2차원 제스춰 인터페이스를 사용하는 한글 문서편집기의 설계와 구현

°이 재혁, 김 진형, 이 윤 준, 김 명 호
한국과학기술원 전산학과

Design and Implementation of Hanguel editor with 2-dimensional gestural interface

Jai Hyuk Lee, Jin Hyung Kim, Yoon Joon Lee and Myoung Ho Kim
Dept of Computer Science, Korea Advanced Institute of Science and Technology

요 약

제스춰란 "의미를 담고있는 몸짓"이라고 정의 할수있으며 기존의 키보드나 마우스등의 입력 장치와 같은 기계본위의 동작과는 다르게 실생활에서 쓰는 몸의 동작을 그대로 기계가 인지하는 인간 본위로의 전환이라는 점에서 중요한 의미를 갖는다 최근에는 키보드난 마우스없이 펜만으로 입력을 하는 펜 컴퓨터가 상품화가 되고있는데 이 논문에서는 기존의 펜 컴퓨터에서의 제스춰를 확장하고 이것을 응용하여 2차원 제스춰 인터페이스를 사용하는 한글문서 편집기를 설계와 구현한다.

1 서론

조기의 컴퓨터에는 키보드가 유일한 사용자와 컴퓨터간의 대화방법이었다 그후 마우스를 도입한 윈도우 시스템이 개발되면서 사용자는 훨씬 더 편리한 대화방법을 사용하게 되었다 그러나 이러한 사용자 대화 장치, 즉 키보드와 마우스만으로는 충분한 대화방법이 되지 않았다 이 장치에 익숙하여질 때까지 상당한 시간이 걸렸고, 사용하기에 쉽지 않으며, Point 장소와 cursor가 공간적으로 분리되어 있기때문에 효율적인 입력 수단이 되지 못하였다. 이러한 이유로보다사용하기 쉽고 자연스러우며 효율적인 입력 수단이 될수 있는 사용자 대화 장치에 대한 요구가 증가하였다 새로운 대화상자는 인간과 컴퓨터의 대화 방식이 실제 인간의 생활에서 사용되던 방식으로부터 도입되는 것이 가장 바람직 하다 그러한 방식이어야만 사용자가 평소 일상생활에서 사용하던 대화 방법을 그대로 컴퓨터에 적용하여 자연스럽고 쉽게 컴퓨터와의 대화를 할 수 있기 때문이다

컴퓨터 용어에 사용자와 컴퓨터간의 대화방법으로 제스춰라는 말이 등장한 것은 아주 최근의 일이다 제스춰란 인간의 일상생활에서 흔히 볼 수있는 어떤 의미를 나타내는 몸짓을 말하는데 최근에 하드웨어 기술의 발전에 힘입어 제스춰를 컴퓨터와의 대화방법으로 사용할수 있게 되었고 제스춰를 사용하는 응용프로그램들이 점점 개발되고 있다 예로는 문서의 편집 명령을 제스춰로 하는 문서 편집기, 제스춰로 조절하는 Speech Synthesizer [Ran90], 음성인식의 보조수단으로 방향 지시방법에 제스춰가 사용되는 PutThatThere [Gor90] 등 여러가지 형태의 시스템이 있다 최근에 가장 널리 소개가 되고있는 PLI (Paper Like Interface) 에서의 제스춰는 기존의 키보드나 마우스가 없이 펜으로 화면에 직접대고 입력하는 방식으로 사용자가 종이위에 펜으로 쓰는 느낌으로 사용하게 하자는 것이다 기존의 PLI 응용 프로그램에서의 문제점은 제스춰 인터페이스가 하나의 특정 응용프로그램 위주로 제작되었기 때문에 다른 응용프로그램에 재사용하기 힘들다는데 있다 이 논문에서는 이러한 문제를 해결하고자 응용프로그램에 비종속적인 제스춰 인터페이스를 제작하고 이것을 한글 문서편집기에 응용하여 전체 시스템을 구현하는데에 있다

