

변형에 의한 필기체 한글의 생성과 이를 이용한 한글 문자인식 시스템의 정량적 평가

(Automatic Generation of Handwritten Hangeul Character Images and its Application to the Evaluation of Hangeul Character Recognition Systems)

朴相泰*, 方勝楊**

(Sang Tae Park and Sung Yang Bang)

要 約

현재 다양한 한글 문자인식 시스템들이 제안되고 있지만, 이들을 평가하는 방법으로는 거의 대부분 임의로 수집된 평가용 데이터에 대해 평균인식율을 측정하는 방법만을 사용하고 있다. 그러나, 평균인식율은 그 인식시스템의 평균적인 인식 성능만을 말해줄 뿐, 세부적인 인식특성을 파악하는 데에는 미흡한 점이 없지 않다. 본 논문에서는 평가용 데이터를 인위적으로 생성하여 평가용 데이터의 난이도를 사전에 알 수 있도록 하는 방법을 제안한다. 이렇게 난이도별로 분류된 평가용 데이터를 가지고 문자인식 시스템을 평가할 경우, 평균인식율만으로는 파악할 수 없었던 난이도에 따른 인식율 변화곡선이나 특정 변형 경향에 대한 인식 특성 등, 인식시스템의 세부적인 성능을 파악할 수 있다. 본 논문에서는 먼저 필기체 한글 문자패턴을 자동적으로 생성해내는 방법을 소개하고, 몇 가지 가상적인 인식시스템에 대한 세부적인 인식 특성을 평가하는 실험을 행함으로써 본 평가방법의 유용성을 검증한다.

Abstract

There is a basic problem with the current evaluation method for character recognition systems. The current method evaluates the average recognition rate by applying the test data to the target system. The average recognition rate tells no more than and no less than the overall performance and it depends on the data. In this paper we propose a testing method which will analyze the target system and point out its strong points and weak points. This can be made possible through using the data which are generated by distorting the standard character images according to a carefully controlled manner. This paper will describe how to automatically generate such distorted images. Also we will show the method is actually effective and useful by applying it to evaluating existing recognition algorithms.

*準會員, **正會員, 浦項工科大学 電子計算學科
(Dept. of Computer Science & Eng.
POSTECH)

(※本 研究는 産業科學技術研究所의 純粹基礎 課題에
서 一部 支援되었음.)

接受日字 1992年 10月 10日

1. 서론

패턴인식 가운데 문자인식 분야는 그 직접적인 실용성 때문에 오래 전부터 연구되었으며 그간 많은 발전을 보아, 지금은 어느 정도 실용화된 상품까지 나오게 되었다. 이는 한글 문자인식의 경우에도 마찬가지이며, 최근에는 기존의 연구방법들이 지니는 한계를 극복하기 위하여 이제까지와는 다른 새로운 방법론, 즉 신경회로망, 퍼지로지, 비전 등으로부터 새로운 이론들을 도입하고 있다.

그러나 다양한 방법론의 문자인식 연구에서 공통적으로 안고 있는 큰 문제점 중의 하나는, 인식방법의 개발, 구현, 인식을 실험 등을 개발자 개개인이 개별적으로 행함으로써 개발된 문자인식 방법의 성능을 객관적으로 검증할 수 없다는 것이다. 이는 또한 서로 다른 방법들간의 객관적인 비교를 어렵게 하며, 각 개발자는 실험에 사용될 문자 데이터를 주변 사람들로부터 얻게 되기 때문에 객관적인 데이터를 얻기가 힘들 뿐만아니라 많은 연구 시간을 데이터 수집에 소비하게 된다.

표준적으로 제시되는 실험용 데이터가 몇 가지 주어진다고 하더라도 제대로 인식 실험을 행하기 위해서는 우선 그 데이터가 다양이어야 한다. 일단 많은 양의 데이터를 얻게 되더라도 그 데이터가 다양한 변형의 문자를 수용하지 못할 때에는 이또한 문제가 된다. 특히 필기체 문자인식의 경우에는 동일한 서체의 문자에 대해서도 많은 변형이 있을 수 있기 때문에 이 모든 변형들을 표준으로 제공해 줄 수는 없는 것이다. 또한 실험용 데이터들이 다양한 변형을 수용하고 있더라도 그 중 선택되는 데이터의 난이도가 규명되어있지 않은 상황에서는 평가용 데이터의 선택이 평가되는 인식시스템의 성능에 많은 영향을 끼친다. 따라서 하나의 표준적인 데이터를 기본으로하여 여러 가지 변형을 능동적으로 얻어내는 방법이 필요하게 된다.

필기체 문자의 인위적 생성에 관한 연구는 국내에서는 거의 찾아보기 어려우나 일본의 경우 오래전부터 이루어져 왔다.^[1, 2] 변형의 대상이 되는 표준문자는 대부분 벡터형태를 가지는데, 일본의 가나문자나 한자의 경우 ETL 8, ETL 9 등 필기체 데이터베이스가 구축되어있어 이들로부터 표준문자를 얻거나^[3], 개인용 컴퓨터에서 널리 쓰이고 있는 비트맵 폰트를 벡터화하여 표준문자를 얻는 방법을 사용하고 있다.^[2, 4] 표준문자로부터 필기체 문자로 변형시키는 방법으로는 먼저 소량의 필기 데이터를 수집한 후 이

들을 벡터화한 후 수집된 문자와 표준문자 사이를 보간하는 방법^[3] 과 표준문자에 직접 여러가지 변형 알고리즘을 적용하는 방법^[2] 의 두 가지가 주로 사용되고 있다.

본 논문에서는 한글이 일본의 가나문자나 한자와는 달리 자소의 조합에 의해 문자를 이룬다는 점에 착안하여 자소의 벡터형을 제작한 후 이를 조합함으로써 벡터형 표준문자를 만들어내는 방법을 사용하였다. 또 변형문자를 생성할 때에는 별도의 데이터 수집이 필요없는 표준문자에 직접 변형 알고리즘을 적용하는 방법을 택하였다. 그러나, 한글의 특성을 감안하여 일본 가나문자나 한자에서와 같이 문자 전체에 대하여 변형 알고리즘을 적용하지 않고 각 자소에 대하여 변형을 적용하였으며, 한글 문자의 구조에 적합한 새로운 변형 알고리즘을 추가하였다.

