

고속 처리용 문자 특징의 성능 비교

최원호 · 최재하
전자공학과

〈要 約〉

DSP 소자등을 사용하여 문자 인식 시스템을 구성하는 경우에 연산량, 인식률의 측면에서 사용 가능한 특징들의 성능을 비교하여 보았다. 비교된 특징들은 메쉬(mesh)와 receptive field를 달리하는 몇 종류의 방향성 에지 연산자들이다.

이 특징들을 문자 인식에 적용하여 성능을 비교하여 본 결과, 메쉬 보다는 방향성 에지 연산자들의 인식률이 높았으며, receptive field를 작게하여 특징 벡터의 크기를 크게 한 것이 인식률이 좋아지지만 연산량은 늘어나게 되어 시스템 구성에 부담이 된다. 본 논문에서 선택한 적절한 특징은 수평 연산자에 수평 방향의 receptive field, 수직 연산자에 수직 방향의 receptive field를 적용한 것으로서, 이들은 벡터 크기와 인식률에서 적절한 것으로 판단된다.

Comparing The Performance of Different Features for High-speed Character Recognition.

Choi, Won-Ho · Choi, Jae-Ha
Dept. of Electronic Eng.

〈Abstract〉

The performance of mesh and directional edge operator with different receptive fields, that are thought acceptable in the recognition rate and in the amount of arithmetic operation, are compared. These features are for high-speed character recognition systems based on DSP components.

The results we got from the experiments of character recognition are as follows. The recognition rate of the directional edge operator is superior to that of the mesh. The directional edge operator with small receptive field explores better recognition rate than the operators with other receptive fields, but the vector size of the former becomes twice of the latter. It means the increasing of the amount of arithmetic operation. As a result, the appropriate feature among the features we tested for high speed character recognition is the mixed feature of horizontal and vertical operator with horizontal and vertical receptive field, respectively.

I 서론

문자 인식을 실시간으로 처리 가능한 시스템을 구성하기 위해서는 고속의 프로세서가 요구되며, 병렬 처리 방식으로하여 문자 추출, 특징 추출, 인식, 후처리등의 업무를 분담하도록 구성할 필요성이 있다. 이때에 인식률을 높이기 위하여 복잡한 특징(feature)을 사용하면 연산량이 증가되어 좋은 성능을 기대하기 어려워지며, 병렬로 구성된 시스템에서 병목 현상이 발생하기도 한다. 따라서 인식률과 복잡성 간의 적절한 타협이 필요하며 이 타협으로 인하여 저하되는 인식률은 후처리 과정에서 보완하는 방법이 바람직하다.

본 논문에서는 DSP 소자들을 사용하여 문자 인식 시스템을 구성하는 경우에, 인식을 위한 적절한 특징을 선택하기 위하여, 메쉬(mesh)와 몇 종류의 방향성 edge 연산자(2, 3)에 의한 특징의 성능을 비교해 보고자 한다.

이들 특징은 연산이 매우 단순하여 실시간 처리(초당20자이상)를 가능토록 해줄 뿐만 아니라 현실적으로 받아들일 수 있을 정도의 문자인식률(90%이상)을 갖고 있는 특징들이다.

알고리즘의 단순성을 유지하기 위하여 특징 추출 시에 mask의한 correlation을 사용하고, 패턴간 유사도 기준을 correlation이나 유클리디안거리(Euclidean Distance)를 이용하였으며, 패턴으로부터 추출되는 모든 특징은 일정한 차원수(dimension)의 벡터로 표현하였다.

II. 문자 추출과 정규화

scanner를 이용하여 문자 패턴을 page단위로 2진 데이터화 한 후 이로부터 개별 문자를 추출한다. 추출 시 글자 간의 접촉이 있거나 특수 문자가 섞여있는 경우에는 다시 특별한 과정을 필요로 하게 되지만 본 논문의 실험에 사용된 문자 패턴은 이러한 것들이 없는 순수 한글의 패턴으로서 단순한 개별 문자 추출 방법이 사용되었다. 즉 행 단위로 스캔하여 후보 문자 열을 선택한후 선택된 행 단위 후보 문자 열을 이번에는 열 단위로 스캔하여 개별 문자 후보를 선택한다. 이 후보 문자에 대하여 한글의 높이와 넓이의 비를 적용하여 개별 문자를 최종적으로 선택하도록 한다.

