



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2017년09월13일  
 (11) 등록번호 10-1777961  
 (24) 등록일자 2017년09월06일

(51) 국제특허분류(Int. Cl.)  
 G06K 9/00 (2006.01) G06F 3/0488 (2013.01)  
 (52) CPC특허분류  
 G06K 9/0002 (2013.01)  
 G06F 3/04883 (2013.01)  
 (21) 출원번호 10-2016-0046915  
 (22) 출원일자 2016년04월18일  
 심사청구일자 2016년04월18일  
 (56) 선행기술조사문헌  
 KR1020140148490 A\*  
 KR1020150117787 A\*  
 \*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
 주식회사 한글과컴퓨터  
 경기도 성남시 분당구 대왕판교로644번길 49, 10층(삼평동, 한컴타워)  
 한국과학기술원  
 대전광역시 유성구 대학로 291(구성동)  
 (72) 발명자  
 이기혁  
 대전광역시 유성구 대학로 291 (구성동, 한국과학기술원)  
 김선준  
 대전광역시 유성구 대학로 291(구성동, 한국과학기술원)  
 (74) 대리인  
 양성보

전체 청구항 수 : 총 10 항

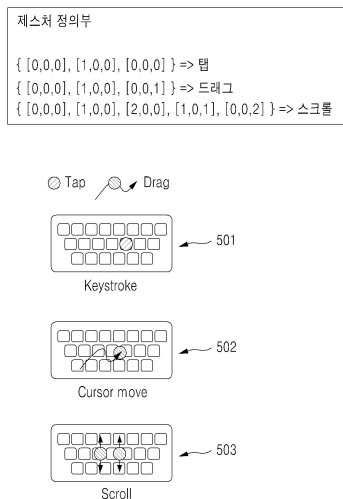
심사관 : 노용완

(54) 발명의 명칭 **멀티 터치 제스처를 인식하는 방법 및 시스템**

**(57) 요약**

멀티 터치 제스처를 인식하는 방법 및 시스템이 개시된다. 제스처 인식 방법은, 터치 포인트 각각에 대해 복수의 터치 단계 중 어느 하나의 터치 단계를 구분하는 단계; 상기 복수의 터치 단계에 대해 각 터치 단계에 해당되는 터치 포인트의 개수를 나타내는 장치 상태를 기록하는 단계; 및 상기 기록된 장치 상태에 대응되는 제스처를 인식하는 단계를 포함한다.

**대표도** - 도5



(52) CPC특허분류  
*G06K 9/00087* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

컴퓨터로 구현되는 제스처 인식 방법에 있어서,

터치 포인트 각각에 대해 복수의 터치 단계 중 어느 하나의 터치 단계를 구분하는 단계;

상기 복수의 터치 단계에 대해 각 터치 단계에 해당되는 터치 포인트의 개수를 나타내는 장치 상태를 기록하는 단계; 및

상기 기록된 장치 상태에 대응되는 제스처를 인식하는 단계

를 포함하고,

상기 기록하는 단계는,

N개의 터치 단계에 대한 장치 상태를, 터치 단계 별 터치 포인트의 개수를 나타내는 N개의 숫자로 표현된 튜플(tuple)로 기록하되, 터치 단계 별 터치 포인트의 개수가 변하는 경우 상기 N개의 터치 단계에 대한 장치 상태의 변화를 튜플의 리스트 형태로 기록하고,

상기 제스처 각각은 장치 상태의 조합으로 사전 정의되는 것으로, 상기 N의 터치 단계에 대해 터치 단계 별 터치 포인트의 개수를 나타내는 N개의 숫자로 표현된 튜플의 리스트로 정의되는 것

을 특징으로 하는 제스처 인식 방법.

#### 청구항 2

제1항에 있어서,

상기 구분하는 단계는,

상기 터치 포인트의 터치 유지 시간과 터치 이동 거리 중 적어도 하나에 따라 해당 터치 포인트의 터치 단계를 구분하는 것

을 특징으로 하는 제스처 인식 방법.

#### 청구항 3

제1항에 있어서,

상기 복수의 터치 단계는 탭 단계와 눌림 단계 및 움직임 단계를 포함하고,

상기 구분하는 단계는,

최초 접촉으로 터치 포인트가 생성되는 경우 해당 터치 포인트의 터치 단계를 상기 탭 단계로 구분하고,

일정 시간 이상 접촉이 유지되는 경우 해당 터치 포인트의 터치 단계를 상기 눌림 단계로 구분하고,

상기 탭 단계 또는 상기 눌림 단계의 터치 포인트가 일정 거리 이상 이동하는 경우 해당 터치 포인트의 터치 단계를 상기 움직임 단계로 구분하는 것

을 특징으로 하는 제스처 인식 방법.

