[5-8]

On-Line 문자 인식에서의 Preprocessing 알고리듬

○
구 본석 김 성훈 김 재희
(연세대학교 전자공학과)

Preprocessing Algorithms for the On-Line Character Recognition

Bon seuk Goo Seong Hoon Kim Jaihie Kim
(Dept. of Electronic Eng. Yonsei University)

Abstract

On-Line input in interactive graphics interface is the interesting field of the new man-machine interface technology. This paper presents a preprocessing module for the on-line character recognition system using the directional features of the character strokes. The HANGUL characters written within the rectangle whose size is 64 x 64 pixels and the English characters, digits, and symbols written within the 48 x 48 rectangle are tested. The results show that the distance of 10 pixels and the $\pi/5$ radian angle are the appropriate parameters for the filtering.

I. 서론

컴퓨터에 데이타를 입력하는 보다 자연스럽고 유용한 방법에 대한 연구가 진행되고 있다. 인간에게는 펜과 종이가 키보드보다 훨씬 친숙한데, 이러한 특징을 살린 입력 방법에는 스타일러스(Stylus), 타블렛(Tablet), LCD판 등을 사용하는 온라인(On-Line) 입력 방식이 있다. 온라인 문자 인식은 이러한 입력장치에 쓰여지고 있는 문자나 부호 등을 인식하는 것을 의미한다[1].

이와 같은 입력 장치를 이용하여 문자, 부호를 인식하는 알고리듬은 컴퓨터 입력 수단의 하나로서 필기(Handwriting)가 가능하도록 만들며, 워드 프로세싱 작업을 필기 방식으로 대체할 수 있게 한다. 본 연구는 워드 프로세싱 작업의 필기 대체를 위한 연구의 일환으로 진행하였다. 즉, 궁극적으로는 필기된 문자, 부호를 인식하는 워드 프로세서를 만들기 위하여 문자 인식기를 개발하고 이를 워드 프로세서에 접목하려고 한다.

문자 인식기는 전처리(Preprocessing) 모듈을 포함한다. 이 전처리 모듈은 인식률을 높이고 인식 단계에서의 데이타 처리량을 줄이기 위하여, 입력 장치에서 들어온 데이타로부터 잡음을 제거하고 인식에 필요한 데이타로 변환하여 인식 모듈에 보내는 역할을 한다. 이 모듈에서의 처리는 인식 단계에서 사용하는 알고리듬과 상관관계가 있으며 전체 시스템의 인식률과 처리 속도에 영향을 미친다[2].

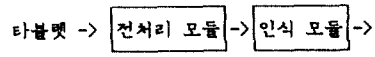


그림 1. 온라인 문자 인식기 블럭 다이어그램

이 논문에서는 스트로크(Stroke)의 방향 성분을 이용하여 스트로크를 분류 인식하는 문자 인식 시스템에서 사용하기 위한 전처리 알고리듬을 제안하고자 한다.

스트로크 간의 상대적인 크기 차이나 문자 전체 크기에 거의 영향을 받지 않으면서 스트로크 형태에 대한 정보를 유지할 수 있는 방향 성분을 사용하여 스트로크를 분류할 수 있다. 전처리를 거친 후 추출된 특징점들은 스트로크의 방향 성분을 결정하는 중요한 점들이다. 즉, 특징점은 스트로크의 방향 성분을 표시할 수 있는 점들로, 그림 2와 같이 획을 포함하지 않는 스트로크의 처음과 마지막, 그리고 꺾이는 부분의 점이다. 이에 대한 자세한 내용은 같은 논문집에 실린 전 병환, 박 충식, 그리고 김 재희의 "On-Line 문자 인식을 위한 Stroke 분류에 관한 연구"를 참고하기 바란다.

