



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0102260
(43) 공개일자 2012년09월18일

(51) 국제특허분류(Int. Cl.)

G06F 3/02 (2006.01) H04B 1/40 (2006.01)

(21) 출원번호 10-2011-0020283

(22) 출원일자 2011년03월08일

심사청구일자 없음

(71) 출원인

삼성전자주식회사

경기도 수원시 영통구 삼성로 129 (매탄동)

서강대학교산학협력단

서울특별시 마포구 백범로 35 (신수동, 서강대학교)

(72) 발명자

오준섭

경기도 수원시 권선구 효원로256번길 7, 409호 (권선동, 지오베르크)

박영희

서울특별시 성북구 북악산로15길 24-15 (돈암동)
(뒷면에 계속)

(74) 대리인

이정순, 권혁록

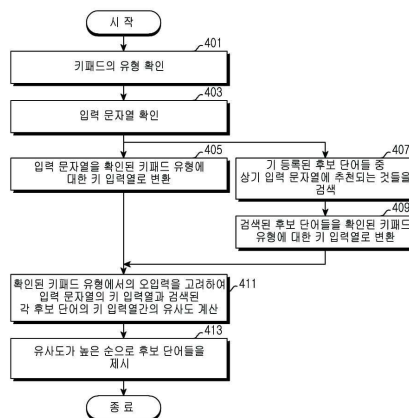
전체 청구항 수 : 총 28 항

(54) 발명의 명칭 휴대용 단말기에서 오입력을 방지하기 위한 방법 및 장치

(57) 요약

본 발명은 휴대용 단말기에서 오입력을 방지하기 위한 방법을 제공하고, 상기 방법은 입력 문자열을 확인하고 등록된 후보 단어들 중 상기 입력 문자열에 추천되는 것들을 검색하는 단계와, 상기 입력 문자열과 검색된 각 후보 단어 간의 유사도를 산출하는 단계 및 유사도가 높은 순으로 검색된 후보 단어들을 제시하는 단계를 포함하는 것을 특징으로 한다.

대표도 - 도4



(72) 발명자

서정연

서울특별시 마포구 백범로 35 (신수동)

김학수

강원도 춘천시 효자2동 192-1

송영길

강원도 춘천시 효자2동 192-1

특허청구의 범위

청구항 1

휴대용 단말기에서 오입력을 방지하기 위한 방법에 있어서,
입력 문자열을 확인하고 기 등록된 후보 단어들 중 상기 입력 문자열에 추천되는 것들을 검색하는 단계;
상기 입력 문자열과 검색된 각 후보 단어 간의 유사도를 산출하는 단계 및
유사도가 높은 순으로 검색된 후보 단어들을 제시하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 2

제 1항에 있어서,
상기 입력 문자열을 확인하고 기 등록된 후보 단어들 중 상기 입력 문자열에 추천되는 것들을 검색하는 단계는,
상기 입력 문자열을 자소 단위로 분리한 후 이와 일정 이상 일치하는 것들을 기 등록된 후보 단어들 중에서 검색함을 특징으로 하는 방법.

청구항 3

제 1항에 있어서,
상기 입력 문자열과 검색된 각 후보 단어 간의 유사도를 산출하는 단계는,
키패드 유형을 확인하는 단계;
확인된 키패드 유형에 상응하게 삽입(insert) 연산, 삭제(delete) 연산 및 치환(replace) 연산의 각 비용(값)이 결정되는 단계 및
상기 각각의 비용이 고려된 연산들을 이용하여 편집 거리(Edit Distance)를 산출하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 4

제 3항에 있어서,
상기 각각의 비용이 고려된 연산들을 이용하여 편집 거리(Edit Distance)를 산출하는 단계는,
상기 입력 문자열과 검색된 후보 단어들을 확인된 키패드 유형에 대한 키 입력열로 변환하는 단계와,
상기 입력 문자열의 키 입력열과 검색된 각 후보 단어의 키 입력열 간의 편집 거리를 산출하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 5

제 4항에 있어서,
상기 입력 문자열과 검색된 후보 단어들을 확인된 키패드 유형에 대한 키 입력열로 변환하는 단계는,
확인된 키패드 유형의 각 키에 부여된 코드를 확인하는 단계 및
상기 입력 문자열 및 후보 단어들이 입력되기 위하여 각 키가 일련의 순서로 선택되어지는 것을 코드열로 나타내는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 6

제 3항에 있어서,

유사도가 높은 순으로 후보 단어들을 제시하는 단계는,

산출한 편집 거리가 작은 것부터 큰 순서로 해당 후보 단어들을 제시함을 특징으로 하는 방법.

청구항 7

제 4항에 있어서,

상기 확인된 키패드 유형에 상응하게 삽입(insert) 연산, 삭제(delete) 연산 및 치환(replace) 연산의 각 비용(값)이 결정되는 단계는,

상기 확인된 키패드 유형에서 발생빈도가 높은 적어도 하나 이상의 오입력 유형을 확인하는 단계;

발생빈도가 높은 적어도 하나 이상의 오입력 유형의 연산 조건이 만족되면 삽입 연산 또는 삭제 연산 또는 치환 연산의 비용을 제1값으로 결정하고, 만족하지 않으면 제1값보다 큰 제2값으로 결정하는 단계 및

발생빈도가 낮은 오입력 유형의 연산 조건 만족 여부에 상관없이 삽입 연산 또는 삭제 연산 또는 치환 연산의 비용을 기본 비용으로 결정하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 8

제 7항에 있어서,

상기 오입력 유형은,

연속 키 눌림에 따른 오입력, 인접 키 눌림에 따른 오입력, 키 생략에 따른 오입력 및 키 입력 순서에 따른 오입력 중 어느 하나임을 특징으로 하는 방법.