#### 청구항 4

삭제

#### 청구항 5

삭제

**청구항 6**

삭제

**청구항 7**

제1항에 있어서,

상기 인식하는 단계는,

상기 기록된 장치 상태를 상기 제스처로 정의된 장치 상태와 비교하여 상기 기록된 장치 상태와 일치하는 제스처를 인식하는 것

을 특징으로 하는 제스처 인식 방법.

**청구항 8**

삭제

**청구항 9**

컴퓨터로 구현되는 방법에 있어서,

터치 포인트 각각에 대해 복수의 터치 단계 중 어느 하나의 터치 단계를 구분하는 단계;

상기 복수의 터치 단계에 대해 각 터치 단계에 해당되는 터치 포인트의 개수를 나타내는 장치 상태의 변화를 기록하는 단계;

상기 장치 상태의 변화에 대응되는 제스처를 인식하는 단계; 및

상기 제스처에 해당되는 동작을 수행하는 단계

를 포함하고,

상기 기록하는 단계는,

N개의 터치 단계에 대한 장치 상태를, 터치 단계 별 터치 포인트의 개수를 나타내는 N개의 숫자로 표현된 튜플(tuple)로 기록하되, 터치 단계 별 터치 포인트의 개수가 변하는 경우 상기 N개의 터치 단계에 대한 장치 상태의 변화를 튜플의 리스트 형태로 기록하고,

상기 제스처 각각은 장치 상태의 조합으로 사전 정의되는 것으로, 상기 N의 터치 단계에 대해 터치 단계 별 터치 포인트의 개수를 나타내는 N개의 숫자로 표현된 튜플의 리스트로 정의되는 것

을 특징으로 하는 방법.

**청구항 10**

컴퓨터 시스템과 결합되어,

터치 포인트 각각에 대해 복수의 터치 단계 중 어느 하나의 터치 단계를 구분하는 단계;

상기 복수의 터치 단계에 대해 각 터치 단계에 해당되는 터치 포인트의 개수를 나타내는 장치 상태를 기록하는 단계; 및

상기 기록된 장치 상태에 대응되는 제스처를 인식하는 단계

를 실행시키기 위해 기록 매체에 기록된 컴퓨터 프로그램으로,

상기 기록하는 단계는,

N개의 터치 단계에 대한 장치 상태를, 터치 단계 별 터치 포인트의 개수를 나타내는 N개의 숫자로 표현된 튜플(tuple)로 기록하되, 터치 단계 별 터치 포인트의 개수가 변하는 경우 상기 N개의 터치 단계에 대한 장치 상태의 변화를 튜플의 리스트 형태로 기록하고,

상기 제스처 각각은 장치 상태의 조합으로 사전 정의되는 것으로, 상기 N의 터치 단계에 대해 터치 단계 별 터

치 포인트의 개수를 나타내는 N개의 숫자로 표현된 튜플의 리스트로 정의되는 것을 특징으로 하는, 컴퓨터 프로그램.

**청구항 11**

컴퓨터로 구현되는 전자 기기의 제스처 인식 시스템에 있어서,  
 상기 컴퓨터에서 관독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서를 포함하고,  
 상기 적어도 하나의 프로세서는,  
 터치 포인트 각각에 대해 복수의 터치 단계 중 어느 하나의 터치 단계를 구분하고,  
 상기 복수의 터치 단계에 대해 각 터치 단계에 해당되는 터치 포인트의 개수를 나타내는 장치 상태를 기록하고,  
 상기 기록된 장치 상태에 대응되는 제스처를 인식하고,  
 상기 적어도 하나의 프로세서는, 상기 장치 상태를 기록하기 위해,  
 N개의 터치 단계에 대한 장치 상태를, 터치 단계 별 터치 포인트의 개수를 나타내는 N개의 숫자로 표현된 튜플 (tuple)로 기록하되, 터치 단계 별 터치 포인트의 개수가 변하는 경우 상기 N개의 터치 단계에 대한 장치 상태의 변화를 튜플의 리스트 형태로 기록하고,  
 상기 제스처 각각은 장치 상태의 조합으로 사전 정의되는 것으로, 상기 N의 터치 단계에 대해 터치 단계 별 터치 포인트의 개수를 나타내는 N개의 숫자로 표현된 튜플의 리스트로 정의되는 것을 특징으로 하는 제스처 인식 시스템.

**청구항 12**

제11항에 있어서,  
 상기 적어도 하나의 프로세서는, 상기 터치 단계를 구분하기 위해,  
 상기 터치 포인트의 터치 유지 시간과 터치 이동 거리 중 적어도 하나에 따라 해당 터치 포인트의 터치 단계를 구분하는 것을 특징으로 하는 제스처 인식 시스템.