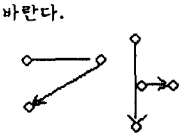


그림 2. 특징점('가'자의 경우)

II. 전처리 알고리듬

타블렛으로부터 들어온 원 데이타는 그대로 인식하기에는 문제점이 많다. 필기 운동의 부정확성과 타블렛의 양자화에 따른 잡음을 포함하게 된다. 입력된 문자의 크기와 필기 속도는 항상 일정한 것은 아니다. 필기를 느리게 한 부분에는 중복분(Redundancy)의 데이타가 생긴다. 펜이 타블렛 면에 닿고 떨어지는 순간에는 문자에 훅(Hook)이 더해지는 경우가 있다[3]. 인식에 앞서 문자나 단어와 같은 필기 단위로 나누어야 한다. 그러므로, 전처리는 외부 분할(External segment), 잡음제거, 스무딩(Smoothing), 필터링(Filtering), 크기 정규화(Size normalization), 디후킹(Dehooking) 등을 포함하여야 한다[1][5].

연속적인 두 점 사이의 거리는 필기 속도에 의하여 제한이 있다. 즉, 사람의 근육이 낼 수 있는 필기 속도에는 제한이 있으므로 한계값(Threshold value) 이상의 거리를 갖는 점은 잡음으로 간주할 수 있다. Tappert는 이러한 원리를 이용하여 잡음을 제거하였다[2].

Tappert는 현재점을 포함한 다섯점에 가중치를 적용한 가중평균값으로 현재점을 대체시킴으로써 스무딩을 행하였다. 쉽게는 앞의 점과의 평균점으로 대체시킬 수도 있다. Ito와 Chui는 Tolerance region내에 포함하는 점을 제거하고 벗어나는 점을 취하는 Piecewise-linear curve fitting 방법을 사용하였다[4]. Mandler는 클러스터링(Clustering) 알고리듬을 사용하여, 적당한 제한 거리이내에 들어오는 점들을 이들의 평균값으로 대체시켰다[6]. 이와 같은 방법들은 스무딩을 따로 처리할 필요가 없게 해 준다.

실험 결과 잡음은 거의 생기지 않아 잡음 제거는 필요없었다. 특징점의 방향 성분을 이용하는 본 인식 시스템의 전처리는 매칭 방법을 이용하는 인식 시스템의 전처리와는 달리 크기 정규화가 필요없다. 외부 분할의 문제는 미리 정해진 정방형 사각형을 화면에 출력시키고 필기자가 주의를 기울여 그 안에 쓰게함으로써 해결하였다.

1. 스트로크

입력 데이타는 펜의 위치를 나타내는 일련의 x,y 좌표열인데, 이 데이타는 타블렛 디지타이저로부터 연속적으로 입력된다. 이때 스트로크는 펜이 타블렛 면에 눌려진 때부터 떨어질 때까지 펜의 이동경로를 따라 샘플링되어 입력된 좌표열로 정의한다. 한 문자는 하나 또는 여러개의 스트로크로 구성될 수 있다.

2. 거리 필터링(Distance Filtering)

특정하게 정한 값 Dmin보다 작은 거리 내에 있는 점은 중복되거나 인식시 불필요한 근접점이므로 제거한다. 이 Dmin 값은 모서리 부분의 특징점을 제거하여 문자를 왜곡시키지 않는 범위에서 가능한한 직선 성분의 불필요한 점과 중복점을 많이 제거하도록 정한다. 이 경우 스트로크의 방향이 급격히 바뀌는 부분의 특징점를 잃게 되는 경우가 많이 발생하는데, 이를 막기 위하여 필기 방향이 바뀌는 부분에는 여러 점이 몰리는 성질을 이용하여 누적 좌표점 갯수가 많은 경우 거리에 상관없이 이를 그대로 취하도록 한다. 누적 좌표점 갯수는 제한값 Cmax와 비교한다.

3. 각 필터링(Angular filtering)

방향이 크게 바뀌지 않는 성분은 인식시 필요하지 않으므로 제거한다. 그림 3과 같이 세 점이 이루는 각 θ을 구하여 이를 θmin값과 비교한다. 이때 비교하는 각 θmin이 너무 커서 곡선을 이루는 부분의 데이타가 많이 제거되어 인식이 불가능하게 해서는 안 된다.

