

# 펜 기반 모바일 컴퓨터에서 제스처 허용 문서 편집기 개발

조미경\*, 조 환규\*\*, 오암석\*

\*동명정보대학교 정보공학부, \*\*부산대학교 정보컴퓨터공학부  
e-mail : mgcho@tit.ac.kr, hgcho@pusan.ac.kr, asoh@tit.ac.kr

## Development of text editor available for gesture on pen-based mobile computers

Mi-Gyung Cho\*, Hwan-Gue Cho\*\*, Am-Seok Oh\*

\*School of Information Engineering, Tongmyong University of  
Information Technology, \*\*School of Computer Science and  
Engineering, Pusan National University

### 요 약

PDA와 같은 휴대용 컴퓨터의 발전은 펜을 이용한 데이터 입력과 처리 기술에 대한 요구를 증가시키고 있다. 하지만 대부분의 PDA용 애플리케이션들은 키보드와 마우스를 주된 입력 도구로 사용하는 데스크 탑 컴퓨터의 GUI와 입력 방식을 그대로 사용하고 있다. 본 논문에서는 펜 기반의 휴대용 컴퓨터에서 필기체 데이터 문서를 편집할 수 있도록 해주는 펜 제스처(pen gesture) 기능 허용 텍스트 에디터를 개발하였다. 삽입, 삭제, 앞뒤 낱말 바꾸기, 줄 바꾸기 등의 제스처 인식 방법과 제스처 처리를 위해 스트로크(strokes)를 GCs(Gesture Components) 단위로 그룹화하기 위한 방법들을 제안하였다. 제스처 허용 에디터의 개발은 펜 기능만으로 문서 편집을 위한 모든 작업을 가능하게 해 줌으로 사용자의 편의를 극대화시켜 준다.

### 1. 서론

PDA와 같은 휴대용 컴퓨터의 발전과 보급으로 펜 컴퓨팅(pen computing)이 사용자들에게 더욱 친숙해지고 있다. 그러나 펜 기반 모바일 컴퓨터에서 사용되는 대부분의 애플리케이션들은 아직도 데스크 탑에서 마우스와 키보드를 이용한 입력 방식과 GUI를 그대로 사용하고 있다. 펜을 주된 입력 도구로 사용하는 시스템에서 데이터 입력 방식은 크게 세 가지로 분류할 수 있다. 첫째는 전자잉크 데이터(electronic ink data) 방식[1,2]과 둘째는 소프트 키보드(soft keyboards)를 이용한 입력 방식, 마지막으로 문자 인식 시스템을 사용하는 것이다.

소프트 키보드 방식은 크기 제한이 있는 모바일 컴퓨터에서 보편적으로 사용되는 방식으로 펜으로 소프트 키보드를 태핑(tapping)함으로써 데이터를 입력한다. 이 방식은 데스크 탑 컴퓨터에서의 키보드 입력 방식을 펜을 이용하여 하도록 고안된 것으로 펜을 사용한 입력 방식으로는 자연스럽지 못하다.

문자 인식 시스템은 현재 대부분의 모바일 애플리케이션에서 문자 입력 수단으로 사용하는 방식으로 펜으로 입력한 스트로크(strokes)를 특정 문자로 인식하여 아스키 코드 값으로 변환하는 방식이다. 이 방식의 장점으로 일단 아스키 코드 값으로 변환하면 데이터 검색과 저장이 전통적인 데스크 탑 컴퓨터에서처럼 효율적으로 처리될 수 있다는 것이다. 하지만 데이터 입력을 문자 인식 시스템에서 제공하는 제한된 영역 내에서만 해야 하고 문자 인식 오류가 발생하면 잘못 인식된 문자를 지우고 다시 데이터를 입력해야 되는 단점이 있다.

전자잉크 데이터 방식은 펜으로 입력한 데이터를 그대로 저장하고 처리하는 방식으로 사용자에게 입력할 수 있는 데이터 형태에 제한을 주지 않는다. 사용자는 문자에서부터 수식, 그림에 이르기까지 펜으로 입력할 수 있는 모든 형태의 데이터를 입력할 수 있다. 이러한 방식은 펜의 특성을 최대한 살리고 사용자에게도 자연스러운 데이터 입력 방식이라고

할 수 있다[1,2]. 전자 잉크 데이터가 사용자에게 더 풍부하고 자연스러운 정보 표현을 제공하는 장점을 가진 입력 매체이지만 이것을 사용하기 위해서는 첫째 전자 잉크 데이터의 검색이 가능해야 하고 둘째 전자 잉크 데이터의 편집이 가능해야 하고 셋째 다양한 플랫폼에서 전자 잉크 데이터를 사용하기 위해 JOT[3]등과 같은 표준 교환 포맷이 개발되어야 한다.

