

펜 기반 PDA 시스템을 위한 제스처(gesture) 환경

박재필, 조환규

부산대학교 전자계산학과

e-mail: {jppark, hgcho}@pearl.cs.pusan.ac.kr

Gesture Environment for Pen-based PDA System

Jae-Pil Park, Hwan-Gue Cho

Department of Computer Science
Pusan National University

요약

제스처(gesture)는 마우스 또는 스타일러스의 단일 표시(single mark)를 이용하여 객체들과 기능들을 식별하는 문서 편집 도구의 강력한 방법들 중 하나이다. 마우스를 위한 제스처는 주로 데스크 탑 기반의 환경을 위해 개발되어지고 스타일러스를 위한 제스처는 보통 휴대용 컴퓨터를 위해 개발되어진다[2]. 현재는 PDA(Personal Digital Assistant) 시스템의 스타일러스를 위한 제스처 개발이 활발하게 이루어지고 있다[3]. 하지만 PDA 시스템에서는 인식에 의한 문자 입력의 보조 도구 정도의 수준에서 제스처가 사용되어진다. 본 논문에서는 PDA 시스템에서 사용되는 키보드의 키 영역을 하는 기존의 제스처와는 달리 사용자 입력과 제스처에 차별을 두지 않고 한 화면에서 직접적인 제스처가 가능한 환경을 개발하고 다양한 제스처 인식 방법을 제안한다. PDA 시스템에서의 제약을 최소화 하기 위해 펜 기반의 입력에서 각 스트로크를 각각의 시퀀스 정보를 최소화 한 뒤 각 스트로크를 양방향 연결 리스트를 이용해 연결하였다. 제스처(gesture)는 삽입, 삭제의 기본적인 것들을 비롯하여 이동, 복사, 붙임과 같은 제스처도 포함되어 있다. 제스처(gesture)의 인식은 각 스트로크의 벡터링 각도와 점들의 위치 정보, 스트로크 생성 순서에 기반하여 이루어지기 때문에, 이로 인해 펜 기반의 문서 에디터 사용에서 요구되는 다른 제약 없이 오로지 펜만으로 모든 편집 환경을 구현할 수 있다.

1 서론

휴대용 컴퓨터가 점점 작아지고 PDA의 사용자가 점점 늘어남에 따라 제스처를 이용한 입력이 키보드에 의한 입력보다 더 일반적인 것이 되어가고 있다[2]. 제스처(gesture) 기반의 시스템은 주로 텍스트 편집 응용프로그램의 지원을 위해 이용되어진다. 이러한 텍스트 편집 응용프로그램에서의 제스처는 두 가지 측면으로 그 기능을 나눌 수 있다. 첫 번째로 텍스트의 입력을 위한 제스처이다. 텍스트의 입력은 거의 대부분의 시스템에서 문자 인식을 통하여 이루어지고 있다. 문자 인식 기술에 많은 발전이 있었지만, 인식률(recognition rate)은 저조한 편이다.

두 번째로 편집 기능을 위한 제스처이다. 기능을 위한 제스처는 제스처 인식(gesture recognition)을 지원하는 다른 응용 프로그램을 요구한다. 지금까지 개발된 제스처 인식 응용 프로그램들 대부분은 인식의 효율을 위해 또 다른 제약을 수반하게 된다. 예를 들어, 입력된 문자를 지우려 할 때 제스처 버튼을 클릭 한 후 이러한 기능을 얻어 내거나 혹은 제스처 입력창을 통해 지우는 기능을 수행하게 된다.

PDA 시스템에서 대부분의 펜 기반 응용 프로그램은 위의 두 가지 기능에 대한 구별을 확실히 하고 있다. 이러한 구별은 좀 더 정확한 상호 작용을 제공할 수 있을지 모르나 사용자의 불편함과 시스템(메모리)의 부담을 유발시킨다.

이러한 문제들을 해결하기 위해 두 가지 종류의 제스처 기능을 하나로 통합하여 사용할 수 있는 환경이 요구된다. 처음에 언급했던 PDA에서 효율적인 텍스트 입력을 위해 handwriting recognition(HWX)를 배제하고 Pen Computing in Ink Domain(PCID) 시스템을 이용한다[5]. 아래는 이러한 HWX 방법을 피하고 PCID 시스템을 사용하는 이유는 다음과 같다.

- PCID 방법이 HWX 보다 정확하고 편리하기 때문에

사용자들은 일상생활 중에 일어나는 많은 일들을 ink domain에서 처리하고 있다.

- 사용할 수 있는 문자 집합에 제한이 없기 때문에 그림, 지도, 도표, 수식 등이 사용 가능하고 여러 가지 언어가 입력되고 처리될 수 있다.