**청구항 13**

제11항에 있어서,  
 상기 복수의 터치 단계는 탭 단계와 눌림 단계 및 움직임 단계를 포함하고,  
 상기 적어도 하나의 프로세서는, 상기 터치 단계를 구분하기 위해,  
 최초 접촉으로 터치 포인트가 생성되는 경우 해당 터치 포인트의 터치 단계를 상기 탭 단계로 구분하고,  
 일정 시간 이상 접촉이 유지되는 경우 해당 터치 포인트의 터치 단계를 상기 눌림 단계로 구분하고,  
 상기 탭 단계 또는 상기 눌림 단계의 터치 포인트가 일정 거리 이상 이동하는 경우 해당 터치 포인트의 터치 단계를 상기 움직임 단계로 구분하는 것을 특징으로 하는 제스처 인식 시스템.

**청구항 14**

삭제

**청구항 15**

삭제

**청구항 16**

제11항에 있어서,

상기 적어도 하나의 프로세서는, 상기 제스처를 인식하기 위해,

상기 기록된 장치 상태를 상기 제스처로 정의된 장치 상태와 비교하여 상기 기록된 장치 상태와 일치하는 제스처를 인식하는 것

을 특징으로 하는 제스처 인식 시스템.

**청구항 17**

삭제

**발명의 설명**

**기술 분야**

[0001] 아래의 설명은 멀티 터치 장치에서 사용하는 제스처를 인식하는 기술에 관한 것이다.

**배경 기술**

[0002] 멀티 터치 장치는 사용자가 통상적으로 하나 이상의 손가락 또는 스타일러스와 같은 기타 접촉 객체를 이용하여 터치 감응 표면을 터치함으로써 만들어지는 제스처들을 사용하여 디스플레이 되는 정보와 상호작용 하도록 한다. 이때, 접촉점들의 개수와 접촉점(들)의 움직임은 멀티 터치 장치에 의해 검출되어 제스처로 해석되고, 이에 응답하여 디바이스는 다양한 작용들을 수행할 수 있다.

[0003] 대표적으로 핑커웍스(FingerWorks)의 초기 멀티 터치 기술이 있었고, 애플이나 마이크로소프트 등의 주요 OS 회사들은 자사 제품에서 다양한 멀티 터치 제스처를 활용하고 있다. 이러한 멀티 터치 제스처는 주로 터치한 손가락의 개수와 움직임의 방향 등을 이용한다.

[0004] 멀티 터치 제스처를 인식하는 방법으로 유한 상태 기계를 사용하는 방법이 있다. 그러나, 유한 상태 기계는 제스처의 수가 늘어날수록 필요한 상태의 수가 늘어나고, 상태의 수가 늘어날수록 복잡도가 지수적으로 증가하기 때문에 쉽게 제스처를 추가하거나 재정의하기가 어렵다. 멀티 터치 제스처를 인식하는 다른 방법으로 터치 포인트들의 여러 속성들을 계산하여 의사결정 트리(decision tree)를 만들거나 기계 학습을 이용하는 방법이 있다. 이는 유한 상태 기계를 사용하는 방법에 비해 제스처 인식기를 제작하기는 쉬우나, 학습 데이터에 따라 인식률이 떨어지거나 제스처의 정의가 모호해진다는 단점이 있다.

**발명의 내용**

**해결하려는 과제**

[0005] 사용자가 직관적으로 쉽게 제스처를 정의할 수 있고 해당 제스처가 수학적으로 명확하게 정의되어 제스처 간 혼동을 발생시키지 않도록 하는 제스처 인식 방법 및 시스템을 제공한다.

**과제의 해결 수단**

[0006] 터치 포인트 각각에 대해 복수의 터치 단계 중 어느 하나의 터치 단계를 구분하는 단계; 상기 복수의 터치 단계에 대해 각 터치 단계에 해당되는 터치 포인트의 개수를 나타내는 장치 상태를 기록하는 단계; 및 상기 기록된 장치 상태에 대응되는 제스처를 인식하는 단계를 포함하는 제스처 인식 방법을 제공한다.

[0007] 일 측면에 따르면, 상기 구분하는 단계는, 상기 터치 포인트의 터치 유지 시간과 터치 이동 거리 중 적어도 하나에 따라 해당 터치 포인트의 터치 단계를 구분할 수 있다.

[0008] 다른 측면에 따르면, 상기 복수의 터치 단계는 탭 단계와 눌림 단계 및 움직임 단계를 포함하고, 상기 구분하는 단계는, 최초 접촉으로 터치 포인트가 생성되는 경우 해당 터치 포인트의 터치 단계를 상기 탭 단계로 구분하고, 일정 시간 이상 접촉이 유지되는 경우 해당 터치 포인트의 터치 단계를 상기 눌림 단계로 구분하고, 상기 탭 단계 또는 상기 눌림 단계의 터치 포인트가 일정 거리 이상 이동하는 경우 해당 터치 포인트의 터치 단

계를 상기 움직임 단계로 구분할 수 있다.