청구항 9

제 8항에 있어서,

발생빈도가 높은 적어도 하나 이상의 오입력 유형의 연산 조건이 만족되면 삽입 연산 또는 삭제 연산 또는 치환 연산의 비용을 제1값으로 결정하고, 만족하지 않으면 제1값보다 큰 제2값으로 결정하는 단계는,

연속 키 눌림에 따른 오입력의 연산 조건이 만족되면 삽입 연산의 비용을 1로 적용하고 그렇지 않으면 3을 적용하거나,

키 생략에 따른 오입력의 연산 조건이 만족되면 삭제 연산의 비용을 1로 적용하고 그렇지 않으면 3을 적용하거나,

키 입력 순서에 따른 오입력의 연산 조건이 만족되면 치환 연산의 비용을 1로 적용하고 그렇지 않으면 3을 적용하거나,

인접 키 눌림에 따른 오입력의 연산 조건을 만족하되 오입력의 가능성이 높으면 1 낮으면 2를 적용하고, 그렇지 않으면 3을 적용함을 특징으로 하는 방법.

청구항 10

제 9항에 있어서,

상기 키패드 유형은,

쿼티(QWERTY) 키패드, 콤비네이션(Combination) 키패드 및 팝업(Pop-up) 키패드 중 어느 하나임을 특징으로 하는 방법.

청구항 11

제 10항에 있어서,

상기 쿼티 키패드는,

인접 키 눌림에 따른 오입력 및 키 입력 순서에 따른 오입력의 발생 빈도가 높음을 특징으로 하는 방법.

청구항 12

제 10항에 있어서,

상기 콤비네이션 키패드는,

연속 키 눌림에 따른 오입력, 인접 키 눌림에 따른 오입력, 키 생략에 따른 오입력 및 키 입력 순서에 따른 오입력의 발생 빈도가 높음을 특징으로 하는 방법.

청구항 13

제 10항에 있어서,

상기 팝업 키패드는,

인접 키 눌림에 따른 오입력, 키 입력 순서에 따른 오입력의 발생 빈도가 높음을 특징으로 하는 방법.

청구항 14

제 9항에 있어서,

상기 입력 문자열의 키 입력열($s[1..i]$)과 검색된 각 후보 단어의 키 입력열($t[1..j]$) 간의 편집 거리를 산출하는 단계에서,

상기 연속 키 눌림에 따른 오입력의 연산 조건은 $s_{i-1} == s_i \& s_i == t_j$ 이고,

키 생략에 따른 오입력의 조건은 $s_i == t_{j+1} \& s_i == t_j$ 이며,

키 입력 순서에 따른 오입력의 조건 $s_i s_{i+1} == t_{j+1} t_j$ 임을 특징으로 하는 방법.

청구항 15

휴대용 단말기에서 오입력을 방지하기 위한 장치에 있어서,

입력을 위한 입력부와, 음성 또는 영상 출력을 위한 출력부와, 데이터를 저장하는 저장부와, 통신을 위한 통신부 및 전반적인 동작의 제어를 수행하는 제어부를 포함하되,

상기 제어부는,

입력 문자열을 확인한 후 기 등록된 후보 단어들 중 상기 입력 문자열에 추천되는 것들을 검색하고, 상기 입력 문자열과 검색된 각 후보 단어 간의 유사도를 산출하여 유사도가 높은 순으로 검색된 후보 단어들을 제시함을 특징으로 하는 장치.

청구항 16

제 15항에 있어서,

상기 제어부는,

입력 문자열을 확인한 후 기 등록된 후보 단어들 중 상기 입력 문자열에 추천되는 것들을 검색하는 경우, 상기 입력 문자열을 자소 단위로 분리한 후 이와 일정 이상 일치하는 것들을 기 등록된 후보 단어들 중에서 검색함을 특징으로 하는 장치.

청구항 17

제 15항에 있어서,

상기 제어부는,

상기 입력 문자열과 검색된 각 후보 단어 간의 유사도를 산출하는 경우,

키패드 유형을 확인하고, 확인된 키패드 유형에 상응하게 삽입(insert) 연산, 삭제(delete) 연산 및 치환(replace) 연산의 각 비용(값)을 결정하며, 상기 각각의 비용이 고려된 연산들을 이용하여 편집 거리(Edit Distance)를 산출함을 특징으로 하는 장치.

청구항 18

제 17항에 있어서,

상기 제어부는,

상기 각각의 비용이 고려된 연산들을 이용하여 편집 거리(Edit Distance)를 산출하는 경우,

상기 입력 문자열과 검색된 후보 단어들을 확인된 키패드 유형에 대한 키 입력열로 변환한 후 상기 입력 문자열의 키 입력열과 검색된 각 후보 단어의 키 입력열 간의 편집 거리를 산출함을 특징으로 하는 장치.

청구항 19

제 18항에 있어서,

상기 제어부는,

상기 입력 문자열과 검색된 후보 단어들을 확인된 키패드 유형에 대한 키 입력열로 변환하는 경우,

확인된 키패드 유형의 각 키에 부여된 코드를 확인하고, 상기 입력 문자열 및 후보 단어들이 입력되기 위하여 각 키가 일련의 순서로 선택되어지는 것을 코드열로 변환함을 특징으로 하는 장치.

청구항 20

제 17항에 있어서,

상기 제어부는,

유사도가 높은 순으로 후보 단어들을 제시하는 경우,

산출한 편집 거리가 작은 것부터 큰 순서로 해당 후보 단어들을 제시함을 특징으로 하는 장치.

