

# 가중 템플릿 정합을 이용한 인쇄체 아라비아 숫자 인식

정민철

상명대학교 컴퓨터시스템공학과, 정보디스플레이연구소  
e-mail: mjung@smu.ac.kr

## Machine-printed Digit Recognition using Weighted Template Matching

Minchul Jung

Dept. of Computer System Engineering,  
Information Display Research Center, Sangmyung University

### 요 약

본 논문에서는 인쇄체 아라비아 숫자를 인식하기 위해 가중 템플릿 정합 방법을 제안한다. 가중 템플릿 정합은 패턴의 특징이 나타나는 영역에 해밍거리(Hamming Distance)의 가중치를 두어 패턴 특징을 강조하여 숫자 패턴의 인식률을 높이는 것이다. 또한 패턴의 표면을 울퉁불퉁한 영상으로 만드는 한 두 픽셀의 랜덤 노이즈를 제거하기 위하여 본 연구에서는 트리밍(trimming) 기법을 적용하였다. 실험에서는 트리밍을 하지 않고 단순 템플릿 정합을 사용했을 때의 혼돈 행렬(confusion matrix)과 트리밍을 한 후 가중 템플릿 정합을 사용했을 때 혼돈 행렬을 서로 비교해 인식률이 크게 향상된 것을 보인다.

### 1. 서론

우리는 정보의 저장이나 전달, 이해를 문서 즉 문자의 구성을 통하여 한다. 멀티미디어의 발달로 문자 외에 소리, 음악, 그래픽, 영상 등에 의해 정보를 처리가 증가하고 있으나, 공식적인 정보 처리는 문서를 통해서 이루어지고 있다. 컴퓨터를 이용하여 방대한 양의 문서 정보를 가장 효율적으로 관리하는 방법은 문서를 단지 영상으로 관리하는 것이 아니라 컴퓨터가 이해하는 형태로 저장하고 관리하는 것이다. 문자 인식 시스템(OCR; Optical Character Recognition)은 영상으로 된 문서 이미지를 컴퓨터가 처리할 수 있는 문자 코드로 바꾸어 문서내의 문자 정보를 처리하는 시스템이다 [1]. 그 응용 분야는 기존 서류들의 자동 입력으로부터 신용카드 전표 인식, 팩스 문서 자동 인식에 이르기까지 매우 광범위하다. 문자 인식 시스템에 대한 초기 연구는 문자 영상을 컴퓨터에 입력하는 방법과 컴퓨터의 능력에 오히려 제한을 받아 그 발전이 늦어졌다. 현재 하드웨어적인 기술상의 문제는 극복되었으나 소프트웨어

적인 기술상의 문제는 아직 완전히 해결되지 않았다. 문자 인식 분야에는 영문자 인식, 한글 인식, 한자 인식, 숫자 인식 등 여러 가지가 있는데 이 중 특히 숫자 인식은 자동차 번호판 인식, 수표, 지로(GIRO) 등의 자동화 처리 등의 활용 범위가 광범위하다. 그러나 숫자 인식은 하나의 예러라도 결과 처리에 있어 치명적이며 또한 포스트프로세싱을 적용하기 매우 힘든 분야이다. 본 논문에서는 인식에 있어 서로 간 혼돈을 가져오는 숫자를 제시하고 이를 해결하는 가중된 템플릿 정합 방법을 제시한다.

### 2. 템플릿 정합

템플릿 정합에서 관찰된 패턴은 각각의 클래스를 나타내는 템플릿 또는 프로토타입과 직접적으로 비교된다. 분류는 최대의 일치도 또는 최소의 불일치도에 따라 이루어진다. 수학적으로 비교는 거리의 측정을 의미한다. 입력 패턴  $x$  와 클래스  $C_i$  의 프로토타입 사이의 거리는 함수  $d(x, C_i)$  에 의해 계산되고  $x$  는 이 함수를 최소화하는 클래스에 할당된

다. 본 연구에서는 이를 위해 해밍 거리(Hamming Distance)가 사용되었다. 해밍 거리는 두 패턴의 같은 위치상의 Boolean 벡터를 서로 비교하여 동일하면 0을 다르면 1을 합계한다. 해밍 거리의 정의는  $\text{sum mod } 2$ 의 연산을 필요로 하는 데 기호,  $\oplus$ 로 나타내면:  $0 \oplus 0 = 0$ ;  $1 \oplus 1 = 0$ ;  $0 \oplus 1 = 1$ ;  $1 \oplus 0 = 1$  이다. 따라서 해밍 거리는

$$D_h[k, l] = \sum_{i=1}^d (x_{ik} \oplus x_{il}) \quad (1)$$

로 나타낼 수 있다.