- [0009] 또 다른 측면에 따르면, 상기 기록하는 단계는, 상기 복수의 터치 단계에 대해 터치 단계 별 터치 포인트의 개수를 모아 하나의 튜플(tuple)로 기록할 수 있고, 상기 기록된 튜플이 상기 복수의 터치 단계에 대한 장치 상태로 정의될 수 있다.
- [0010] 또 다른 측면에 따르면, 상기 기록하는 단계는, 상기 터치 단계 별 터치 포인트의 개수가 변하는 경우 상기 복수의 터치 단계에 대한 장치 상태를 튜플의 리스트 형태로 기록할 수 있다.
- [0011] 또 다른 측면에 따르면, 상기 제스처는 상기 복수의 터치 단계에 대해 터치 단계 별 터치 포인트의 개수를 나타내는 장치 상태의 조합으로 정의될 수 있다.
- [0012] 또 다른 측면에 따르면, 상기 제스처는 상기 복수의 터치 단계에 대해 터치 단계 별 터치 포인트의 개수를 나타내는 장치 상태로 사전 정의되고, 상기 인식하는 단계는, 상기 기록된 장치 상태를 상기 제스처의 장치 상태와 비교하여 상기 기록된 장치 상태와 일치하는 제스처를 인식할 수 있다.
- [0013] 또 다른 측면에 따르면, 상기 제스처는 상기 복수의 터치 단계에 대해 터치 단계 별 터치 포인트의 개수를 나타내는 장치 상태의 리스트로 사전 정의되고, 상기 기록하는 단계는, 상기 터치 단계 별 터치 포인트의 개수가 변하는 경우 상기 복수의 터치 단계에 대한 장치 상태를 리스트 형태로 기록하고, 상기 인식하는 단계는, 상기 기록된 장치 상태의 리스트를 상기 제스처의 장치 상태의 리스트와 비교하여 상기 기록된 장치 상태와 일치하는 제스처를 인식할 수 있다.
- [0014] 터치 포인트 각각에 대해 복수의 터치 단계 중 어느 하나의 터치 단계를 구분하는 단계; 상기 복수의 터치 단계에 대해 각 터치 단계에 해당되는 터치 포인트의 개수를 나타내는 장치 상태의 변화를 기록하는 단계; 상기 장치 상태의 변화에 대응되는 제스처를 인식하는 단계; 및 상기 제스처에 해당되는 동작을 수행하는 단계를 포함하는 방법을 제공한다.
- [0015] 컴퓨터 시스템과 결합되어, 터치 포인트 각각에 대해 복수의 터치 단계 중 어느 하나의 터치 단계를 구분하는 단계; 상기 복수의 터치 단계에 대해 각 터치 단계에 해당되는 터치 포인트의 개수를 나타내는 장치 상태를 기록하는 단계; 및 상기 기록된 장치 상태에 대응되는 제스처를 인식하는 단계를 실행시키기 위해 기록 매체에 기록된 컴퓨터 프로그램을 제공한다.
- [0016] 컴퓨터로 구현되는 전자 기기의 제스처 인식 시스템에 있어서, 상기 컴퓨터에서 판독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는, 터치 포인트 각각에 대해 복수의 터치 단계 중 어느 하나의 터치 단계를 구분하고, 상기 복수의 터치 단계에 대해 각 터치 단계에 해당되는 터치 포인트의 개수를 나타내는 장치 상태를 기록하고, 상기 기록된 장치 상태에 대응되는 제스처를 인식하는 것을 특징으로 하는 제스처 인식 시스템을 제공한다.

**발명의 효과**

- [0017] 본 발명에서 제안하는 제스처 정의는 짧은 길이의 숫자를 나열하는 것만으로 특정 제스처를 명확하게 정의할 수 있다는 장점이 있다. 또한, 제스처의 인식에 있어 직관적이고 단순한 방법을 사용하기 때문에 개발자뿐만 아니라 일반 사용자들도 쉽게 임의의 제스처를 추가하고 삭제하는 것이 용이하다. 또한, 제스처의 인식 방법이 복잡하지 않고 명료하기 때문에 제스처 인식 정확도가 높다는 장점이 있다.

**도면의 간단한 설명**

- [0018] 도 1은 본 발명의 일 실시예에 있어서, 제스처 인식 시스템의 내부 구성을 설명하기 위한 블록도를 도시한 것이다.
- 도 2는 본 발명의 일 실시예에 있어서, 제스처 인식 시스템의 프로세서가 포함할 수 있는 구성요소의 예를 도시한 도면이다.
- 도 3은 본 발명의 일 실시예에 있어서, 제스처 인식 방법의 예를 도시한 도면이다.
- 도 4는 본 발명의 일 실시예에 있어서, 터치 단계 전이도를 설명하기 위한 예시 도면이다.
- 도 5는 본 발명의 일 실시예에 있어서, 제스처 정의 및 제스처 별 기능의 예를 도시한 도면이다.
- 도 6 내지 도 7은 본 발명의 일 실시예에 있어서, 제스처 정의 및 제스처 별 기능의 다른 예를 도시한

도면이다.

