

손동작을 이용한 운전 안전성을 높이기 위한 전화 다이얼 인터페이스 설계

장원앙*, 이도훈

*부산대학교 컴퓨터공학과

e-mail: {wonang, dohoon}@pusan.ac.kr

Dialing Interface Design for Safe Driving using Hand Gesture

WonAng Jang*, DoHoon Lee

*Dept of Computer Science & Engineering, Pusan National University

요 약

운전 중에 주의를 분산시키는 요소는 대부분 인터페이스 조작에 있으며 교통사고의 직접적인 원인이 된다. 스마트 자동차에 대한 관심이 높아지면서 운전자 안전에 대한 다양한 연구가 모색되고 있다. 순간의 시선이동으로 인해 판단력과 조작능력을 상실 할 수 있는 현재의 인터페이스는 안전성이 보장되지 못한다. 본 논문에서는 이러한 운전자의 주의를 분산시키는 요소로부터 안전성을 확보하기 위해서 차량 내 카메라를 이용하여 손동작을 인식하여 직관적인 제스처로 전화번호를 입력하거나 검색할 수 있는 안전한 인터페이스를 제안한다. 제안한 시스템은 직관적 동작과 TTS(Text To Speech)를 활용하여 사용자 편의성과 안전성을 높였다.

1. 서론

스마트폰이나 내비게이션 DMB 등이 대중화되면서 정보 활용 능력은 높아졌지만 조작을 위한 복잡함은 증가하고 있다. 이런 복잡한 요소는 운전자의 주의를 분산시켜 안전성에 문제를 야기할 수 있다. 주의를 분산시키는 행동으로는 운전 중 전화, 내비게이션 조작 등이 있으며 이런 행동들은 교통사고의 직접적인 원인이 될 수 있다.

이런 사고 원인의 중요한 요소가 운전자의 시선이다. 운전자는 정면을 주시하면서 안전운전과 방어운전을 병행해야하는데 스마트폰이나 내비게이션을 조작하다보면 시선을 빼앗기게 된다. 시선을 빼앗긴 운전자는 그 짧은 순간에 판단력과 조작능력을 상실할 수 있으며 이런 상황은 매우 위험한 결과를 초래할 수 있다.

운전 중 안전한 전화를 위해 마이크와 스피크의 도움을 받는 핸즈프리를 이용한다. 핸즈프리는 전화를 수신할 때는 별다른 입력이 없기 때문에 시선을 빼앗기지 않지만 송신 할 때는 전화번호부에서 저장된 전화번호를 찾거나 직접 번호를 입력해야 하기 때문에 시선을 빼앗길 수 있다.

스마트 자동차에 대한 관심이 높아지면서 안전을 위한 연구가 병행되고 있다. 손동작 인식과 이를 활용한 이메일 시스템을 제어하기 위한 방법을 제안하거나[1, 2] 상업적 컴퓨터 게임이나 교육적인 환경에서 응용되기도 하였다[3]. 또한 손동작은 신체적, 정신적 장애가 있는 사람이나 고령자의 안전을 위한 연구와 음성인식과 결합하여 응용된 결과가 발표되었다[4, 5]. 그 외 손동작을 이용한 TV

컨트롤 시스템과 메시지에 응용한 사례와[6, 7] 손동작 자체의 인식률에 대한 연구도 활발하며 Deterministic Boosting을 이용한 실시간 손동작 인식에 대한 연구가 있다[8].

본 논문은 스마트 자동차 기능의 하나인 손동작 인식을 통해 안전성을 높일 수 있는 다이얼 인터페이스를 제안한다. 운전자는 차량 내 카메라를 부착하여 전방을 주시하면서 핸들을 잡지 않은 다른 손의 제스처를 통해 명령을 내릴 수 있으며 차량 내 장비들과 상호작용 할 수 있다. 즉 제시하는 인터페이스는 자동차와 의사소통이 가능하며 입력은 운전자의 손동작이고 출력은 TTS로 알려준다.