청구항 21

제 18항에 있어서,

상기 제어부는,

상기 확인된 키패드 유형에 상응하게 삽입(insert) 연산, 삭제(delete) 연산 및 치환(replace) 연산의 각 비용(값)을 결정하는 경우,

상기 확인된 키패드 유형에서 발생빈도가 높은 적어도 하나 이상의 오입력 유형을 확인한 후 상기 발생빈도가 높은 적어도 하나 이상의 오입력 유형의 연산 조건이 만족되면 삽입 연산 또는 삭제 연산 또는 치환 연산의 비용을 제1값으로 결정하고 그렇지 아니하면 제1값보다 큰 제2값으로 결정하고,

발생빈도가 낮은 오입력 유형의 연산 조건 만족 여부에 상관없이 삽입 연산 또는 삭제 연산 또는 치환 연산의 비용을 기본 비용으로 결정함을 특징으로 하는 장치.

청구항 22

제 21항에 있어서,

상기 오입력 유형은,

연속 키 눌림에 따른 오입력, 인접 키 눌림에 따른 오입력, 키 생략에 따른 오입력 및 키 입력 순서에 따른 오입력 중 어느 하나임을 특징으로 하는 장치.

청구항 23

제 22항에 있어서,

상기 제어부는,

발생빈도가 높은 적어도 하나 이상의 오입력 유형의 연산 조건이 만족되면 삽입 연산 또는 삭제 연산 또는 치환 연산의 비용을 제1값으로 결정하고, 만족하지 않으면 제1값보다 큰 제2값으로 결정하는 경우,

연속 키 눌림에 따른 오입력의 연산 조건이 만족되면 삽입 연산의 비용을 1로 적용하고 그렇지 않으면 3을 적용하거나,

키 생략에 따른 오입력의 연산 조건이 만족되면 삭제 연산의 비용을 1로 적용하고 그렇지 않으면 3을 적용하거나,

키 입력 순서에 따른 오입력의 연산 조건이 만족되면 치환 연산의 비용을 1로 적용하고 그렇지 않으면 3을 적용하거나,

인접 키 눌림에 따른 오입력의 연산 조건을 만족하되 오입력의 가능성이 높으면 1 낮으면 2를 적용하고, 그렇지 않으면 3을 적용함을 특징으로 하는 장치.

청구항 24

제 23항에 있어서,

상기 키패드 유형은,

쿼티(QWERTY) 키패드, 콤비네이션(Combination) 키패드 및 팝업(Pop-up) 키패드 중 어느 하나임을 특징으로 하

는 장치.

청구항 25

제 24항에 있어서,

상기 쿼티 키패드는,

인접 키 눌림에 따른 오입력 및 키 입력 순서에 따른 오입력의 발생 빈도가 높음을 특징으로 하는 장치.

청구항 26

제 24항에 있어서,

상기 콤비네이션 키패드는,

연속 키 눌림에 따른 오입력, 인접 키 눌림에 따른 오입력, 키 생략에 따른 오입력 및 키 입력 순서에 따른 오입력의 발생 빈도가 높음을 특징으로 하는 장치.

청구항 27

제 24항에 있어서,

상기 팝업 키패드는,

인접 키 눌림에 따른 오입력, 키 입력 순서에 따른 오입력의 발생 빈도가 높음을 특징으로 하는 장치.

청구항 28

제 23항에 있어서,

상기 제어부가 상기 입력 문자열의 키 입력열($s[1..i]$)과 검색된 각 후보 단어의 키 입력열($t[1..j]$) 간의 편집 거리를 산출하는 경우,

상기 연속 키 눌림에 따른 오입력의 연산 조건은 $s_{i-1} == s_i \& s_i == t_j$ 이고,

키 생략에 따른 오입력의 조건은 $s_i == t_{j+1} \& s_i == t_j$ 이며,

키 입력 순서에 따른 오입력의 조건 $s_i s_{i+1} == t_{j+1} t_j$ 임을 특징으로 하는 장치.

명세서

기술분야

[0001] 본 발명은 휴대용 단말기에서 오입력을 방지하기 위한 방법 및 장치 관한 것으로서, 특히, 오입력에 따른 유사 단어를 제시하는 방법 및 장치에 관한 것이다.

배경기술

[0002] 현재 전자 통신 산업의 발달로 말미암아 이동 통신 단말기(셀룰러폰), 전자수첩, 복합 단말기 등의 휴대용 단말

기는 현대 사회의 필수품이 되어가면서, 빠르게 변화하는 정보 전달의 중요한 수단이 되고 있다.

[0003] 일반적으로 상기 휴대용 단말기는 입력 수단인 키패드를 구비한다. 주지하다시피, 휴대용 단말기는 점차 소형화되고 있고, 이러한 키패드를 제한된 공간에 구비하기 어려워지고 있다. 또한, 이러한 제한된 공간에 구비된 키패德拉 할지라도 키의 크기가 작거나 키 간격이 조밀하여 의도하지 않은 오입력이 심심찮게 발생하는게 사실이다. 결국 사용자는 오입력을 줄이기 위하여 신중할 수밖에 없으며, 이는 사용자에게 부담을 주게 된다. 일반적으로 이러한 문제를 해결하기 위하여 오입력된 단어와 유사한 단어들을 제시하는 방법이 제안되고 있다. 기존의 이러한 방법은 입력된 문자열 기 등록되어 있는 올바른 문자열과 비교하는 것으로서 문자적 형태에 의존적이라는 단점이 있다. 게다가, 키패드의 유형에 따라 오입력되는 상황이 다르므로 이에 유연하게 유사한 단어를 제시할 수 있는 방안이 모색된다. 일 예로, 콤비네이션 키패드는 하나의 키에 두 가지 이상의 자소가 할당되어 있고 키를 누를 때마다 자소가 바뀌어 입력되는 방식으로, 공지된 쿼터 키패드에 비해 키의 수가 적기 때문에 인접하는 키가 오입력되는 경우보다는 키가 눌러지는 횟수에 따른 오입력이 자주 발생한다.