**발명을 실시하기 위한 구체적인 내용**

- [0019] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.
- [0020] 본 실시예들은 멀티 터치 장치에서 사용하는 제스처를 인식하는 제스처 인식 방법 및 시스템에 관한 것으로, 이는 멀티 터치 장치를 가진 대부분의 전자 기기에 적용 가능하며, 예컨대 스마트폰(smart phone), 태블릿(tablet), 터치 컴퓨터(touch computer), 웨어러블 컴퓨터(wearable computer) 등의 다양한 장치에 적용될 수 있다.
- [0021] 도 1은 본 발명의 일 실시예에 있어서, 제스처 인식 시스템의 내부 구성을 설명하기 위한 블록도를 도시한 것이다.
- [0022] 도 1을 참조하면, 제스처 인식 시스템(100)은 디스플레이(110), 터치 장치(120), 통신 모듈(130), 프로세서(140), 전력 모듈(150) 및 스토리지(160)를 포함할 수 있다.
- [0023] 디스플레이(110)는 디스플레이 화면에 시각화를 위한 콘텐츠를 출력할 수 있다. 또한, 디스플레이(110)는 프로세서(130)에서 실행되는 정보 또는 프로세서(130)의 제어 명령에 기초하여 콘텐츠를 출력할 수 있다. 예를 들어, 콘텐츠는 각종 텍스트, 이미지, 동영상 등 시각화 가능한 모든 정보를 포함할 수 있다.
- [0024] 터치 장치(120)는 임의 개수의 손가락이나 스타일러스와 같은 접촉 객체의 터치를 감지할 수 있는 장치이며, 예컨대 디스플레이(110)와 결합된 터치스크린이나 터치패드, 터치 감지가 가능한 키보드 형태의 장치 등을 포함할 수 있다. 제스처 인식 시스템(100)에 대한 입력 신호는 터치 장치(120)에 대한 터치 입력을 기본으로 할 수 있다.
- [0025] 통신 모듈(130)은 외부 디바이스와 다양한 프로토콜을 사용하여 통신을 수행하고, 이를 통해 데이터를 송/수신할 수 있다. 또한, 통신 모듈(130)은 유선 또는 무선으로 네트워크에 접속하여 콘텐츠 등의 디지털 데이터를 송/수신할 수 있다.
- [0026] 프로세서(140)는 데이터 통신을 통해 수신된 콘텐츠, 또는 스토리지(160)에 저장된 콘텐츠 등을 실행할 수 있다. 또한, 프로세서(140)는 제스처 인식 시스템(100)의 각 유닛들을 제어하며, 유닛들 간의 데이터 송수신을 제어할 수 있다. 특히, 프로세서(140)는 제스처 인식 시스템(100)의 터치 장치(120)를 통해서 디택트한 입력 신호에 기초하여 제스처 인식 시스템(100)의 동작을 제어할 수 있다.
- [0027] 전력 모듈(150)은 제스처 인식 시스템(100) 내부의 배터리 또는 외부 전원과 연결되는 파워 소스로, 제스처 인식 시스템(100)에 파워를 공급할 수 있다. 전력 모듈(150)은 제스처 인식 시스템(100)의 컴포넌트의 전부 또는 일부로 전력을 공급할 수 있다. 예를 들어 전력 모듈(150)은 전력 관리 시스템, 배터리나 교류(AC) 등과 같은 하나 이상의 전원, 충전 시스템, 전력 실패 감지 회로(power failure detection circuit), 전력 변환기나 인버터, 전력 상태 표시자 또는 전력 생성, 관리, 분배를 위한 임의의 다른 컴포넌트들을 포함할 수 있다.
- [0028] 스토리지(160)는 동영상, 오디오, 사진, 애플리케이션 등 다양한 디지털 데이터를 저장할 수 있다. 스토리지(160)는 프로세서(140)의 처리 및 제어를 위한 프로그램을 저장할 수 있고, 입/출력되는 데이터들의 임시 저장을 위한 기능을 수행할 수도 있다. 스토리지(160)는 일례로 고속 랜덤 액세스 메모리(high-speed random access memory), 자기 디스크, 에스램(SRAM), 디램(DRAM), 롬(ROM), 플래시 메모리 또는 비휘발성 메모리를 포함할 수 있다.
- [0029] 스토리지(160)는 제스처 인식 시스템(100)의 동작에 필요한 소프트웨어 모듈, 명령어 집합 또는 그밖에 다양한 데이터를 포함할 수 있다. 이때, 프로세서(140)나 다른 컴포넌트에서 스토리지(160)에 액세스하는 것은 프로세서(140)에 의해 제어될 수 있다.
- [0030] 제스처 인식 시스템(100)이 수행하는 다양한 동작들은 프로세서(140)에 의해 제어될 수 있다. 또한, 제스처 인식 시스템(100)의 동작들은 스토리지(160)에 저장되어 프로세서(140)에 의해 구동되는 애플리케이션에 의해 제어될 수도 있다. 편의를 위해, 도면 및 이하의 설명에서는 이러한 동작들을 통칭하여 제스처 인식 시스템(100)이 수행/제어하는 것으로 도시 및 설명하도록 한다.
- [0031] 이러한 도 1의 실시예는, 제스처 인식 시스템(100)의 일례일 뿐이고, 제스처 인식 시스템(100)은 도 1에 도시된



