

As described in the above section, a plane curve $f(t) = (x(t),\ y(t))$ is extracted from the binary image of a hand written decimal digit and is divided into arc segments at points where the curve makes a sharp turn or at points where the direction of the curvature vector changes. In this section, we describe how an arc segment of the smooth curve can be represented by a fuzzy set. The fuzzy set will be defined as a Cartesian product of four fuzzy sets; the first is for the segment length and the second is for the overall direction of the arc, the third is for its shape, the fourth is a crisp number indicating whether it has a junction and whether it has an end point of the stroke.

For the first component, we fuzzify the arc length by using the fuzzy sets shown in fig.5. Recall that the curve was normalized so that min$\{x_n\}$=min$\{y_n\}$=0 and max $\{x_n\}$=min $\{y_n\}$=1, and hence an arc segment with maximum length would correspond to a half circle with diameter 1. Based on this, we define five fuzzy sets; 0,1,...,4 as shown in fig.4 and use them to fuzzify the arc length. We will use two more fuzzy sets; fuzzy set '7' for the union of sets '0', '1'and '2', and fuzzy set '8' for the union of '2', '3', '4'. These are used to take care of deviations due to different writings.

For the direction of an arc segment, we use the average of the unit tangential vectors on the arc and the angular coordinate $\theta \in |\,0.2\pi\,|$ of the resulting vector is used for its direction. The angle θ is represented by fuzzy sets shown in fig.6. The tangential vector is computed by

$$\xi_1 = dr/ds \approx (x_{j+1} - x_j,\ y_{j+1} - y_j)/d$$

where $d = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2}$, and the average of these unit vectors ξ_1 at each point on the arc is used for the direction. The angle θ for this direction is assumed to be 0 when the direction is along the positive x-axis and $\pi/2$ when it is along the positive y-axis. We use the spike functions or the triangular sets shown in fig.6 as fuzzy sets for fuzzification of the direction angle, where fuzzy set number corresponds to the angle $\frac{k\pi}{4}$.

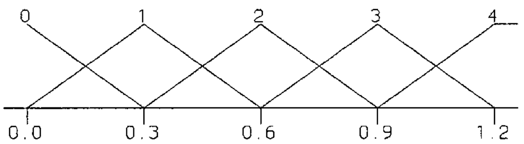


fig.5 Fuzzy sets for arc length

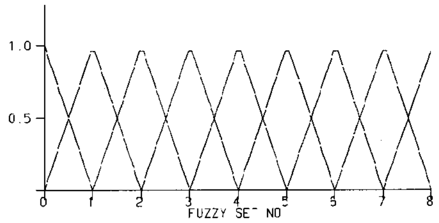


Fig.6 Fuzzy sets for direction of arc segments

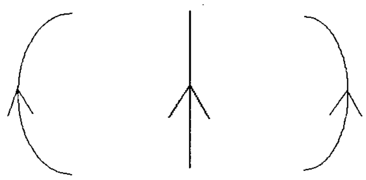


Fig. 7 Fuzzy sets for arc shapes (for writing from top to bottom)

For the arc shape, we use the 3 different shapes shown in fig.7. Note that the 'S' type arc segments are broken into two parts by using the direction change of the curvature vector. To determine the shape, we rotate the arc segment so that it lies on the positive x-axis with the beginning point at the origin. Then we compute the finite difference approximation of the derivative of the unit tangential vector ξ_1 with respect to s, where s is the arc length coordinate. Note that the result $\frac{d\xi_1}{ds}$ is $\frac{1}{\rho}\xi_2$, where ρ is the curvature and ξ_2 is the unit curvature vector. Since the arc segment lies on the positive x-axis and starts from the origin, the center of curvature lies in the upper half plane, i.e. $y > 0$ when the arc is curved toward left of the moving direction. Therefore, the second component of ξ_2 is positive for the part of the arc curved toward left. We take the average of the magnitude of $\frac{d\xi_1}{ds}$, equivalently the average of $\frac{1}{\rho}$ as the membership value.

The last is a crisp number which is '0' when there is no junction in the arc segment and no end point of the stroke, '1' when it has a junction point but no end point, '2' when it has an end point but no junction point, '3' when it has both a junction point and an end point. Note that a junction point must belong to at least two arc segments. We use the convention, however, that only the first arc segment is marked to have a junction point. We will use two more crisp numbers for this fuzzy set; '4' for the union of '0' and '1', and '5' for the union of '2' and '3'.