인위적으로 생성된 문자 패턴을 문자인식 시스템의 평가용으로 적용한 사례도 국내에서는 아직 없었지만, 일본에서는 문자의 난이도에 대한 안정성 등을 평가하는 방법 등이 이미 연구되었다.^[3] 본 논문에서는 한결음 더 나아가 난이도 변화에 따른 인식율의 변화를 추적할 뿐만아니라 한글 필기에 자주 나타나는 변형들 각각에 대한 인식시스템의 인식 특성을 관찰함으로써 인식시스템의 보다 세부적인 특성을 파악할 수 있도록 하였다. 즉 이를 이용하여 인식시스템이 어떤 변형에 강하고 약한지를 파악할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 변형의 대상이 되는 표준문자의 생성과정을 밝히고, 3장에서는 이들을 필기체 형태로 변형시키는 방법에 관하여 기술하며, 4장에서는 가상적인 4 가지 인식시스템을 구성하고 인위적으로 생성된 데이터를 이용하여 이들의 성능을 정밀하게 분석하는 실험을 행한다. 끝으로 5장에서는 결론을 맺는다.

II. 표준문자의 생성

여기서 말하는 표준문자는 변형의 대상이 되는 문자를 말하는데, 각종 변형이 용이하게 이루어지게 하기 위해서 이들 표준문자는 벡터 형태로 표현하였다. 모든 한글 문자에 대한 벡터 형태를 직접 제작한다는 것은 그 하나만으로 너무 부담이 큰 일이므로, 본 논문에서는 모든 한글 자소의 벡터형을 제작한 다음 이들을 조합함으로써 한글 문자의 벡터형을 손쉽게 얻었다. 자소를 조합할 때에는 해당 문자의 형태에 따라 그림 1과 같이 분류하여, 조합하였다. 벡터형의

경우 확대, 축소 및 위치 변화가 간단하므로, 문자형태의 수를 늘이는 것도 비트맵 자형을 제작할 때보다 훨씬 간단하다.

생성된 표준문자에 대해 보다 상세히 알아보자. 하나의 한글 문자는 종성의 유무에 따라 2 또는 3개의 자소로 구성되고, 각 자소는 최소 1획에서 최대 8획(예, ㅁ)으로 구성된다. 여기서 하나의 획은 40개의 좌표로 정의하였다. 본 논문에서 제작한 자소는 초성 19자(ㄱ, ㅋ, ㆁ, ..., ㅎ)의 경우 우측 모음용과 아래 모음용으로 모두 2벌이고, 중성 21자(ㅏ, ㅑ, ..., ㅣ)와 종성 27자(ㄱ, ㅋ, ..., ㅎ)의 경우에는 각각 1벌뿐이다. 생성된 자소들을 이용하여 표준문자를 생성하는 것은 해당 한글 문자의 형태에 따라 그림 1에 나타낸 18가지의 문자형태 가운데 하나를 선택한 다음, 이 형태에 맞게 초성, 중성, 종성의 위치와 크기를 정함으로써 쉽게 해결된다.

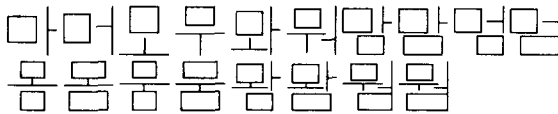


그림 1. 18가지 문자 형태 분류

Fig. 1. 18 character shapes for combination.

III. 변형문자의 생성

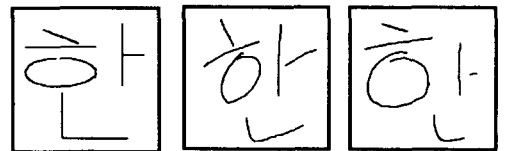
필기체에는 표준이 없다. 즉, 글씨를 쓰는 개개인이 서로 다른 필체를 가지고 있을 뿐만 아니라 같은 사람이 쓰는 경우라 하더라도 그 필기가 일어나는 환경에 따라 다른 글씨가 발생하게 된다. 따라서 모든 한글 필기를 만들어낼 수 있는 변형기법이란 존재하지 않는다. 그러므로 여기서는 실제의 한글 필기에서 자주 나타나는 변형들을 가능한 한 포괄적으로 수용하는 변형기법을 찾는 데 주력하였다. 그렇다면 실제의 한글 필기가 앞절에서 말한 표준문자와 다른 점은 무엇인가. 여러가지 차이가 있지만, 본 논문에서는 다음과 같은 두 가지 요인에 초점을 맞추었다.

첫째, 실제 한글 필기에서 사용되는 자소의 모양은 표준문자의 그것과 많이 달라진다. 즉, 전체적인 기울기, 꺾인 점에서의 부드러움 정도, 필체에 따른 특정 방향으로의 치우침, 획의 굵기 등등이 모든 면에서 정규화된 표준문자의 자소 모양과는 사뭇 다르다.

둘째, 실제의 필기에서는 각 자소의 상대적 위치나 크기가 표준문자에서와 같이 고정되어 있지 않다. 즉, 필체에 따라 초성, 중성, 종성의 위치와 상대적

크기가 천차만별로 나타나게 되는데, 이것이 한글 필기 문자의 형태를 결정하는 가장 큰 요인이 된다.