일부 컴포넌트가 생략되거나, 도 1에 도시되지 않은 추가의 컴포넌트를 더 구비하거나, 2개 이상의 컴포넌트를 결합시키는 구성 또는 배치를 가질 수 있다. 제스처 인식 시스템(100)에 포함 가능한 컴포넌트들은 하나 이상의 신호 처리 또는 어플리케이션에 특화된 집적 회로를 포함하는 하드웨어, 소프트웨어, 또는 하드웨어 및 소프트웨어 양자의 조합으로 구현될 수 있다.

- [0032] 도 1에 도시된 제스처 인식 시스템(100)은 일 실시예에 따른 블록도로서, 분리하여 표시한 블록들은 제스처 인식 시스템(100)의 엘리먼트들을 논리적으로 구별하여 도시한 것이다. 따라서, 상술한 제스처 인식 시스템(100)의 엘리먼트들은 디바이스의 설계에 따라 하나의 칩 또는 복수의 칩으로 장착될 수 있다.
- [0033] 본 실시예들은 터치 포인트의 터치 단계를 구분하는 방법, 각 터치 단계에 있는 터치 포인트의 개수를 세는 방법, 및 터치 포인트의 단계 변화에 따라 제스처를 인식하는 방법을 포함한다.
- [0034] 본 발명에서는 제스처의 정의를 위해 다음의 개념을 적용한다.
- [0035] 1) 터치 단계: 손가락 등의 터치를 인식함에 따라 터치 포인트가 생성되게 된다. 터치 포인트가 생성되고 터치 포인트가 사라질 때까지 터치 포인트는 유지 시간과 이동 거리 등에 따라 세 가지 단계 중 하나의 단계를 나타내게 된다. 터치 포인트의 유지 시간과 이동 거리 등을 바탕으로 "터치 단계"를 구분할 수 있으며, 일 예로 터치 포인트가 처음 생성되면 '탭 단계'로 볼 수 있고, 터치 포인트가 일정 시간 유지되면 '눌림 단계'로 볼 수 있으며, 터치 포인트가 일정 거리 이상 이동하게 되면 '움직임 단계'로 볼 수 있다. 따라서, 터치 포인트의 "터치 단계"는 탭 단계, 눌림 단계, 움직임 단계로 구분될 수 있다.
- [0036] 2) 장치 상태: 터치 장치(120)에 대해 각각의 터치 단계에 있는 여러 개의 터치 포인트가 있을 수 있다. 각 터치 단계에 있는 터치 포인트의 수를 세어 나열한 것을 "장치 상태"라고 정의한다. 예를 들어, 터치 장치(120)에서 일어난 터치 포인트들 중 탭 단계에 있는 터치 포인트가 a개, 눌림 단계에 있는 터치 포인트가 b개, 움직임 단계에 있는 터치 포인트가 c개라고 할 때, "장치 상태"는 [a, b, c]가 된다. 즉, N개의 터치 단계를 가지는 시스템의 "장치 상태"는 N개의 숫자가 나열된 리스트(list)로 표시될 수 있다.
- [0037] 3) 제스처: 터치 장치(120)에서의 터치 단계가 변화함에 따라 장치 상태가 변화할 수 있다. 제스처는 장치 상태의 변화로 정의될 수 있다. 좀 더 구체적으로, 제스처는 장치 상태의 조합으로 정의되는 것으로, 장치 상태를 원소로 하여 임의의 길이를 가지는 리스트로 정의될 수 있다. 예를 들어, {[0,0,0], [1,0,0], [0,0,0]}은 터치 장치(120)에 터치된 손가락이 없다가(=[0,0,0]), 한 손가락의 터치로 탭 단계에 있는 터치 포인트가 하나 생성되고(=[1,0,0]), 이어 손가락의 터치를 모두 해제하는(=[0,0,0]) 형태의 제스처를 의미한다. 이러한 형태의 리스트 {[0,0,0], [1,0,0], [0,0,0]}를 탭핑(tapping)을 의미하는 제스처로 정의할 수 있다.
- [0038] 상기한 개념들을 활용한 제스처 인식 방법으로, 터치 장치(120)에서 일어나는 장치 상태의 변화를 리스트에 기록해 두었다가, 리스트의 마지막 부분이 미리 정의된 특정 제스처의 리스트와 일치하는 경우 해당 제스처가 수행된 것으로 판단하는 방법을 적용할 수 있다.
- [0039] 본 발명의 실시예에서는 터치 포인트 각각에 대해 세 가지의 터치 단계 중 어느 하나를 결정하고 각 터치 단계에 있는 터치 포인트의 개수를 세어 장치 상태로 정의할 수 있으며, 이러한 장치 상태의 변화를 나열한 것을 제스처로 정의함으로써 장치 상태의 변화 기록과 일치하는 제스처를 인식한 것으로 간주할 수 있다.
- [0040] 도 2는 본 발명의 일 실시예에 있어서, 제스처 인식 시스템의 프로세서가 포함할 수 있는 구성요소의 예를 도시한 도면이다.
- [0041] 도 2에 도시된 바와 같이, 제스처 인식 시스템(100)의 프로세서(140)는 구성요소들로서 제스처 정의부(201), 장치 상태 관리부(202), 장치 상태 변화 기록부(203), 제스처 비교부(204), 및 제스처 실행부(205)를 포함할 수 있다. 이러한 프로세서(140) 및 프로세서(140)의 구성요소들은 제스처 인식 방법을 수행하도록 제스처 인식 시스템(100)을 제어할 수 있다. 이때, 프로세서(140) 및 프로세서(140)의 구성요소들은 스토리지(160)가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다. 또한, 프로세서(140)의 구성요소들은 프로세서(140)에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 예를 들어, 프로세서(140)가 터치 포인트와 관련된 장치 상태를 관리하기 위해 동작하는 기능적 표현으로서 장치 상태 관리부(202)가 사용될 수 있다.
- [0042] 터치 장치(120)는 임의 개수의 손가락 접촉을 감지할 수 있는 장치이며, 이는 터치스크린, 터치패드, 터치 감지가 가능한 키보드 형태의 장치 등이 될 수 있다. 도 3을 참조하면, 터치 장치(120)는 각 터치 포인트( $T_1$ ,  $T_2$ ,  $T_3$ , ...,  $T_n$ )마다 개별적인 터치 단계(탭 단계, 눌림 단계, 움직임 단계)를 부여 및 관리할 수 있으며, 터치 단계

