

지로 서식 문서의 인쇄체 숫자 인식*

김진숙[○] 변영철 김경환[†] 최영우[‡] 이일병
연세대학교 컴퓨터과학과, [†]서강대학교 전자공학과, [‡]숙명여자대학교 전산학과

Printed Numeric Character Recognition on Giro Form

Jinsook Kim[○] Yungcheol Byun Gyungwhan Kim[†] Yeong-Woo Choi[‡] Yillbyung Lee
Dept. of Computer Science, Yonsei University
[†]Dept. of Electronic Engineering, Sogang University
[‡]Dept. of Computer Science, Sookmyung Women's University

요 약

본 논문에서는 일상 생활에서 쉽게 접할 수 있는 지로(Giro) 서식 상에 있는 인쇄체 숫자열 인식 방법으로서 템플릿 매칭 방법에 대해 설명한다. 지로 서식 문서 상의 인쇄체 숫자는 인쇄시의 오류로 인하여 숫자의 굵기나 높이, 그리고 폭이 다를 수는 있지만 기본적으로 폰트의 유형이 한가지라는 것과 나타날 수 있는 오류의 유형이 몇 가지로 제한되어 있다는 특징을 갖는다. 따라서 이러한 데이터 특징을 효율적으로 수용할 수 있도록 템플릿을 정의한 후 매칭 방법을 통해 숫자를 인식하는 템플릿 매칭 방법에 대해 설명한다. 실험 결과 비교적 간단한 방법을 이용하더라도 인쇄체 숫자열을 효율적으로 인식할 수 있었다.

1 서 론

일상 생활에서 대금 결제 수단으로 많이 사용되는 서식 문서 중의 하나가 지로 서식이다. 지로(Giro)란 은행 대체(대체 저금)에 의한 결제로서 임의의 사용자가 지로를 개설한 후 은행을 통하여 대금을 결제하는 제도를 말한다. 현재 우리나라에서는 이 제도를 위해 금융 결제원이 설립되어 있다.

지로 용지 인식은 금융 결제원의 한 업무로서 현재까지 많은 변화와 발전을 거듭해 왔다. 초기의 지로 용지의 처리 방법은 마그네틱 잉크 문자 판독기(Magnetic Ink Character Reader)와 광학 문자 판독기(Optical Character Reader)를 사용였다. 이 영향으로 말미암아 지금 시중에서 사용되는 지로 용지를 보면 크게 MICR과 OCR로 나누어진다. <그림 1>은 OCR 지로 용지의 예이다. MICR의 경우 마그네틱 잉크의 주사 방법에 따라 특별히 고안된 폰트를 사용하는데, 이 방법으로는 경계 활동 영역의 세분화와 팽창으로 인한 지로 용지 사용의 급증을 수용할 수 없기때문에 금융 결제원에서는 영상 처리 시스템을 도입하게 되었다.

금융 결제원에서 사용하고 있는 영상 처리 시스템이 지로 용지를 처리하는 방법은 대략적으로 설명하면 다음과 같다. (1) 우선, 처리될 서식 용지는 처리에 앞서 MICR과 OCR로 분류되어지며, (2) 각각의 지로 서식은 용지의 크기에 맞게 설계된 스캐너에 의해 스캔된다. 그 다음, (3)인식할 숫자 열로서 지로 번호, 고객 번호, 금액 등을 추출한다. 이 경우 숫자 열의 위치는 고정되어 있기 때문에 잡음이 있다라도 비교적 안정적인 방법으로 숫자열을 추출할 수 있다. (4)마지막으로, 추출한 숫자열은 숫자열 분할 과정을 거쳐 낱자로 분할되고 숫자 낱자에 대한 인식이 수행된다. 낱자 인식에서는 라인 접촉에 의한 잡음 및 인쇄 잡음, 그리고 기타 이유에 의해 발생한 잡음으로 말미암아 오

류가 발생할 수 있다. 지로 용지 상의 지로 번호, 고객 번호, 금액 등은 매우 중요한 데이터로서, 한자리 인식 오류만으로도 심각한 문제를 야기할 수 있기 때문에 처리 시간 및 인식을 면에서 이를 보다 효율적으로 처리할 수 있는 방법이 필요하다. 따라서 본 연구에서는 인식 대상으로 데이터의 특징을 살펴본 후 그러한 데이터를 효율적으로 인식할 수 있는 방법에 대해 설명하고자 한다.