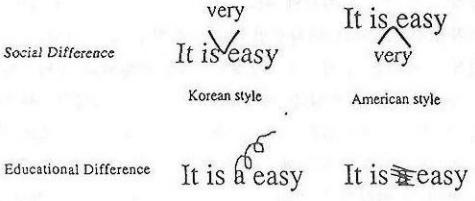
2 제스춰 입력의 특성

PLI에서는 모든 펜의 입력은 점(X,Y)의 나열로 이루어

진 획이다. 하지만 이것을 해석하는 방법에 따라 문자, 제스쳐, 도형, 낙서 등 여러가지 object로 나눌 수있다. 이중에서 제스쳐가 갖는 특징과 방향성은 다음과 같다.

2.1 제스쳐 심볼

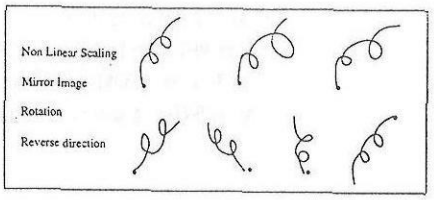
어떠한 심볼을 제스쳐로 사용할 것이가는 아주 중요한 문제이다. 문자의 경우는 이미 사회적으로 일정한 규약이 있어서 그것을 따르면 되지만, 제스쳐의 경우는 어떤 행동에 대한 대표되는 심볼이 없기 때문이다. 하지만 제스쳐는 인간의 일상생활에서 쓰는 동작을 그대로 응용했을때 가장 큰 효과를 볼수있다. 이런 심볼선정 문제 때문에 IBM에서는 PLI 상에서의 워드프로세서에 필요한 제스쳐 심볼 선정을 위해 설문조사를 했다[Cat87]. 그 내용은 문자의 삭제, 삽입, 등의 문자 편집기 명령어들을 종이위의 문장위에 써보게 하는것이었다. 그 결과는 삽입이나 삭제같은 잘 알려진 명령에 대해서는 많은 사람들이 원고지에서 사용하는 교정기호를 써서 응답을 했다. 따라서 제스쳐 심볼의 선정은 한 명령에 대하여 많은 사람들이 사용하는 심볼 몇개를 선정하여 해결할수 있다는 것이다. 이미 상품화 되어있는 PLI의 경우에는 제스쳐 심볼로서 국가표준안의 교정부호를 단순화하여 미리 교육을 받지 않았더라도 쉽게 배워 사용할수 있게 하는데 노력을 했다[Ste88],[Rob91]. 하지만 여기에는 몇가지 문제가있다. 이러한 설문조사를 통한 심볼의 선정은 그 대상에 따라서 큰 차이를 보인다. 지역이나 사회적 배경이 다른 곳에서의 심볼 선정은 다른 결과를 가져올 것이고 위의 워드 프로세서의 경우에는 교정기호 교육을 받았는가 의 차이에 의해서 또한 다른 결과를 가져올것이기 때문이다.



<그림 1> 제스쳐 심볼사용의 사회적 교육적 차이

또한 교정부호만으로 모든 워드프로세서 명령을 대치할 수 없다. 원고지와는 다르게 워드프로세서 고유의 명령이 많이 존재하기 때문이다. 다음 페이지(Page Down)와 같은 화면 이동, 폰트를 바꾸는 명령은 표준화된 교정부호에 들어있지 않으나 제스쳐화 해야하는 명령이다. 또한 PLI 에서 워드프로세서만 제스쳐를 사용하는 것이 아니라 여러 응용프로그램이 제스쳐를 사용하기 때문에 각 응용프로그램의 제스쳐 심볼도 선정해야한다. 물론 워드프로세서와 같은 교정부호로 제스쳐 심볼을 사용할 수도 있지만 응용프로그램의 특성을 살리기 위해서는 각 응용프로그램에 알맞은 심볼을 사용해야한다. 그러므로 교정부호를 일률적으로 제스쳐 심볼로 사용하는 것은 알맞지 않다. 예를들어 스프레쉬 시트의 경우에 합(sum)이란 함수에 대한 제스쳐 심볼을 교정기호에서는 찾을수없으며 그대신 Σ로 사용하는것이 바람직하다.