이 방법으로 추출된 문자는 크기와 굵기등의

모양이 여러가지 형태를 가지게 된다. 일정한 크기의 특징 벡터를 만들기 위해서는 크기를 균등하게 할 필요성이 있는데 이를 위하여 정규화 과정을 필요로 한다. 본 논문에서는 정규화 과정으로서 bilinear interpolation방법(1)을 사용했다. 정규화된 개별 문자의 크기는 32×32 이다. 추출된 임의 크기의 문자영상을 $f(n_1, n_2)$, $n_1=0, \dots, N_1-1$, $n_2=0, \dots, N_2-1$ 로 표현하면 정규화된 문자 영상은 다음과 같다.

$$F(x, y) = (1-dx)(1-dy)f(n_1, n_2) \\ + (1-dx)dyf(n_1, n_2+1) \\ + dx(1-dy)f(n_1+1, n_2) \\ + dx dyf(n_1+1, n_2+1) \quad (1)$$

여기서,

$$dx = \frac{x/31 - n_1 T_1}{T_1}, \quad T_1 = \frac{1}{N_1 - 1} \\ dy = \frac{y/31 - n_2 T_2}{T_2}, \quad T_2 = \frac{1}{N_2 - 1}$$

III. 특징 추출

서론에서 지적한 바와 같이 문자 인식에서 소요되는 시간을 줄이기 위해서는 알고리즘 자체가 단순 명료하여야 하고 병렬 처리, 학습 등을 효과적으로 수행할 필요성이 있는데 이를 위하여 특징은 단순한 벡터 형식의 것이 바람직하다. 이러한 특징으로서 스캔 방식에 의한 투영(projection)특징, 그리고 mesh특징 등을 들 수가 있다. 스캔 방식에 의한 투영 특징은 그림 1과 같이 수평 또는 수직 에지 성분을 발생하는 scanner의 특성에 의한 noise로 인하여 전처리(preprocessing)과정으로서 세선화(thinning)를 하지 않으면 심각한 에러를 발생하기도 한다. 그러나 전처리 과정이 많으면 많을 수록 인식 시간은 늘어나게 되며 더구나 이 특징은 벡터 크기가 커서 연산 시간이 많이 소요되는 단점도 안고 있다.

추출된 문자 패턴을 일정한 영역으로 나누어서 그 block 내의 흑화소 수를 세어 분포를 파악하는 것이 mesh특징이다. mesh특징은 noise에 대해 비교적 강하며, 특징 추출이 단순하고, 추출된 특징 벡터의 크기가 작아서 빠른 속도로 처리할 수 있는 장점을 갖고 있다.

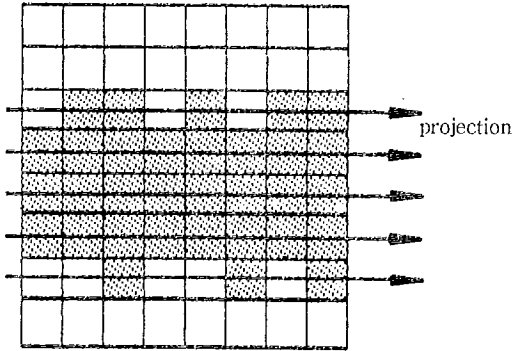


그림 1. projection 특징

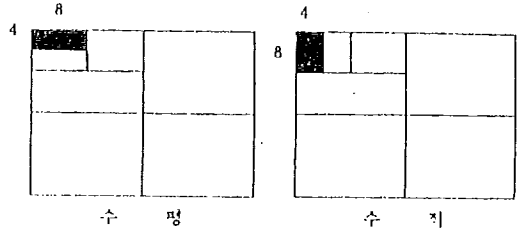
그러나 방향의 구별이 어렵고 패턴 변형, 위치 변동, 문자의 폭 변화를 흡수하지 못하는 단점을 갖고 있다. 따라서 사용하고자 하는 특징은 다음과 같은 성격을 갖고 있는 것이 좋다.

- 1) 추출 방법이 단순하고 용이하여야 한다.
- 2) 방향 특징의 표현이 포함되어야 한다.
- 3) 패턴 변형, 위치 변동을 부분적으로 흡수할 수 있어야 한다.
- 4) 문자폭의 변화를 흡수할 수 있어야 한다.

