

# 획의 일부분이 손상된 숫자가 포함된 필기체 숫자 열의 인식

김계경\*, 김진호\*\*, 조수현\*, 지수영\*, 정연구\*

\*한국전자통신연구원

\*\*경일대학교 전자정보공학과

e-mail : (kyekyung, jsh62860, chisy, ykchung)@etri.re.kr, kjinho@bear.kyungil.ac.kr

## Recognition of numeral strings with broken digits

Kye-Kyung Kim \*, Jin-Ho Kim \*\*, Soo-Hyun Cho\*, Soo-Young Chi\*, Yun-Koo Chung\*

\* Electronics and Telecommunications Research Institute

\*\*Dept. of Electronic Engineering, Kyungil University

### 요 약

본 논문에서는 획의 일부분이 손상된 숫자(broken digit)나 붙은 숫자(touching digits)와 같은 비정형 숫자들이 포함된 필기체 숫자 열을 인식할 수 있는 방법에 대하여 제안하였다. 비정형 숫자들은 분류(pre-segmentation) 단계에서 숫자들의 구조적인 특징 정보를 이용하여 정형인 개별 숫자(isolated digit)로부터 획의 일부분이 손상된 숫자 또는 붙은 숫자들로 분류된다. 획의 일부분이 분리된 숫자의 결합 및 붙은 숫자들의 분할 단계를 거쳐 인식을 시도하였다. 제안된 방법의 타당성을 증명하기 위하여 NIST SD19 데이터베이스를 이용하여 시뮬레이션 해 보았다.

### 1. 서론

문자 인식 분야 가운데 필기체 숫자 열의 인식은 응용 분야가 넓기 때문에 가장 활발히 연구되어 온 분야 [1-5] 가운데 하나이다. 필기체 숫자 열의 인식은 독립된 개별 숫자(isolated individual digit)를 인식하는 것과는 달리 획의 일부분이 손상된 숫자 (broken digit) 및 붙은 숫자(touching digits)들을 인식하는 것을 포함하고 있다. 그러나, 숫자 열 안에서 이러한 숫자들을 인식하는 것은 매우 어려우며 오인식의 가장 큰 원인으로 여겨지고 있다 [3].

숫자 열을 인식하는 방법은 숫자 열에 존재하는 숫자들을 개별 숫자 또는 붙은 숫자로 미리 분류(pre-segmentation)하여 인식하는 방법 [3,4] 과 사전 분류 없이 인식하는 방법 [5] 으로 나눌 수 있다. 미리 분류하여 인식하는 전자의 방법에서는 분할 기반 방법(segmentation-based method)으로 숫자들을 개별 숫자 또는 붙은 숫자 그룹 가운데 하나로 분류한 다음 숫자를 인식한다. 이때 필기자의 습성에 따라 획의 일부분이 쉽게 분리되는 숫자들을 검출하여 분리된 획을 결합하여 인식하고자 하는 시도가 이루어졌다. 한편,

붙은 숫자들은 분할 기반 방법 즉, 인식 무기반 분할 방법(recognition-free segmentation method), 인식 기반 분할 방법(recognition-based segmentation method)을 이용하여 인식되어 왔다. 또한, 분할로 인한 오류를 줄이기 위하여 무분할 인식 방법(holistic method)이 제안되기도 하였다.

반면, 사전 분류 없이 숫자를 인식하는 후자의 방법에서는 숫자들로부터 분할 특정 점들을 구하여 얻어진 획들을 조합하여 인식 기반 방법에 의해 숫자들을 인식하여 왔다. Dynamic programming 은 개별 숫자에 가장 가까운 인식 결과를 가지는 획들의 조합을 찾아내는데 사용되어 왔다. 그러나, dynamic programming 은 계산량이 많고 특히 숫자 열 안의 숫자가 증가할수록 오인식 확률도 같이 증가한다는 단점이 있다.

본 논문에서는 오인식의 원인이 되는 손상된 획을 가지는 숫자나 붙은 숫자를 포함하는 숫자 열을 인식할 수 있는 방법을 제안하였다. 손상된 획을 검출하여 결합하고 붙은 숫자를 분할하기 위하여 인식 기반 분할 방법을 이용하였다. 숫자의 구조적인 특징 정보를 이용하여 숫자 열 안의 숫자 분류나 붙은 숫자를 분할하도록 하였다. 제안된 방법의 타당성을 검증하기

위하여 NIST SD19 데이터베이스를 이용하여 시뮬레이션 해보았다.