본 논문에서는 전자 잉크 데이터 중심의 데이터 처리를 위한 방안으로 한글 필기체 데이터를 편집하기 위한 텍스트 에디터를 개발하였다. 개발된 에디터는 펜의 특성을 최대한 살리기 위해 펜 제스처 (pen gesture) 기능을 허용한다. 데이터 입력을 수행하는 도중에도 언제든지 정의된 제스처를 입력하면 바로 제스처 기능이 수행되는 직접(direct) 제스처 방식을 제공하는 장점을 지니고 있다. 에디터 개발을 위해 몇가지 제스처를 인식하기 위한 방법과 제스처 기능을 수행하기 위해 스트로크 단위의 필기 데이터를 문자 단위로 그룹화하는 방법을 제안하였다.

논문의 구성은 다음과 같다. 2장에서는 인식 가능한 제스처의 종류와 인식 방법에 대해 설명하고 3장에서는 스트로크 단위의 데이터를 문자 단위로 그룹화 하는 방법을 제안할 것이다. 마지막으로 4장에서는 구현 결과를 보이고 결론을 맺는다.

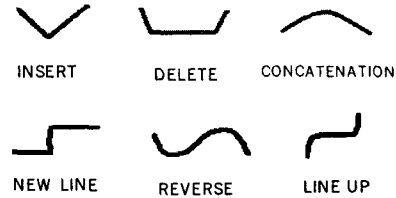
2. 제스처 종류와 인식 방법

펜 제스처는 스타일러스 펜이나 마우스를 이용하여 어떤 기능을 수행하도록 하는 강력한 방법[4,5]으로 문자와는 달리 그래픽 객체(graphic object)로 표현된다. 크기가 작고 펜을 주된 입력 도구로 사용하는 PDAs와 같은 휴대용 컴퓨터에서는 제스처 기능의 허용은 데이터 편집을 위한 강력한 도구가 되며 키보드 입력의 대안이 될 수 있다.

펜 제스처를 사용하기 위해서는 입력하려는 문자 데이터와 구분하여 제스처를 인식해야 되는데 제스처 인식은 필기체 인식과는 다른 문제이다. 왜냐하면 제스처는 제한된 문자가 아닌 기하학적인 모양을 가지고 있기 때문이다. 현재 개발된 몇 개의 제스처 인식 알고리즘[4,5]에서 제스처의 모양은 단일 스트로크(single stroke)으로 표현할 수 있는 형태, 즉 펜 다운(pen down)에서 펜 업(pen up)까지의 연속적인 단일 스트로크를 사용하였다. 이는 제스처를 입력 데이터와 구분하여 쉽게 인식하기 위해서이다.

개발한 텍스트 에디터에서 사용한 제스처의 모양

은 원고지 교정 부호를 참조하여 정의하였으며 구현된 여섯 가지의 제스처들은 그림 1과 같다.



[그림 1] 펜 제스처(pen gesture)의 종류

제스처 INSERT는 글자 사이에 새로운 데이터를 입력하는 기능이며 DELETE는 입력한 데이터를 삭제할 경우, CONCATENATION은 문자사이의 공백을 제거하고 데이터를 연결하는 기능을 수행한다. 줄을 바꾸는 NEW LINE과 두 문자의 순서를 앞뒤로 바꾸어 주는 REVERSE, 두 줄을 한 줄로 연결해 주는 LINE UP 기능이 제공된다.

제스처를 입력 문자와 구분하기 위해 텍스트 에디터는 베이스라인(baseline)을 가진다. 입력 데이터는 베이스라인과 베이스라인 사이에 작성하도록 허용된다. 반면 제스처는 베이스라인에 교차되게 입력하도록 하였다. 따라서 펜으로 입력된 스트로크가 베이스 라인과 만나는 교차점이 생기면 시스템은 입력된 스트로크가 제스처인지 아닌지를 검사하게 된다. 표 1은 여섯 가지 제스처 인식에 사용된 정보들이다.

[표 1] 제스처 인식에 사용된 정보들

제스처 종류 정보들	INSERT	DELETE	CONCATENATION	REVERSE	LINE UP	NEW LINE
시작점(SP)/끝점(EP)/중간점(MP)의 좌표	SP.y EP.y << MP.y	SP.y EP.y << MP.y	SP.y EP.y >> MP.y	SP.y MP.y << EP.y	SP.y >> EP.y	SP.y MP.y >> EP.y
교차점 생성 여부	x	x	x	o	o	o
곡률에 의한 획 분리 개수	2	3	≥1	≥2	3	3
분리된 획방향	down, up	down, right, up	right 의	down 의	up, right, up	right, up, right