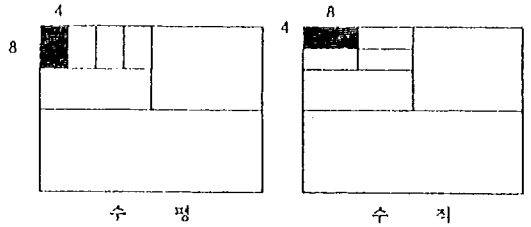
인간 혹은 포유류의 경우 낮은 레벨에서의 주요 시각 정보 처리 과정은 edge와 방향성의 추출로 이루어진다. (4) 이를 위하여 방향성 edge연산자를 적용하면 문자 폭의 변화를 흡수할 수 있고 방향 특징을 표현할 수 있으며 마스크에 의한 연산자 적용이 주 연산이므로 추출이 단순 용이하다. 위치 변동, 패턴 변형, 문자 폭의 변화등에 무관한 특징을 선택하는 것은 매우 어려운 문제이며, 특징의 벡터 표현 시 한 벡터 요소의 값을 결정하기 위해 그 대상으로 삼는 블록의 모양, 즉 receptive field의 모양이 관건이 된다. (5) 본 논문에서는 방향성 에지 연산자에 여러 receptive field를 적용하여 이들의 성능을 비교 분석하고자 한다. 사용된 방향성 에지 연산자는 그림 2와 같이 수평 방향과 수직 방향의 변화를 검출할 수 있는 단순한 연산자이다.

-1 -1 -1	1 0 1
0 0 0	-1 0 1
1 1 1	-1 0 1

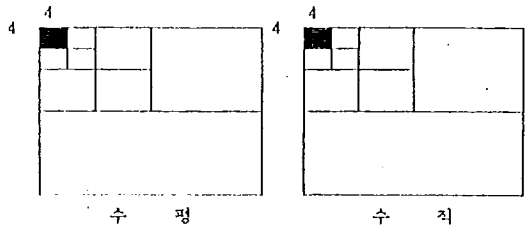
수평방향 수직방향
그림 2. 방향성 에지 연산자



(a) receptive field 1



(b) receptive field 2



(c) receptive field 3

그림 3. 방향성 에지 연산자를 적용한 receptive field의 모양

이 연산자를 이용하여 에지를 검출한 다음 그림 3과 같이 receptive field를 여러 모양으로 취하였다. (a)그림에서 수평 연산자를 취해서 얻은 값에 4x8 receptive field를, 수직 연산자에는 8x4 receptive field를 사용하여 특징 벡터를 구하였

고, (b)그림 receptive field 2는 (a)와 반대로 수평 연산자의 결과 값에 8×4 를, 수직 연산자에는 4×8 을 각각 취하였다. 다음 (c)그림의 receptive field 3는 수평, 수직연산자 모두에 4×4 를 취한 경우이다.

이렇게하면 (a)와 (b)의 경우에는 한 문자에 대하여 64개의 원소로 구성된 특징 벡터를 구할 수 있으며 (c)의 경우에는 특징 벡터가 128개의 원소로 구성된다.

IV. 분류 실험 및 결과 고찰

4.1 실험 방법

본 논문에서 고속처리가 가능한 특징들의 성능을 파악하기 위하여 한글 패턴을 분류해 보았다. 실험의 과정은 그림 4에서 보는 바와 같이 먼저 스캐너를 이용하여 데이터를 받은 다음 이 중에서 공백은 버리고 문자만을 추출한다. 추출된 문자를 32×32 화소에 맞도록 정규화 작업을 한다.

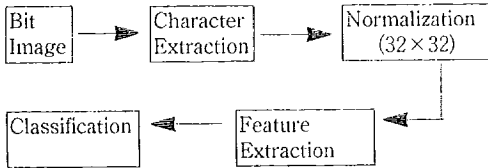


그림 5. 문자 인식 방법

정규화된 개별 문자로 부터 앞에서 설명한 특징들을 추출한다. 추출된 성능을 간략히 평가하기 위하여 패턴 분류 알고리즘 중 가장 단순한 EDC(Euclidean Distance Classifier)를 사용하여 분류하였다. 이 EDC방식은 각 클래스별로 학습 패턴 벡터의 평균을 취하여 구한 template 벡터(prototype, exemplar)를 갖고 있다가 테스트시에 입력되는 임의의 패턴 벡터와 이 template 벡터와의 유클리디안 거리를 계산하여 최소 거리를 갖는 클래스로 분류하는 방식이다. (6) 즉 벡터 j 와 입력 패턴 벡터에 대하여 다음 식의 거리를 계산하여 거리가 가장 작은 클래스로 분류하게 된다.

$$D_j^2 = \sum_{i=0}^{M-1} (x_i - t_{ji})^2 \quad (2)$$

여기서

$$T_j = \{t_{j0}, t_{j1}, \dots, t_{jM-1}\}^T$$

$$X = \{x_0, x_1, \dots, x_{M-1}\}^T$$

이고 M 은 특징 벡터의 차수(dimension)이다.