2. 분류

분류는 숫자 열 안에 포함된 숫자가 세 가지 숫자 그룹 즉, 독립된 개별 숫자, 붙은 숫자 또는 손상된 획을 가진 숫자인가를 인식 전에 미리 구분하는 단계이다. 먼저, 숫자 열 안에 존재하는 연결 요소(connected component; CC)를 찾고 각각의 연결 요소에 대하여 다음과 같은 숫자의 구조적인 특징 정보와 인식 결과를 이용하여 숫자 그룹을 분류하도록 하였다.

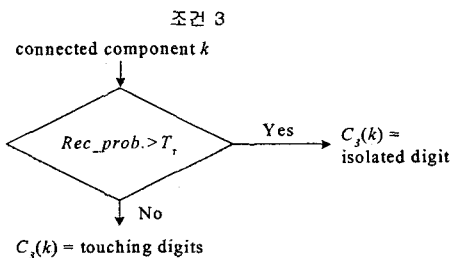
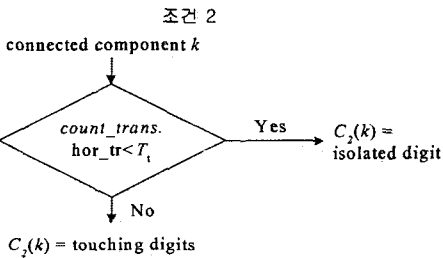
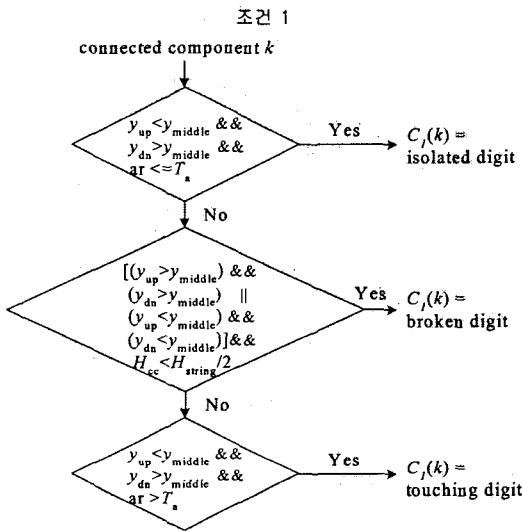


그림 1. 숫자 열 안의 숫자 분류

Fig. 1. Classification of digits in numeral string.

그림 1에서  $y_{up}$ 는 연결 요소의 상한 경계선(upper

boundary),  $y_{dn}$ 는 하한 경계선(lower boundary),  $y_{middle}$ 는 숫자 열의 1/2 높이에 해당하는 좌표 값을 나타낸 것이다.  $H_{cc}$ 는 연결 요소의 높이,  $H_{string}$ 은 숫자 열의 높이를 나타낸 것이다.

조건 1은 연결 요소  $k$ 가  $y_{middle}$  좌표 값을 지나는지 여부와 또한, aspect ratio를 계산하여 연결 요소  $k$ 를 숫자의 세가지 그룹 가운데 한 그룹으로 분류한다. 조건 2와 3은 각각 수평 전이 값(horizontal transition) 및 인식 확률 값(recognition probability)을 계산하여 각 숫자 그룹으로 분류한다. 이때  $T_a, T_t$  및  $T_r$ 은 임계치(threshold)로 각 숫자 클래스에 따른 개별 숫자들의 aspect ratio, 전이 값 및 인식 확률 값을 계산한 것이다. 조건 1-3의 결과들로부터 연결 요소  $k$ 는 다음 식 (1)의 분류 결정치  $C(k)$ 에 의해 한 개의 숫자 그룹으로 분류된다.

$$C(k) = F[C_1(k), C_2(k), C_3(k)] \quad k = 1, 2, \dots, n_{cc} \quad (1)$$

$$F(\cdot) = C_3(k)[C_1(k) + C_2(k)] + C_1(k)C_2(k)$$

$$C_1(k) = \Gamma(a_k, T_a(i))$$

$$C_2(k) = \Gamma(t_k, T_t)$$

$$C_3(k) = \Gamma(r_k, T_r)$$

(2)

$$\text{where } \Gamma(a, b) = \begin{cases} 1 & \text{if } a \geq b \\ 0 & \text{o.w.} \end{cases}$$

$$T_a(i) = \frac{1}{N_d} \sum_{j=0}^{N_s} a_{ij}$$

$$T_t = \frac{1}{N_s} \sum_{j=0}^{N_s} t_{ij}$$

$$T_r = \frac{1}{N_s} \sum_{j=0}^{N_s} r_{ij} \quad i = 0, 1, 2, \dots, N_d \quad (3)$$

