

사용자에 친밀한 문서 편집기의 한가지

One of User Friendly Word Editors

심상덕, 김인순, 이상욱, 추명경, 손영선
동명정보대학교 정보공학부 정보통신공학과

Sang-Duck Sim, In-Soon Kim, Sang-Wook Lee, Myung-Kyung Chu, Young-Sun Sohn
Communication Engineering, TongMyoung University of Information Technology
E-mail : yssohn@tmic.tit.ac.kr

ABSTRACT

본 논문에서는 Windows 환경에서 동작하는 문서 편집기상에서 사용자의 명령을 음성으로 입력받아 글자 크기 및 화면의 이동 속도를 조절하는 시스템을 연구하였다. 입력된 명령으로부터 추론된 글자 크기 값을 중심으로 font size 변경 범위를 1/2씩 줄여 감으로써 사용자가 만족하는 글자 크기로 접근해 가는 방법을 사용하였다. 화면의 이동 속도는 사용자가 선택한 명령 부근에서 원하는 이동 속도로 근접할 것이라는 개념을 도입하여 사용자가 원하는 속도로 접근하는 방법을 사용하였다.

1. 서론

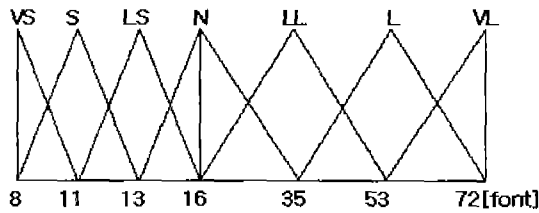
최근, 컴퓨터가 인간생활에 널리 보급되면서 컴퓨터 사용 인구가 증가하게 되었다. 그에 따라 인간에게 친밀감을 주면서, 더 편리한 컴퓨터 시스템을 만들기 위해 많은 연구가 행하여지고 있다[1-4]. 사람과 컴퓨터사이의 인터페이스는 키보드, 마우스, 조이스틱 등과 같은 하드웨어 부분과 한자 변환 소프트웨어, 그래픽 작성 소프트웨어 등의 소프트웨어 부분으로 생각할 수 있다[5]. 이와 같은 여러 가지 인터페이스 중에서도 우리가 사용하는 자연 언어를 이용한 연구가 계속적으로 행하여지고 있다[2,4]. 본 논문에서는 인간의 음성을 입력으로 하여 문서 편집기의 글자크기 및 화면의 이동 속도를 조절하는 인터페이스를 구현하였다.

2 추론 알고리즘

2-1. 글자 크기에 대한 추론

일반적인 문서 편집기에서 지원하는 글자 크기가 8~72 font이므로, 구현한 시스템에서도 글자 크기에 대한 멤버쉽 함수의 범위를 동일하게 하였으며, 이 범위 안에서 문서를 읽기 적당한 글자크기로서 16 font를 초기 기준으로 정하였다. 구현한 인터페이스 화면 상에는 글자 크기의 변화가 1 font 단위로 지원되어지기 때문에 퍼지 추론에 의해서 얻어진 글자크기를 동일한 단위로 연계 하였다. 처음 명령이 입력되어지면 전체범위에서 추론이 이루어지며 차기 명령이 입력되어질 때마다 추론범위를 이전 범위의 반으로 좁혀 나간다. 처음 화면에 문서가 보여질 경우의 멤버쉽 함수가 그림 1에 보여진다.

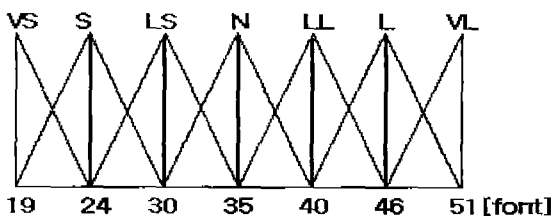
초기 중심 값을 기준으로 큰 쪽의 영역이 작은 쪽의 영역보다 크기 때문에 큰 영역을 선택했을 때 재 추론하는 범위를 어느 정도 포함 시켜주면서, 역 방향으로 이동하는 경우를 고려하기 위해서 전체범위의 1/2씩 줄여나가는 방법을 사용하였다. 이러한 일련의 과정을 통하여 사용자가 원하는 글자 크기에 근접하도록 하였다.