실험에 사용한 데이터는 문자 모양, 크기를 다양하게 선택하였으며 500자의 문자를 1 set로 하여 학습 데이터로 6 set, 테스트 데이터로 12 set를 사용하였다.

4.2 Mesh 특징의 분류 실험

표 1. Mesh 특징에 의한 분류 결과

	1	2	3
1	325(65.0)	83(81.6)	29(87.4)
2	384(76.8)	60(88.8)	30(94.8)
3	396(79.2)	71(93.4)	16(96.6)
4	412(82.4)	56(93.6)	18(97.2)
5	412(82.4)	55(93.4)	17(96.8)
6	379(75.8)	73(90.4)	22(94.8)
7	252(50.4)	83(67.0)	50(77.0)
8	397(79.4)	64(92.2)	21(96.4)
9	390(78.0)	69(91.8)	17(95.2)
10	428(85.6)	46(94.8)	11(97.0)
11	402(80.4)	53(91.0)	24(95.8)
12	378(75.6)	72(90.0)	22(94.4)
평균	75.9	89.0	93.6

Mesh 특징은 방향성 에지 연산자를 취하지 않고 4×4 의 receptive field에 대해 흑화소의 수를 세어 특징 값으로 삼는다. 특징의 크기는 64이다. 이 특징에 대해 4.1절에서 설명한 분류 방법을 사용하여 12 set (12×500 문자)의 데이터를 분류해 본 결과를 표 1에 나타내었다. 이 표는 후보 갯수가 하나일 때, 또 둘, 셋 일 때의 분류율을 모두 보여주고 있다.

괄호 왼쪽에 있는 숫자는 500자 중 정분류의 갯수이고 괄호안은 누적 분류율이다. 후보 갯수의 의미는 1인 경우 최소 거리에서 정분류된 경우이고 2인 경우는 두번 째의 최소 거리에서 정분류된 경우이다.

4.3 RF1의 방향성 에지 연산자에 의한 특징의 분류 실험

그림 3의 방향성 에지 연산자를 정규화된 문자 패턴에 correlation을 취하여 구한 값에 그림 4(a)의 receptive field 1을 이용하여 특징 벡터를 구한다. 이 경우 mesh와 마찬가지로 특징의 크기는 64이다.

이 특징 벡터들을 분류 실험에 적용해 본 결과를 표 2에 나타내었다.

표 2. RF1을 이용한 분류 결과

	1	2	3
1	401 (80.2)	56 (91.4)	22 (95.8)
2	426 (85.2)	55 (96.2)	11 (98.4)
3	440 (88.0)	39 (95.8)	10 (97.8)
4	448 (89.6)	35 (96.6)	12 (99.0)
5	454 (90.8)	30 (96.8)	14 (99.6)
6	377 (75.4)	60 (87.4)	22 (91.8)
7	364 (72.8)	71 (87.0)	24 (91.8)
8	438 (87.6)	48 (97.2)	9 (99.0)
9	444 (88.8)	39 (96.6)	7 (98.0)
10	446 (89.2)	35 (96.2)	14 (99.0)
11	461 (92.2)	27 (97.6)	9 (99.4)
12	367 (73.4)	52 (83.8)	26 (89.0)
평균	84.3	93.5	96.5

이 표에서 알 수 있듯이 방향성 에지 연산자를 사용한 것이 mesh보다 인식률에서 전반적으로 향상되었다.

4.4 RF2의 방향성 에지 연산자에 의한 특징의 분류 실험

그림 2와 같은 방향성 에지 연산자를 사용하고 receptive field를 그림 4(b)와 같은 RF2로 바꾸어 구한 특징 벡터를 이용하여 분류 실험을 하였다. 이 특징 벡터의 크기도 64이다.

receptive field 2를 사용하는 경우에는 mesh보다 인식률을 높일 수 있었으나 receptive field 1을 사용하는 경우보다는 인식률이 낮아진 것을 알 수 있다.