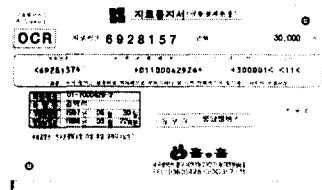


그림 1: OCR 지로 용지의 예

이를 위해, 다음 장에서는 데이터 특징 및 제안하는 방법에 대해 간단히 설명한다. 3장에서는 인식 방법으로서 템플릿 기반 방법을 설명한 후 4장에서 실험 결과에 대해 설명하도록 한다. 그리고 마지막으로 본 연구에 대한 결론 및 개선점은 5장에서 논하고자 한다.

2 데이터 특징 및 제안하는 방법

이 절에서는 본 연구에서 대상으로 하는 인쇄체 숫자열의 특징을 살펴본 후 제안하는 방법에 대해 설명하고자 한다.

* 본 연구는 정보통신연구진흥원 과제의 일부로 이루어졌음

2.1 대상 문자의 특징

지로 서식 문서에 있는 인쇄체 숫자의 가장 큰 특징은 폰트 유형이 한가지라는 것이다. 인쇄시의 오류로 인하여 숫자의 굵기나 높이, 그리고 폭이 다를 수는 있지만 기본적인 구조는 동일하다. 또한 발생할 수 있는 오류의 유형도 몇 가지로 제한된다. (그림 3)과 같이 처리할 숫자열의 위 부분과 아래 부분에 선분배 의한 잡음이 있을 수 있다. 따라서 (1)첫째, 인쇄시 잘못된 위치 선정에 의해, 혹은 낮은 해상도로 지로 서식을 스캔할 경우 라인과 숫자열이 접촉할 수가 있다. (2)둘째, 비교적 온전한 데이터에 비해 상대적으로 문자의 굵기가 가는 경우가 있을 수 있으며, 또한 높이와 폭이 작은 경우가 있다. (3)셋째, 숫자에 새로운 획이 잡음 형태로 첨가되거나 혹은 반대로 잘려져 나간 경우가 있을 수 있다. 비록 잡음이 있는 데이터라 할지라도 대부분 이러

0123456789

그림 2: 비교적 온전한 영상

0123456789

그림 3: 변형된 영상

한 범주로 분류할 수 있었으며, 또한 회전 및 크기에 의한 변형이 거의 존재하지 않는다. 즉, 나타날 수 있는 잡음의 형태가 어느 정도 한정된다. 따라서 다음과 같은 방법을 제안하고자 한다.

2.2 제안하는 방법

본 연구에서는 앞서 설명한 데이터 특성을 효율적으로 수용할 수 있도록 템플릿을 정의한 후 매칭 방법을 통해 숫자를 인식하는 템플릿 매칭 방법을 설명하고자 한다. 템플릿 매칭 방법은 예외적인 상황이 어느 정도 한정되어 있을 경우, 그리고 회전과 크기의 변화가 적을 경우 효과적이다[1][2]. 본 연구에서 대상으로 하는 지로 서식 상의 숫자열은 폰트가 하나가 유형이며, 크기에 의한 변형과 회전에 의한 변형이 거의 존재하지 않는다. 또한 앞서 설명했듯이 오류의 유형도 몇가지로 제한되기 때문에 입력 데이터를 수용할 수 있는 적절한 템플릿을 고안할 수 있다면 템플릿 매칭 방법을 이용함으로써 효율적으로 서식 문서를 처리할 수 있다.