제스쳐 심볼 선정에 있어서는 다음의 몇가지 기준을 따라야한다. 첫째 제스쳐 심볼이 다른 입력들, 즉 문자 도형의 심볼과 충돌이 일어나지 않아야한다. 어떤 입력이 왔을때 이것이 문자인가 제스쳐인가 도형인가를 구분하는 문제도 어려운 문제이기 때문에 제스쳐 심볼의 선정과정에서 이러한 충돌을 최대한 없애야한다. 둘째로 되도록 사용시 심볼의 일치도가 높은 심볼을 선정해야한다. 심볼의 일치도라는것은 inter user consistency와 intra user consistency를 말하는것으로 사용자의 교육적 수준, 사회적 또는 국가적배경이 달라도 높은 일치도를 나타내는 심볼을, 한 사용자의 사용시기, 감정 상태가 달라도 높은 일치도를 나타내는 심볼을 선정해야한다. 세째로 배우기 쉬운 심볼을 사용해야한다. 또한 제스쳐 심볼은 사용시에 문자와는 틀린 몇가지 특징을 갖는데, 한 심볼의 모양에 대한 심볼의 회전 (rotation), 비규칙적인 확대 또는 축소 (Nonlinear scaling), 대칭 모양 (mirror image), 역필순 (reverse direction)등의 변형이 많이 존재한다는 것이다. [Kim88] 따라서 제스쳐 인식기는 심볼에 따라 이러한 변형등을 흡수할수 있는 기능이 있어야하는데 이점이 제스쳐 인식기의 개발을 어렵게 하고 있다.



<그림 2> 제스쳐 심볼의 특징

제스쳐 심볼의 또하나의 특징은 그 기호에 따른 동작이 일어날 위치가 정해져 있다는 것이다. 이 위치는 그 기호가 나타내는 의미에 따라 다르게 되는데 제스쳐 인식기는 입력에 대한 동작 위치도 또한 찾아 낼수 있어야 한다.

2.2 제스쳐 대화 방식

제스쳐를 실제 시스템에 적용하기 위해서는 제스쳐와 실제 시스템의 결합방법, 즉 대화 방법이 고려 되어야 한다. 이 대화 방식에는 "Direct Manipulation", "Switch to gesture mode", "region Selection and Menu", "Regoin Selection and Handwritten" 등이 있다. [Jam86],[Jim87]

Direct Manipulation이란 제스쳐를 입력하고 펜을 때면 바로 이것을 인식하여 그 동작에 대한 결과를 보여주는 것으로, 가장 바람직한 대화 방법이다. 하지만, 모든 제스쳐는 펜을 가지고 모든것을 행하기 때문에 펜을 이용하는 문자, 도형과 모양에서 충돌이 생길 수 있다. 그래서, 이러한 충돌이 생길 가능성 이 있을 때는 이것을 피하기 위해서 다른 대화 방식이 사용되어야 한다. Switch to Gesture Mode란 바로 이러한 충돌을 없애기 위해서 제스쳐를 사용할 때에는

먼저 모드를 제스춰로 전환하는 제약을 가하는 것이다 이것은 사용하기에는 불편하지만, 제스춰 기호의 충돌 문제를 해결할 수 있기 때문에 이 모드로 전환하는 행동을 최소화하는 선에서 고려 되어 사용할 만하다 Region Selection and Menu는 기존의 마우스 기반의 시스템에서 사용하던것을 응용한 것으로 아주 분확실한 제스춰, 특별한 모양으로 기호화 하기 힘든 제스춰가 있을 때에 일단은 그 동작의 Source 범위를 먼저 선택한 후 동작에 해당하는 메뉴로 선택하게 하는 방법인데, 이러한 제스춰가 아주 많을 때에는 힘들지만 몇개 안되는 제스춰가 기호화 하기 힘든 경우에, 사용할 수 있다 Region Selection and Hadwritten은 동작의 범위를 선택한 후에, 그 명령을 기호와 약간의 handwritten으로 입력하는 대화 방식이다 보통의 제스춰 심볼로 표현하기 힘들지만 제스춰에 약간의 문자나 숫자를 도입하여 더 알기쉬운 제스춰 심볼을 만들수 있을때 사용한다 예를들어 컴퓨터 상에서 화면의 크기 제약으로 목표지점에 직접 제스춰 심볼을 사용할수 없을때 목표지점을 나타내는 숫자를 그 목표지점의 대용으로 생각하게 할수있다 제스춰 대화 방식은 PLI에서의 주요 사용자 인터페이스로서 사용자에게 큰 영향을 주므로 매우 신중히 고려되어야 할 부분이다