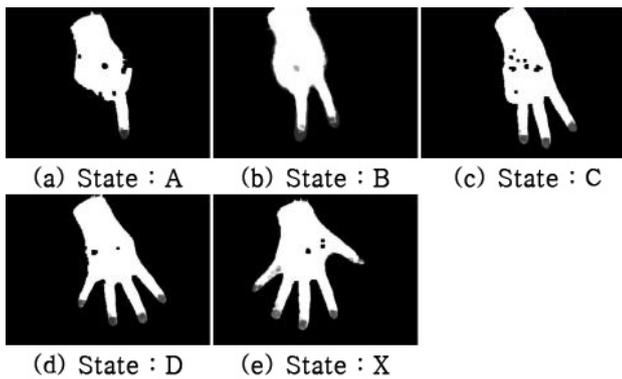
2. 설계

손동작을 본 논문에서 State라고 정의한다. State는 차량 내 카메라를 이용하여 손가락 끝을 검출하여 5가지로 나눈다. Action은 State의 순서 열로 구성되며 2개 혹은 3개의 State를 조합하여 표현한다. 각각의 상황에 따라 다양한 Action이 있으며 직관적이게 구성하였다. 즉 따로 배우지 않아도 명령에 해당하는 손동작을 쉽게 표현하여 전화번호를 입력할 수 있으며 전화번호 목록에서 원하는 번호를 선택할 수도 있다.

TTS는 미리 입력해놓은 문장을 컴퓨터에서 음성신호로 변환하여 오디오로 출력해주는 기술을 말하며 운전자의 손동작이 정상적으로 인식되었는지 여부와 명령이 성공적으로 수행되었는지 여부를 알려준다. 또한 한 State를 입력하고 다음 State를 입력하기 전에 미리 조합 가능한

State를 음성으로 안내하여 불필요한 정보를 줄이고 정확한 명령을 입력할 수 있게 도와주는 Action 및 State 완료 상태 정보를 제공한다.

차량 내 부착된 카메라로부터 영상처리를 거쳐 손 끝점을 추출한 뒤 그림 1과 같이 5개의 State를 정의한다. State는 A, B, C, D, X로 구성되며 A는 그림 1의 (a)와 같이 손가락을 하나 펼친 동작이다. B는 그림 1의 (b)와 같이 손가락을 2개 펼친 동작이며 C와 D도 동일하게 구성된다. 그림 1의 (e)에 표현된 5개의 손가락을 펼친 손동작은 State X로 구분하며 A, B, C, D와는 다른 용도로 사용한다.



(그림 1) 제스처를 구성하는 5개의 State. (a)부터 (e)까지는 각각 1부터 5를 의미한다.

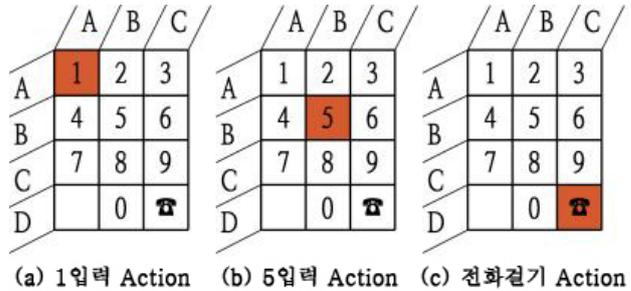
Action은 식 1에서와 같이 State의 순서 열로 2개 혹은 3개의 State를 조합하여 구성한다. 즉 손동작을 연속해서 2번 혹은 3번 입력해야 하나의 Action이 완성되며 하나의 Action은 하나의 명령을 수행한다. Action의 순서 열은 5가지 State를 3번 연속해서 표현할 수 있는 경우이므로 5^3 인 125가지로 조합 가능하다. Action의 순서 열을 행렬로 구성하고 행렬의 인덱스를 State로 표현하여 State들의 조합으로 Action을 정의한다. 즉 1, 2, 3을 표시할 수 있을 정도의 직관적 상태를 선택했다. 인터페이스 설계에서는 전화걸기에 주목적이 있으며 전화 걸기는 전화번호를 직접 입력하는 것과 전화번호부에서 검색하는 것으로 한정한다. 즉 직접 입력과 전화번호 검색으로 두 가지 과정으로 구분한다. 일반적으로 Action은 State의 순서열로 다음과 같이 정의한다.