그림 1은 해밍 거리를 이용한 템플릿 정합의 예를 보인다.

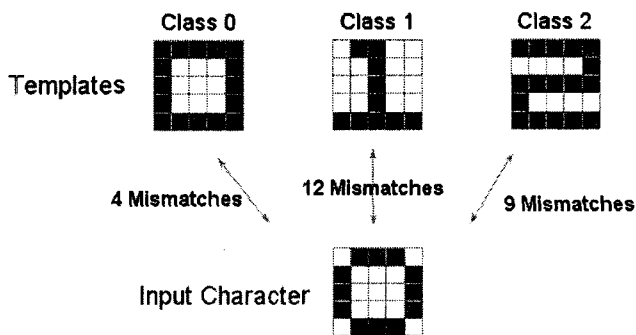


그림 1. 해밍 거리를 이용한 템플릿 정합. 비교되는 패턴의 모든 픽셀에 대해 일치될 때 0 이 불일치될 때 1 이 합산된다. 본 예에서는 첫 번째 템플릿이 최소 불일치로 입력 글자에 대한 결과로 선택된다.

### 3. 혼돈 숫자와 골격화

숫자는 글자 내부의 구조가 단순하다. 단순한 만큼 몇몇 글자는 스캔되고 영상 이진화한 후에는 글자의 내부 구조가 비슷해 OCR 시스템에게는 혼돈을 일으켜 에러를 발생시킨다. 폰트에 따라 정도의 차이는 있으나 “5 와 6”, “6, 8 과 9” 는 문자 내부적 구조가 유사하다. 그림 2 은 이러한 혼돈을 주는 숫자들의 예를 보여준다. OCR 시스템에서는 문자의 구조를 알아내기 위해 골격화(skeletonization) 처리를 사용한다. 문자의 골격은 문자의 내재된 구조를 설명하기 때문에 중요한 측정치가 된다. 보통 이러한 구조를 측정하고 나서 이미 알려진 구조와 이 구

조를 비교하여 문자의 특성을 알아내 문자를 인식할 수 있다. 그러나 이러한 숫자의 구조적 유사함은 그림 3 와 같이 문자 내의 획이 서로 접합될 경우 골격화 같은 기하학적인 특징 추출은 무의미해진다 [2,3].

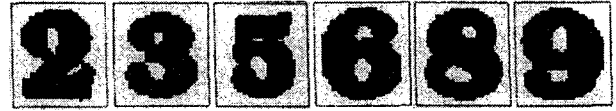


그림 2. 혼돈 숫자: 위의 모든 숫자는 내부 구조적으로 두 개의 hole 을 가진다.

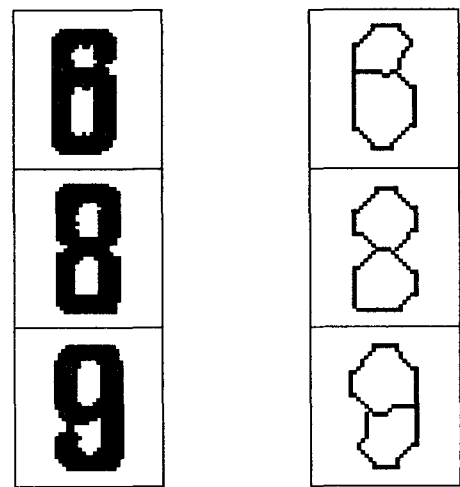


그림 3. 골격화의 한계: 우측의 골격화를 한 후 영상은 구조적으로 완전히 동일하다.

### 4. 가중 템플릿 정합

템플릿 정합은 영상의 모든 픽셀에 대해 일률적으로 해밍 거리를 취한다. 그러나 이 경우 패턴의 주요한 특징을 나타내는 부분의 해밍 거리가 그렇지 못한 부분의 해밍 거리와 합해져 그 결과는 패턴 고유의 특징을 나타내지 못한다. 더구나 그림 4 에서와 보는 바와 같이 해밍 거리 오류를 발생시킨다. 그림 4 에서 보면, 그림 4(c)의 “8”과 그림 4(a)의 “8”사이의 해밍 거리  $D_h$  는 32 인 반면 그림 4(c)의 “8”과 그림 4(b)의 “3”사이의 해밍 거리  $D_h$  는 23 이다. 마찬가지로 그림 4(d)의 “3”과 그림 4(a)의 “8” 사이의 해밍 거리  $D_h$  는 22 인 반면 그림 4(d)의 “3”과 그림 4(b)의 “3”사이의 해밍 거리  $D_h$  는 28 이다 [4]. 이를 해결하기 위해 본 연구에서는 가중 템플릿 정합을 개발하였다. 가중 템플릿 정합은 패턴의 특징이 나타나는 영역에 해밍 거리의 가중치를 두어 패