발명의 내용

해결하려는 과제

[0004] 본 발명의 목적은 휴대용 단말기에서 사용자의 입력을 편의하게 하기 위한 방법 및 장치를 제공하는데 있다.

[0005] 본 발명의 다른 목적은 휴대용 단말기에서 오입력을 줄이기 위한 방법 및 장치를 제공하는데 있다.

[0006] 본 발명의 또 다른 목적은 휴대용 단말기에서 키패드별 오입력의 유형을 고려하여 오입력된 단어와 유사한 단어를 제시하는 방법 및 장치를 제공하는데 있다.

과제의 해결 수단

[0007] 상기 과제를 해결하기 위한 제 1 견지로서, 본 발명은 휴대용 단말기에서 오입력을 방지하기 위한 방법에 있어서, 입력 문자열을 확인하고 기 등록된 후보 단어들 중 상기 입력 문자열에 추천되는 것들을 검색하는 단계와, 상기 입력 문자열과 검색된 각 후보 단어 간의 유사도를 산출하는 단계 및 유사도가 높은 순으로 검색된 후보 단어들을 제시하는 단계를 포함하는 것을 특징으로 하는 방법을 제공한다.

[0008] 상기 과제를 해결하기 위한 제 2 견지로서, 본 발명은 휴대용 단말기에서 오입력을 방지하기 위한 장치에 있어서, 입력을 위한 입력부와, 음성 또는 영상 출력을 위한 출력부와, 데이터를 저장하는 저장부와, 통신을 위한 통신부 및 전반적인 동작의 제어를 수행하는 제어부를 포함하되, 상기 제어부는 입력 문자열을 확인한 후 기 등록된 후보 단어들 중 상기 입력 문자열에 추천되는 것들을 검색하고, 상기 입력 문자열과 검색된 각 후보 단어 간의 유사도를 산출하여 유사도가 높은 순으로 검색된 후보 단어들을 제시함을 특징으로 하는 장치를 제공한다.

발명의 효과

[0009] 본 발명은 키패드별 오입력의 유형을 고려하여 유사한 단어를 제시함으로 오입력을 줄여 사용자 입력을 돕는다.

도면의 간단한 설명

[0010] 도 1은 본 발명의 일 실시 예에 따른 휴대용 단말기의 블록 사시도;
 도 2는 본 발명의 일 실시 예에 따른 키패드 유형을 보여주는 도면;
 도 3은 본 발명의 일 실시 예에 따른 오입력 유형을 보여주는 도표;
 도 4는 본 발명의 일 실시 예에 따른 입력 문자열에 대한 유사 단어를 제시하는 절차의 순서도;
 도 5는 본 발명의 일 실시 예에 따라 키 입력열을 도출하기 위하여 키패드 유형별로 각 키에 매핑된 코드를 보여주는 표;

도 6은 본 발명의 일 실시 예에 따라 입력 문자열과 각 후보 단어간의 유사도를 계산하는 알고리즘;

도 7은 본 발명의 일 실시 예에 따라 편집 거리 산출시 각 연산에 적용되는 비용을 도시한 표;

도 8은 본 발명의 일 실시 예에 따라 콤비네이션 키패드에서 입력 문자열과 각 후보 단어간의 편집 거리를 구하는 것을 개략적으로 보여주는 도면 및

도 9, 도 10 및 도 11은 본 일 실시 예에 따라 각 키패드 유형별로 인접 키 눌림에 따른 오입력에 대하여 치환 연산에 적용되는 비용이 제시된 표.

발명을 실시하기 위한 구체적인 내용

- [0011] 이하 첨부된 도면을 참조하여 본 발명의 동작 원리를 상세히 설명한다. 하기에서 본 발명을 설명에 있어 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다. 그리고 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.
- [0012] 본 발명은 휴대용 단말기에서 오입력을 방지하기 위한 방법 및 장치 관한 것으로서, 특히, 키패드별 오입력의 유형을 고려하여 유사 단어를 제시하는 방법 및 장치에 관한 것이다.
- [0013] 도 1은 본 발명의 일 실시 예에 따른 휴대용 단말기의 블록 사시도이다.
- [0014] 도 1을 참조하면, 본 발명에 따른 휴대용 단말기는 입력을 위한 입력부(11)와, 음성 또는 영상 출력을 위한 출력부(12)와, 데이터를 저장하는 저장부(13)와, 통신을 위한 통신부(14) 및 전반적인 동작의 제어를 수행하는 제어부(15)를 포함한다.
- [0015] 상기 입력부(11)는 입력 신호를 제어부(15)로 출력하고, 상기 출력부(12)는 상기 제어부(15)의 제어에 따라 입력 신호에 상응하는 음성 또는 표시 데이터를 입력받아 출력한다. 상기 저장부(12)는 전반적인 동작을 제어하는 소정의 프로그램과 제어동작이 수행될 때 입출력되는 각종 데이터를 저장한다.
- [0016] 상기 제어부(15)는 단말기의 전반적인 동작을 제어하고, 이하 도면을 참조하여 상기 제어부(15)가 본 발명의 일 실시 예에 따른 키패드별 오입력의 유형을 고려하여 유사 단어를 제시하는 방법에 대해 설명한다.
- [0017] 도 2는 본 발명의 일 실시 예에 따른 키패드 유형을 보여주는 도면이다.
- [0018] 도 2를 참조하면, 키패드의 유형은 쿼티(QWERTY) 키패드, 콤비네이션 키패드, 팝업 키패드 등이 있다. 일 예로, 터치스크린 단말기는 이러한 키패드들을 가상으로 모두 구현할 수 있다. 상기 쿼티 키패드는 PC(Personal Computer) 키보드의 자판 배열과 동일하고, 하나의 키에 하나의 문자(또는 자소)가 매핑되기 때문에 여러 키가 구성된다. 따라서, 상기 쿼티 키패드는 의도하지 않게 인접키를 누를 수 있는 오입력을 자주 발생시킨다. 또한, 상기 콤비네이션 키패드는 하나의 키에 두 가지 이상의 문자가 할당되어 있고 키를 누를 때마다 문자가 바뀌어 입력되는 방식으로, 상기 쿼티 키패드에 비해 키의 수가 적기 때문에 인접하는 키가 눌러져 오입력되는 경우보다 키가 눌러지는 횟수에 따른 오입력이 자주 발생한다. 더욱이, 상기 팝업 키패드는 자소가 정의된 키를 눌렀을 때 이 키에 정의된 자소와 조합 가능한 자소들이 정의된 키들을 드래그 또는 터치로 선택할 수 있는 창이 제시되는 방식으로, 인접하는 키가 눌러져 오입력이 발생하기 쉽다.
- [0019] 도 3은 본 발명의 일 실시 예에 따른 오입력 유형을 보여주는 도표이다.
- [0020] 도 3을 참조하면, 오입력의 유형은 해당 키를 누르는 횟수에 따라 오입력이 발생하는 것과, 인접하는 키가 눌러져 오입력이 발생하는 것과, 키의 눌림이 생략되어 오입력이 발생하는 것 및 키 입력의 순서가 바뀌어 오입력이 발생하는 것으로 나눌 수 있다. 상기 콤비네이션 키패드에서는 연속으로 키를 누름에 따른 오입력과 키 생략에 따른 오입력이 자주 발생한다. 더욱이, 모든 키패드 유형에서는 인접하는 키를 누름에 따른 오입력과 키 입력 순서에 따른 오입력이 자주 발생한다. 즉, 키패드 유형별로 발생 빈도가 높은 오입력의 유형이 다르다. 따라서,