는 접촉 시간, 접촉 후 이동 거리, 접촉 후 접촉 유지 시간이나 누르는 압력 등의 요인에 따라 결정될 수 있다. 도 3의 터치 장치(120)는 터치 포인트( $T_1, T_2, T_3, \dots, T_n$ ) 각각에 대해 3가지의 터치 단계( $P_1, P_2, P_3$ )를 가지는 장치를 표현한 것이고, 각 터치 포인트( $T_1, T_2, T_3, \dots, T_n$ )는  $P_1, P_2, P_3$  셋 중 하나의 터치 단계를 가진다.

[0043] 장치 상태 관리부(202)는 터치 장치(120)를 통해 감지된 터치 포인트와 터치 포인트 별 터치 단계를 바탕으로 장치 상태를 표현 및 관리할 수 있다. 일 예로, 장치 상태 관리부(202)는 각 터치 단계 별로 해당 터치 단계에 있는 터치 포인트의 수를 세어 이를 하나의 튜플(tuple)로 표현할 수 있다. 튜플의 원소 개수는 터치 단계의 가짓수, 즉 3개이다. 도 3을 참조하면, 장치 상태 관리부(202)는 터치 장치(120)의 감지 결과를 튜플(tuple)로 표현 및 관리하되,  $P_1$  단계에 있는 터치 포인트가 A개,  $P_2$  단계에 있는 터치 포인트가 B개,  $P_3$  단계에 있는 터치 포인트가 C개인 경우 이를 [A, B, C] 형태의 튜플로 표현할 수 있다.

[0044] 장치 상태 변화 기록부(203)는 장치 상태 관리부(202)에서 표현된 장치 상태를 기록하는 역할을 할 수 있다. 특히, 장치 상태 변화 기록부(203)는 장치 상태 관리부(202)를 관찰하다가 장치 상태가 변하는 경우(즉, 터치 포인트의 개수가 변하는 경우) 이를 리스트의 형태로 기록할 수 있다. 도 3에 도시한 바와 같이, 장치 상태 변화 기록부(203)는 장치 상태의 변화를  $\{[A_1, B_1, C_1], [A_2, B_2, C_2], [A_3, B_3, C_3], \dots\}$  와 같은 형태의 튜플 리스트로 기록할 수 있다.

[0045] 제스처 정의부(201)는 제스처 인식 시스템(100)이 구현된 장치에서 사용할 수 있는 제스처를 사전 정의를 통해 저장 및 유지할 수 있다. 제스처 각각은 터치 단계 별 터치 포인트의 개수로 표현된 튜플의 리스트와 해당 제스처의 고유 식별자를 포함할 수 있다. 다시 말해, 제스처 각각은 장치 상태(각 터치 단계에 있는 터치 포인트의 수를 나열한 것)를 원소로 하여 임의의 길이를 가지는 리스트로 정의될 수 있다.