그림 2에서 표준문자와 실제 한글 필기를 비교하였는데, 여기서 나타난 바와 같이 한글의 필기에서는 필기의 기본 단위가 하나의 문자가 아니라 한 개의 자소가 된다. 즉, 일본의 가나문자나 한자와 같은 음절문자에서는 필기의 기본단위가 한 개의 문자가 되지만, 한글은 자소들의 이차원적 조합으로 문자를 구성하기 때문에 각각의 자소가 필기의 기본단위가 되는 것이다. 따라서 실제의 한글 필기에 가까운 변형 문자를 얻기 위해서는 일본의 관련 문헌들에서와 같이 변형 알고리즘을 문자전체에 대하여 적용하기보다는 각각의 자소에 대하여 적용하는 것이 바람직하다. 또한 이와 같이 자소들에 대한 변형이 끝난 이후에는 문자전체에 대한 변형에서는 고려할 필요가 없었던 조합시의 변형을 고려해야만 한다. 즉, 앞서 언급한 자소의 상대적 위치 및 크기의 변형이 여기서 말하는 조합시의 변형에 해당한다. 그러면 각 자소들을 실제 필기에 나타나는 여러 형태로 변형시키는 기법들을 구체적으로 알아보자.



(a) 표준문자

(b) 실제의 한글 필기

그림 2. 표준문자와 실제 한글 필기의 비교

Fig. 2. Comparison between standard character pattern and real hand written character.

변형의 대상이 되는 표준문자의 한 자소를 J라 하고, J가 N(J)개의 획으로 구성되어 있다고 하면, J는 다음과 같은 벡터로 나타낼 수 있다.

$$J = (S_1, S_2, \dots, S_{N(J)}) \tag{1}$$

이때, $S_i (1 \leq i \leq N(J))$ 는 자소 J를 구성하는 i번째 획이다. 또한 S_i 가 M개의 좌표로 구성되었다면, 이를 다음과 같은 벡터로 나타낼 수 있다.

$$S_i = (S_{i1}, S_{i2}, \dots, S_{iM}) \tag{2}$$

본 논문에서는 모든 획이 40개의 좌표로 구성되도

록 고정시켰으므로, 여기서의 M은 상수 40이 되고, s_{ij} 는 획 S를 구성하는 번째 좌표가 된다. 하나의 좌표 s_{ij} 를 다음과 같이 나타낸다.

$$s_{ij} = (s_{ij,x}, s_{ij,y}) \tag{3}$$

이와 같이 나타나는 자소 J에 대하여 다음과 같이 회전, 블록화, 오목화, 기울이기, 역사다리꼴 변형 등을 적용한다.

1. 회전 (Rotation)

이러한 변형의 목적이 실제의 필기와 유사한 필기체를 인위적으로 생성하는 것인 만큼 실제의 필기에서 나타나지 않는 과도한 변형은 제거하였다. 즉, 회전 변형의 경우, 실제 필기에서 시계방향으로 회전하는 경우는 거의 없기 때문에 반시계방향으로의 회전만을 허용하였다. 또한 30도이상 회전되는 경우도 거의 없으므로 이또한 배제하여 회전 변형의 범위를 0도에서 30도 사이로 제한하였다. 이때, 회전 변형의 정도를 α 라 하고 그 범위를 $0.0 \leq \alpha \leq 1.0$ 와 같이 주면, α 에 따른 실제 회전각도 t 는

$$t = 30 \times \alpha \tag{4}$$

와 같고, 회전 변형 방법은 다음과 같다. 먼저 자소 J의 중심 $O = (O_x, O_y)$ 를 다음과 같이 구한다.

$$\text{Min}_x = \text{MIN} \{s_{ij,x} \mid s_{ij} \in S_i, S_i \in J, 1 \leq i \leq N(J), 1 \leq j \leq M\} \tag{5-a}$$

$$\text{Min}_y = \text{MIN} \{s_{ij,y} \mid s_{ij} \in S_i, S_i \in J, 1 \leq i \leq N(J), 1 \leq j \leq M\} \tag{5-b}$$

$$\text{Max}_x = \text{MAX} \{s_{ij,x} \mid s_{ij} \in S_i, S_i \in J, 1 \leq i \leq N(J), 1 \leq j \leq M\} \tag{5-c}$$

$$\text{Max}_y = \text{MAX} \{s_{ij,y} \mid s_{ij} \in S_i, S_i \in J, 1 \leq i \leq N(J), 1 \leq j \leq M\} \tag{5-d}$$

$$O_x = (\text{Max}_x + \text{Min}_x) / 2 \tag{5-e}$$

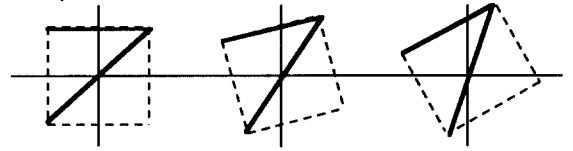
$$O_y = (\text{Max}_y + \text{Min}_y) / 2 \tag{5-f}$$

$s_{ij} \in S_i, S_i \in J$ 인 모든 좌표 s_{ij} 가 다음과 같이 s'_{ij} 으로 이동하면, 실제 회전 변형이 이루어진다.

$$s'_{ij,x} = (\cos(t) * (s_{ij,x} - O_x) - \sin(t) * (s_{ij,y} - O_y)) + O_x \tag{6-a}$$

$$s'_{ij,y} = (\sin(t) * (s_{ij,x} - O_x) + \cos(t) * (s_{ij,y} - O_y)) + O_y \tag{6-a}$$

그림 3에서 회전 변형의 예를 보여준다.



(a) $\alpha=0.0$ 인 경우 (b) $\alpha=0.5$ 인 경우 (c) $\alpha=1.0$ 인 경우

그림 3. 회전 변형

Fig. 3. Rotation distortion.