표 1에 사용된 정보에서 교차점은 스트로크의 시작점과 끝점을 연결했을 때 생성되는 교차점을 말하며

INSERT, DELETE, CONCATENATION 제스처의 경우 교차점이 생성되지 않지만 NEW LINE, LINE UP, REVERSE 제스처에서는 생성되므로 제스처를 구분하는 특징이 될 수 있다. 곡률은 스트로크를 구성하는 점들(points)의 휨 정도를 나타내는 값으로 곡률이 일정한 임계치 이상의 값을 가지는 위치에서 획을 분리했을 때 분리되는 획의 개수를 제스처 구분의 특징으로 사용하였다. 또한 분리된 획들에 대해 방향 정보도 제스처를 구분하는 정보로 사용하였다. 예를 들어 DELETE 제스처의 경우 세 개의 획으로 분리되는데 각 획들의 주된 방향은 down, right, up의 방향을 가진다. 제스처의 종류를 구분하기 위한 인식 알고리즘은 다음과 같다.

**[알고리즘] 제스처 인식 알고리즘**

입력 : 펜으로 입력된 온라인 스트로크  
출력 : 스트로크의 종류

```

[1] if ( Is there intersections between an end point
      and a start point of an input stroke )
      goto [2]
    else goto [3]
[2] if ( SP.y << EP.y )
      stroke_type = REVERSE; return;
    else {
      if (stroke direction == <RIGHT,UP,RIGHT>)
        stroke_type = NEW_LINE; return;
      else (stroke direction == <UP, RIGHT,UP>)
        stroke_type = LINE_UP; return;
    }
  stroke_type = NO_GESTURE
[3] if ( (SP.y >> MP.y && EP.y >> MP.y) )
      stroke_type = CONCATENATION; return;
    else { if ( separating stroke number == 2 )
            stroke_type = INSERT; return;
          else if (separating stroke number == 3)
            stroke_type = DELETE; return;
          }
  stroke_type = NO_GESTURE
    
```

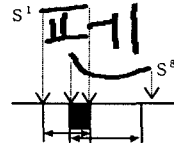
텍스트 에디터에서 제스처 기능을 처리하기 위해서는 먼저 입력된 스트로크들에 대한 그룹화가 이루어져야 한다. 다음 절에서는 입력된 스트로크를 문자 단위로 그룹화 방법에 대해 설명할 것이다.

**3. 스트로크의 문자 단위 그룹화**

스트로크의 그룹화는 입력된 스트로크들을 제스처 기능을 처리하기 위한 데이터 단위로 묶는 것을 의미한다. 제스처 기능은 문자나 단어 단위로 이루어져야 하지만 펜으로 입력한 전자 잉크 데이터에는 문

자나 단어에 대한 구분이 전혀 없다. 따라서 제스처 기능을 수행하기 위해 스트로크를 문자나 단어가 되도록 묶어주는 작업이 필요하다.

한글은 초성, 중성, 종성으로 구성되어 있고 종성이 있는 글자의 경우 획의 x좌표의 값을 수직으로 투영시켰을 경우 필기체 데이터에서 대부분 초성과 중성에 해당되는 획과 종성에 해당되는 획이 겹치게 된다. 일반적으로 x 좌표값이 중복되는 스트로크들을 OCs(Overlapped Components)로 정의한다[6]. 그림 2의 데이터는 여덟 개의 입력 스트로크들로 구성되어 있지만 x 좌표값을 수직으로 투영시켰을 때 하나의 OCs가 생성된다. 그림은 첫 번째 스트로크(S<sup>1</sup>)과 마지막 스트로크(S<sup>8</sup>)이 색칠된 사각형 영역만큼 겹쳐지는 것을 보여준다.



[그림 2] 스트로크에 대한 OCs

중성이 없는 한글 데이터인 경우 초성과 중성이 각각 하나의 OCs가 되는 경우가 대부분이기 때문에 초성과 중성을 하나로 묶어주는 과정이 필요하다. 스트로크를 그룹화하는 방법은 문자 인식 분야에서 다양한 방법들이 제안되었는데 가장 좋은 결과를 주는 방법이 convex hull을 이용한 방법으로 알려져 있다[6]. 이 방법은 OCs를 구성하는 스트로크들의 점들에 대한 convex hull을 구하여 이것의 중심점과 이웃한 convex hull의 중심점을 연결할 때 생기는 두 개의 교차점사이의 거리를 계산한다. 그리고 임계치 이하의 거리값을 가지는 경우 두 OCs를 묶어준다.

본 논문에서는 convex hull 방법을 이용하여 스트로크를 그룹화하였다. 그림 3은 각 OCs에 대한 교차점 사이의 계산된 거리값을 보여 준다. 거리값이 일정한 임계치 이하인 경우 두 OCs를 그룹화하여 제스처 기능을 처리하기 위한 단위 GCs(Gesture Components)를 생성한다. 그림 3의 경우 네 개의 OCs로 구성되어 있지만 d<sub>2</sub>가 임계치 이하의 값을 가지므로 두 번째 세 번째 OCs를 묶어서 하나의 GCs가 된다. 결과적으로 그림의 데이터는 세 개의 GCs가 생성된다.