차후 설명하겠으나, 키패드 유형별로 오입력을 고려하여 입력된 문자열과 후보 단어들의 유사도가 계산되며, 유사도가 높은 후보 단어들이 우선순위로 사용자에게 제시된다.

- [0021] 도 4는 본 발명의 일 실시 예에 따른 입력 문자열에 대한 유사 단어를 제시하는 절차의 순서도이다.
- [0022] 도 4를 참조하면, 상기 제어부(15)는 키패드 유형을 확인한다(401 단계). 전술하였듯이, 상기 키패드 유형은 쿼티 키패드, 콤비네이션 키패드, 팝업 키패드 중 어느 하나가 될 수 있다. 일 예로, 터치스크린은 세 가지 키패드 유형을 가상으로 모두 제공할 수 있으며, 사용자는 그 중 하나를 선택하여 사용할 수 있다.
- [0023] 다음, 상기 제어부(15)는 사용자에게 의해서 입력되는 입력 문자열을 확인한다(401 단계).
- [0024] 이후, 상기 제어부(15)는 상기 입력 문자열을 상기 확인된 키패드 유형에 대한 키 입력열로 변환한다(405 단계). 상기 키 입력열(또는, 코드열)은 상기 문자열이 입력되기 위하여 각 키가 눌러지는 일련의 순서를 나타낸다.
- [0025] 도 5는 본 발명의 일 실시 예에 따라 키 입력열을 도출하기 위하여 키패드 유형별로 각 키에 매핑된 코드를 보여주는 표이다. 5를 참조하면, 쿼티 키패드는 한글 자소가 할당된 각 키를 알파벳 코드로 매핑한다. 일 예로, '감기' 문자열은 'RKARL' 키 입력열로 변환된다. 또한, 콤비네이션 키패드는 한글 자소가 할당된 각 키를 숫자 코드 매핑한다. 일 예로, '감기' 문자열은 '4120041' 키 입력열로 변환된다. 즉, 상기 키 입력열(또는, 코드열)은 해당 문자열이 입력되는 실제 키들이 차례로 눌러지는 순서를 제시한다. 또한, 상기 키 입력열은 해당 문자열이 입력되는 실제 키들의 자판상 위치를 제시한다. 따라서, 오입력에 해당하는 문자열의 키 입력열과 정답에 해당하는 문자열의 키 입력열을 비교한다면 어떤 종류의 오입력인지 확인할 수 있다. 일 예로, 정답 문자열 '감기'와 오입력 문자열 '감기'는 문자열 자체로만 본다면 'ㄱ' 대신 'ㅋ'이 입력된 것이다. 그러나 콤비네이션 키패드에서 '감기' 문자열은 '4120041' 키 입력열로 '감기' 문자열의 '44120041' 키 입력열로 표현되기 때문에 'ㄱ/ㅋ'키가 연속으로 눌러져 생긴 오입력이라는 것을 알 수 있다.
- [0026] 더욱이, 상기 제어부(15)는 기 등록된 후보 단어들 중 상기 입력 문자열에 추천되는 것들을 검색한다(407 단계). 입력 문자열에 추천되는 후보 단어들을 검색하는 방법은 입력 문자열을 자소 단위로 분해하여 이와 일정 이상 일치하는 것 들을 기 등록된 후보 단어들 중에서 검색할 수 있다. 이후, 상기 제어부(15)는 검색된 후보 단어들을 전술하였듯이 확인된 키패드 유형에 대한 키 입력열로 변환한다.
- [0027] 이후, 상기 제어부(15)는 확인된 키패드 유형에서의 오입력을 고려하여 입력 문자열의 키 입력열과 검색된 각 후보 단어의 키 입력열간의 유사도를 계산한다. 도 6은 본 발명의 일 실시 예에 따라 입력 문자열과 각 후보 단어간의 유사도를 계산하는 알고리즘이다. 도 6을 참조하면, 상기 알고리즘은 원본 문자열을 원하는 문자열로 바꾸는 최소 편집 횟수, 즉, 편집 거리(Edit Distance)를 구하는 것이다. 편집 거리를 구하는 알고리즘으로 공지된 레빈슈타인 거리(Levenshtein Distance)가 있다. 일반적으로 상기 레빈슈타인 거리는 편집 명령의 종류와 대상에 관계없이 일정한 횟수만을 계산하는 알고리즘이다. 즉, 일반적으로 레빈슈타인 거리는 삽입(insert), 삭제(delete), 치환(또는, 수정)(replace)에 대한 각 연산에 동일한 값(또는, 비용) 1을 적용한다. 그러나 전술하였듯이, 키패드 유형별로 발생빈도가 높은 오입력의 유형이 다르므로, 본 발명은 편집 거리를 구할 때 삽입, 삭제, 치환에 대한 각 연산에 키패드 유형별로 상이한 비용을 적용한다. 산출한 편집 거리가 작으면 작을수록 두 문자열이 더 유사한 것으로 판단된다. 따라서, 입력 문자열의 키 입력열($s[1..i]$)과 검색된 각 후보 단어의 키 입력열($t[1..j]$) 간의 유사도는 편집 거리가 작으면 높다.
- [0028] 도 7은 본 발명의 일 실시 예에 따라 편집 거리 산출시 각 연산에 적용되는 비용을 도시한 표이다. 도 7을 참조하면, 기본적으로 모든 연산은 비용 3이 적용되되, 해당 키패드 유형별로 발생 빈도가 높은 오입력 유형을 고려하여 삽입, 삭제, 치환 연산에 대한 비용이 차등된다. 일 예로, 연속 키 눌림에 따른 오입력은 쿼티 키패드와 팝업 키패드에서 발생하기 어려우나 콤비네이션 키패드에서는 발생하기 쉬우므로, 콤비네이션 키패드의 삽입 연산은 비용 1 또는 3이 적용되고 나머지 키패드 유형은 비용 3이 적용된다. 즉, 콤비네이션 키패드의 경우 연속