2. 블록화 (Barrel Distortion)

블록화 변형의 요점은 자소의 중심을 구하고 이 중심으로부터 거리가 먼 점일수록 중심쪽으로 많이 이동시키는 것이다. 그러나 여기서 말하는 중심은 (식 5)에서와 같은 위상적인 중심보다는 무게의 중심을 뜻한다. 무게중심 $G = (G_x, G_y)$ 는 다음과 같이 구해진다.

$$G_x = \frac{1}{N(J) * M} \sum_i^{N(J)} \sum_j^M s_{ij,x} \tag{7-a}$$

$$G_y = \frac{1}{N(J) * M} \sum_i^{N(J)} \sum_j^M s_{ij,y} \tag{7-b}$$

중심을 구한 다음에는 각 점 s_{ij} 를 다음과 같이 s'_{ij} 으로 이동시키면 변형이 이루어진다. 이때 블록화 변형의 정도를 β ($0.0 \leq \beta \leq 1.0$)로 나타내자.

$$x = s_{ij,x} - G_x \tag{8-a}$$

$$y = s_{ij,y} - G_y \tag{8-b}$$

$$s'_{ij,x} = \beta * x * (R / \sqrt{(x^2 + y^2)} - 1) + x + G_x \tag{8-c}$$

$$s'_{ij,y} = \beta * y * (R / \sqrt{(x^2 + y^2)} - 1) + y + G_y \tag{8-d}$$

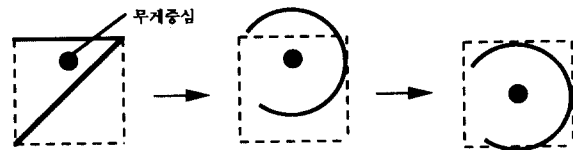


그림 4. ㄱ에 대한 블록화 변환($\beta=1.0$)

Fig. 4. Barrel distortion.

여기서, R은 미리 정의된 상수로서 블록화가 극단적일 경우, 즉 β 가 1일 경우 사각형의 경계가 원으로 변할 때, 그 원의 반지름이 된다. 실제로 블록화를 적용할 때에는 자소 J를 정사각형이 되도록 가로

세로의 크기를 조절한 후 (식 7), (식 8)를 적용하고, 모든 점에 대한 변형이 끝난 후 다시 원래의 크기 및 위치로 복귀시키는 방법을 사용한다. 그림 4는 볼록화 변형의 예를 보여준다.

3. 오목화 (Pincushion Distortion)

오목화 변형은 볼록화의 반대 개념으로 중심으로부터의 거리가 먼 점일수록 더 멀리 떨어지도록 변형하는 것이다. 따라서, 중심을 구하는 과정은 볼록화에서와 같이 (식 7)을 사용하고 각 점의 변형방법만 (식 8)로부터 약간 수정한다. 변형의 정도를 x ($0.0 \leq x \leq 1.0$)로 나타내면, 각 점 S_{ij} 의 변형식은 다음과 같다.

$$x = S_{ij,x} - G_x \tag{9-a}$$

$$y = S_{ij,y} - G_y \tag{9-b}$$

$$S'_{ij,x} = \chi * x * (\sqrt{(x^2 + y^2)} / R - 1) + x + G_x \tag{9-c}$$

$$S'_{ij,y} = \chi * y * (\sqrt{(x^2 + y^2)} / R - 1) + y + G_y \tag{9-d}$$

그림 5는 오목화의 예를 보여준다. 오목화나 볼록화는 회전을 비롯한 다른 변형들과는 달리 비선형 변형이기 때문에, 변형 알고리즘 자체는 다소 복잡하지만 실제의 다양한 필체를 나타내는 데에는 커다란 역할을 한다.

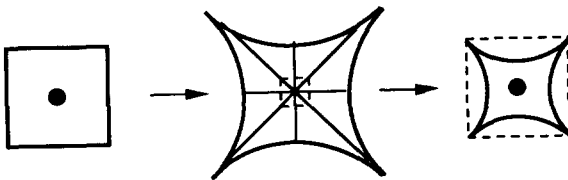


그림 5. □에 대한 오목화 변환 ($\chi = 1.0$)

Fig. 5. Pincushion distortion.

4. 기울이기 (Skew Distortion)

기울이기 변형은 그림 6에서 보는 바와 같은 간단한 선형 변형이다. 그러나, 그 효과는 오목화와 회전 변형을 동시에 한 결과와 비슷해진다. 변형의 요점은 x 축 방향으로 중심과 멀수록 y 축 방향으로의 이동량이 많아지고, y 축 방향으로 중심과 멀수록 x 축 방향으로의 이동량이 많아지는 것이다. 변형의 결과는 그림 6에서 보는 바와 같이 정사각형을 기울어진 평행사변형으로 바꾸는 것인데, 실제의 필기에서 x , y 축 각 방향에 대하여 30도이상 기울어지는 경우는 거의 없기 때문에 기울이기 변형의 양도 x , y 축 양 방향에 대하여 각각 -30도에서 30도 사이로 제한한다. 따라

서 변형의 정도를 x 축 방향과 y 축 방향을 구분하여 각각 δ_1, δ_2 ($0.0 \leq \delta_1, \delta_2 \leq 1.0$)로 나타내면, 각 점 S_{ij} 의 변형은 먼저 (식 5)에서와 같이 $Min_x, Min_y, Max_x, Max_y$ 를 구한 후, 다음과 같이 이루어진다.

$$sx = \pm (\delta_1 * 0.5) \tag{10-a}$$

$$sy = \pm (\delta_2 * 0.5) \tag{10-b}$$

$$x = S_{ij,x} - Min_x \tag{10-c}$$

$$y = S_{ij,y} - Min_y \tag{10-d}$$

$$S'_{ij,x} = x + y * sx \tag{10-e}$$

$$S'_{ij,y} = y + x * sy \tag{10-f}$$

이때, sx, sy 는 각각 x 축, y 축 방향으로의 기울기 를 나타낸다.

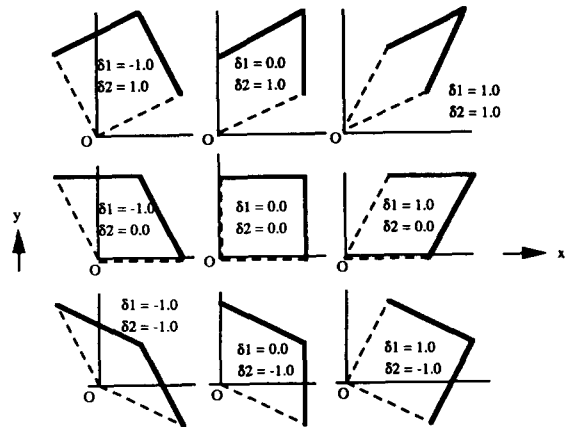


그림 6. □에 대한 기울이기 변환

Fig. 6. Skew distortion.