$$Action = \begin{cases} \langle State1, State2 \rangle, & \text{State가 2개일때} \\ \langle State1, State2, State3 \rangle, & \text{State가 3개일때} \end{cases}$$

2.1 직접입력 인터페이스 설계

전화번호 집적 입력은 그림 2와 같이 미리 정의된 Action으로 전화번호를 하나씩 입력할 수 있는 제스처의 순서 열이다. 직접입력에는 0부터 9까지 숫자의 입력에 대한 10가지 Action과 전화걸기, 1개 번호 삭제, 2개 번호 삭제, 취소에 대한 4가지 Action으로 총 14개의 Action이 정의되어 있으며 가장 인식하기 쉬운 형태로 구성한다. 전

화번호를 입력해야 하기 때문에 기존의 휴대전화나 유선 전화의 숫자패드 디자인을 그대로 행렬로 표현한다. 예를 들어 전화번호 1을 입력하고 싶을 경우 Action은 그림 2의 (a)와 같이 <State A, State A>로 구성된다. 숫자 5를 입력하고 싶은 경우 그림 2의 (b)와 같이 <State B, State B>를 연속적으로 입력하면 된다. 전화번호 입력이 모두 끝난 뒤 전화를 걸 때는 그림 2의 (c)와 같이 <State D, State C>를 연속적으로 입력하면 된다.



(그림 2) 직접입력 예. 흔히 사용하는 휴대전화나 유선 전화의 키패드의 배치를 그대로 행렬로 표현하였기 때문에 직관적이다. 2개의 State Sequence를 입력하면 행렬에서 색으로 표현된 Action이 선택된다.

전화번호 집적 입력은 사용자가 잘못 입력하거나 다른 State로 인식할 수 있기 때문에 입력뿐만 아니라 삭제 혹은 종료에 해당하는 Action도 정의해야 할 필요가 있다. 그 Action을 Action_op라 하고 식 1과 같이 정의한다. 연속해서 해당하는 2개의 State를 입력하면 명령을 수행한다.

$$Action_op = \begin{cases} \langle StateX, StateA \rangle, & \text{1개 번호 삭제} \\ \langle StateX, StateB \rangle, & \text{2개 번호 삭제} \\ \langle StateX, StateX \rangle, & \text{종료} \end{cases}$$

(식 1)

2.2 전화번호 검색 인터페이스 설계

전화번호 검색은 그림 3에서와 같이 운전자의 휴대전화에 저장된 전화번호를 검색할 수 있는 제스처의 순서 열이다. 전화번호 검색은 최소한의 Action으로 원하는 항목을 검색할 수 있어야 하며 목록의 상, 하 이동이 가능해야 한다. 한글은 자음과 모음의 조합에 따라 정렬이 가능하며 그러한 특징을 이용하며 효율적인 검색을 위해서 한글의 자음을 검색할 수 있는 Action들로 행렬을 구성한다. 그림 3의 (a)와 같이 <State A, State A>로 구성된 Action은 전화번호부 상에 ㄱ(기역)으로 시작하는 항목 중 제일 첫 번째 것을 검색하여 선택한다. 그림 3의 (b)와 같이 <State B, State C>로 구성된 Action은 ㅅ(시옷)으로 시작하는 항목 중 제일 첫 번째 것을 검색하여 선택하며 마찬가지로 그림 3의 (c)와 같이 <State D, State C>로 구성된 Action은 현재 선택된 항목의 한 단계 아래에 있는 항목을 선택하게 된다. 원하는 항목을 성공적으로 검

색한 뒤 그림 3의 (d)와 같이 <State X, State C>로 구성된 Action은 전화를 건다.