키 눌림에 따른 오입력의 조건($s_{i-1} == s_i \& s_i == t_j$)이 만족되면 삽입 연산의 비용을 1로 적용하고 그렇지 아니하면 3을 적용한다. 나머지 키패드 유형은 연속 키 눌림에 따른 오입력 조건의 만족 여부에 상관없이 삽입 연산의 비용을 3으로 적용한다. 또한, 키 생략에 따른 오입력은 쿼티 키패드와 팝업 키패드에서 발생하기 어려우나 콤비네이션 키패드에서는 발생하기 쉬우므로, 콤비네이션 키패드의 삭제 연산은 비용 1 또는 3이 적용되고 나머지 키패드 유형은 비용 3이 적용된다. 즉, 콤비네이션 키패드의 경우 키 생략에 따른 오입력의 조

건($s_i == t_{j+1} \& s_i == t_j$)이 만족되면 삭제 연산의 비용을 1로 적용하고 그렇지 아니하면 3을 적용한다. 나머지 키패드 유형은 키 생략에 따른 오입력 조건의 만족 여부에 상관없이 삭제 연산의 비용을 3으로 적용한다. 게다가, 키 입력 순서에 따른 오입력은 모든 키패드 유형에서 발생하기 쉬우므로, 모든 키패드 유형의 치환 연산은 비용 1 또는 3이 적용된다. 즉, 모든 키패드 유형은 키 입력 순서에 따른 오입력의 조건

($s_i s_{i+1} == t_{j+1} t_j$)이 만족되면 치환 연산의 비용을 1로 적용하고 그렇지 아니하면 3을 적용한다. 더욱이, 인접 키 눌림에 따른 오입력은 모든 키패드 유형에서 발생하며 치환 연산은 키의 물리적 위치를 고려하여 각 키패드 유형별로 해당 비용이 적용된다. 도 9, 도 10 및 도 11은 본 일 실시 예에 따라 각 키패드 유형별로 인접 키 눌림에 따른 오입력에 대하여 치환 연산에 적용되는 비용이 제시된 표들이다. 도 9, 도 10 및 도 11을 참조하면, 기본 비용을 3으로 하되, 인접 키 눌림에 따른 오입력에 대한 치환 연산은 인접해 있으나 오입력 가능성이 높은 키는 비용 1이 적용되고, 인접해 있으나 오입력 가능성이 낮은 키는 비용 2가 적용된다.

[0029] 다음, 상기 제어부(15)는 입력 문자열과 유사도가 높은 순으로 후보 단어들을 배열하여 제시한다(413 단계).

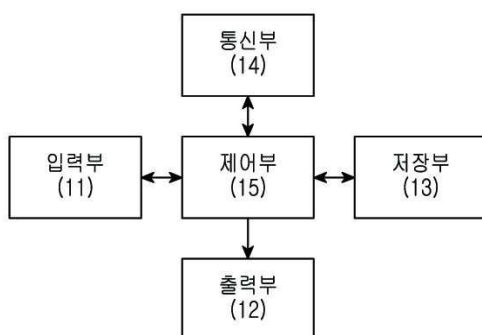
[0030] 도 8은 본 발명의 일 실시 예에 따라 콤비네이션 키패드에서 입력 문자열과 각 후보 단어들의 편집 거리를 구하는 것을 개략적으로 보여주는 도면이다.