5. 역사다리꼴 변형 (Trapezoidal Distortion)

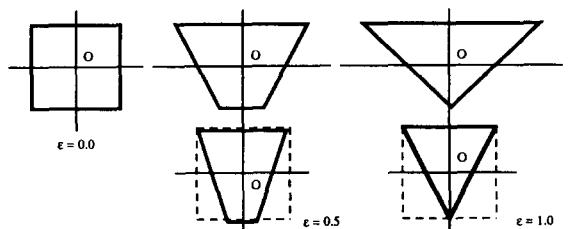


그림 7. □에 대한 역사다리꼴 변환

Fig. 7. Trapezoidal distortion.

역사다리꼴 변형은 실제 한글 필기에서 자소의 윗 부분이 아랫부분보다 더 커지는 점을 반영하는 변형

이다. 실제로 α 값을 쓸 경우를 상상한다면, 이러한 변형이 얼마나 현실과 부합되는지를 알 수 있다. 그림 7에서 보듯이 변형의 결과가 역사다리꼴 형태를 유지하려면, 변형량 d 는 0에서 자소크기의 반사이의 값을 가져야 한다. 변형의 중심 (O_x , O_y)는 (식 5)에서와 같이 구한다. 그러므로, 변형의 정도를 ϵ ($0.0 \leq \epsilon \leq 1.0$) 으로 나타내면, 다음과 같이 변형이 이루어진다.

$$d = \epsilon * (\text{Max}_x - \text{Min}_x) / 2 \quad (11-a)$$

$$S_{i'}^x = (S_{i'}^x - O_x) * (1 + 4 * d * (S_{i'}^y - O_y) / (\text{Max}_x - \text{Min}_x)^2) + O_x \quad (11-b)$$

$$S_{i'}^y = S_{i'}^y \quad (11-c)$$

6. 자소의 상대적 위치와 크기 변형

전체문자의 위상 변화에 미치는 영향에 비하여 이 변형은 너무나 간단하게 이루어진다. x , y 각 방향으로의 위치 변형의 정도를 Φ_1 , Φ_2 , 크기 변형의 정도를 γ_1 , γ_2 라 하자. 본 논문에서는 위치 변형과 크기 변형 각각의 한계를 해당 방향 자소 크기의 10%로 제한하였다. 변형은 먼저 (식 5)에서와 같이 Min_x , Min_y , Max_x , Max_y , O_x , O_y 를 구한 후, 다음과 같이 하면 된다.

$$dx = 0.1 * \Phi_1 * (\text{Max}_x - \text{Min}_x) * \text{random}(0, 1) \quad (12-a)$$

$$dy = 0.1 * \Phi_2 * (\text{Max}_y - \text{Min}_y) * \text{random}(0, 1) \quad (12-b)$$

$$sx = 0.1 * \gamma_1 * (\text{Max}_x - \text{Min}_x) * \text{random}(0, 1) \quad (12-c)$$

$$sy = 0.1 * \gamma_2 * (\text{Max}_y - \text{Min}_y) * \text{random}(0, 1) \quad (12-d)$$

$$S_{i'}^x = (S_{i'}^x - O_x) * sx + O_x + dx \quad (12-e)$$

$$S_{i'}^y = (S_{i'}^y - O_y) * sy + O_y + dy \quad (12-f)$$

그림 8은 각 자소의 상대적 위치 및 크기 변화가 문자 전체의 모양에 미치는 영향을 다양한 예를 통하여 보여준다.

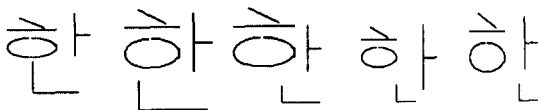


그림 8. 자소의 상대적 위치 및 크기 변형

Fig. 8. Distortion for the relative position and size of each Jaso.

7. 변형문자의 난이도

앞 절에서 언급한 각 변형기법에서 사용한 변형 정도들의 합을 변형율, DR (Distortion Rate) 로 나타내면, DR은 다음과 같은 식이 된다.

$$DR = \alpha + \max(\beta, \gamma) + \delta_1 + \delta_2 + \epsilon + \theta_1 + \theta_2 + \gamma_1 + \gamma_2 \quad (13)$$

이 때, 전체 변형율에서 각 변형 정도의 비는 난수를 통하여 선정한다. 변형율이 클수록 변형이 심해져서 문자패턴의 왜곡이 커지므로 변형율은 해당 문자패턴의 난이도로 볼 수 있다. 따라서 본 논문에서는 생성된 문자패턴의 난이도를 변형율로 나타낸다.

변형에 의해 생성된 필기체 한글 문자패턴의 예가 부록 1에 나와 있으며, 이 때 초성을 변형하기 위해 사용된 각 변형 요인들의 변형 정도는 부록 2에 나와 있다.

IV. 실험

이제까지 보고된 인식 알고리즘들 가운데 몇 가지를 가상적인 한글 필기체 인식 시스템으로 두고 이들을 비교평가하였다. 특징 추출방법으로는 단순하게 32 x 32 격자 (Mesh) 를 사용하였다. 인식 대상이 되는 생성된 문자 이미지가 128 x 128 의 크기이므로, 각 문자 패턴마다 도합 16차원의 입력특징이 만들어진다. 인식방법으로는 신경회로망의 하나인 APC(Adaptive Pattern Classifier)^[5], 최근사이웃 방법(NNC : Nearest Neighbor Classifier)^[6], K-최근사이웃 방법(KNNC)^[6], 최소거리 방법(MDC : Minimum Distance Classifier)^[7]의 네 가지를 고려하였다.