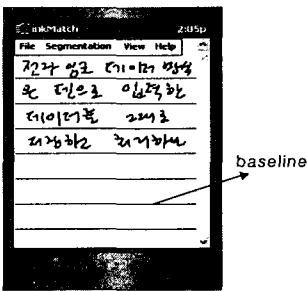
4. 구현 결과 및 결론



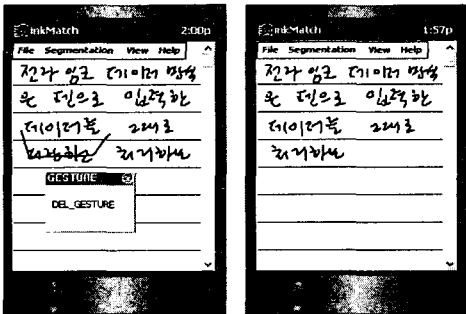
[그림 3] convex hull을 이용한 그룹화

본 논문에서 제안된 펜 제스처 인식 알고리즘과 스트록의 문자 단위 그룹화 기능을 이용한 제스처 허용 문서 편집기는 Pocket PC(Cassiopea E-125 model)에서 구현되었다.

펜 데이터의 입력은 베이스라인 사이에서 이루어 지도록 하였다. 그리고 제스처는 입력 데이터와 구별하기 위해 베이스라인에 걸쳐 입력하도록 하였다. 따라서 베이스라인과 교차되는 스트록이 입력되면 제스처 인식 알고리즘에 의해 제스처 유무를 결정하고 제스처인 경우 제스처 기능을 수행하기 위해 스트록들을 GCs 단위로 그룹화하는 단계를 거치게 된다. 제스처의 종류에 따라 GCs 단위로 해당 기능이 수행되는데 그림 4는 삭제 제스처 기능이 수행된 결과를 보여 준다.



(a) 편집중인 문서



(b)DELETE 제스처 입력 (c) 제스처 수행 결과

[그림 4] DELETE 제스처의 수행 화면

본 논문에서 제시한 제스처 허용 텍스트 에디터의 장점은 첫째 펜 기반 모바일 컴퓨터에서 펜의 기능을 최대한 이용하여 문서를 편집할 수 있도록 하였기 때문에 사용자에게 문서 편집을 위한 자연스럽게 편리한 인터페이스를 제공하는 것이다. 둘째 제스처 삽입을 위한 모드를 따로 지정하지 않더라도 문서를 편집하는 과정에서 언제든지 제스처를 입력할 수 있는 직접 제스처 방식을 제공하는 것이다. 셋째 제스처의 모양(shape)이 원고지 교정 부호를 참조하였기 때문에 사용자에게 친숙하고 빨리 익힐 수 있다는 장점이 있다.

향후 연구 과제로는 첫째 멀티 스트록 (multi-strokes) 제스처를 정의하여 사용하는 것이다. 본 논문에서 사용한 제스처는 단일 스트록으로 획의 개수를 제한하였다. 하지만 단일 스트록으로 표현할 수 있는 제스처의 종류는 다양하지 못하기 때문에 멀티 스트록(multi-stroke)으로 제스처를 정의하고 인식하므로 다양한 모양의 제스처를 정의하여 사용할 수 있도록 그 기능을 확장할 계획이다.

참고문헌

- [1] W. Aref, D. Barbara, D. Lopresti, and A. Tomkins, "Ink as a first-class datatype in multimedia databases," Multimedia Database, Springer-Verlag, 1995.
- [2] Walid G. Aref, Ibrahim Kamel, and Daniel P. Lopresti, "On Handling Electronic Ink," ACM Computing Surveys, Vol. 27, No 4. pp. 564-567, 1995.
- [3] Slate Corporation, Scottsdale, AZ. JOT:A Specification for Ink Storage and Interchange Format, 1993.
- [4] Landay, J.A and Myer, B.A. Extending an existing user interface toolkit to support gesture recognition. *Adjunct Proceedings of INTERCHI '93:Human Factor in Computing System*, pp. 24-29, April 1993.
- [5] Long, Jr., Landy, J.A and Rowe, L.A. PDA and gesture user in practice : insights for designers of pen-based user interfaces. *Technical Report CSD-97-976, U.C.Berkeley*, 1997
- [5] S.H. Kim, S.Jeong, G.S Lee and C.Y.Suen, "Gap metrics for handwritten Korean word segmentation," IEEE Electronics Letters Vol.m 37, No. 14, pp.892-893, 2001.