Ink domain에서 작업하는 Pen Computing 방법은 사용자에게 풍부하고 자연스러운 표현 방법을 제공하는 장점이 있는 반면, 입력되는 스트로크 전체에 대한 정보를 저장하는데 있어서는 비효율적이다. 그와는 반대로 컴퓨터 연산 중심의 고정된 문자 집합(ASCII)을 사용하는 표현방법은 저장에 효율적인 장점이 있다. 특히 PDA는 적은 메모리를 가지기 때문에 이러한 문제는 심각한 것일 수 있다. 이 문제를 해결하기 위해 스트로크 간소화(stroke simplification) 방법을 이용한다[1].

제스처(gesture) 인식의 번거로움을 해결하기 위해 본 논문에서는 입력과 제스처의 차별 없이 입력 자체의 제스처 인식을 통해 직접적인 조작(direct manipulation) 방법을 이용한다.

2 Pen-Based PDA에서 텍스트 입력

PDA에서의 텍스트 입력은 주로 PDA 자체에서 제공하는 키보드와 문자 입력창에 의하여 이루어지고 한글의 입력을 위해서는 또 다른 한글 인식 응용 프로그램을 설치해야만 가능하다. 문자 인식률이 크게 향상되지 않는 한 이러한 시스템은 항상 원하는 텍스트의 입력을 위해 수차례 오류를 수정해야 한다. 이러한 문자 인식을 통한 입력방식의 불편함은 Pen Computing in Ink Domain 시스템의 이용과 펜으로 입력된 정보를 효과적으로 저장할 수 있는 스트로크 간소화로 해결될 수 있다.

PDA에서 스타일러스 펜을 통해 입력된 스트로크는 간소화 작

입을 거쳐서 폴리라인(polyline)의 집합으로 구성하여 양방향 연결 리스트(doubly linked list)로 입력된 순서대로 연결된다.

2.1 입력 스트록의 간소화

입력된 각 폴리라인에 대해 각과 길이의 두 가지 비교 연산을 이용하여 간소화한다[1]. 첫 번째 비교는 폴리라인을 구성하는 점들 사이의 각을 비교하는 것이다. 하나의 폴리라인에 저장되어 있는 세 점 P_{i-1} , P_i , P_{i+1} 사이의 각 $\angle P_{i-1}P_iP_{i+1}$ 이 사용자가 정의한 임계각 θ 보다 크다면, 그것은 하나의 직선 위에 있는 것으로 간주하고, 폴리라인으로부터 점 P_i 를 삭제한다. 이러한 각의 비교 연산은 모든 폴리라인에서 사용자가 정의한 임계각 θ 보다 큰 각이 존재하지 않을 때까지 계속하게 된다.

두 번째 비교는 각 폴리라인에 저장되어 있는 점들간의 거리를 비교하는 것이다. 각 폴리라인에 저장되어 있는 세 점 P_{i-1} , P_i , P_{i+1} 을 연결한 삼각형의 높이의 길이가 사용자가 정의한 임계값 L 보다 작을 경우 점 P_i 를 삭제한다. 이것은 매끈한 선으로 만들어주기 위해 필요한 작업이다. 아래의 그림 1은 입력 스트록의 간소화를 그림으로 표현한 것이다. 이리

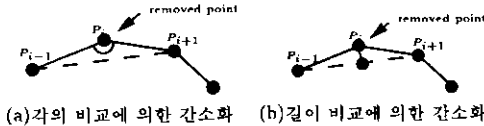


그림 1: 입력 폴리라인의 간소화

한 입력 스트록의 간소화는 PDA 시스템의 하드웨어 제약인 메모리 부족에 많은 도움을 줄 수 있고 메일로의 전송시 네트워크의 부하도 줄일 수 있게 된다.

2.2 스트록 바운딩 박스의 생성과 한 음절의 통합

PDA에 입력되는 스트록은 점들로 구성된 폴리라인만 가지고는 텍스트 입력과 제스처의 입력의 구분에 어려움이 따른다. 이를 위해 각 스트록의 구분을 위해 각 스트록에 바운딩 박스를 설정한다. 바운딩 박스는 입력되는 한 폴리라인의 점들 가운데 최대, 최소의 x 좌표값과 y 좌표값을 구해 박스의 왼쪽 상단의 한 점과 오른쪽 하단의 한 점으로 사각형을 그려 폴리라인 전부를 포함하는 바운딩 박스를 생성한다.