- VS(Very Small) : 매우 작게
- S(Small) : 작게
- LS(Little Small) : 조금 작게
- N(Normal) : 보통
- LL(Little Large) : 조금 크게
- L(Large) : 크게
- VL(Very Large) : 매우 크게

<그림 1. 글자크기에 대한 초기 멤버쉽 함수>

그림 1에서 입력된 명령을 기준으로 전체 범위의 절반을 다시 전체 범위로 보고 중심을 기준으로 양쪽 범위를 동일하게 나눔으로써, 현재의 글자 크기에서 큰 영역과 작은 영역을 같게 고려하여 멤버쉽 함수를 구성하였다. 추론과정에서, 명령이 입력되어지면 그림 2에서 볼 수 있듯이 해당 멤버쉽 함수의 중심 값을 font size로 변경시킨 후, 그 값을 기준으로 새로운 멤버쉽 함수를 구성하였다.



< 그림 2. 변화된 멤버쉽 함수 >

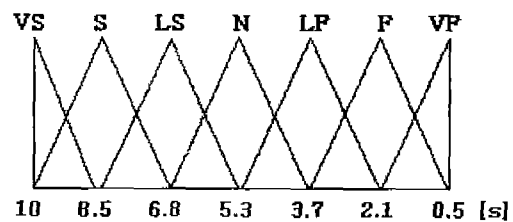
설정된 범위가 초기의 전체범위(8~72)를 초과 할 경우에는 최대 값 또는 최소 값으로

범위를 제한시켰다. 또한, 추론에 의해 나온 중심이 최대 또는 최소치와 2 font 이하로 근접해 있을 경우, 멤버쉽 함수가 구성되지 않는다. 따라서 더 이상 추론이 이루어지지 않으므로 같은 쪽의 명령이 다시 입력되면 최대 또는 최소치 값을 가지게 하면서 사용자에게 더 이상의 글자변환이 이루어 질 수 없음을 알려주도록 하였다.

한편, 추론 범위를 이전 범위의 반으로 좁혀 가는 알고리즘을 사용하기 때문에 명령이 입력되어 질수록 멤버쉽 함수의 범위는 좁아진다. 이때, 사용자가 연속적으로 '매우 크게' 나 '매우 작게'의 명령을 2회 입력하면 현 font에서 크게 변화하기를 원한다고 판단하여 현재 범위의 타당성 여부를 확인하도록 하였다. 현재 범위 이상 일 경우에는, 현재 화면에 나타난 글자 크기를 기준으로 처음 추론 범위로 멤버쉽 함수를 재구성하여 추론하도록 하였다.

2-2. 화면의 이동 속도에 대한 추론

화면의 이동 속도는 한 라인을 읽는데 걸리는 시간(second)으로 0.5초와 10초 사이에 멤버쉽 함수를 구성하였다.



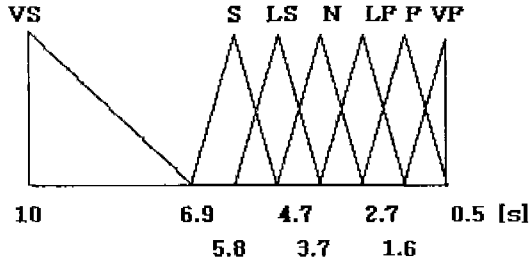
- VS(Very Slow) : 아주 느리게
- S(Slow) : 느리게
- LS(Little Slow) : 조금 느리게
- N(Normal) : 보 통(중 심)
- LF(Little Fast) : 조금 빠르게
- F(Fast) : 빠르게
- VF(Very Fast) : 아주 빠르게

<그림3. 화면 이동 속도에 대한 멤버쉽 함수>

멤버쉽 함수의 초기 상태가 그림 3에 보여지며, 사용자가 명령을 입력하기 전까지는

화면의 이동이 없다. 명령이 입력되면 추론에 의해 한 라인씩 이동하는 속도가 구해진다. 추론과정에서, 어떤 명령이 입력되어지면, 해당 멤버쉽 함수의 범위를 공유하는 인접한 멤버쉽 함수를 함께 고려하여 중심범위에 의해 추론하도록 하였다.