식 (1)에서  $n_{cc}$ 는 숫자 열 안의 연결 요소 개수를 나타낸 것이다. 식 (2)에서  $a_k, t_k$  및  $r_k$ 는 연결 요소  $k$ 의 가로 세로의 비(aspect ratio), 수평 전이 값 및 인식 확률 값을 나타낸 것이다. 식 (3)은 학습시킨 숫자 클래스  $N_d$ 에 대한 샘플  $N_s$ 로부터 숫자의 가로 세로의 비, 수평 전이 값 및 인식 확률 값의 임계치를 계산한 것이다.

그림 2는 손상된 획을 가진 숫자 열의 예를 나타낸 것이다. 획의 일부분이 손상된 숫자는 필기 도구에 따른 저해상도 숫자나 셀 안에 쓰여진 숫자를 분리해내는 과정에서 많이 발생하게 된다.

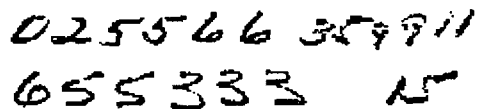


그림 2. 손상된 획을 가진 숫자들을 포함한 숫자 열의 예

Fig. 2. Example of numeral strings with broken digits.

일반적으로 획의 일부분이 손상된 숫자는 숫자 "4"와 숫자 "5"에 대해서 주로 해석되어져 왔다. 숫자 "4" 및 숫자 "5"는 흔히 수직 방향의 획과 수평 방향의 획이 각각 숫자 부분에서 분리되어진다. 또한, 해상도가 낮은 숫자나 특정 블록 안에 쓰여진 숫자를 분리하는 단계에서 획의 일부분이 손상된 숫자가 나타나기도 한다. 분류단계에서는 이러한 숫자들을 인식 단계 전에 미리 해석하여 오인식의 원인을 줄일 수 있도록 하였다.

그림 3 은 숫자의 구조적인 특징 정보를 이용하여 손상된 획을 가진 숫자를 검출하고 보상 결합(composite merging)시킨 숫자들을 나타낸 것이다.

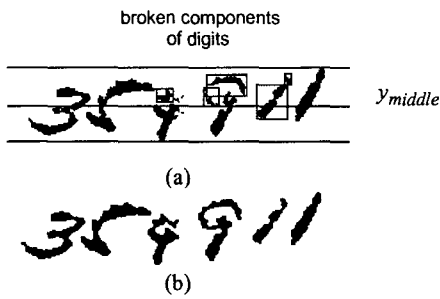


그림 3. (a) 분리된 획을 가진 숫자들의 검출과 (b) 보상 결합된 숫자 열

Fig. 3. (a) Detection of broken digit and (b) numeral string merging by composite method.

숫자로부터 분리된 획을 가진 숫자들은 그림 1 의 조건 1 에 나타난 획의 위치 및 높이 정보를 이용하여 그림 3 (a)와 같이 검출되었다. 분리된 획은 결합 알고리즘(merging algorithm) [6] 을 이용하여 그림 3 (b)와 같이 이웃하는 연결 요소와 결합시킨다.

숫자로부터 분리된 획을 결합하기 위하여 기존 연구에서는 이웃하는 연결 요소와의 거리차를 계산하여 가장 근접한 연결 요소와 결합하는 방법을 사용하였다. 그러나, 필기자에 따라 획의 일부분이 분리된 숫자보다 이웃하는 숫자와의 거리차가 더 가까운 경우가 발생할 수도 있다. 따라서, 본 논문에서는 획의 일부분이 손상된 숫자에 대해서 결합 알고리즘을 사용하여 보상 결합할 수 있는 방법을 사용하였다. 결합 알고리즘은 획의 일부분이 손상된 숫자가 나타나지 않을 때까지 실행하도록 하였다.