이렇게 생성된 바운딩 박스는 제스처 기능의 제공을 위해 한 자의 글자임을 인식할 필요가 있다. 한 글자의 인식은 두 가지 과정을 반복하여 한 글자를 포함하는 새로운 바운딩 박스를 생성한다. 먼저 각 스트록 바운딩 박스의 중점이 이전의 스트록 바운딩 박스의 중점과 사용자가 정의한 임계값 L 보다 작으면 두개의 바운딩 박스를 하나의 박스로 구성한다. 그리고, 하나의 바운딩 박스로 통합될 때 마다 그 박스의 중점을 구하고 다음 바운딩 박스와의 중점 비교를 한다. 만약 크다면 통합 바운딩 박스를 저장하고 다음 스트록의 바운딩 박스부터 처음 과정을 시작한다. 여기서 사용자의 임계값 L 은 노트의 배경 라인의 크기에 따라 자동 조정되어진다. 다음 그림 2는 스트록에 대한 바운딩 박스의 생성과 한 글자를 위한 바운딩 박스 생성을 보여주고 있다.

3 Direct Gesture의 인식

제스처(gesture)의 강점은 객체(Object), 기능(Operation), 파라미터(Parameter)들의 단일 표시(single mark) 집합을 결정짓는 능력이다. 일반적으로 강력한 제스처들은 한 획(single stroke)으로 결정되어질 수 있다. 즉 펜이 제스처



(a)한 스트록의 바운딩 박스 (b)한 글자의 바운딩 박스

그림 2: 스트록 바운딩 박스의 생성

를 취하는 동안에는 펜을 들지 않는다는 것을 의미한다. 현재 사용되어지고 있는 응용 프로그램의 제스처는 이러한 한 획으로 결정되지 않는 경우가 대부분이다. 예를 들어, 제스처 입력시 제스처를 입력하는 다른 전용 창에 해야 한다거나 제스처 모드(gesture mode)의 버튼을 클릭한 상태에서 입력하는 방법들이 주로 사용된다. 본 시스템에서 사용하고 있는 방법은 이러한 간접적인 제스처 방법이 아니라 사용자가 펜으로 텍스트나 그림 등을 입력하고 있는 위치에서 별도의 작업 없이 바로 제스처를 사용한다. 이러한 방법은 자칫 제스처 인식면에서 많은 제약을 가질 수 있지만 제스처의 기능적 특성을 이용한다면 이 문제 또한 줄어들 수 있다. 예를 들어, 문자 삽입을 위해 "L" 모양의 직접적인 제스처(direct gesture)를 사용할 때 영어 알파벳의 "L"과의 구분을 위해 사용자에게 제약을 주는 것이다. 일반적으로 삽입은 사용자가 기존에 입력한 텍스트 위에서 행해지며 삽입 제스처 "L" 모양에 대한 스트록의 위치 및 중심각의 크기 등의 기능적인 특성을 가진다. 이러한 다양한 특성들을 적용시켜 직접적인 제스처(direct gesture)를 구현할 수 있다. 보통은 이러한 문제를 해결하기 위해 학습을 통하여 제스처 인식자(gesture classifier)를 생성한다[4]. 하지만 이러한 제스처 인식자 또한 사용자의 훈련이 필요하며 훈련으로 저장된 제스처를 다시 매칭(matching) 시켜야하는 번거로움이 있다. 그림 3은 직접적인 제스처(direct gesture)를 위한 진행 순서를 보여주고 있다. 이러한 제스처들은 간단한 드로

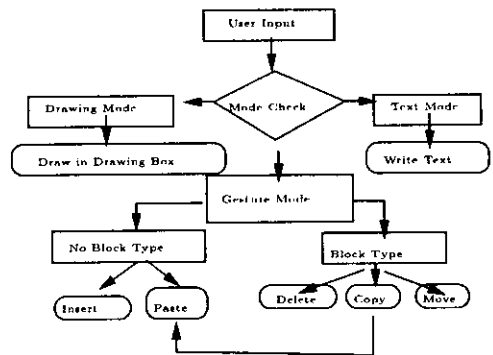


그림 3: Gesture 처리 flow chart

잉(drawing) 프로그램에서 사용되어진다. 사용자의 입력에 대한 모드체크(mode check)는 먼저 정의된 각 제스처들의 기능적 특성에 관한 조건부터 검사하게 된다. 예를 들면, 한

개의 스트로크가 입력되었을 때 불럭 표시를 위한 것인지 아닌지 부터 검사한 후 기존의 그림모드 박스에 위치하는지 확인하고 각 제스처의 특성인 점들간의 각도나 시작점과 끝점의 위치에 대한 정보, 바운딩 박스들과의 관계에 대한 위치적 정보들을 이용해서 제스처임을 확인한다. 그림 4는 본 시스템에서 구현된 제스처의 종류들이다. 보통 삽입, 삭제, 붙이기

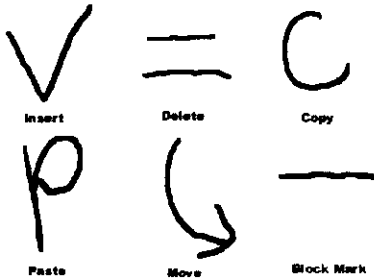


그림 4: 구현된 Pen Gesture의 종류

제스처는 불럭 지정의 전처리 과정을 가질 필요가 없지만 복사와 이동 제스처는 이러한 과정이 필요하다. 즉, 불럭 지정이 필요한 제스처는 한 획으로 결정되어지지 않는다. 이러한 불럭의 지정은 한 개의 수평선으로 표현 할 수 있으며 반드시 입력되어 있는 스트로크와 겹쳐져야 한다.