4. Representation of decimal digits

The fuzzy sets defined by the method described in the previous section for the ten decimal digits from '1' through '0' are shown in Table 2 and 3. Each of the four digit numbers in the tables represents a fuzzy set which is a Cartesian product of four fuzzy sets corresponding to a digit. For example, the number '7210' represents the first arc segment of digit '3' whose length corresponds to fuzzy set '7' which is a union of fuzzy sets '0', '1', '2' shown in fig.5. The second fuzzy set number '2' is for the arc direction shown in fig.6, where '2' is for the direction of $\frac{2\pi}{4}$.

The third fuzzy set number '1' is for the arc shape indicating the arc is curved toward right as shown in fig.7. The last digit '0' is for a crisp number indicating that there is no junction point and no end point of the stroke in this arc segment.

Table 2. Fuzzy set representation of decimal digits - 1

1	2	3	4	5	6
7220	7210	7210	8620	8724	7430
4725	5714	8714	4123	9710	5734
7125	7210	7210	4720	7413	8330
	8120	8710		8120	7520
		7410			

Table 3. Fuzzy set representation of decimal digits - 2

7	8-1	8-2	9	0
7723	8731	7430	8630	5730
4120	8731	8731	8230	5330
4720	8310	8310	9720	
	8330	8330		

All of the ten digits are represented by two to five fuzzy sets or equivalently by two to five arc segments. Note that for letters '1', '4', '5', and '7', there are two continuous components and hence there are segments with last digit '3' or '5' for these digits in Table 2 and 3. In order to check that the above provides an acceptable representation, we have to verify that they are disjoint, i.e. any pair of digits can be distinguished. Two sequences of fuzzy sets are different when

1) the number of continuous components are different
2) the number of fuzzy sets are different
3) there exists a pair of corresponding fuzzy sets that are disjoint

We will say two fuzzy sets are disjoint if any of the four component fuzzy sets are mutually disjoint. For the case of the fuzzy sets for arc shape shown in fig.7, we can see that the fuzzy sets are disjoint only when one is '1' and the other is '3'. Similarly for the arc length and for the arc direction, any pair of fuzzy sets will be considered disjoint only when the corresponding fuzzy set numbers differ by 2 or more. Note that the fuzzy set 8 for the arc direction must be considered as a neighbor set to fuzzy set 1.

Theorem 1. Any pair of fuzzy set representations of decimal digits shown in Table 2 and 3 are mutually disjoint except for the two different representations of '8'.

Proof. Note that if the number of segments or fuzzy sets for two different digits is different, then they must be considered as disjoint and that the fuzzy sets starting with arc length '7' as the first digit of the four digit number must be considered for the two different cases; one as a zero-length arc, i.e. none-existing arc segment, and the other as a none-zero length arc segment.

First, we consider the case of 5 segments. There is only one case of digit '3' with none of the segment with length '7' is zero and hence it is disjoint from others. Next, we consider the case of 2 segments only. There are four cases;

"1-1"=(4723, 7123), "1-2"=(7220,4723),
"3"=(8714,8710), "7"=(4120,4720),
"0"=(5730,5330).

"1-1" is disjoint from others in the direction of 2nd arc segment, "1-2" is disjoint from others in the direction of 1st arc or the length of the 1st arc. "3" is disjoint from "7" and "0" in the direction of 1st and 2nd arc respectively. "7" and "0" are disjoint in the direction of 1st arc.

Next, we consider the cases where there are 3 arc segments, i.e. the following 13 cases;

"1"=(7220,4725,7125), "2-1"=(5714,7210,8120),
"2-2"=(7210,5714,8120), "3-1"=(8714,8710,7410),
"3-2"=(8714,7210,8710), "3-3"=(7210,8714,8710),
"4"=(8620,4123,4720), "5-1"=(8724,9710,8120),
"6-1"=(5734,8330,7520), "6-2"=7430,5734,8330),
"7"=(7723,4120,4720), "8-1"=(8731,8710,8210),
"9"=(8630,8230,9720).

"1" is the only digit with junction in the 3rd arc and hence it is disjoint from others. "2-1" is disjoint from "3-3" in the direction of 1st arc, disjoint from "6-1","6-2","8-1","9" in the 1st arc shape, disjoint from "3-1", "3-2", "4", "7" in the direction of 3rd arc. "2-2" is disjoint from all others in the direction of 1st arc except for "3-3" whose direction in the 3rd arc is disjoint. Similar arguments can be used to prove that all the digit representations with three arc segments are mutually disjoint.