2.3 제스춰 인식

일반적으로 제스춰 인식의 난이도는 제스춰의 자유도가 높을수록, 인식할 제스춰의 수가 많을수록, 제스춰의 속성이 연속적일수록 높아지게 된다 자유도가 높다면 입력의 종류가 많아져서 보아야할 특징이 많아지게 된다 현재까지의 인식 기술로 볼때 이렇게 되면 결국 여러 특징들간에 혼란이 생겨서 각 제스춰가 갖는 고유한 특징들을 상실할 우려가 있다

제스춰의 수가 많아져도 각 제스춰 모양을 구분짓는 특징을 찾기가 힘들어지게 된다 또한 연속적인 속성을 갖는 제스춰가 더 어렵다는것은 당연한 일이다 왜냐하면 정적인 제스춰에 비해 연속적인 제스춰를 인식하기위해서는 연속성을 인식하기 위해서 자유도가 더 증가해야 하기 때문이다

PLI에서의 제스춰 인식기는 제스춰 장치로부터 입력을 받아 그것을 인식하여 그 제스춰의 종류, 시작 위치의 범위, 목적 위치의 세가지 정보를 병행 출력하는 것이다 즉 기존의 온라인 문자 인식기와는 다르게 그것의 연산이 일어날 위치도 함께 알아내야 한다 이 연산위치는 심볼의 종류나 응용프로그램에 따라 다르게 나타난다 제스춰를 성공적으로 인식했더라도 이 연산위치를 잘못 계산하게되면 결국 입력한 제스춰 명령을 취소하고 다시 입력해야하므로 제스춰 인식을 못한 것 보다도 못한 상황이 일어난다 그러므로 인식 못지않게 이 연산위치를 계산하는 방법도 중요하다.

제스춰 인식에 관한 연구가 시작된것은 아주 최근의 일이며 아직까지는 큰 연구성과가 나오고 있지 않다 대부분의 연구방향은 지금까지 문자 인식에서 사용되었던 많은 방법은 그대로 적용하되, 제스춰 심볼의 특성인 회전, 불규칙한 확대및 축소, 반사된 모양, 역필순 등을 새로이 고려하여 이것을 반영하는 것이다 제스춰 인식의 흐름은 크게 두가지로 볼수있다 첫째는 각 심볼의 특징을 분석해서 각각의 고유의 특징을 알아낸다음 그 특징을 근거로 인식 알고리즘을 구현하는 Feature analysis방법이다 이러한 접근 방법은 구현자의 특징 추출 능력에 따라 인식기의 성능이 달라지게 되는데 제스춰 의 수가 아주 소수일때 알맞은 방법이다 [Kim88],[Stc88],[Rog90] 두번째 방법은 학습에 의한 제스춰 인식 방법이다 이것은 일정 범위내에서사용자의 변형을 흡수 할수있기 때문에 사용자간에 넓은 일치도를 갖는 제스춰 인식에 적합한 방법이라고 할수있다 또한 최근의 제스춰 인식 방법의 경향이 이 방향으로 흐르고 있다 [Jam91], [Dea91]