4 실험 결과 및 향후 과제

본 논문에서 제안한 PDA에서의 PCID 방법의 입력과 간소화(simplification)를 바탕으로 하여 직접적인 제스처(direct gesture)의 방법을 통해 펜 기반 PDA 시스템의 gesture 환경의 개선을 제안하였다.

시스템은 Pentium III 600MHz에서 Visual C++ 6.0을 통해 구현되고 Embedded Visual C++ 3.0을 사용하여 Embedded 환경으로 재 구성하여 실행과일을 생성한 후 Windows CE 기반의 Pocket-PC를 위한 Platform SDK 에서 Desktop Pocke PC Emulator를 이용하여 테스트하였다. 그림 4는 본 논문에서 제안한 환경으로 구성된 에뮬레이터에서의 제스처 사용을 보여준다.

본 논문은 문자 인식의 단점을 피하기 위해 펜 기반의 PDA(Personal Digital Assistant) 시스템에서 잉크 시스템을 사용하여 입력을 하고 그것을 간소화 하여 적은 양의 데이터로 만드는 방법과 직접적인 제스처(direct gesture) 방법을 이용하여 문서 편집 환경의 개선을 제시하고 있다.

기존의 PDA의 문서 편집 환경에서의 입력되는 방식은 펜 제스처가 문자 인식의 과정을 통해 입력되거나 여러 절차를 통해 펜으로 직접 입력되는 방식을 사용하고 있다. 제스처 입력은 점점 더 인간 중심적으로 발전되어 가고 있으며 특히 PDA와 같은 휴대용 단말기의 경우에는 그 중요성이 더욱 크다고 할 수 있다. 하지만 이러한 입력은 어떠한 방식으로 이루어 질 수 있지만 그에 수반되는 제스처의 인식 작업 없이는 힘들 것이다.

제스처(gesture)의 인식에 있어 제약을 최소화하고 각 사용자 별로 정해진 기능에 대해 제스처를 정의하고 훈련을 통한 제스처 집합을 형성하여 그것이 곧바로 제스처로 인식될 수 있는 연구도 이루어지고 있다. 하지만 이러한 것조차도 사용자의 입장에서 보면 번거로운 일일 수 있다.

어떠한 제스처 표준을 통해 사용자들의 직접적인 제스처(direct gesture)가 이루어진다면 향후에는 마치 종이에 펜

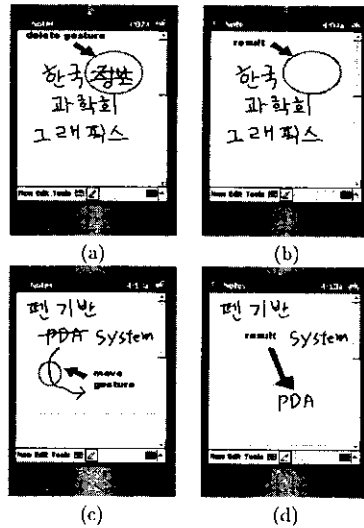


그림 5: (a)(b): 삭제 제스처의 입력과 실행 결과 (c)(d): 이동 제스처의 입력과 실행 결과

을 가지고 쓰는 것 같은 효과를 얻을 수 있을 것이다.

5 감사의 글

본 연구는 정보통신연구진흥원에서 지원한 '멀티 미디어 단말기 그래픽 및 GUI(Graphics User Interface) 기술' 과제의 연구 결과입니다.

참고문헌

- [1] 김해선, 진희정, 유영중, 조환규. 경계선 간략화를 이용한 일러스트레이션 기법. 한국정보과학회 2000, 27(2):484-486, September 2000.
- [2] Landay, J.A and Myer, B.A. Extending an existing user interface toolkit to support gesture recognition. *Adjunct Proceedings of INTERCHI '93: Human Factors in Computing System*, pages 24-29, April 1993.
- [3] Long, Jr., Landay, 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.
- [4] Long, Jr., Landay, J.A and Rowe, L.A. Implications for a gesture design tool. *Human Factors in Computing Systems (SIGCHI Proceedings)*, ACM, ACM Press, 40-47, May 1999.
- [5] Daniel Lopresti and Andrew Tomkins. Computing in the ink domain. *proc. CHI '99*, pages 543-548, May 1999.