

# 접촉 숫자쌍 인식기를 이용한 필기 숫자열 인식

최순만\* · 오일석

전북대학교 컴퓨터과학과

smchoi@cs.chonbuk.ac.kr, isoh@moak.chonbuk.ac.kr

## Recognition of Handwritten Numeral Strings Using Touching Numeral Pair Recognizer

Soon-Man Choi and Il-Seok Oh

Department of Computer Science, Chonbuk National University

### 요 약

임의 길이 숫자열을 인식하기 위해서는 우선 숫자열 영상을 인식기가 다룰 수 있는 형태로 변환해야 한다. 만일, 사용하는 인식기가 낱자 단위 인식기라면 낱자 단위로 분할하여야 하는데, 두자 이상의 숫자들이 접촉한 경우 정확한 분할이 어렵다. 이 논문은 이러한 문제를 해결하기 위하여 접촉 숫자쌍을 분할하지 않고 통째로 인식하는 방법을 사용한다. 필기 숫자열을 인식하기 위해 제안한 방법은 두 개의 인식기를 이용한다. 숫자열에서 분할된 패턴이 낱자일 경우 낱자 인식기가, 접촉 숫자쌍일 경우 접촉 숫자쌍 인식기가 인식한다. NIST 데이터베이스에 대한 실험 결과 2~10개의 숫자를 포함한 숫자열에 대하여 83.76%의 숫자열 인식률을 보여 접촉 숫자열 패턴을 낱자 단위로 분할하지 않고도 효과적으로 인식할 수 있음을 확인할 수 있었다.

### 1. 서 론

임의 길이 숫자열을 인식하기 위해서는 우선 숫자열을 인식기에서 다룰 수 있는 형태로 변환하는 과정이 요구된다. 사용하는 인식기가 낱자 단위 인식기라면 낱자 단위로 분할하여야 한다. 지금까지 진행되어온 대부분의 연구들이 사용하고 있는 방법[Cheo99, Fern91, Ha98, Mart93, Mata92, Nish94, Shri86, 이95]이며, 이들의 성능은 낱자 인식기의 성능도 중요하지만, 낱자 숫자들로 분할해 주는 분할 알고리즘이 매우 중요하다.

숫자열에는 낱자 숫자뿐만 아니라 두자 이상의 숫자가 접촉하는 경우가 발생하게 되는데 다중 접촉, 깨짐, 리가취, 잡음등으로 인하여 정확한 분할을 어렵게 하고 오분할을 야기 시켜 인식률을 저하시키고 있다.

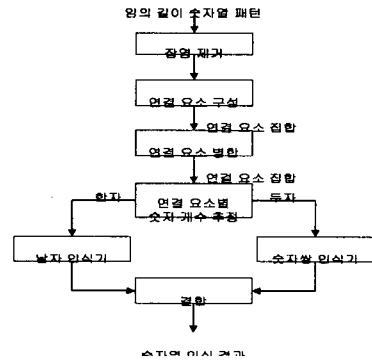
이러한 문제를 해결하기 위하여 본 논문은 접촉된 숫자열을 분할하지 않고 인식하는 방법을 사용한다. 이 방법은 숫자열 내에 포함되어 있는 접촉 숫자쌍(touching numeral pair)을 하나의 패턴으로 간주하고 낱자 인식기처럼 하나의 인식기를 만들어 인식한다. 즉, 두개의 인식기를 이용하여 숫자열을 인식하는 것이다. 숫자열에서 낱자 패턴일 경우 낱자 인식기가 인식하고, 접촉 숫자쌍일 경우 숫자쌍 인식기가 인식한다.

제안한 방법의 성능 평가를 위하여 NIST 데이터베이스의 숫자열 패턴 11,116개를 사용하였다. 실험 결과 2~10개의 숫자를 포함한 숫자열에 대하여 83.76%의 숫자열 인식률을 보여 접촉 숫자쌍을 낱자 단위로 분할하지 않고도 효과적으로 인식할 수 있음을 확인할 수 있었다.

### 2. 숫자열 인식

임의 길이를 갖는 숫자열 인식에서 숫자열 내의 낱자 숫자의 개수가 알려진 경우와 알려지지 않은 경우의 차이는 또 다른 문제의 복

잡성과 관련된다. 지금까지 대부분의 연구들은 알려진 경우에서 많이 진행되었으며, 특히 접촉 숫자쌍에 대한 연구가 활발히 진행되어 왔다. 본 논문은 숫자열 개수를 모르는 경우를 다루며, 전체 인식 과정은 그림 1과 같은 단계를 거치며 이루어진다.



