

필기 숫자열 인식을 향상을 위한 초기 처리에 관한 연구 *

윤성수[○], 변영철, 김경환[†], 최영우[‡], 이일병

연세대학교 컴퓨터과학과, [†]서강대학교 전자공학과, [‡]숙명여자대학교 전산학과

Early Processings for an Improvement in Handwritten Digit String Recognition

Sungsoo Yoon, Yungcheol Byun, Gyeonghwan Kim[†], Yeong-Woo Choi[‡], Yillbyung Lee

Dept. of Computer Science, Yonsei Univ.

[†]Dept. of Electronic Engineering, Sogang Univ.

[‡]Dept. of Computer Science, Sookmyung Women's Univ.

요 약

필기 숫자열의 인식성능을 향상시키기 위해서는 물론 인식기 자체의 성능 개선도 필요하지만 인식기에서 필요로 하는 정보를 제공해주는 초기단계의 개선 역시 매우 중요하다. 낱자와는 달리 숫자열 인식에서는 인식기에서 필요한 단위로 입력 데이터를 분할해야만 하는데 잡영, 기울어짐, 접촉 등의 원인에 의해서 쉽게 분할해내기 어렵기 때문이다. 본 논문에서는 이런 문제점들을 극복하기 위한 방법들은 제시하였으며 NIST 숫자열 데이터에 적용해 본 결과 16%의 성능 향상을 보였다.

1. 서론

실생활에 사용되는 필기 숫자를 인식하기 위해서는 우선 숫자열을 인식기에서 다룰 수 있는 형태로의 변환과정이 요구된다. 사용되는 인식기가 직접 숫자열을 처리하는 경우가 아니라면 이 과정은 반드시 필요할 뿐 아니라 인식결과에도 많은 영향을 끼치게 된다. [1]에서와 같이 숫자열을 직접처리하는 경우를 제외하면 숫자열을 미리 정해진 단위로 분할해야 한다. 대부분의 경우 이 단위는 낱자 숫자가 되지만 다중 분할을 기본처리 방법으로 삼는 경우는 그렇지 않을 수도 있다[2]. 그러나 다중분할 방법의 경우에도 역시 숫자들 사이를 분리하는 곳을 포함하고 있어야만 한다. 인쇄숫자의 경우에는 등간격에 크기도 거의 동일하기 때문에 분할 과정에 고려해야할 요소가 많지 않지만 필기 숫자의 경우에는 많은 변수가 작용한다. 우선 문자의 크기와 간격이 일정하지 않고 또한 문자가 기울어져 있는 경우가 많으며 빼친 획에 의해서 인접한 문자가 접촉하는 경우가 흔히 발생한다. 이런 상황에서 정확히 숫자들 사이를 분리해 내는 일은 쉬운 일이 아니다. 특히 입력 영상의 질이 좋지 않은 경우에는 더 더욱 고려해야할 요소가 많아진다. 예를 들어 숫자열 내에서 몇개의 숫자가 2~3개의 조각으로 분리되어 있고 조각난 획들이 또 이웃 숫자와 접촉되어 있다면 숫자들 간의 분리점을 찾기가 매

우 어려워진다. 또한 두 개 이상의 숫자들간에 접촉이 일어날 수 있기 때문에 몇개의 숫자가 접촉되어 있는지에 대한 정보가 숫자들간의 정확한 분할에 필요하게 된다.

본 논문에서는 필기 숫자열 인식을 향상해 필요한 초기 처리 방법들을 살펴보고 이런 방법들이 실제로 얼마나 도움이 되는지 실험을 통해서 살펴 보겠다.