턴 특징을 강조하여 패턴의 구별을 확실하게 하는 것이다. 가중 템플릿 정합은 다음과 같이 나타낼 수 있다.

$$D_{wh}[k,l] = \sum_{i=1}^{d_1-1} (x_{ik} \oplus x_{il}) + w \sum_{i=d_1}^{d_2} (x_{ik} \oplus x_{il}) + \sum_{i=d_2+1}^d (x_{ik} \oplus x_{il}) \quad (2)$$

(2)에서  $d_1$  과  $d_2$  사이는 패턴의 특징을 나타내는 영역으로  $w$  만큼 가중된다.

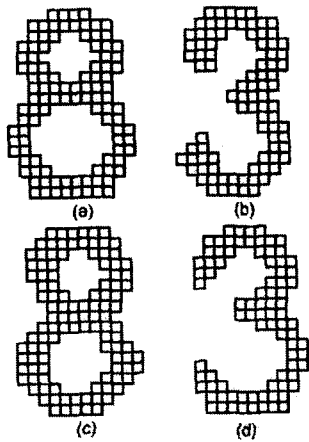


그림 4. 해밍 거리 오류. (a) 와 (b)는 프로토타입 문자. (c)와 (d)는 프린트한 후 스캔된 이미지. (참고 문헌 [4]에서 발췌한 그림)

### 5. 트리밍(trimming)을 사용한 랜덤 노이즈 제거

인쇄된 문자 패턴을 스캔하여 영상 이진화하면, 우리가 보는 종이에 인쇄된 문자와는 전혀 다른 영상이 형성된다. 이는 스캔할 때 랜덤 노이즈가 첨가되거나 영상 이진화 자체 알고리즘의 문제로 이진화 영상을 입력 영상으로 하는 OCR 시스템에게는 불가피한 현상이다. 이러한 한 두 픽셀의 랜덤 노이즈는 그림 5(a)에서 보듯이 표면이 울퉁불퉁한 영상을 만든다. 이러한 한 두 픽셀의 랜덤 노이즈는 보기에 도 안 좋을 뿐만 아니라 템플릿 정합을 사용할 때 전체적인 픽셀을 랜덤 노이즈만큼 자리 이동시켜 두 패턴의 같은 위치상의 Boolean 벡터를 서로 비교하는 해밍 거리의 오차를 크게 하여 신뢰도를 저하시킨다. 패턴의 표면을 울퉁불퉁한 영상으로 만드는 이러한 한 두 픽셀의 랜덤 노이즈를 제거하기 위하여 본 연구에서는 트리밍(trimming) 기법을 사용하였다. 이는 입력 패턴의 상하 좌우 한 픽셀라인을 단순히 깎아 내는 것으로 트리밍 후에는 크기 정규

화(size normalization)를 한다. 그림 5(b)는 크기 정규화 하기전의 트리밍을 한 영상으로 그림 6(a)에 있던 표면의 울퉁불퉁한 픽셀들이 제거되었다.

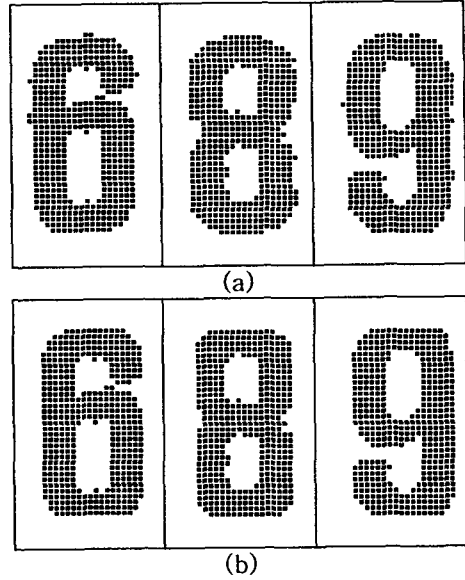


그림 5. 트리밍(trimming)을 사용한 랜덤 노이즈 제거 (a) 트리밍 전의 패턴과 (b)트리밍 후의 패턴

### 6. 실험 결과

그림 6은 본 연구에서 사용된 숫자 영상의 예를 보여준다. 이러한 숫자 영상을 각각 100개씩 입력으로 하여 단순 템플릿 정합과 가중 템플릿 정합을 각각 사용하여 두 방법을 비교 실험하였다.