[그림 1] 숫자열 인식 단계

#### 2.1 전처리

필기 숫자열 인식률 향상을 위한 전처리로는 잡음 제거, 조각난 영상 보정, 기울어짐 보정, 연결요소 재구성 등과 같은 방법[윤99]들이 있다. 본 논문에서는 이들 중 잡음 제거와 연결요소 재구성 방법을 이용하였다. 잡음 제거는 구원이 쉽고 숫자 패턴내의 잡음을 효과적

으로 제거할 수 있는 Ray[Ray88] 논문을 구현하여 사용하였다. 그림 2는 입력 숫자열 패턴과 잡음 제거 후의 결과이다.



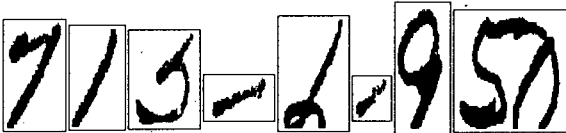
(a) 입력 숫자열 패턴



(b) 잡음 제거된 숫자열 패턴

[그림 2] 입력 숫자열 패턴과 잡음 제거

숫자열 패턴에는 그림 2의 '3'과 '2'처럼 하나의 숫자가 여러 연결 요소로 떨어져 있는 경우가 발생하는데 이들을 하나의 패턴에 포함되도록 결합할 필요가 있다. 연결요소 재구성엔 먼저 8-이웃의 연결요소(connected component)를 구하면 그림 3과 같이 레이블된 각 요소들을 얻을 수 있다. 여기서 각 요소들의 외각 사각형(bounding rectangle) 정보와 각 요소들간의 평균 간격(gap)을 구한 다음 각 획들의 구조적인 정보를 이용해서 결합이 이루어지도록 한다. 그림 4는 재구성된 결과이다.



[그림 3] 레이블된 각각의 연결 요소



[그림 4] 재구성된 각각의 연결 요소

### 2.2 낱자 수 추정

재구성된 각각의 연결요소(그림 4)에 대해서 한자인지 두자인지 판정한다. 판정은 각 연결 요소에 대한 숫자열 내의 위치, 크기, 획 두께, 화소 수 등의 정보를 이용해서 판정한다[윤99]. 그림 4에서 첫 번째에서 다섯 번째까지의 연결요소는 한자로 판정되고, 마지막 연결요소는 두자로 판정된다. 한자인 경우 낱자 인식이 두자인 경우 두자 인식이 인식을 담당하게 된다. 판정된 수를 모두 더하면 숫자열의 전체 숫자 개수가 된다. 그림 5는 숫자열에서 낱자 수를 옮겨 판정한 경우이고, 그림 6은 낱자 수를 틀리게 판정한 경우이다.

$$5407 \Rightarrow 2+1+1 = 4\text{개}$$

$$29344 \Rightarrow 1+1+1+2 = 5\text{개}$$

[그림 5] 숫자열에서 낱자 수를 옮겨 판정한 예

$$92174 \Rightarrow 1+1+1+1 = 4\text{개}$$

$$648985 \Rightarrow 1+1+1+1+1+2 = 7\text{개}$$

[그림 6] 숫자열에서 낱자 수를 틀리게 판정한 예

### 2.3 낱자 인식기

낱자 인식기는 낱자 수 추정에서 한자라고 판정한 것에 대한 인식을 담당한다. 이 인식기 구조는 모듈러 신경망[Oh96]이고 사용된 특징은 DDD[오98]이다. 훈련은 NIST 데이터베이스의 낱자 18,000패턴과 오류 역전파 알고리즘을 이용하였다. 이 인식기의 훈련 집합에 대한 인식률은 99.8%이고 검사 집합에 대해서는 97.0%이다.

### 2.4 접촉 숫자쌍 인식기

낱자 수 추정에서 두자라고 판정된 패턴은 숫자쌍 인식기[최00]를 이용하여 인식한다. 이 인식기는 숫자쌍을 하나의 패턴으로 간주하여 통째로 인식하는 것으로 최순만[최00]에서 분할-기반 방법보다 우수함을 보였으며, 구조는 모듈러 신경망이고 사용된 특징은 DDD이다. 인식기의 훈련을 위해 실제 접촉 숫자쌍을 NIST 데이터베이스에서 수집하여 데이터베이스를 구축하였으나, 부류별 편차가 심하고 개수가 불충분하여 훈련 집합으로 사용할 수 없다. 그래서 낱자 숫자를 이용하여 합성한 접촉 숫자쌍 50,000패턴을 인식기의 훈련 집합으로 사용하였다. 접촉 숫자쌍 인식기의 훈련 집합에 대한 인식률(필드 인식률)은 95.3%이고 검사 집합에 대해서는 78.0%이다.