3 템플릿 기반 인식 방법

3.1 템플릿 매칭

템플릿은 가급적이면 예외적인 경우를 많이 수용할 수 있도록, 즉, 입력 데이터의 변형을 수용할 수 있도록 정의되어야 한다[2]. 템플릿을 이용한 매칭 방법으로서, 가장 간단한 방법은 단순히 템플릿과 입력 데이터를 비교하여 서로 매칭되는 픽셀 수를 스코어로 정의한 후 이를 고려하는 것이다. 매칭되는 픽셀 수와 매칭되지 않는 픽셀 수를 더하여 정규화하는 방법도 있다. (그림 2)를 보면 숫자 1의 영상은 다른 숫자 영상 보다 검은 화소가 적다. 이러한 경우는 템플릿과 숫자 영상을 비교할 때 단지 검은 픽셀이 매칭되는 것과 매칭되지 않는 경우를 고려하는 것보다는 배경에 해당하는 흰 화소가 매칭되는 경우와 그렇지 않은 경우를 고려하여 정규화하는 방법도 있다.

마지막으로 영상에서 템플릿과 매칭할 영역을 결정하는 문제가 있다. (그림 3)를 보면 라인 접촉으로 인하여 0, 2, 3, 5에 해당되는 데이터는 템플릿과 그대로 할 경우 잘못 인식될 가능성이 크다. 따라서 이러한 경우에는 템플릿을 이동하는 위치 정규화를 수행하여 매칭을 수행하는 방법이 바람직하다. 이동 방향은 먼저 라인이 위 부분, 위와 아래 부분에 있을 경우에는 템플릿을 아래로 이동해야 하며, 영상에서 숫자의 위 부분이 잘려진 경우에는 템플릿을 위로 이동해야 한다. 그리고 영상에서 숫자의 왼쪽에 공백이 있을 경우에는 오른쪽으로 이동해야 한다.

3.2 템플릿 결정 방법

앞서 설명했듯이 템플릿은 입력 데이터의 오류, 즉 예외적인 경우를 수용할 수 있어야 한다. 이를 구현하기 위해서는 예외적인 경우를 하나의 템플릿으로 보고 템플릿 집합에 추가로 정의할 수도 있지만 하나의 템플릿에 예외적인 경우를 표현할 수도 있다. 본 논문에서는 가급적 템플릿 수를 줄이면서 효율적으로 입력 문자를 인식하기 위하여 입력 데이터에서 유사한 데이터들을 분류하는 K-Means 알고리즘을 사용한다[3].

숫자 영상들은 비슷하지만 약간씩 차이가 있을 수 있는데, K-Means 알고리즘을 사용함으로써 유사한 특징을 갖는 데이터들을 동일한 부류로 분류할 수 있다. 이 경우 클러스터의 수가 많을수록 한 클러스터에 속하는 데이터들 간의 유사도는 더욱 높아지며 클러스터 간의 유사도는 낮아진다. K-Means 알고리즘을 이용하여 템플릿을 정의하는 과정은 다음과 같다. (1)0에서 9까지의 각각의 학습 데이터에 대해 수평/수직 히스토그램을 구한 후 이를 K-Means의 입력으로 이용하여 클러스터링을 수행한다. 히스토그램을 이용하는 이유는 문자에 대해 FFT를 수행할 경우 히스토그램이 그 문자의 주요 특징을 이루고 있음을 알 수 있기 때문이다[4][5]. (2)K-Means 알고리즘을 이용하여 클러스터링을 수행한 결과 오류값이 클 경우에는 K 값을 증가시켜 클러스터 내의 데이터에 대해 다시 클러스터링을 수행한다. K-Means 알고리즘은 클러스터의 수를 제시할 수 있기에 원하는 만큼의 템플릿을 정의하는데 효과적이다. (3)클러스터링을 수행한 후 각각의 클러스터에 속한 영상을 이용하여 템플릿을 정의한다. 이때 템플릿은 클러스터에 속한 모든 학습 영상들을 겹친 후 서로 대응되는 위치에 있는 검은 픽셀의 수를 누적시킨 값을 학습 영상의 수로 나누어 정규화하여 정의한다. 따라서 템플릿은 이차원 배열 형태로 표현된다. 이처럼 모든 학습 데이터를 이용하여 템플릿을 구함으로써 템플릿에 어느 정도 고정적인 형태의 잡음에 의한 예외적인 특성도 내포하도록 한다. (4)이러한 연산은 0에서 9까지의 모든 학습 데이터에 대해 수행한다. 본 연구에서는 각각의 숫자에 대해 300개의 영상을 이용하여 템플릿을 구하였다.