2.4 사용자 정의 제스춰

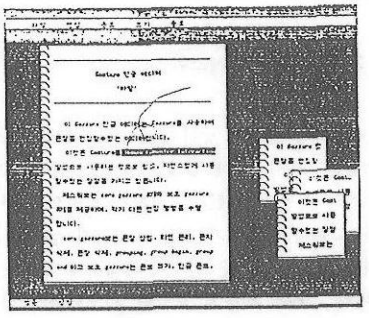
지금까지 본바와 같이 현재의 PLI에서 제스춰 사용하는 시스템은 그 시스템에 알맞는 심볼을 선정하고 그 심볼을 제스춰의 특성을 갖게끔 여러가지 방법으로 구현하는 것이었다 그러나 제스춰가 갖는 여러가지 심볼의 특징은 반영했지만 제스춰 심볼을 인식기 제작시에 고정을 시키게 되므로 미리 정해진 심볼의 모양을 사용한다는 점에서 사용자는 실생활에서의 제스춰와 같이 자연스럽게 제스춰를 사용하지 못할수가 있다 또한 여러가지 응용 프로그램 의 제작시에 각 응용프로그램 마다 서로 다른 모양을 갖는 제스춰를 사용할수가 있는데 이경우에 제스춰 인식기를 각 응용프로그램에 맞게 새로 제작해야하는 문제점이 있다 이런 이유로 제스춰 심볼의 선정 권한을 제스춰 인식기 제작자가 아니라 사용자나 응용프로그램 제작자에게 주는게 바람직하며 이 논문에서는 이러한 사용자 정의 제스춰 인식기 제작하고 그것의 응용프로그램으로 제스춰를 사용하는 한글 에디터를 구현한다

사용자 정의 제스춰 인식기는 다음의 여러가지 요구사항을 만족해야한다 첫째로 제스춰 심볼이 갖는 특성 (non linear scaling, rotation reverse direction, mirror image)을 흡수할수 있어야 한다 둘째로 사용자가 정의한 심볼을 학습하는데있어서 제스춰가 갖는 특성을 포함하면서도 학습률이 빨라야 한다 세째로 인식 속도가 대화식(interactive)으로 사용하는데 지장이 없을 정도로 빨라야 한다 네째로 정의 심볼의 모양에 관한 제한이 없어야 한다 마지막으로는 인식률이 매우 높아야 한다는 점이다 위의 요구사항을 완전히 만족하는 인식기를 개발하기는 쉬운일이 아니다 하지만 앞으로의 제스춰 연구 방향은 사용자 정의 제스춰와 같이 인간위주로의 시스템 즉 사용자가 미리 정해진 심볼에 길들여져 가는 것이 아니라 기계가 사용자의 의미를 이해하고 그것에 충실하려는 시스템이 되어야 한다

3 PLI환경에서의 문서 편집기 개발

PLI환경에서는 키보드나 마우스를 사용하지 않고 모든 것을 펜으로 처리해야하기 때문에 기존의 편집기와는 다르게 사용자 인터페이스가 정의 되어야 한다 즉 키보드와 마

우스를 사용할때의 개념을 펜에 맞는 대화형식으로 재정의 해야한다. 개발한 편집기에서는 기존의 편집기중 19개의 명령을 여러가지 제스춰 심볼과 제스춰 대화 형식으로 대치하였다. 19가지 명령은 기본적인 편집명령과 화면 이동 명령, 폰트 변환같은 편집기 고유의 명령에 해당한다. 대화 형식도 Direct manipulation을 위주로 하면서 HandWritten을 약간 도입한 형태, 공간적인 버퍼링으로 기존의 명령을 더욱 쉽고 편하게 대치했다. 편집기는 영문, 한글을 편집할수 있고, 다중 폰트, 다중 크기의 문서를 편집할수 있는 문서 편집기이다



<그림 3> 제스춰 인터페이스 한글문서편집기

3.1 명령 코드와 심볼의 정의

명령 코드는 단순히 0에서 18까지의 숫자로 정의 하였고 사용한 제스춰 심볼은 3개의 set으로 각각 13,6,26개의 심볼로 구성된다. 첫번째 심볼 set은 사용자가 정의하는 제스춰로 학습에 의한 인식을한다. 두번째 set는 core제스춰로 많은 응용프로그램에서 사용하는 제스춰로 인식률을 높이기 위하여 학습에 의존하지않고 심볼을 고정시켜 특징분석으로 해결한다. 세번째 set는 단순한 문자,숫자 set로 제스춰중 문자, 숫자의 의미가 필요할때 도입해서 쓸수있는 set로 각각 영문,숫자 인식기로 대치할수 있는 set이다. 예로는 1번 한글폰트로 바꾸라는 명령이 있다. 문서 편집기는 상황에 따라 세가지 set중 한가지 set에서의 인식 결과를 얻는다.