[0031] 도 8을 참조하면, 사용자에게 의하여 입력된 입력 문자열과 상기 입력 문자열에 상응하는 각 후보 단어들은 콤비네이션 키패드에 대한 키 입력열로 변환된다. 이후, 도 6 및 7에서 전술한 바와 같이, 콤비네이션 키패드에서 두 입력열의 편집 거리를 계산하며, 이때 발생 빈도가 높은 오입력의 유형을 고려하여 삽입 또는 삭제 또는 치환의 연산에 적용되는 비용이 결정된다. 입력 문자열의 입력열과 각 후보 단어의 입력열과의 편집 거리들(A, B, C,...)이 산출되고, 편집 거리가 작은 것부터 큰 순서로 해당 후보 단어들이 사용자에게 제시된다.

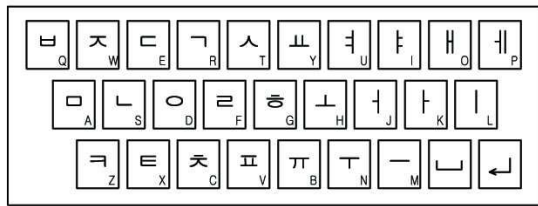
[0032] 한편 본 발명의 상세한 설명에서는 구체적인 실시 예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능하다. 그러므로 본 발명의 범위는 설명된 실시 예에 국한되어 정해져서는 아니 되며 후술하는 특허청구의 범위뿐만 아니라 이 특허청구의 범위와 균등한 것들에 의해 정해져야 한다.

도면

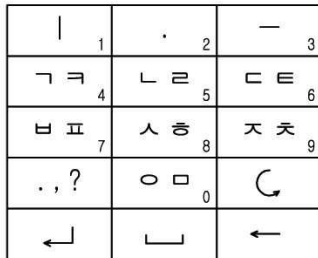
도면1



도면2



<QWERTY 키패드>



< COMBINATION 키패드 >

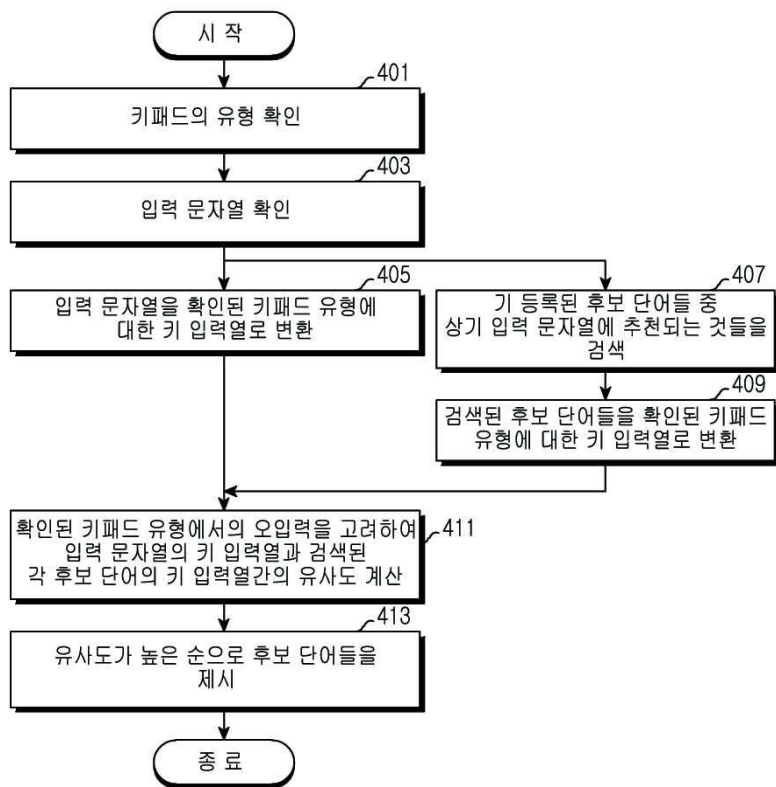


< POP-UP 키패드 >

도면3

오입력 유형	예	해당 오입력의 발생 빈도가 높은 키패드 유형
연속 키 눌림	등잔산 -> 등잔ㅅ산	COMBINATION
인접 키 눌림	삼일전력공사 -> 삼일전렬공사	QWERTY, COMBINATION, POP-UP
키 생략	효성기업사 -> 효싱기업사	COMBINATION
키 입력 순서	장호원유치원 -> 장호원유치위.ㄴ	QWERTY, COMBINATION, POP-UP

도면4



도면5

< QWERTY KEYPAD >

키	매핑 문자
ㅂ	Q
ㅈ	W
ㄷ	E
ㄱ	R
ㅅ	T
ㅊ	Y
ㅋ	U
ㅌ	I
ㅎ	O
ㄱ	P
...	...

< COMBINATION KEYPAD >

키	매핑 문자
₁	1
· ₂	2
— ₃	3
ㄱ ㅋ ₄	4
ㄴ ㄹ ₅	5
ㄷ ㅌ ₆	6
ㅂ ㅍ ₇	7
ㅅ ㅎ ₈	8
ㅈ ㅊ ₉	9
ㅇ ㅁ ₀	0
...	...