3.3 템플릿을 이용한 인쇄체 숫자 인식

템플릿 매칭에 있어서 입력 영상과 템플릿의 0이 아닌 값, 즉 전경 픽셀만을 비교할 수도 있고, 배경 픽셀에 해당되는 픽셀을 비교할 수도 있다. 전경 픽셀만을 비교하는 방법은 배경 픽셀만을 비교할 경우와 다른 결과를 가져온다. 배경 픽셀만을 비교한 결과 8이 주로 6으로 오인되는데 반해 전경 픽셀만을 비교할 경우에는 8이 0으로 오인식되었다. 한편, 템플릿과 입력 영상이 모두 0이 아닌 경우만을 고려하기도 하고 값이 다른 경우까지 고려하는 방법도 있다[2]. 본 연구에서 제안하는 방법에 의해 만들어진 템플릿, 즉, K-Means 알고리즘에 의해 클러스터링된 데이터를 이용하여 평균값을 취한 템플릿은 숫자 부분에 대해서

는 0보다 큰 값을 갖고 나머지 부분, 즉 배경에 대해서는 0으로 표현된다. 템플릿을 이용하여 인쇄체 숫자를 인식하기 위하여 입력 영상의 0이 아닌 부분과 템플릿의 0이 아닌 부분을 곱한 값을 모두 더한 결과값을 유사도로 이용한다.

4 실험 결과

4.1 실험 환경 및 데이터

제한하는 방법의 성능을 평가하기 위하여 Pentium 300 상에서 C++ 언어를 이용하여 인식 시스템을 구현하였다. 그리고 0에서 9까지의 각 숫자에 대해 잡음이 있다고 생각되는 데이터들을 대상으로 실험하였다. <표 1>은 각 숫자 마다 두 개의 템플릿을 정의한 후 실험한 결과이다. 실험 결과 2와 8의 경우 다른 숫자

표 1: 각 숫자에 대해 두 개의 템플릿을 정의한 결과

정인식/숫자 개수	인식률(%)
0	197/202 97.5
1	132/137 96.4
2	71/109 65.1
3	215/225 95.6
4	218/219 99.0
5	112/118 94.9
6	48/51 94.1
7	152/158 96.2
8	52/82 64.4
9	149/156 95.5

에 비해 상대적으로 낮은 인식률을 보였다. 2의 경우 오인식한 데이터 38개 중 27개를 7로 오인식하였고, 8의 경우에는 오인식한 데이터 33개 중 32개를 0으로 오인식하였다.

2와 7, 그리고 8과 0에 대해 변별력을 높임으로써 혼동되는 문제를 해결하기 위하여 2와 8의 템플릿을 3개로 정의하여 실험해 보았다. 우선, 2의 템플릿만을 3개 정의한 후 실험한 결과 숫자 2에 대한 인식률이 92.7%로 향상되었다. 2와 8의 템플릿을 모두 3개를 정의하여 실험한 결과 다른 숫자의 인식률은 그대로 유지되면서 8에 대한 인식률이 97.6%로 향상되었다. 또한 3에 대한 템플릿의 수를 3개로 증가하여 실험한 결과 3의 인식률이 95.6%에서 99.1%로 증가하지만 1의 인식률이 96.4%에서 92.0%로 감소하였다. 숫자 1의 경우 오인식한 데이터 11개 중 10개를 3으로 오인식하였다. 다음으로 2, 3, 5, 8의 템플릿을 3개 사용했을 때는 2, 3, 8의 템플릿을 3개씩 사용했을 때와 별 차이 없지만 1, 2, 3, 5, 8의 템플릿을 3개씩 사용할 경우 9의 인식률이 80.8%로 감소하였다. 이것 또한 오인식한 43개의 데이터 중 40개를 1로 오인식하였다. 이는 템플릿의 수가 증가함에 따라 템플릿의 각 원소의 값이 증가하였기 때문에 나타났기 때문이다. 따라서 본 논문에서는 템플릿 수의 증가는 다른 숫자의 오인식을 초래하지 않으면서 단지 변별력만을 높여야 된다고 보고 템플릿 수를 1, 3, 4, 5, 6, 7, 9는 2개, 2, 8은 3개로 하였다. 다음은 위치 이동을 고려하여 템플릿을 이동시켜 매칭시킨 결과이다.