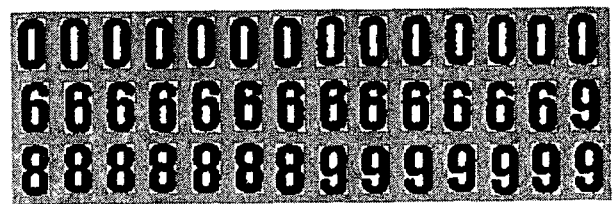


그림 6. 실험에 사용된 숫자 영상의 예

표 1은 트리밍을 하지 않고 단순 템플릿 정합을 사용했을 때의 혼돈 행렬(confusion matrix)을 나타낸다. 표 2는 트리밍을 한 후 가중 템플릿 정합을 사용했을 때 이러한 혼돈 행렬상의 에러는 모두 크게 향상된 것을 알 수 있다. 실험에서 Reject 은 모두 0로 하여 해밍거리의 신뢰도가 낮은 경우에도 결과를 출력하게 하였다. 따라서 각 표에서 해밍거리의 신뢰도가 낮은 경우에는 Error 발생률이 높아졌다.

표 1. 단순 템플릿 정합을 사용했을 때의 혼돈 행렬

Recognized as:					
True ID	0	6	8	9	Error
0	<b>93</b>	5	2	0	7
6	2	<b>94</b>	7	0	9
8	0	4	<b>96</b>	0	4
9	2	3	1	<b>94</b>	6

표 2. 가중 템플릿 정합을 사용했을 때의 혼돈 행렬

Recognized as:					
True ID	0	6	8	9	Error
0	<b>100</b>	0	0	0	0
6	0	<b>98</b>	2	0	2
8	0	1	<b>99</b>	0	1
9	0	0	0	<b>100</b>	0

### 7. 결론

본 논문에서는 템플릿 정합 방법을 개선한 가중 템플릿 정합 방법을 제안하였다. 아라비아 숫자는 스캔되고 영상 이진화 되면 문자 내부적 구조가 유사해진다. 문자의 구조를 알아내기 위해 골격화 (skeletonization)처리를 사용하는 OCR 시스템은 이러한 숫자를 서로 혼돈하여 에러를 발생시킨다. 따라서 숫자의 기하학적 구조보다는 획의 굵기 변화 같은 미묘한 차이를 특징으로 추출해야하는데 각각의 픽셀을 특징으로 하는 템플릿 정합 방법이 간단하면서도 신속하게 숫자 인식을 할 수 있다. 그러나 템플릿 정합 방법은 모든 픽셀을 동일하게 취급함으로써 패턴 고유의 특징을 나타내지 못한다. 본 논문에서 제안된 가중 템플릿 정합 방법은 패턴의 특징이 나타나는 영역에 해밍 거리의 가중치를 두어 패턴 특징을 강조하여 패턴의 구별을 확실히 하며, 이는 실험 결과에서 검증되었다. 템플릿 정합 방법의 한계이자 제안된 방법의 한계로서는 템플릿으로 저장된 프로토타입의 폰트를 벗어난 다양한 폰트의 입력에 대해 인식률이 저하된다는 것이다. 사실 다양한 폰트로 쓰인 문서의 문서 인식에서 높은 인식률을 계속 유지하는 OCR 시스템을 만드는 것은 아직 풀어야 할 연구과제이다. 본 연구에서는 각각 다른 폰트에 대해 다른 프로토타입을 만들어 이 문제를 해

결한다. 그러나 여전히 몇몇 알려진 폰트에 대해서 이방법이 가능하며 새로운 폰트에 대해 새로운 프로토타입을 만들어야 하는 단점이 있다.

### 참고 문헌

- [1] G. Nagy and S. Seth, "Modern Optical Character Recognition," *The Froehlich/Kent Encyclopedia of telecommunications*, F. Froehlich and A. Kent, eds., Vol. 11, pp. 473-531, Marcel Dekker, Inc., 1996.
- [2] M. Bokser, "Omnidocument technologies," *Proceedings of the IEEE*, Vol. 80, No. 7, pp. 1066-1078, 1992.
- [3] D. Trier and A. K. Jain and T. Taxt, "Feature extraction methods for character recognition a survey," *Pattern Recognition*, Vol. 29, No. 4, pp. 641-662, 1996.
- [4] Morton Nadler and Eric P. Smith, *Pattern Recognition Engineering*, John Wiley & Sons Inc. 1993.
- [5] S. N. Srihari, "High-performance reading machines," *Proceedings of the IEEE*, Vol. 80, No. 7, pp. 1120-1132, 1992.
- [6] 임길택, "문자형식 분류 기반의 인쇄체 문자인식에 관한 연구," 전자공학회 논문지, 40권, 5호, pp. 266-279, 2003.