도면6

< Edit Distance >

$$\text{edit}(i, j) = \min \begin{bmatrix} \text{edit}(i-1, j) + \text{insert}(s_{i-1}, s_i, t_i) \\ \text{edit}(i, j-1) + \text{delete}(s_i, t_i, t_{j+1}) \\ \text{edit}(i-1, j-1) + \text{replace}(s_i, s_{i+1}, t_j, t_{j+1}) \end{bmatrix}$$

도면7

유형		조건	편집 거리 산출시 각 연산에 적용되는 비용(값)		
			QWERTY	COMBINATION	POP-UP
insert(s_{i-1} , s_i , t_i)	연속 키 늘림	if $s_{i-1}==s_i$ & $s_i==t_i$	3	1	3
		otherwise	3		
delete(s_i , t_i , t_{i+1})	키 생략	if $s_i==t_{i+1}$ & $s_i==t_i$	3	1	3
		otherwise	3		
replace(s_i , s_{i+1} , t_i , t_{i+1})	.	if $s_i==t_i$	0		
		if $s_{i+1}==t_{i+1}$ & t_i	1		
	키 입력 순서	if NEAR(s_i , t_i)	인접해 있으며 오입력 가능성이 높은 키 = 1 인접해 있지만 오입력 가능성이 많지 않은 키 = 2		
	인접 키 늘림	otherwise	3		

도면8

<Edit Distance in Combination Keypad>																															
입력 문자열				후보 문자열																											
강기 ⇨ 44120041				강기 ⇨ 4120041							강기 ⇨ 412041							검기 ⇨ 4120041													
				4	1	2	0	0	1	4	1	2	0	4	1	4	2	1	0	0	4	1									
				...																											
				...																											
				...																											
4																															
4																															
1																															
2																															
0																															
0																															
4																															
1																															
											A																				
</																															

도면9

< QWERTY KEYPAD >

자소	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	Sh
a	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	3	1	3	3	3	3	3	3	2	1
b	3	0	3	3	3	3	2	2	3	3	3	3	3	1	3	3	3	3	3	3	3	1	3	3	3	3	1
c	3	3	0	2	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	3	1	3	3	1
d	3	3	2	0	1	1	3	3	3	3	3	3	3	3	3	3	3	1	3	3	3	3	2	3	3	1	
e	3	3	3	1	0	3	3	3	3	3	3	3	3	3	3	3	1	3	3	3	3	1	3	3	3	3	1
f	3	3	3	1	3	0	1	3	3	3	3	3	3	3	3	3	1	3	3	3	2	3	3	3	3	1	
g	3	2	3	3	3	1	0	1	3	3	3	3	3	3	3	3	3	3	1	3	2	3	3	3	3	1	
h	3	2	3	3	3	3	1	0	3	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	1	3	1
i	3	3	3	3	3	3	3	0	3	1	3	3	3	1	3	3	3	3	3	1	3	3	3	3	3	1	
j	3	3	3	3	3	3	3	1	3	0	1	3	2	2	3	3	3	3	3	3	1	3	3	3	3	3	1
k	3	3	3	3	3	3	3	1	1	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	1	
l	3	3	3	3	3	3	3	3	3	3	1	0	3	3	1	3	3	3	3	3	3	3	3	3	3	1	
m	3	3	3	3	3	3	3	3	3	2	2	3	0	1	3	3	3	3	3	3	3	3	3	3	3	1	
n	3	1	3	3	3	3	2	3	3	2	3	3	1	0	3	3	3	3	3	3	3	3	3	3	3	1	
o	3	3	3	3	3	3	3	3	1	3	3	1	3	3	0	1	3	3	3	3	3	3	3	3	3	1	
p	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	0	3	3	3	3	3	3	3	3	3	1	
q	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	3	3	3	3	3	1	3	3	3	1	
r	3	3	3	3	1	3	3	3	3	3	3	3	3	3	3	3	0	3	1	3	3	3	3	3	3	1	
s	1	3	3	1	3	3	3	3	3	3	3	3	3	3	3	3	3	0	3	3	3	1	2	3	2	1	
t	3	3	3	3	3	3	1	3	3	3	3	3	3	3	3	3	1	3	0	3	3	3	3	1	3	1	
u	3	3	3	3	3	3	3	3	1	1	3	3	3	3	3	3	3	3	3	3	0	3	3	3	1	3	1
v	3	1	1	3	3	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	0	3	3	3	3	1	
w	3	3	3	3	1	3	3	3	3	3	3	3	3	3	3	1	3	1	3	3	3	0	3	3	3	1	
x	3	3	1	2	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3	3	0	3	1	1	1	
y	3	3	3	3	3	3	3	1	3	3	3	3	3	3	3	3	3	3	1	1	3	3	3	0	3	1	
z	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3	3	1	3	0	1	
Sh	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	

도면10

< QWERTY KEYPAD >

자소		ㅇ/ㅁ	ㅣ	·	ㅡ	ㄱ/ㅋ	ㄴ/ㄷ	ㄷ/ㅌ	ㅂ/ㅍ	ㅅ/ㅈ	ㅊ/ㅊ	[]
	코드	0	1	2	3	4	5	6	7	8	9	a
ㅇ/ㅁ	0	0	3	3	3	3	3	3	3	3	3	3
ㅣ	1	3	0	1	3	1	1	3	3	3	3	3
·	2	3	1	0	1	3	1	3	3	3	3	3
ㅡ	3	3	3	1	0	3	3	1	3	3	3	3
ㄱ/ㅋ	4	3	1	3	3	0	1	3	3	3	3	3
ㄴ/ㄷ	5	3	3	3	3	1	0	1	3	3	3	3
ㄷ/ㅌ	6	3	3	3	1	3	1	0	3	3	1	3
ㅂ/ㅍ	7	3	3	3	3	1	3	3	0	1	3	3
ㅅ/ㅈ	8	3	3	3	3	3	1	3	1	0	1	3
ㅊ/ㅊ	9	3	3	3	3	3	3	1	3	1	0	3
[]	a	3	3	3	3	3	3	3	3	3	3	0

도면11

< POU-UP KEYPAD >

자소		—	·	l	⊥	⌢	†	‡
	코드	n	m	j	8	2	4	6
—	n	0	2	2	1	1	1	1
·	m	2	0	2	1	1	1	1
l	j	2	2	0	1	1	1	1
⊥	8	1	1	1	0	3	3	3
⌢	2	1	1	1	3	0	3	3
†	4	1	1	1	3	3	0	3
‡	6	1	1	1	3	3	3	0