보통 글을 읽을 경우, 각 개인의 초기 글 읽는 속도에서 크게 변화시키지 않으므로, 그림 4에 보여지듯이 초기 상태에서 1차 명령 후 작은 쪽 범위를 3등분 한 범위로 각 멤버쉽 함수의 범위를 동일하게 구성함으로써 현재 속도의 느낌에서 미세한 속도의 변화가 가능하도록 하였다. 이런 추론 과정을 반복하여 사용자가 원하는 속도에 접근할 수 있도록 하였다. 사용자가 현재 이동하는 속도에서 반대 방향의 범위로 이동시킬 경우를 고려하여 VF 또는 VS 멤버쉽 함수는 최대, 최소치를 함수의 중심으로 두고 구성하였다.



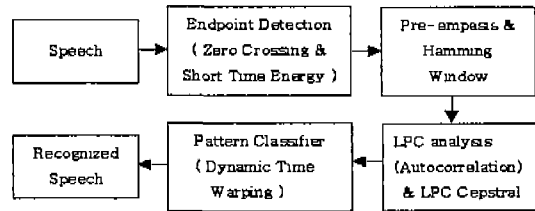
< 그림 4. 변화된 멤버쉽 함수 >

현재의 화면 이동 속도가 4.0~6.5초 사이일 경우, 느린 쪽과 빠른 쪽의 범위가 거의 비슷하기 때문에 각 멤버쉽 함수는 큰 범위를 가지게된다. 따라서 사용자의 요구에 의해 세밀하게 조절하기 위하여 VS와 VF를 제외한 나머지 멤버쉽 함수들은 4.0~6.5 범위 안에서 중심 값을 기준으로 위와 동일한 방법으로 멤버쉽 함수를 구성하였다.

3. 음성인식 알고리즘

명령에 대한 음성패턴의 특징을 추출하여 표준패턴을 만든 뒤 음성명령이 입력되면 저

장된 표준패턴과 비교하여 가장 유사한 패턴을 찾아 인식하는 패턴 매칭 방법을 사용하였다. 그림 5에서 전체적인 블록도가 보여진다.



< 그림 5 음성 인식 블록도 >

3-1 실시간에서의 음성구간 추출

실시간으로 음성 구간을 추출하기 위해 short time energy 와 zero-crossing 방법을 사용하였다.

3-2 LPC(Linear Prediction Coding)처리 과정

LPC 분석이란 과거 p개의 sample을 가지고 현재의 sample을 예측하는 방식이다. LPC 분석을 하기 전에 pre-emphasis를 처리함으로써 구간내의 미세한 파형을 강조시켰으며, Hamming windowing를 사용하여 분석구간의 음성 신호를 점차 감소시켜 신호의 잡음을 줄이도록 하였다. 특정 파라미터를 산출하는 과정에서 LPC계수를 얻는데 Durbin's method를 사용하였다.

3-3 Cepstrum

LPC Cepstrum으로 음성이 갖는 정보를 스펙트럼 포락 정보와 세부구조 정보를 분리시켰으며, tempered window 기법을 사용하여 cepstrum 계수의 민감도를 완화시켰다. 더 향상되고 확장된 스펙트럼의 표현을 위해 시차를 적용한 delta cepstrum을 사용하였다.