2. 필기 숫자열 인식을 향상해 도움이 되는 초기 처리과정들

만일 숫자열이 많은 작은 조각으로 나뉘어 있다면 각각의 낱자에 대한 정보도 얻기가 힘들고 잡영으로 오인되어서 중요한 정보를 잃어 버릴 수도 있다. 그러므로 가능한 원래 영상을 보존하면서 조각난 숫자들을 묶어줄 필요가 있다. 필기 숫자열의 경우 필기자의 습관에 따라 기울어지는 경향을 보이는데 이런 경우 화소수에 비해 낱자의 폭이 증가되어 다양한 필기자의 숫자열을 처리할 때 일관성 있는 처리가 곤란해진다. 또한 분할과정에서도 기울어진 숫자열을 처리하는 것 보다 바로선 숫자열을 다루는 것이 상대적으로 용이하다. 숫자의 경우 필기자의 습관에 따라서 차이는 있지만 대부분 하나의 획으로 이루어져 있으나 4, 5, 7은 두 개의 획으로 이루어지는 경우가 많다. 이 경우는 영상의 질과는 상관없이 흔히 나타나는 문제이기 때문에 두 획

* 본 연구는 정보통신연구진흥원의 지원으로 수행되었음

이 하나의 숫자에 속한다는 정보를 주는 것은 인식물에 더 더욱 중요한 요소로 작용한다. 필기 숫자열의 경우 이웃한 숫자간에 접촉이 일어나는 경우가 자주 있으므로 연결된 화소집합이 하나의 숫자인지, 만일 그렇지 않다면 몇개의 숫자가 포함되어 있는지에 대한 정보는 분할과정에서 중요한 처리 근거로 사용된다. 접촉된 숫자열에 포함된 낱자의 수를 정확히 추측한다면 인식물 뿐만 아니라 전체 시스템의 속도면에서도 많은 도움을 줄 수 있다. 다음의 각 절에서 각각의 방법을 설명한다.

2.1. 조각난 영상 보정

떨어진 조각들을 잇기 위해서는 어느 정도 떨어진 조각까지를 이어야할지를 결정해야 한다. 먼 거리까지 포함시킨다면 이웃한 문자와 근접한 획까지 이을 위험이 있고 너무 가까운 거리만 고려한다면 떨어진 조각들을 충분히 이을 수 없다. 이어야할 조각들은 획들 사이에 존재하므로 획의 방향성을 찾아서 경우에 따라서 다른 방향의 윈도우를 사용한다. 즉, 검출된 방향에 따라서 0°, 45°, 90°, 135°, 4개의 7×5 윈도우를 적용한다. 방향의 검출은 이으려는 화소의 15×15 윈도우와 각각의 방향으로 한 개와 두 개 화소만큼 움직였을때의 윈도우와 NAND값을 계산하여 합한 값이 가장 작은 값을 보이는 방향을 선택한다. 결정된 7×5 윈도우 안에서 조각들을 잇는 방법은 다음과 같다[3].

1. 결정된 방향으로 검은 화소가 존재하는 최상단과 최하단을 결정한다.
2. 최상단과 최하단의 사이에 존재하는 각 열(행)에서 좌/우측(위/아래) 끝에 존재하는 검은 화소를 검출한다.
3. 만일 최상단과 최하단이 윈도우의 중앙이면 그 열(행)의 검은 화소를 모두 지우고 대응되는 위치의 결과영상에 검은 화소를 설정한다. 그렇지 않은 경우 검은 화소 사이의 흰 화소를 모두 검은 화소로 설정하고 결과영상에도 검정화소로 설정한다
4. 최상단과 최하단 사이의 좌/우측(위/아래)언어진 검은 화소의 위치를 다음 방법으로 새로 정한다.
 - $newleftmost(i) = \min\{ leftmost(i), (leftmost(i-1) + leftmost(i+1)) / 2 \}$
 - $newrightmost(i) = \max\{ rightmost(i), (rightmost(i-1) + rightmost(i+1) + 1) / 2 \}$
5. 새로 언어진 끝점 사이의 화소들은 입력영상에서는 모두 흰 화소로 바꾸고 결과영상은 검정화소로 바꾼다.
6. 검은 화소가 있는 경우에 최상단과 최하단의 좌측 끝점을 새로운 시작점으로 위의 과정을 반복한다.

위와 같은 방법에서는 조각들을 모으는 과정에서 이웃한 숫자를 붙일 가능성도 있기 때문에 이으려는 화소 집단들간의 크기 정보를 이용하여 임계치 이상의 큰 집단끼리의 연결은 막도록 한다(그림 1).