인식대상은 한글 자기 순위 500자로 하였다. 문자 인식 시스템들의 학습 패턴으로는 전체적인 변형율을 10단계로 나누어 각 변형 단계마다 10개씩의 패턴을 생성하고, 여기에 표준 문자패턴을 첨가하여 각 문자당 101 개의 패턴을 사용하였다.

MDC에서는 MaxiMin 알고리즘^[7] 을 이용하여 각 문자마다 일정량의 표본 패턴 (prototype)을 생성한 후 최소거리 인식 알고리즘^[7] 을 적용하였는데, 최종적으로 6041 개의 표본을 사용하였다. NNC와 KNNC에서는 각 문자당 학습패턴 101개 모두를 표본 패턴으로 하여 총 50500 개의 표본을 사용하였다. KNNC에서는 K를 3으로 두고, 세 개의 최근사이웃 가운데 가장 많이 들어있는 것을 해당 입력 패턴의 부류로 보도록 하였다. APC에서는 APC 학습 알고

리듬¹⁵⁾에 따라 필요한 만큼의 중간층 뉴런을 스스로 생성하도록 하였는데, 최종적으로 23406 개의 중간층 뉴런을 생성하였다. 참고로 APC에서는 하나의 중간층 뉴런이 하나의 표본 패턴에 해당한다.

1. 전체적인 인식특성의 평가

평가용 패턴은 각 변형율마다 2 세트씩 생성하여 각 인식 시스템에 적용하였다. 종래의 방법을 이용할 경우 20개의 평가용 데이터 세트에 대하여 <표 1>에 나타낸 것과 같이 평균적인 인식률밖에 얻을 수 없지만, 본 논문의 방법을 이용할 경우, 그림 9에 나타낸 바와 같은 그래프를 인식 결과로 얻을 수 있다. 즉, 평가용 데이터의 난이도에 따른 인식율의 변화 양상을 알아낼 수 있다. 그래프를 분석해보면, 변형율이 가장 작은 학습한 데이터에 대한 인식율의 경우 APC나 NNC, MDC가 KNNC보다 높은 인식율을 가지는 반면, 변형율이 커짐에 따라 MDC와 APC의 성능 저하가 KNNC보다 뚜렷하게 나타남을 알 수 있다. 동일한 학습패턴을 사용하였다는 것을 고려할 때, NNC나 KNNC가 APC나 MDC보다 더 좋은 인식 특성을 가지고 있음을 알 수 있다. 물론, 이는 NNC나 KNNC가 APC나 MDC보다 훨씬 더 많은 표본을 가지고 있기 때문이라 볼 수도 있으며, 메모리 사용량을 고려하였을 경우에는 MDC가 오히려 더 효과적으로 패턴공간을 분할하고 있다고 볼 수도 있다.

표 1. 평균 인식율

Table. 1. Average correct recognition rate.

인식 시스템	평균 인식율
APC	57.4%
NNC	72.8%
KNNC(K=3)	69.8%
MDC	54.9%

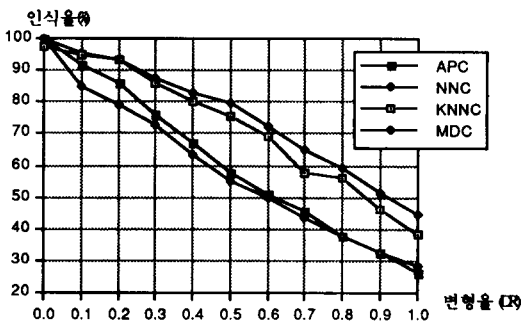
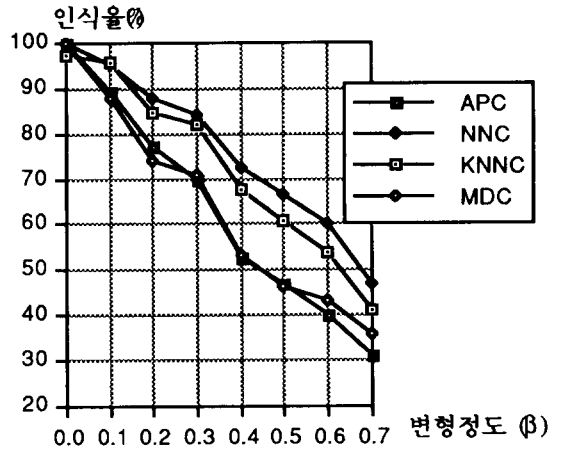


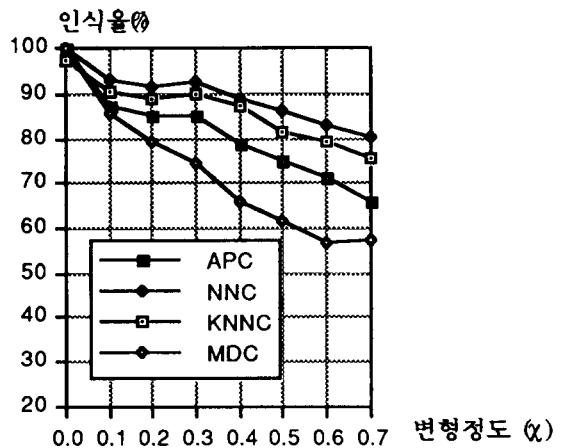
그림 9. 변형율에 따른 인식율의 변화
Fig. 9. Recognition rate vs. distortion rate.