### 3. 실험

실험은 Sun Microsystems사의 Enterprise I(UltraSPARC 168MHz, 128MByte)에서 C 언어를 이용하여 구현하였다. 사용한 데이터는 NIST 데이터베이스의 숫자열 데이터 11,116개를 사용하였다. 숫자열 데이터에 포함된 낱자 숫자의 개수는 2개에서 10개까지로 구성되어 있다. 실험 결과는 표 1과 같으며 그림 7과 8은 옮겨 인식한 경우와 틀리게 인식한 경우의 예이다.

[표 1] 숫자열 인식 성능

총 숫자열 데이터 개수	11,116
숫자 개수를 옮겨 추정된 숫자열 개수	10,032 (10,032/11,116 = 90.25%)
정인식 숫자열 개수	9,311 (9,311/11,116 = 83.76%)

$$0123456789 \Rightarrow 0123456789$$

$$1258 \Rightarrow 1258$$

[그림 7] 숫자열을 옮겨 인식한 경우

$$4558 \Rightarrow 4858$$

$$832657 \Rightarrow 832657$$

[그림 8] 숫자열을 틀리게 인식한 경우

숫자열 인식 시스템의 속도도 측정하였다. 처리 속도는 숫자열 인식의 단계별로 측정하였으며 그 결과는 표 2와 같다. 인식기 속도는 낱자는 1초당 약 191.80패턴, 접속 숫자쌍은 1초당 약 90.36패턴을 인식하였다. 길이가 10인 숫자열인 경우 총 0.492초가 소요되어 초당 약 2개의 숫자열을 처리할 수 있다. 이중 연결 요소 병합이 0.39초로서 전체 처리시간의 약 80%의 시간을 소요하는데 이 부분을 개선하여 속도 향상을 꾀할 수 있다.

[표 2] 평균 처리 속도 측정(단위 초)

단계 숫자 개수	잡영 제거	연결요소 병합	숫자 개수 추정	인 식	합 계
2자 (낱자 두개)	0.01	0.10	0.0	0.0100	0.120
2자 (접속쌍)	0.01	0.10	0.0	0.0110	0.121
5자	0.02	0.23	0.0	0.0262	0.276
7자	0.03	0.25	0.0	0.0367	0.317
10자	0.05	0.39	0.0	0.0524	0.492

[표 3] 기존 연구의 임의 길이 숫자열 인식에 대한 실험 결과

방 법	분할 성공률	숫자열 인식률	데이터 및 개수	비 고
[Cheo99]	-	85%	KAIST 검사: 600	2~7자의 숫자 개수
[Fenr91]	97.6%		CEDAR 검사: 176	1~6자의 숫자 개수
[Ha98]	-	92.7%	NIST 검사: 4,925	2~6자의 숫자 개수 낱자 인식기: 99.45%(훈련)
		83.6%	CEDAR 검사: 1,764	5자 또는 9자 낱자 인식기: 98.69%(훈련)
[Nish94]	-	88%	자체 수집 검사: ?	2~5자의 숫자 개수 낱자 인식기: 99.11%(훈련)
[Mart93]	-	75.6%	NIST 훈련: 20,000 검사: 5,000	2~6자의 숫자 개수
[Mata92]	-	73.85%	CEDAR 훈련: 10,000 검사: 2,585	5자 또는 9자
[Shri86]	-	93%	자체수집 검사: 500	1~3자의 숫자 개수 낱자 인식기: 99%(훈련)
[이95]	-	82.44%	NIST 훈련: 20,000 검사: 5,000	2~6자의 숫자 개수
본 논문	-	83.76%	NIST 검사: 11,116	2~10자의 숫자 개수 낱자 인식기: 99.8%(훈련) 접속 숫자쌍 인식기: 95.3%(훈련)

임의 길이 숫자열의 인식에 대한 기존 연구들의 인식률과 사용된 데이터베이스를 조사하여 표 3에 제시하였다. 많은 연구들이 표준 데이터베이스인 NIST와 CEDAR를 사용하였으며, NIST 데이터베이스를 사용한 연구들의 인식률이 높은 것으로 조사되었다. CEDAR 데이터베이스에서 낮은 인식률을 보이는 이유는 숫자열 데이터 내부에 밑줄, 인쇄된 박스 그리고 숫자 외의 다른 문자(예 '-') 등이 포함되어 있기 때문으로 추정된다. 조사된 문헌들에서는 이와 같은 문제들에 대한 대처 방안은 제시하고 있지 않다. 대체적으로 숫자열 인식률은 85% 정도를 보이고 있고, [Ha98]이 NIST와 CEDAR에서 각각 92.7%와 83.6%로 좋은 인식률을 보이고 있다. 이 방법은 2~6개의 숫자 개수를 포함하는 숫자열만을 대상으로 한 인식률이다. 본 논문의 인식률은 표 3에서 NIST 데이터베이스에 대해 83.76%를 보이고