	A	B	C	D
A	가	나	다	라
B	마	바	사	아
C	자	차	카	타
D	파	하	↑	↓
X	-	-	☎	End

(a) 가 검색 Action (b) 사 검색 Action (c) Down Action (d) 전화걸기 Action

(그림 3) 전화번호 검색의 행렬표현

3. 구현

인터페이스의 구현은 크게 하드웨어와 소프트웨어 부분이 있다. 하드웨어는 Windows 7이 설치된 펜티엄4 PC와 웹캠이 사용되었다. 시뮬레이션 환경에서 PC는 처리기 역할을 하고 웹캠은 차량 내에 지정된 위치에 설치되어 사용자의 손동작을 실시간으로 입력받게 된다. 소프트웨어는 OPENCVSharp, Interop.SpeechLib 2가지 라이브러리가 사용되었으며 개발언어는 Visual Studio 2008 C#이다. OPENCVSharp은 C++ 라이브러리인 OPENCV를 C#에서 동작할 수 있게 래핑된 라이브러리이며 2.3.1버전으로 구현하였다. OPENCVSharp은 그림 5의 D_1 과 같이 웹캠으로 부터 입력받은 운전자의 손 이미지를 실시간으로 처리하여 손끝을 추출하는데 사용된다.

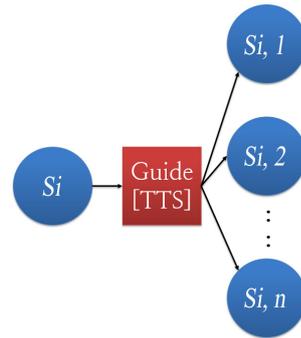
설계된 인터페이스의 안전한 조작을 위해 시선의 전방 주시를 보장하는 기능이 필요하였다. 각 Action이 완료되었음을 인터페이스를 보면서 확인하는 대신, 소리로 인지할 수 있다면 안전성이 크게 개선됨을 착안하여 TTS기능을 첨가하였다. TTS는 미리 입력해놓은 문장을 컴퓨터에서 음성신호로 변환하여 오디오로 출력해주는 기술을 말하며 운전자의 손동작이 정상적으로 인식되었는지 여부와 명령이 성공적으로 수행되었는지 여부를 알려준다. 또한 한 State를 입력하고 다음 State를 입력하기 전에 미리 조합 가능한 State를 음성으로 안내하여 불필요한 정보를 줄이고 정확한 명령을 입력할 수 있게 도와주는 Action 및 State 완료 상태 정보를 제공한다.

TTS는 운전자와 핸즈프리 사이에서 원활하고 안전한 의사소통을 보장해 준다. 2개 혹은 3개의 State를 연속적으로 조합해야 하나의 명령으로 인정되기 때문에 입력하고자하는 State가 정상적으로 인식되었는지, 그리고 최종적으로 조합된 Action이 정확하게 입력되었는지 여부를 사용자에게 인지토록 한다. 즉 제안한 인터페이스는 운전자가 명령을 입력만 하는 단방향 통신이 아니라 시스템과의 의사소통이 가능한 양방향 통신이다.

예를 들어, 그림 2의 (a)에 해당하는 Action을 입력하고자 할 때 운전자는 2개의 State를 입력하게 된다. 이해하기 쉽게 첫 번째 손동작, 두 번째 손동작이라고 할 때 첫 번째 손동작은 State A이고 두 번째 손동작은 State A가 된다. 운전자가 첫 번째 손동작을 입력 한 후 인터페이스

가 음성으로 인식의 성공여부를 안내하면 첫 번째 입력한 손동작이 정상적으로 입력되었는지, 언제 두 번째 State를 입력해야하는지 여부를 쉽게 알아차릴 수 있다. 또한 그림 4에서와 같이 첫 번째 손동작의 입력 후에 조합 가능한 두 번째 손동작을 추려서 음성으로 안내하면 입력해야 할 경우의 수가 줄어들게 된다. 음성으로 안내할 때 조합 가능한 State와 더불어 조합된 Action에 대한 정보까지 음성으로 안내하여 훨씬 이해하기 쉽고 직관적이다.