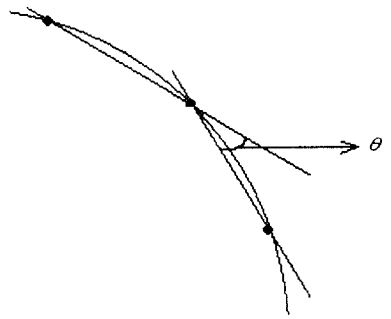


그림 3. 세 점이 이루는 각

4. 디후킹(Dehooking)

훅은 스트로크의 처음과 끝 부분에서 발생하는데, 이는 인식에 장애가 되므로 반드시 제거되어야 한다. 대개의 경우 이러한 훅 성분은 전체 스트로크 길이에 비하여 상당히 작은 길이로 스트로크의 처음과 마지막 부분에 나타난다. 그러므로 스트로크의 처음과 끝 부분에서 제한값 Ddeh보다 거리가 짧은 성분을 제거하면 된다. 여기서 Ddeh는 스트로크 길이에 비하여 상당히 짧은 거리가 되도록 정한다.

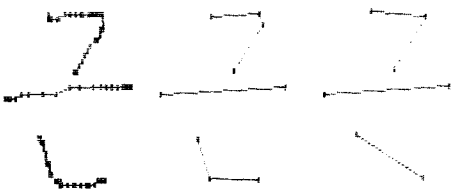
이상의 세 모듈로 구성된 전처리 모듈의 주요 파라미터는 Dmin, Cmax, θmin, Ddeh이다.

III. 실험 및 고찰

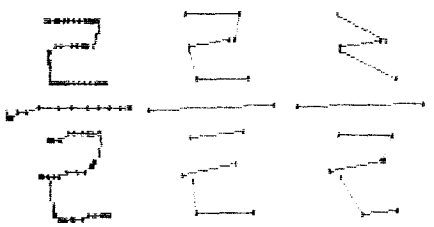
본 실험에서 사용한 처리 기종은 IBM PC/AT 호환기종이며, 알고리듬은 C-language를 사용하여 구현하였다. 타블렛 디지타이저(resolution : 510 points/inch)로부터 샘플링 속도(Sampling Rate) 240 samples/sec로 입력 받은 데이타를 화일로 보관한다. 이때 필기를 하는 사람에게 한글의 경우 64 x 64 pixels, 영문자, 숫자, 부호의 경우 48 x 48 pixcels 크기의 정방형 사각형안에 흘림체가 아닌 정체로 쓸 것을 요구하였다. 이 화일을 위의 알고리듬으로 구성한 전처리 모듈에 입력하고 처리된 결과를 화면에 출력시켰다.

출력 결과를 살펴보면 다음과 같다. 그림 4는 Dmin 값이 너무 큰 경우 왜곡 현상이 생김을 보여주고 있다. 그러나 Dmin 값이 작게 할수록 데이타 처리량이 늘고 처리 속도가 증가하게 된다. 그림 5는 점이 몰리는 부분의 특징점 추

출을 무시하면 스트로크의 방향이 바뀌는 부분에서 왜곡 현상이 생김을 보여준다. 이는 누적좌표점 갯수를 살펴서 꺾이는 부분의 특징점을 추출하여야 함을 보여주는 것이다. 그림 6은 θmin값을 크게하면 곡성 성분에서 왜곡 현상이 생김을 보여 주고 있다. 즉, 이들 파라미터 값을 무한정 늘려 처리량을 줄일 수는 없고, 글자 모양을 심하게 왜곡시키지 않는 범위로 제한하여야 한다.