Finally, for the representations with four segments, we have;

"2"=(7210,5714,7210,8120),
"3-1"=(8714,7210,8710,7410),
"3-2"=(7210,8714,8710,7410),
"3-3"=(7210,8714,7210,8710),
"5"=(8724,9710,7413,8120),
"6"=(7430,5734,8330,7520),
"8-1"=(8731,8731,8310,8330),
"8-2"=(7430,8731,8710,8210)

A routine check to verify that these fuzzy sets are mutually disjoint is omitted. Q.E.D.

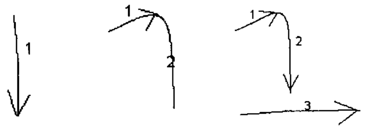


Fig. 8 Arc Segments of the Digit '1'

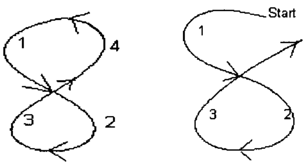


Fig. 9 Arc Segments of the Digit '8'

To see how the deviations due to different writings are modeled by our representation, consider the three different writings in fig.8. There is no doubt that the 3rd writing that has 3 arc segments, is represented by '7220', '4723', '7123'. The 2nd writing that has only two arc segments is also represented by the 3 fuzzy sets by interpreting the 3rd segment '7123' as an arc segment with length 0. If we consider both '7220' and '7123' as arc segments with zero length, then the 1st writing can also be represented by the same three fuzzy sets. Similar arguments for different writings of digit '3' in fig.4, and of digit '8' in fig.9 are omitted.

5. Conclusion

We have shown how gray scale images of hand written decimal digits can be thinned, how the curve functions can be extracted, and how arc segments are derived from the curves after smoothing. We also showed that a sequence of two to five fuzzy sets can be used to represent a handwritten decimal digit and that a certain range of deviations can be modeled by using these fuzzy sets. Our numerical representation is obtained by looking at how a man writes the digit and no training is used for the representation. For the identification of a digit, we only need to take fuzzy set intersections with predefined sequences of fuzzy sets for ten decimal digits.

References

[1] B.S. Moon, et al., "Representation of Hand Written English Characters by a Sequence of Fuzzy Sets", 2002 Annual Meeting of the North American Fuzzy Information Processing Soc. Proc. 322-326.

[2] Ling-Hwei Chen, "A new approach for handwritten character stroke extraction", Computer Processing of Chinese and Oriental Languages 6(1) (1992) pp.1-17.

[3] C. Lee and B. Wu, "A Chinese-character-stroke-extraction algorithm based on contour information", Pattern Recognition, 31(6) (1998) pp.651-663.

[4] J. Zhou, A. Krzyzak C. Suen, "Verification - a method of enhancing the recognizers of isolated and touching handwritten numerals", Pattern Recognition, 35(3) (2002) pp.1179-1189.

[5] C. Remi, C. Frelicot, P. Courtellemont, "Automatic analysis of the structuring of children's drawings and writings", Pattern Recognition, 35(3) (2002) pp.1059-1069.

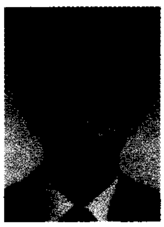
[6] S. Djeziri, et al., "Learning handwriting with pen-based systems: Computational issues", Pattern Recognition, 35(3) (2002) pp.1049-1057.

[7] B. S. Moon, "A representation of digitized patterns and an edge tracking thinning algorithm", Pattern Recognition, 34 (2001) pp.2155-2161.

[8] Alberto S. Aguado, et al., "Invariant characterization of the Hough transform for pose estimation of arbitrary shapes", Pattern Recognition, 35(3) (2002) pp.1083-1097.

[9] B.S. Moon, "A curve smoothing method by using fuzzy sets", Fuzzy Sets and Systems, 96 (1998) pp.353-358.

Byung Soo Moon
1974 - Univ of Illinois,
PhD in Mathematics
1974-1978 Sargent & Lundy Enginners (Chicago)
1978- Korea Atomic energy Research Institute
Areas of Interests: Fuzzy Systems, Neural Networks
Phone : 042-868-2980
Fax : 042-868-8916
E-mail : bsmoon@kaeri.re.kr



In Koo Hwang
1982-1886 InHa Univ of Korea, BS in Electrical Engineering
1987-1990 InHa Univ of Korea, MS in Electrical Engineering
1986 - Korea Atomic Energy Research Institute
Areas of Interests: Signal Processing, Instrumentation Sensors
Phone : 042-868-2925
Fax : 042-868-8916
E-mail : ikhwang@kaeri.re.kr