2. 특정 변형에 대한 인식 특성의 평가

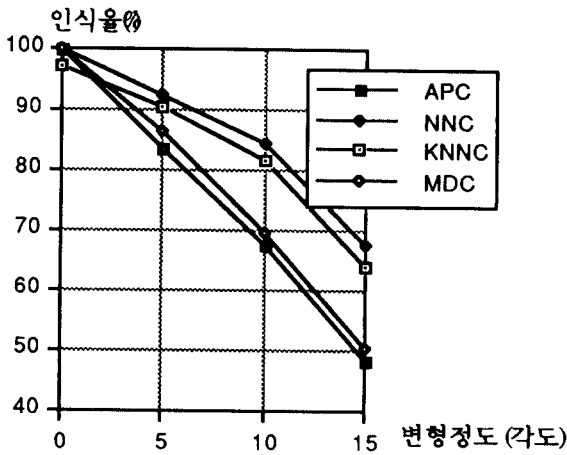
각각의 변형 요인에 대한 인식 특성을 평가하기 위하여 각 변형 요인별로 여러 단계의 변형 정도를 가진 평가용 패턴을 생성하였다. 각 변형 요인에 대한 변형 정도에 따른 인식률의 변화가 그림 10과 같이 나타나 있다. 이 그래프에서, MDC의 경우 오목화 변형에 대하여 다른 인식 시스템들에 비하여 지나치게 민감하여 난이도에 따른 인식율 저하가 급격하게 나타남을 알 수 있다. 반대로 회전이나 위로 기울이기 변형에 대해서는 APC보다도 더 좋은 성능을 보여주고 있음을 알 수 있다. 이와 같이 그래프들을 분석함으로써 각 인식 시스템이 어떠한 변형에 강하고 약한지를 분명히 파악할 수 있다. 이는 종래와 같이 평균인식률만으로 평가를 행할 경우 도저히 알아낼 수 없는 정보이다.



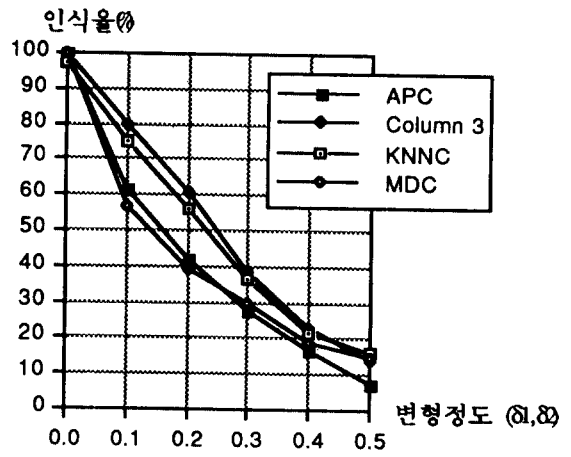
(a) 블록화 변형에 대한 인식율 변화



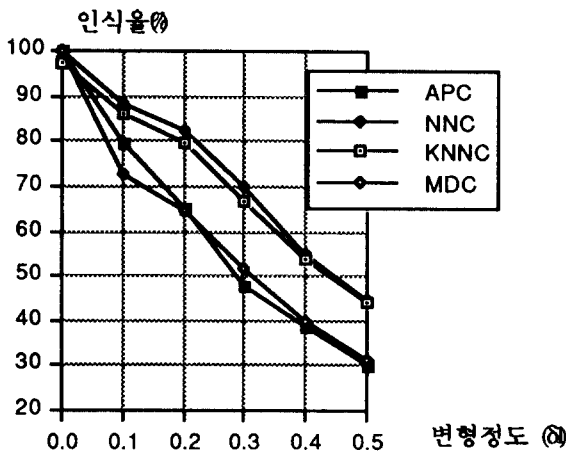
(b) 오목화 변형에 대한 인식율 변화



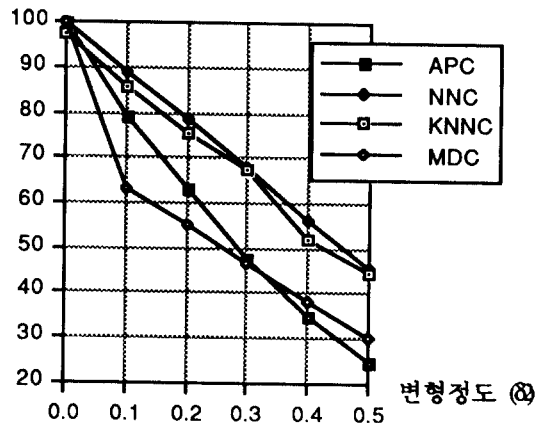
(c) 회전 변환에 대한 인식율 변화



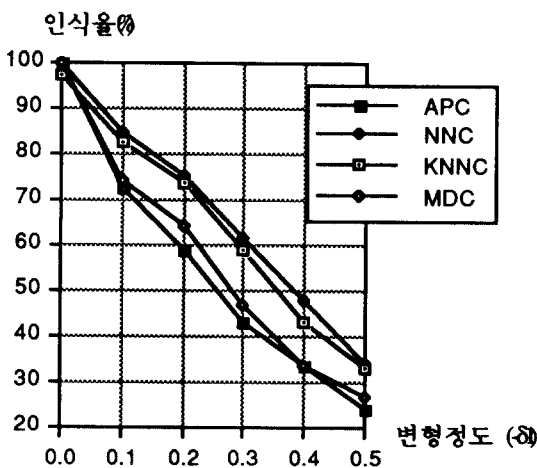
(f) 우측 위로 기울이기 변환에 대한 인식율 변화



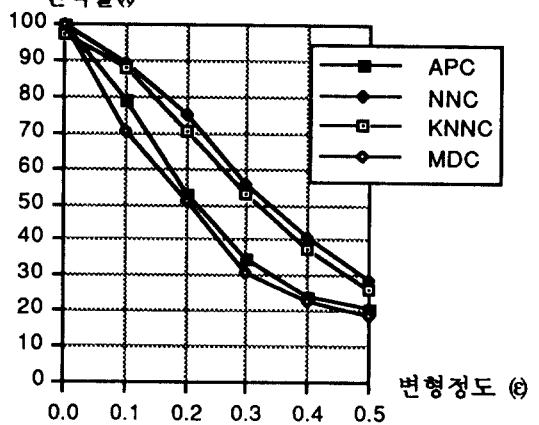
(d) 우측 기울이기 변환에 대한 인식율 변화



(g) 위로 기울이기 변환에 대한 인식율 변화



(e) 좌측 기울이기 변환에 대한 인식율 변화



(h) 역사다리꼴 변환에 대한 인식율 변화

그림 10. 특정 변형의 변형량에 따른 인식율의 변화
Fig. 10. Recognition characteristics for particular distortion.