<표 2>를 보면 <표 1> 보다는 인식률이 향상되었음을 알 수 있다. 그러나 아직도 1, 6, 7 등은 인식률이 다른 숫자에 비해서 떨어진다. 이는 문자의 구조가 파손된 경우와 문자의 폭이나 높이가 작기 때문이었다. 문자의 구조가 파손된 경우는 외판으로 보기에 다른 문자와 혼동할 수 있지만 폭이나 높이가 작아진 경

표 2: 위치 이동을 고려하여 매칭한 결과

정인식/숫자 개수	인식률(%)
0	200/202 99.0
1	134/137 97.8
2	108/109 99.1
3	220/225 97.8
4	216/219 98.6
5	117/118 99.2
6	48/51 94.1
7	152/158 96.2
8	81/82 98.8
9	158/156 98.7

우에는 크기 정규화나 문자의 구조적 특징을 이용함으로써 이러한 문제를 해결할 수 있을 것으로 보인다[6].

5 결론 및 개선점

지로 서식 문서 상의 인쇄체 숫자는 인쇄시의 오류로 인하여 숫자의 굵기나 높이, 그리고 폭이 다를 수는 있지만 기본적으로 폰트의 유형이 한가지라는 것과 나타날 수 있는 오류의 유형이 몇 가지로 제한되어 있다는 특성을 갖는다. 따라서 본 연구에서는 이러한 데이터 특성을 효율적으로 수용할 수 있도록 템플릿을 정의한 후 매칭 방법을 통해 숫자를 인식하는 템플릿 매칭 방법에 대해 설명하였다.

K-means 알고리즘을 이용하여 학습 데이터를 클러스터링한 후 동일한 클러스터에 속한 데이터를 이용하여 템플릿을 정의하였다. 실험 데이터에 대해 매칭을 수행하여 인식한 결과 비교적 적은 수의 템플릿 만으로도 긍정적인 결과를 얻을 수 있다. 템플릿 매칭 방법에서 오인식에 가장 큰 영향을 미치는 경우는 문자의 구조가 파손된 경우와 문자의 크기가 작은 경우이다. 이러한 문제를 해결하기 위하여 문자 크기를 정규화하기 위한 연구와 구조적 특징에 기반한 결정 트리(decision tree)를 이용하는 방법에 관한 연구를 수행하고 있다.

참고 문헌

- [1] Robert M. Haralic, Linda G. Shapiro, "Computer and Robot Vision," Addison-Wesley, Washington, 1992.
- [2] Parker, "Algorithms for Image Processing and Computer Vision," John Wiley & Sons, New York, 1997.
- [3] Ruby L. Kennedy, "Solving Data Mining Problems through Pattern Recognition," Printice Hall, New Jersey, 1995-1997.
- [4] 김상우, 전운호, 최종호, "신경회로망을 이용한 인쇄체 한글 문자의 인식," 전자공학회 논문집, 2권, 2호, pp.228-234, 1990.
- [5] 이성환, "오프라인 필기체 문자 인식 기술의 현황," 정보과학회지, 11권, 5호, pp.51-65, 1993.
- [6] 김원우, 백중현, 이관용, 이일병, 변혜란, "구조적 특징 분석을 통한 무제약 필기체 숫자의 검증기 구현," 한국정보과학회 춘계 학술 논문집, pp.277-280, 1996.