3.2 Semantic filter

정확한 연산 위치를 요구하는 제스춰에 연산위치를 계산하기가 모호할때는 심볼을 인식했더라도 실행 불가로 Reject해야된다. 또한 같은 심볼 모양이라도 그 위치에 따라 연산위치 뿐만이 아니라 명령의 해석이 틀려지게 할수도 있는데 삽입기호의 경우 편집기 내에서 입력했을때는 문장 삽입이란 명령이 되나 이외의 경우에는 화면의 스크롤 다운 명령이 된다. 이렇게 심볼의 모양의 차이는 없더라도 그것의 위치관계에 따른 에러를 찾아내는 부분을 semantic filter라고 하며, 응용프로그래머가 입력획과 응용프로그램 내의 위치비교로 이루어진 간단한 프로그램으로 작성된다.

4. 결론

제스춰는 실생활에서의 동작을 그대로 컴퓨터와의 대화에 이용하는것이다. 그러므로 실생활에서 개인의 개성에 따라 큰 차이를 보이는 제스춰의 심볼은 본질적으로 사용자가 직접 정의하고 사용하는 사용자 정의 시스템이어야 한다. 또한 제스춰는 문자에 비해서 심볼의 변형이 매우 크다. 따라서 기존의 문자인식방법과는 인식방법이 있어야하며 또한 인식시에 심볼의 종류뿐만이 아니라 연산위치도 인식해야한다. 새로개발된 제스춰 인터페이스는 여기에서 응용된 한글 문서편집기 뿐만이 아니라 심볼set가 다른 여러 응용프로그램에서도 사용할수 있었다.

앞으로 더 연구해야할 과제로는 제스춰와 문자입력과의 충돌문제 해결에 있다. 한글,영문 문자, 제스춰, 도형등을 제대로 구분해주는 구분기가 있어야 더욱 진전된 사용자 인터페이스를 제작할 수있기때문이다.

참고 문헌

[Cat87] Catherine G. Wolf, and Palmer Morrel Samuels, "The use of hand-drawn gestures for text editing," Man-Machine Studies 27, pp. 91-102, 1987.

[Cat87]Catherine G. Wolf, and James R. Rhyne, "A Taxonormic Approach to Understanding Direct Maxipulation," RC 13104(#58210), Human Factors, 1987.

[Dea91]Dean Rubine, "Specifying Gestures by Example," Computer Graphics, Vol. 25, No. 4,July 1991.

[Dou91]Doug Gephurdt, and Mark C. Klonower,"Destination Laptop," BYTE, pp. 239-250,Febrary, 1991.

[Gor90]Gordon Kurtenbach and Eric A .Hulteen,"Gestures in Human Computer Communication," The Art of Human-Computer Interface Design, pp. 309-318, 1990.

[Jam86]James R. Rhyne and Catherine G. Wolf, "Gestural Interfaces forInformation Processing Applications," RC 12179 (#54544), Computer Science, 1986.

[Jam91]James S. Lipscomb,"A trainable Gesture Recognizer," Pattern Recogniton, Vol. 24, No. 9, pp 895-907, 1991.

[Jim87]Jim Rhyne,"Dialogue Management for Gestural Interfaces," Computer Graphics, Vol. 21, No. 2, April, 1987.

[Kim88]Joonki Kim, "On-line Gesture Recognition by feature analysis,"Proceedings of Vision Interface, June, pp. 6-10, 1988.

[Min89]M. Minsky, "Manipulating Simulated Objects with Real-world Gestures using a Force and Position Sensitive Screen,"Computer Graphics 18, No. 3, pp. 195-203, 1989

[Ran90]Randy Pausch, and Ronald D. Williams, "Creating Custom User Interfaces Based on