V. 결 론

본 논문에서는 변형에 의해 필기체 한글 문자를 인위적으로 생성하는 방법을 개발하였다. 또 인위적으로 생성된 문자를 평가용 데이터로 사용함으로써 문자인식 시스템의 성능을 보다 객관적이고 정밀하게 비교 분석할 수 있었다. 즉, 이 방법에서는 사용되는 평가용 데이터의 품질을 인위적으로 생성된 문자 각각에 대하여 정량적으로 부여함으로써 기존에 보편적으로 사용되던 평가 척도인 평균인식률 이외에 문자 품질에 따른 해당 문자인식 시스템의 전체적인 인식 특성이나 특정 변형에 대한 인식 특성 등도 함께 알아낼 수 있었다. 본 방법의 효용성을 보이기 위하여 기존에 제안된 인식 알고리즘을 바탕으로 몇 가지 한글 필기체 문자인식 시스템을 가상적으로 구성하고 각 인식 시스템의 전체적인 인식 특성과 특정 변형에 대한 인식 특성을 알아내는 실험을 행하였다.

현재의 문제점과 앞으로의 연구방향을 정리하면, 먼저 한글 필기의 특성을 좀더 분석하여 보다 완전한 한글 필기체를 생성할 수 있도록 변형기법에 관한 연구가 보완되어야 한다. 특히 흘림에 의한 획간의 연결을 나타낼 수 있는 변형 기법의 연구가 필요하며 이 경우에는 필순의 효과도 고려의 대상이 되어야 한다. 또한 지금까지는 모든 획의 굵기를 1로 통일하였지만 획의 굵기에 대한 변형도 고려해야 한다.

다음으로 개인의 필기 성향을 자동적으로 파악함으로써 파라미터화시키는 방법에 대한 연구가 필요하다. 이것은 필적 감정 등 다양한 응용 상품 개발이 가능할 뿐만 아니라 필기 문자의 내부적인 특성을 알아내는 데 커다란 기여를 할 수 있으므로 매우 유익한 연구가 될 것이다.

마지막으로 실험에 사용된 인식 알고리즘들 가운데 최근 많이 연구되고 있는 신경회로망 BP(Back-propagation) 방식을 메모리 사정으로 포함시키지

못한 점과 인식 알고리즘 뿐만아니라 입력으로 사용되는 여러 가지 특징(feature)의 효과를 함께 알아내는 실험을 추가하지 못한 점이 아쉽다.

參 考 文 獻

- [1] K. Ishii, "변형문자패턴 생성법과 그 응용," *일본전자정보통신학회 논문지*, vol. J66-D, no.11, pp.1270-1277, Nov. 1983.
- [2] M. Shiono, "A method for generation of handwritten style character patterns using nonlinear geometrical distortion," *일본전자정보통신학회 논문지*, vol. J74-D-II, no.2, pp.209-219, Feb. 1991.
- [3] S. Kurakake, K. Ishii, "Evaluation of Handprinted Character Recognition Systems Using Distorted Characters," *일본전자정보통신학회 논문지*, vol. J74-D-II, no.9, pp.1208-1216, Sep. 1991.
- [4] Y. Uehara, T. Agui, K. Tahara, and M. Nakajima, "Conversion from Kanji ROM Data to Vector Data and Its Application," *일본전자정보통신학회 논문지*, vol. J70-D, no.7, pp.1407-1412, Jul. 1987.
- [5] 박 상태, 방 승양, "패턴 분류용 신경회로망 APC-II," *한국정보과학회 춘계학술발표논문집*, 제 19권 제 1호, pp.685-688, 1992.
- [6] T. M. Cover, P.E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. on Information Theory*, vol. IT-13, no. 1, pp.21-27, Jan. 1967.
- [7] J. T. Tou, R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, Massachusetts, 1974.

부 록 1. 생성된 한글 문자 예

Appendix 10. Example characters which are generated by distortion algorithms.

- (1) 이의다는에하을고한가
- (2) 이의다는에하을고한가
- (3) 이의다는에하을고한가
- (4) 이의다는에하을고한가
- (5) 이의다는에하을고한가
- (6) 이의다는에하을고한가

부 록 2. 1의 초성들에 대한 각 변형 요인의 변형량

Appendix 10. Used parameters to generate the first Jasos of the characters in Appendix 1.

번호	α	β	χ	δ_1	δ_2	ϵ	ϕ_1	ϕ_2	γ_1	γ_2	DR
(1)	0	0	0	0	0	0	0	0	1	1	0.0
(2)	0.033333	0.001042	0	0.039063	0.017708	0.019271	0	0	0.982000	0.988800	0.2
(3)	0.033333	0.200000	0	-0.051351	0.001351	0.028378	4	0	0.974400	1.036800	0.4
(4)	0.033333	0.058647	0	0.164662	0.027068	0.049624	3	5	0.992800	1.004800	0.6
(5)	0.200000	0	0.242837	-0.076190	-0.076190	0.011905	-1	-3	1.076800	0.923200	0.8
(6)	0.100000	0.343891	0	0.033937	0.119910	0.110860	-1	-1	0.976000	1.088000	1.0

著 者 紹 介



朴 相 泰 (準會員)
 1968年 12月 23日生. 1991年 포항
 공과대학 전자계산학과 학사. 1993
 年 포항공과대학 전자계산학과 석
 사. 현재 포항공과대학 정보통신연
 구소 연구원. 주관심분야는 패턴인
 식, 신경회로망, HCI 등임.

方 勝 楊 (正會員) 第 29卷 B編 第 2號 參照
 현재 포항공과대학 전자계산학과
 교수.