Interop.SpeechLib는 TTS를 위한 라이브러리며 미리 입력해놓은 문장을 음절단위로 처리하고 음성으로 합성하여 오디오로 출력해주는 역할을 한다.

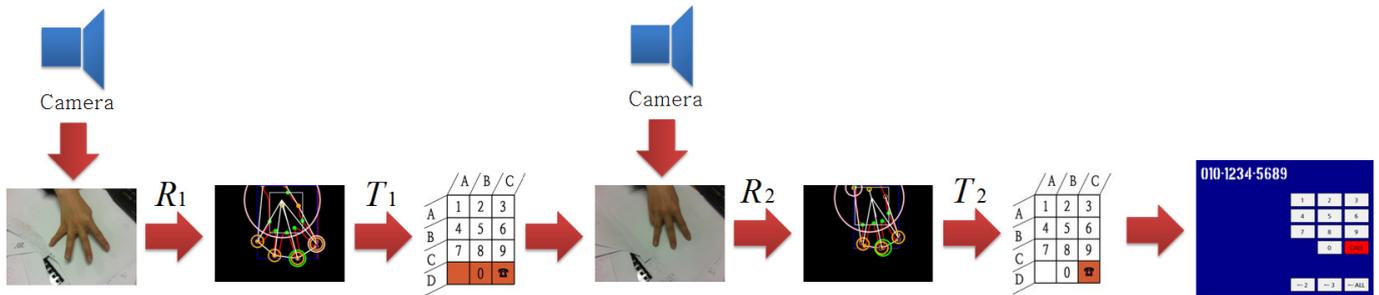


(그림 4) TTS의 발생 시점을 보여주는 그림. State (S_i)와 State($S_{i,1}, \dots, S_{i,n}$) 사이에 안내 TTS를 통해 예측 동작을 안내하여 편의성을 높임.

카메라에서 입력받은 손동작을 인식하는 임의의 단계를 R_i 라 한다. R_1 에서 각종 영상처리를 거쳐 5가지 State중 한가지로 인식된다. 인식이 완료되면 TTS로부터 안내하는 임의의 한 과정을 T_i 라 하고 첫 번째 T_1 단계에서 인식결과와 함께 Action으로 조합 가능한 State를 알려주고 2번째 입력을 기다린다. 운전자가 두 번째 State를 입력하면 R_2 에서 5가지 State중 한가지로 인식되고 T_2 에서 완성된 Action을 운전자에게 알려준 뒤 미리 정의된 행렬과 비교한다. 미리 설정된 프로세스에 따라 Action과 행렬이 비교되며 행렬에 매핑되는 Action이 존재하면 TTS로 인식 결과를 안내한 후 명령을 실행한다. 정상적으로 명령이 수행되면 다시 입력 대기 상태로 돌아간다.

4. 결론 및 향후 과제

본 논문에서는 운전자의 주의를 분산시키는 요소로 부터 안전성을 높여주는 전화 다이얼 인터페이스를 제안했다. 카메라를 이용하여 사용자의 손동작을 인식하고 2번 혹은 3번으로 연속된 손동작의 순서열로 전화번호를 직접 입력하거나 전화번호부를 검색할 수 있었다. 손동작은 State로 표현하며 입력된 영상으로부터 손끝을 추출하여 인식된 결과를 A, B, C, D, X 5가지 상태로 구분했다. 손동작의 순서열은 Action으로 표현하며 하나의 Action은 하나의 명령으로 표현했다. 손동작을 이용한 전화걸기에 주목적이 있으며 전화 걸기는 전화번호를 직접 입력하는



(그림 5) 구현된 인터페이스의 흐름도. 카메라에서 입력된 운전자의 손동작은 각종 인식단계를 거쳐 핸드프리의 전화번호를 입력하거나 전화번호부를 검색할 수 있는 명령이 된다. 그 과정에서 TTS를 통해 인터페이스와 운전자가 상호작용이 가능하다. R 은 인식과정이고 T 는 동작인식을 인지해주는 단계이다.