표 3. RF2를 이용한 분류 결과

	1	2	3
1	361 (72.2)	57 (83.6)	28 (89.2)
2	436 (87.2)	34 (94.0)	15 (97.0)
3	435 (87.0)	28 (92.6)	11 (94.8)
4	437 (87.4)	42 (95.8)	10 (97.8)
5	466 (93.2)	26 (98.4)	3 (99.0)
6	426 (85.2)	41 (93.4)	11 (95.6)
7	315 (63.0)	59 (74.8)	32 (81.2)
8	441 (88.2)	36 (95.4)	12 (97.8)
9	432 (86.4)	41 (94.6)	7 (96.0)
10	423 (84.6)	41 (92.8)	15 (95.8)
11	455 (91.0)	34 (97.8)	9 (99.6)
12	430 (86.0)	38 (93.6)	12 (96.0)
평균	84.3	92.2	95.0

4.5 RF3의 방향성 에지 연산자에 의한 특징의 분류 실험

수평과 수직 모두 4×4의 receptive field를 적용한 특징을 추출하여 분류 실험을 하였다. 특징의 크기는 다른 특징들의 두 배인 128이다.

표 4. RF3를 이용한 분류 결과

	1	2	3
1	428 (85.6)	49 (95.4)	8 (97.0)
2	447 (89.4)	38 (97.0)	9 (98.8)
3	464 (92.8)	28 (98.4)	5 (99.4)
4	467 (93.4)	23 (98.0)	8 (99.6)
5	482 (96.4)	13 (99.0)	4 (99.8)
6	427 (85.4)	43 (94.0)	16 (97.2)
7	404 (80.8)	62 (93.2)	15 (96.2)
8	463 (92.6)	20 (98.8)	3 (99.4)
9	474 (94.8)	20 (98.8)	3 (99.4)
10	467 (93.4)	22 (97.8)	9 (99.6)
11	482 (96.4)	15 (99.4)	2 (99.8)
12	407 (81.4)	49 (91.2)	20 (95.2)
평균	90.2	96.7	98.5

4 X 4 receptive field를 적용하면 인식률이 비교적 많이 상승되는 것을 알 수 있다. 그러나 특징 벡터의 크기가 두 배이므로 그만큼 연산량이 늘어나는 단점을 갖고 있다.

4.6 결과 비교

mesh와 receptive field 1, 2, 3의 결과를 표 5에 나타내었다.

표 5. 분류 결과의 비교

특징 추출 방식	후 보 겐 수			특징 크기
	1	2	3	
mesh특징	75.9	89.0	93.6	64
RF1	84.3	93.5	96.5	64
RF2	84.3	92.2	95.0	64
RF3	90.2	96.7	98.5	128

mesh의 특징은 후보 겐수를 3개로 하더라도 인식률이 94%이하로서 후처리 과정을 거친다해도 좋은 인식률을 기대하기 어렵다. 특징의 크기가 같은 RF1과 RF2를 비교해 볼 때 RF1의 인식률이 약간 높은 것을 알 수 있다. 이 두 특징 벡터의 크기가 같고 특징 추출시 연산량도 같으므로 인식 소요 시간은 똑같다고 볼 수 있다.

RF3에 있어서는 분류율이 가장 좋으나 특징의 크기가 다른 특징의 두배이고 따라서 연산량도 그 만큼 커지게 된다.

V. 결 론

본 논문에서 DSP 소자들을 이용하여 문자 인식 시스템을 구성할 때 고속 처리가 가능하고 추출이 용이한 몇 개의 특징들의 성능을 비교해 보

았다. 물론 이 특징들 보다 우수한 성능의 특징이 많이 개발되었지만 여기서 실험한 특징들은 상용화가 가능하도록 하기 위하여 추출과 처리시의 소요 시간이 작은 단순한 특징들만을 선택하였다. 결과에서 알 수 있듯이 3개의 후보를 갖고 다시 후처리 과정을 거친 후의 인식률이 95~96%를 요구하는 경우에는 RF1의 특징이 적절하며, 97~98%의 인식률이 요구되는 경우에는 연산 시간이 다소 길어지더라도 RF3의 특징을 선택하는 것이 적절하다.

참 고 문 헌

1. A. Rosenfeld, A. C. Kak, *Digital Picture Processing, vol I, II*, Academic Press, 1982.
2. J. T. Too, R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, 1974.
3. R. O. Duda, P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, 1973.
4. M. D. Levine, *Vision in Man and Machine*, McGraw-Hill, 1985.
5. T. Y. Young, K. S. Fu, ed., *Handbook of Pattern Recognition and Image Processing*, Academic Press, 1986.
6. 최원호, 최동혁, 이병래, 박규태, "한글 인식을 위한 신경망 분류기의 응용," 대한전자공학회 논문지, 제27권 8호, pp. 93-103, 1990.