### 3. 붙은 숫자들의 분할 및 인식

붙은 숫자들은 숫자의 구조적인 특징 정보를 이용하여 후보 분할 점(candidate segmentation points) [4] 들을 찾아 분할하도록 하였다. 개별 숫자로의 분할은 특징 점들 가운데 최왼쪽(leftmost) 숫자를 분할해 나가는 방법을 사용하였다. 이때 숫자의 구조적인 특징 정보 및 인식 결과로 구성된 분할 신뢰도(reliability value)를 이용하여 분할하도록 하였다.

그림 4 는 숫자 열에서 붙은 숫자 그룹으로 분류된 연결 요소에 대하여 후보 분할 점들을 찾아 분할해 나가는 과정을 나타낸 것이다.

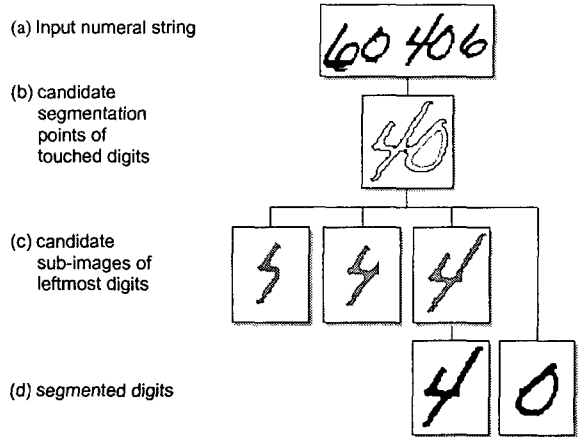


그림 4. 붙은 숫자들의 분할 과정  
Fig. 4. Segmentation of touching digits.

그림 4(b)는 붙은 숫자 그룹으로 분류된 연결 요소에 대해 네 가지 종류의 후보 분할 점들(candidate segmentation points)을 찾는 것이다. 개별 숫자로의 분할은 후보 분할 점들에 의해 최왼쪽 숫자로 가상 분할된 이미지 가운데 분할 신뢰도가 가장 높은 부 이미지 숫자를 선택하도록 하였다. 분할 신뢰도  $R(i)$  는 아래의 식으로부터 구할 수 있다.

$$R(i) = \Gamma[R_1(i), R_2(i), R_3(i)]$$

$$\text{where } \Gamma(a, b, c) = (a^2 + b^2 + c^2) \quad (4)$$

$$R_1(i) = 1 - \left( \frac{a_i - T_a(j)}{\max a_i} \right) \quad (5)$$

$$R_2(i) = 1 - \left( \frac{t_i - T_t(j)}{\max t_i} \right) \quad (6)$$

$$R_3(i) = 1 - (r_i - T_r(j))$$

$$i = 1, 2, \dots, n_s, \quad j = 0, 1, \dots, N_d \quad (7)$$

식에서  $i$  는 후보 분할 점에 의해 분리된 부 이미지(sub-image),  $j$  는 숫자 클래스를 나타낸다. 분할 단계에서는 식 (5)-(7)과 같이 부 이미지의 가로 세로 비  $a_i$ , 수평 전이 값  $t_i$  및 인식 결과 값  $r_i$  을 이용하였다.  $T(\cdot)$  는 숫자의 구조적인 특징 정보를 훈련 샘플을 이용하여 구한 임계치이다.

### 4. 실험 및 고찰

제안된 방법의 타당성을 살펴보기 위하여 NIST SD19 데이터베이스의 숫자 열을 가지고 시뮬레이션 해왔다. 테스트용 샘플로는 획의 일부분이 손상된 숫

자를 포함하거나 붙은 숫자를 포함하는 숫자 열 샘플 239 개를 사용하였다. 사용된 특징들은 메쉬(mesh), 수평 전이값, 경계로부터의 거리값, hole 의 개수, 8 방향 코드 성분(chain code direction)을 사용하였다.

그림 5 는 필기체 숫자 열에서 인식된 숫자들과 오인식된 숫자들을 나타낸 것이다.