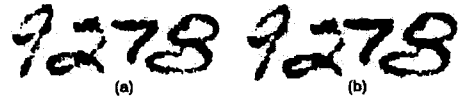


그림 1. 조각난 영상 보정 전(a)과 후(b)

2.2. 기울어짐 보정

숫자열에서 숫자들의 기울어짐을 찾기 위해서 각 숫자의 윤곽들에 대해서 gradient 연산자를 적용해서 각각의 화소에서의 기울기 정보를 얻고 이들 중에서 50° ~ 130°, 230° ~ 310° 사이에 화소들만을 선택한다. 화소들의 연속된 정도를 가중치로 사용하면 각 화소의 기울기값을 평균하여 문자열 전체의 기울기를 얻는다. 얻어진 기울기 θ 를 사용해서 입력영상을 보정한다[4](그림 2).

$$x_n = x - y \cdot \tan(\theta)$$

$$y_n = y$$

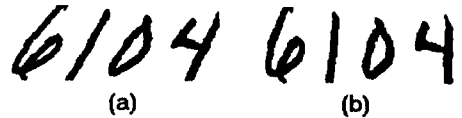


그림 2. 기울기 보정 전(a)과 후(b)

2.3. 연결요소 재구성

물리적으로 떨어져 있는 획중에서 하나의 숫자에 포함되는 획들은 논리적으로 결합할 필요가 있다. 인식단계 전에 구조적인 정보를 이용해서 논리적으로 합친다면 인식률과 시스템 효율에 기여할 수 있다. 다음 조건을 만족하는 획들을 같은 낱자의 것으로 추정한다.

1. 획들간에 수평으로 겹친 경우중에서 다음 식을 만족하지 않는 경우[5]

(BVT and BVCH) or (BT and BHFA)

- BVT: 두 획이 매우 높은 경우($c.h \geq 0.6 \times ech$)
- BVCH: 두 획이 매우 유사한 높이의 경우, $(\min(c1.h, c2.h) / \max(c1.h, c2.h)) > 0.9$
- BT: 두 획이 큰 경우($c.h \geq 0.3 \times ech$)
- BHFA: 두 획이 수평으로 많이 떨어져 있는 경우, $abs(c1.x - c2.x) \geq (0.3 \times ech)$
- c.h: 획의 높이
- c.x: 획의 중심 수평 위치
- ech: 숫자열에서 가장 큰 획의 높이

2. 두 획간의 거리가 획 폭보다 작고 둘 중 하나가 숫자열의 무게 중심보다 높고 두 획간의 크기 차이가 2배 이상인 경우

2.4. 숫자열에 포함된 낱자 수 추정

입력 숫자열에 대해서 연결요소(connected component)를 구하고 각 연결요소에 대한 위치, 크기 정보를 얻어 이를 바탕으로 낱자, 합쳐야할 획들 그리고 2개 이상의 숫자가 접촉된 것들로 구분한다. 이웃한 숫자와 접촉된 연결요소인지를 결정하기 위해서 획 폭, 낱자 높이 그리고 화소 수를 이용해서 다음 식에 의해서 결정한다.

$$\begin{aligned}
 ssc &= (-2.11)ar + 5.8 & (1) \\
 ar &= \frac{w}{ech} \\
 ssa &= esw \times ech \\
 ssc &= \frac{p}{ssa}
 \end{aligned}$$

- w : 연결요소의 폭
- ech : 숫자열에서 가장 큰 연결요소의 높이
- esw : 추정된 획 폭
- p : 연결요소의 화소 수

식 (1)은 ssc와 ar을 축으로하는 2차원 평면에서 직선을 나타내며 낱자의 경우 이 두 특징값에 의한 좌표값이 이 직선의 아래쪽에 놓이는 경향을 보인다[5]. 이런 특성을 이용해서 접촉된 숫자열을 검출하고 분할하고 다시 검출하는 과정을 반복적으로 적용해서 포함된 낱자 수를 추정한다.