있다. 본 논문은 숫자열내의 숫자 개수가 2~10개로 [Ha98]보다 많은 수의 숫자 개수를 포함하는 숫자열을 대상으로 한 것이다. 한편 숫자열에서 낱자 수를 옮겨 추정한 숫자열에서 숫자열을 정인식한 인식률은 92.81%(9,311/10,032 = 92.81)를 보이고 있어 숫자열 내의 낱자 수를 추정하는 개선된 알고리즘을 사용한다면 보다 좋은 성능을 얻을 수 있을 것이다.

4. 결론

본 논문에서는 접속 숫자쌍을 낱자 단위로 분할하지 않고 인식하는 인식기를 임의 길이 숫자열 인식에 활용하였다. 제안한 방법은 숫자열 내의 낱자 패턴은 낱자 인식기 그리고 접속 숫자쌍 패턴은 접속 숫자쌍 인식기로 인식하였다. NIST 데이터베이스를 이용한 성능 평가 실험에서 접속 숫자쌍을 낱자 단위로 분할하지 않고도 효과적으로 인식할 수 있음을 보였다. 향후 연구 과제로는 낱자 수 추정 알고리즘과 접속 숫자쌍 인식기의 성능 향상과 세자이상 접속한 경우의 처리 등이 있다.

참고문헌

[Cheo99] C.E. Cheong, H-Y. Kim, J-W. Suh, and J.H. Kim, "Handwritten numeral string recognition with stroke grouping," *Proceedings of ICDAR'99*, pp. 745-748, 1999.  
 [Fenr91] R. Fenrich, "Segmentation of automatically located handwritten words," *Proceedings of IWFHR II*, pp. 33-44, 1991.  
 [Ha98] T.M. Ha, M. Zimmermann, and H. Bunke, "Off-line handwritten numerals string recognition by combining segmentation-based and segmentation-free method," *Pattern Recognition*, Vol. 31, No. 3, pp. 257-272, 1998.  
 [Mart93] G.L. Martin, M. Rashid, and J.A. Pitman, "Integrated segmentation and recognition through exhaustive scans or learned saccadic jumps," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 7, No. 4, pp. 831-847, 1993.  
 [Mata92] O. Matan et al., "Reading handwritten digits: a ZIP code recognition system," *IEEE Computers*, Vol. 25, No. 7, pp. 59-63, 1992.  
 [Nish94] H. Nishida and S. Mori, "A model-based split-and-merge method for character string recognition," in *Document Image Analysis* (Edited by Bunke et al.), World Scientific, pp. 209-226, 1994.  
 [Oh96] I-S. Oh, J-S. Lee, S-M. Choi, and K-C. Hong, "Class-expert approach to unconstrained handwritten numeral recognition," *Proceedings of IWFHR V*, pp. 95-102, 1996.  
 [Ray88] S. Ray, "A heuristic noise reduction algorithm applied to handwritten numeric character," *Pattern Recognition Letters*, Vol. 7, No. 1, pp. 9-12, 1988.  
 [Wang98] X. Wang, V. Govindaraju, and S. Srihari, "Holistic recognition of touching digits," *Proceedings of IWFHR VI*, pp. 295-303, 1998.  
 [오98] 오일석, C.Y. Suen, "광학 문자 인식을 위한 거리 특징," *한국정보과학회논문지(B)*, Vol. 25, No. 7, pp. 1028-1043, 1998.  
 [윤99] 윤성수, 변영철, 김경환, 최영우, 이일병, "필기 숫자열 인식률 향상을 위한 초기 처리에 대한 연구," *한국정보과학회 1999 추계학술발표회지*, pp. 455-457, 1999.  
 [이95] 이성환, 이용재, "순환 신경망을 이용한 연속 필기 숫자열의 통합적 분할 및 인식," *한국정보과학회 1995 추계학술발표회지*, pp. 447-450, 1995.  
 [최00] 최순만, 오일석, "두자 접속 숫자열의 분할 자유 인식," *한국정보과학회논문지(B)*, Vol 27, No. 5, pp. 563-574, 2000.