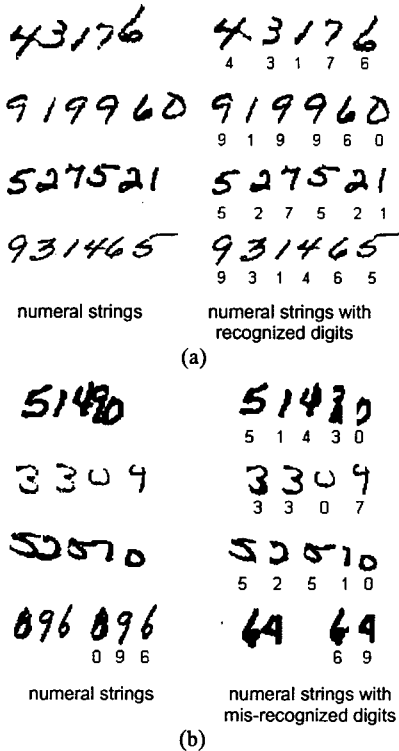


그림 5. (a) 인식된 숫자들과 (b) 오인식된 숫자들을 포함하고 있는 숫자 열의 예

Fig. 5. Example of numeral strings with (a) recognized digits and (b) mis-recognized digits.

그림 5 에서 해상도가 낮은 숫자나 다중 분할 점 (multi-touched segmentation points)을 가지는 붙은 숫자들을 분할하여 정확하게 인식했음을 볼 수 있다. 반면, 그림 5(b)에 나타난 일부 숫자 열에 대해서는 오분할로 인한 오인식 및 분할은 정확히 이루어졌으나 인식 알고리즘의 문제로 인하여 오인식된 숫자도 확인할 수 있다. 표 1 은 필기체 숫자 열에 대한 인식 결과를 나타낸 것이다.

표 1. 숫자 열에 대한 인식 결과

Table 1. Recognition result of numeral strings

	2NS	3NS	4NS	5NS	6NS
No. of digits	86	132	124	205	234
No. of recognized digits	79	113	105	189	207
Rec. rates(%)	96.7	96.5	96.1	96.5	96.4

(NS : numeral string)

표에서 나타난 결과로부터 제안된 알고리즘은 획의 일부분이 손상된 숫자뿐만 아니라 붙은 숫자도 효과적으로 인식할 수 있다는 것을 보여주고 있다. 또한, 인식 기반 분할 기법을 이용한 숫자 열 인식에 있어서 숫자 열 안에 포함된 숫자의 개수가 증가함에 따라 오인식률도 같이 증가되는 문제점을 개선할 수 있음을 알 수 있었다.

### 5. 결론

본 논문에서는 손상된 획의 일부분이 포함된 숫자나 붙은 숫자들과 같은 비정형 숫자들이 포함된 필기체 숫자 열을 인식하는 방법에 대해서 제안하였다. 숫자 열 안에 포함된 숫자들은 세 가지 다른 종류의 숫자 그룹 가운데 한 그룹으로 분류되며, 이때 숫자의 구조적인 특징 정보가 이용되었다. 또한, 숫자의 구조적인 특징 정보는 붙은 숫자들을 개별 숫자로 분할하는데 이용되는 분할 신뢰도를 결정하는 구성요소로 쓰인다. 제안된 방법은 기존의 인식 기반 분할 방법에서 숫자 열 안의 숫자의 개수가 증가하면 오인식률도 같이 커지는 단점을 보완해 주었다. 향후, 분할 기반 인식 방법과 무분할 기반 인식 방법을 이용하여 상호 보완 능력을 가질 수 있는 숫자 열 인식 방법을 시도할 것이다.

### 참고문헌

[1] Z. Lu, Z. Chi, W. Siu, and P. Shi, "A background-thinning based approach for separating and recognizing connected handwritten digit strings", *Pattern Recognition* 32, pp. 921-933, 1999.

[2] Z. Shi, V. Govindaraju, "Segmentation and recognition of connected handwritten numeral strings", *Pattern Recognition* 30, pp. 1501-1504, 1997.

[3] T. M. Ha, M. Zimmermann, and H. Bunke, "Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods", *Pattern Recognition* 31, pp. 257-272, 1998.

[4] K.K. Kim, J.H. Kim and C.Y. Suen, "Recognition of Unconstrained Handwritten numeral String by Composite Segmentation Method", *International Conference on Pattern Recognition*, vol. 2, pp. 594-597, 2000.

[5] D. Nishiwaki, K. Yamada, "A new numeral string recognition method using character touching type verification", *Proceedings of the Sixth International Workshop on Frontiers in Handwriting Recognition*, pp. 567-576. 1998.

[6] D. Cheng, and H. Yan, "Recognition of broken and noisy handwritten characters using statistical methods based on a broken-character-mending algorithm", *Optical Engineering*, vol. 36, no. 5, pp. 1465-1479, 1997.