(a)원 데이타 (b)처리후 결과 (c)처리후 결과
(b)Dmin=10pixels, Cmax=5, θmin=π/3rad, Ddeh=3pixcels
(c)Dmin=15pixels, Cmax=∞, θmin=π/3rad, Ddeh=3pixcels
그림 4. Dmin값에 따른 결과의 차이

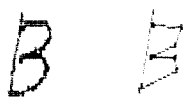


(a)원 데이타 (b)처리후 결과 (c)처리후 결과
(b)Dmin=10pixels, Cmax=5, θmin=π/3rad, Ddeh=3pixels
(c)Dmin=10pixels, Cmax=∞, θmin=π/3rad, Ddeh=3pixels
그림 5. Cmax값에 따른 결과의 차이

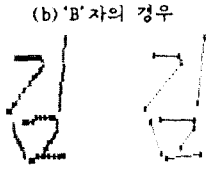
(a)원 데이타 (b)처리후 결과 (c)처리후 결과
(b)Dmin=10pixels, Cmax=5, θmin=π/5rad, Ddeh=3pixcels
(c)Dmin=10pixels, Cmax=5, θmin=π/3rad, Ddeh=3pixcels
그림 6. θmin값에 따른 결과의 차이

외부 분할은 필기자의 도움을 받기 위하여 미리 정의한 크기의 사각형안에 쓰게 하는 방법이 시스템 처리의 단순화를 위하여 사용될 전망인데, 이 실험에 의하면 그림 7에서 보인 바와 같이 각 파라미터는 64 x 64 pixels 크기의 경우 Dmin = 10 pixels, Cmax = 5, θmin = π/5 radian, Ddeh = 3 pixels이 적당하다. 이러한 파라미터를 사용한 결과, 입력된 좌표점이 각 단계에서 전체 데이타의 약 73%, 10 %, 1%가 감소되었고, 전체적으로 약 84%가 감소되었다. 이는 필기 속도에 크게 좌우된다.

(a)'A'자의 경우



(b)'B'자의 경우



(c)'김'자의 경우

그림 7. 처리된 결과의 예

VI. 결론

이 논문에서 제안한 온라인 문자 인식 시스템을 위한 전처리 모듈은 원 데이타에서 중요한 특징점만을 추출하는 데 적합하고, 인식 모듈에서의 처리 부담을 줄일 수 있었다. 전처리 후의 데이타는 원 데이타의 중요한 특징점만을 포함하는 극히 적은 양으로 감소되었고, 이러한 특징점을 연결하여 얻은 경로는 실제 필기 운동시의 펜 경로를 크게 왜곡시키지 않았다.

이 논문의 실험 결과는 필기 방향을 이용하여 스트로크를 인식하는 문자 인식 시스템에 응용될 수 있을 것이다. 그러나 인식 시스템의 인식 대상 차이에 따라 파라미터 값을 약간씩 조정할 필요가 있고, 처리 속도의 개선을 위해 원 패스 알고리듬으로 변형시켜야 할 것이다.

이후에는 다른 시스템의 전처리 모듈과의 차이를 분석하고, 실제적인 응용을 위해서 여러가지 프로토타입을 구성하여 처리 속도, 처리량 감소율, 전체 시스템의 인식률 등에 미치는 영향에 대한 비교 분석할 예정이다.

V. 참고 문헌

[1] Charles C.Tappert, Ching Y.Suen, Toru Wakahara, "The State of the Art in On-Line Handwriting Recognition," IEEE Tran. Pattern Analysis and Machine Intelligence, VOL. 12, NO. 8, August 1990.

[2] C.C.Tappert, "Speed, accuracy, flexibility trade-offs in on-line character recognition," IBM research Report RC13228,October 1987.

[3] J.R.Ward and M.J.Phillips, "Digitizer Technology : performance characters and the effects on the user interface," IEEE Trans. Computer Graphics and Appl., pp. 31-44, April 1987.

[4] M.R.Ito and T.L.Chui, "On-line computer recognition of proposed standard ANSI handprinted characters," Pattern Recognition, VOL. 10, pp. 341-349, 1987.

[5] K.Okada, H.Arakawa and I.Masuda, "On-line recognition of handwritten characters by approximating each stroke with several points," Proc. IEEE Trans. System. Man and Cybernetics, VOL. 12, pp. 893-903, 1982.

[6] Eberhard Mandler, "Advanced Preprocessing Technique for On-Line recognition of Handprinted Symbols," Computer Recognition and Human Production of Handwriting, pp. 19-36, 1989.