것과 전화번호부에서 검색하는 것으로 한정 했다. 전화 걸기는 전화번호의 숫자패드를 입력하듯이 0부터 9까지의 숫자를 손동작으로 입력할 수 있으며 전화번호부 검색은 가부터 하까지 14가지의 한글 자음을 손동작으로 검색할 수 있었다. 각 Action이 완료되었음을 인터페이스를 보면서 확인하는 대신, 소리로 인지할 수 있다면 안전성이 크게 개선됨을 착안하여 TTS기능을 사용했다. 운전자는 직관적인 제스처를 통해 시선을 빼앗기지 않을 수 있고 운전자가 전방을 주시할 수 있도록 도와주어 교통사고예방에 기여 할 수 있을 것이라 기대한다.

인터페이스의 성능은 카메라의 화소와 손동작의 인식률에 따라 결정 된다. 카메라의 화소가 좋으면 고해상도로 촬영된 영상에서 원하는 특징을 뽑아내어 결과물을 출력할 수 있지만 처리하는데 시간이 많이 들고 단가가 올라간다. 손동작의 인식은 카메라 입력중 살색영역을 추출하여 가능하며 색온도와 광량에 민감하다. 색온도가 높은 광원은 피사체를 파란색 계열로 비추고 색온도가 낮은 광원은 피사체를 붉은색 계열로 비추게 된다. 그리고 차광내부는 광량이 부족하기 때문에 정확한 살색 영역의 추출이 어렵다. 손동작의 인식문제를 개선하기 위해서는 인체의 Skeleton을 추적하여 손목이상의 영역으로부터 손동작을 추출하거나 Depth Map을 이용하여 특정 범위안에 움직이는 객체를 추적하여 손동작으로 인식하는 연구가 이루어져야 할 것이다. 또한 손동작 인식에 대한 새로운 알고리즘이나 라이브러리가 개발된다면 운전 중이더라도 제스처를 통해 안전하게 각종 조작을 할 수 있을 것이다.

참고문헌

- [1] Christoph Endres and Svilen Dimitrov "Using a Theremin for Micro-Gesture Recognition in an Automotive Environment", Proc. of The 2nd Int. Conf. on Automotive User Interfaces and Interactive Vehicular, Pennsylvania, USA, November 11-12, 2010
- [2] Andreas Riemer, Michael Rossbory and Alois Ferscha, "Natural DVI based on intuitive hand gestures", Springer LNCS, pp. 62-66, September 2011.
- [3] J. Bernardes, R. Nakamura and R. Tori,

"Comprehensive Model and Image-Based Recognition of Hand Gestures for Interaction in 3D Environments" Proc. of The International Journal of Virtual Reality, Vol. 10(4), pp. 11-23, 2011

- [4] Dimitra Anastasiou "Gestures in assisted living environments", Proc. of the 9th International Gesture Workshop, May 25-27, Athens, Greece, 2011
- [5] Nutan D Sonwane, Sharda Chhabria and Dr.R.V.Dharaskar, "Combine Approach for Speech and Gesture Recognition" International Journal of Computer Applications(IJCA), Vol. pp 16--19, 2012
- [6] 손명규, 이상현, 김병민, 이장우, 박지호 "손 동작 인식을 이용한 인터랙티브 TV 컨트롤 시스템(ITCS)의 설계 및 구현", 한국정보과학회 한국컴퓨터 학술발표논문집 제37권 제2호(C), Vol. pp 207--2011, 2010
- [7] 이원주 "웹캠을 이용한 동적 제스처 인식 기반의 감성메신저 구현 및 성능 분석", 한국컴퓨터정보학회논문지 제15권제7호, pp. 75--81 2010
- [8] Raymond Lockton and Andrew W. Fitzgibbon "Real-time gesture recognition using deterministic boosting" Proc. of British Machine Vision Conference, pp 817--826, 2002