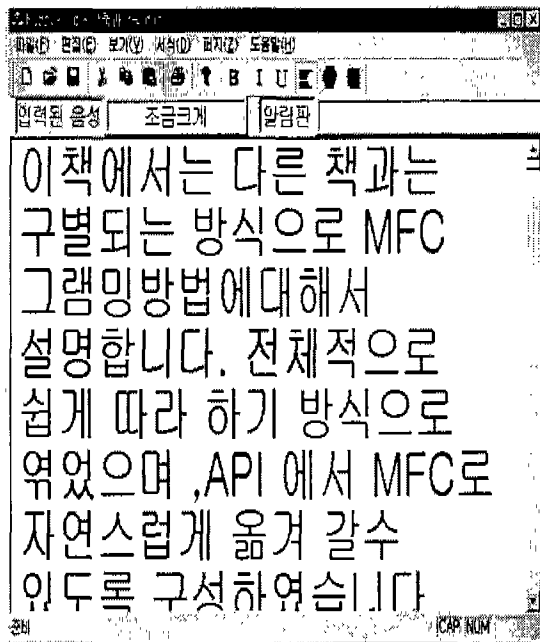
3-4 DTW 방식

DTW방식은 비 선형적인 시간축의 변동 패턴을 선형적으로 정규화 시키는 방식이다. 정규화 거리는 표준 패턴과 입력된 패턴 사이의 오차 거리로부터 계산되어진다. 따라서 DTW는 연속단어 보다는 고립단어에 많이 사용되며 고립단어에서 더욱 인식률이 높다. 본 시스템에서 사용되는 명령들은 짧은 단어 이기에 DTW방식을 사용하였다. 입력되어진

명령패턴과 표준패턴을 비교하여 오차 거리가 가장 작은 패턴을 해당 명령으로 인식한다.

4. 화면 구성 및 기능

인터페이스 화면은 그림 6에 보여지듯이 Windows에서 제공되는 워드패드 기능과 유사하게 문서를 작성할 수 있도록 구현하였다. 보여진 화면은 '조금 크게'라는 명령이 입력되어 글자크기가 변경된 화면이다.



<그림 6. 인터페이스 화면>

'font size', '속도' 명령에 의해 글자크기와 화면 이동속도를 구분하여 입력하도록 하였으며 '정지' 명령에 의해 움직이던 화면을 정지시키도록 하였다. 'page down', 'page up' 명령에 의해 문서를 page단위로 이동시켰으며, 메시지 박스에 대한 응답으로 '예', '아니오' 명령을 입력하도록 하였다.

처음 호출된 문서는 초기 값의 font size로 글자를 보여주며, 저장된 문서를 다시 불러왔을 때는 저장시킬 때의 글자 크기로 화면 상에 보여준다. '알림판' 창에는 경고나 특별 메시지를 사용자에게 알려주도록 하였다.

5. 결론 및 향후의 과제

본 논문에서는 음성으로 명령을 입력받아, 글자크기 및 화면의 이동 속도를 사용자의 요구에 따라 조절할 수 있는 문서 편집기를 구현하였다. 기존의 시스템에서 제공하는 마우스 또는 키보드를 대신하여 음성을 사용함으로써 각 개인의 시력에 따른 불편함 해소와 장애자들도 접근하기 쉬운 인터페이스를 구현하여 보았다.

이 후의 과제로서는 PC 사용자를 위한 사용자 인식 문제 및 시력 장애자를 위한 촉각을 이용한 인터페이스 구현 등이 고려되어야 하겠다.

6. 참고 문헌

- [1] 손영선 외 4명, "퍼지 추론에 의한 커서의 조작" 한국 퍼지 및 지능시스템 학회 1999년도 추계학술대회 학술논문발표논문집, Vol.9, No.2, pp. 239-242, 1999
- [2] 김주홍 외 3명, "제스처 및 음성 인식을 이용한 윈도우 시스템 제어에 관한 연구" 대한 전자공학회 1998년도 추계종합학술대회논문집, 11 v.21, n.2, pp. 1289-1292, 1998
- [3] 이근배 외 2명, "LVQ-HMM-FSN 기반의 음성 명령 결합 다중양식 한글문서편집기의 구현" 정보과학논문(C) = Journal of KISS(C): Computing Practices v.2, n.2, pp.206-217, 1996
- [4] 한영원 외 2명 "Windows95 환경에서의 음성 인터페이스 구현" 전자공학회 논문지-S, v.34-S, n.5, pp. 86-93, 1997
- [5] 田村博編, "휴먼 인터페이스", ohmsah, 1998
- [6] 오영환, "음성언어정보처리", 흥릉과학출판사, 1998
- [7] 本多中二, 大里有生, 최용엽 번역, "퍼지공학 입문", 웅보 출판사, 1999