3. 실험 및 결과

숫자열에 포함된 낱자 수 추정 결과가 숫자열 인식 시스템의 성능 향상에 도움을 준다는 가정하에서 위에서 설명한 방법들이 낱자수 추정 결과에 미치는 정도를 실험했다. 실험에는 NIST DB special 3에서 추출한 3 ~ 10의 낱자를 포함하는 10976개의 숫자열 데이터를 사용했다. <표 1>에서 실험 결과를 보이고 있다.

표 1. 제시한 방법들을 적용한 후의 낱자 추정 실험 결과

방법	옳게 추정한 문자 수				증가 비
낱자 수 추정	2332	890	1431	4227	-
연결요소 재구성	2667	1057	1663	4866	15.5%
기울기 보정	2670	1059	1675	4866	15.7%
조각난 영상 보정	2670	1063	1688	4879	16.0%

위 결과는 각 방법을 누적해서 적용한 결과이다. 기울기 보정이나 조각난 영상 보정이 연결요소 재구성 결과와 비교해서 좋지 못한 이유는 실험에 사용된 입력 데이터가 모두 기울기 보정이나 조각난 영상 보정이 필요한 것은 아니었기 때문이다. 그러나 다양한 데이터에 대해서 신뢰성을 보장할 수 있었다. 숫자열에서 낱자 수 추정 실험은 분할 과정을 포함하지 않아 세 자 이상 접촉된 경우는 고려하지 않았다.

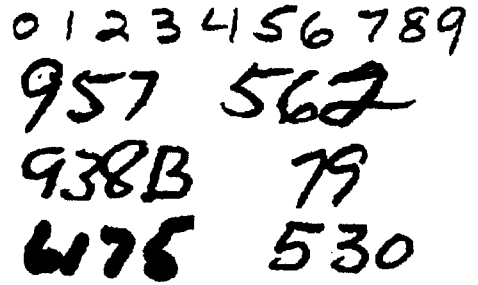


그림 3. 실험에 사용한 데이터

4. 결론

대부분의 인식 시스템에서와 같이 숫자열 인식 시스템의 초기 과정은 인식률에 직접적인 영향을 주게 된다. 초기 과정들의 오류를 극복하기 위해서 반복적으로 순환해서 처리할 수도 있지만 초기 과정의 결과에서 크게 벗어나지 못한다. 그러므로 초기 과정에서 효율성을 해치지 않는 범위 내에서 가능한 신뢰성 있는 처리를 하는 것이 인식률을 높이는 직접적 방법이다. 본 논문에서는 필기자의 습관에 의해 기울어진 숫자열에 대한 처리, 숫자열이 작은 조각으로 나뉘어진 좋지 않은 영상에 대한 처리, 하나의 숫자에 포함된 획이 둘 이상으로 나뉘어진 경우에 대한 처리 그리고 이웃한 숫자가 접촉되어 있는 경우 이것을 검출하여 분할할 수 있는 정보를 제공하는 처리 방법을 제시하고 이 방법들의 유용성을 실험을 통해서 확인하였다.

참고문헌

- [1] Xian Wang, Venu Govindaraju, Sargur Srihari, *Holistic Recognition of Touching Digits*, 6th IWFHR'98, Korea, 295-303, 1998
- [2] N Arica, F T Yarman-Vural, *A New Scheme for Off-Line Handwritten Connected Digit Recognition*, 14th ICPR'98, Australia, 1127-1129. 1998
- [3] Zhixin Shi, Venu Govindaraju *Character image enhancement by selective region-growing*, Pattern Recognition Letters, 523-527, 1996.
- [4] Gyeonghwan Kim, Venu Govindaraju, *Efficient chain code based image manipulation for handwritten word recognition*, In Proc. of the SPIE symposium on electronic imaging science and technology, vol. 2660, 262-272, February. 1996.
- [5] M. D. Garris, *Component-Based Handprint Segmentation Using Adaptive Writing Style Model*, NIST Internal Report 5843, June 1996.