[0046] 제스처 비교부(204)는 장치 상태 변화 기록부(203)에서 기록된 리스트를 제스처 정의부(201)에 저장된 제스처들과 비교하여 특정 제스처를 인식할 수 있다. 구체적으로, 제스처 비교부(204)는 사전 정의된 제스처가 k개의 튜플로 이루어지고 장치 상태 변화 기록부(203)에서 기록된 리스트 중 가장 마지막에 기록된 튜플을 기준으로 k개의 튜플이 사전 정의된 제스처의 튜플과 정확히 일치하는 경우 해당 제스처를 인식한 것으로 판단한다. 제스처 정의부(201)에 보관된 제스처에 따라 임의의 개수의 제스처가 인식될 수 있다. 제스처 비교부(204)는 인식된 제스처가 하나 이상인 경우 해당 제스처의 고유 식별자 집합을 제스처 실행부(205)로 전달한다.

[0047] 제스처 실행부(205)는 제스처 비교부(204)로부터 전달받은 제스처의 고유 식별자를 확인한 후 해당 제스처에 대응되는 작업을 선택하여 해당 작업의 애플리케이션에 명령을 보낸다. 이러한 작업의 예는 마우스 움직이기, 키 입력하기, 마우스 버튼 클릭하기, 특정 기능 실행하기 등이 있을 수 있으며, 애플리케이션의 예는 워드 프로세서, 스프레드 시트, 프리젠테이션, 그래픽 편집기 등이 있을 수 있다.

[0048] 이하에서는 터치 스크린에서 사용 가능한 제스처 키보드를 구현한 예를 설명한다.

[0049] 도 4를 참조하면, 터치 장치(120)에서 터치 포인트의 터치 단계( $P_1, P_2, P_3$ )는 터치 단계 전이도에 따라 결정될 수 있다. 손가락이 최초 접촉하는 경우 해당 터치 포인트는 탭 단계( $P_1$ )에 해당된다. 손가락의 접촉이 일정 시간( $\tau$ ) 이상 지속되는 경우 해당 포인트는 눌림 단계( $P_2$ )에 해당된다. 그리고, 탭 단계( $P_1$ )나 눌림 단계( $P_2$ )에 있는 터치 포인트가 일정 거리( $\delta$ ) 이상 이동하는 경우 해당 터치 포인트는 움직임 단계( $P_3$ )에 해당된다.

[0050] 이때, 장치 상태 관리부(202)는 터치 장치(120)의 인식 결과를 [#탭, #눌림, #움직임]의 형태로 장치 상태를 표현한다. 여기서, #는 해당 단계에 있는 터치 포인트의 개수를 의미한다.

[0051] 도 5는 사전에 정의된 제스처들과 제스처 별 수행 작업들의 예시를 도시한 것이다.

[0052] 도 5를 참조하면, 먼저  $\{[0,0,0], [1,0,0], [0,0,0]\}$ 은 손가락을 가볍게 두드리는 탭 제스처를 의미하고, 장치 상태 변화 기록이  $\{[0,0,0], [1,0,0], [0,0,0]\}$ 과 같을 때 해당 터치 포인트의 위치에 있는 키에 해당하는 키 입력 동작(501)을 실행할 수 있다. 그리고,  $\{[0,0,0], [1,0,0], [0,0,1]\}$ 은 한 손가락을 대고 움직이는 드래그 제스처를 의미하고, 장치 상태 변화 기록이  $\{[0,0,0], [1,0,0], [0,0,1]\}$ 과 같을 때 드래그의 방향에 따라 텍스트 커서를 움직이는 동작(502)을 실행할 수 있다. 또한,  $\{[0,0,0], [1,0,0], [2,0,0], [1,0,1], [0,0,2]\}$ 는 두 손가락을 대고 움직이는 드래그 제스처를 의미하고, 장치 상태 변화 기록이  $\{[0,0,0], [1,0,0], [2,0,0], [1,0,1], [0,0,2]\}$ 와 같을 때 손가락의 움직임 방향에 따라 문서의 스크롤 바를 움직이는 동작(503)을 실행할 수 있다.

- [0053] 기존 제스처 인식 기술에서는 단순하게 한 손가락을 움직이는 제스처로 커서 이동, 두 손가락을 함께 움직이는 제스처로 스크롤 동작을 수행하는 장치를 만들 경우, 두 손가락을 대고 떼는 과정이 동시에 일어나지 않을 경우 한 손가락만 움직이는 상황이 만들어질 수 있다. 구체적으로, {손가락 1 내려놓기, 손가락 2 내려놓기, 손가락 1과 2 움직이기, 손가락 1 떼기, 손가락 2 떼기}의 과정이 순차적으로 일어날 경우, 손가락 1 떼기와 손가락 2 떼기 사이에 손가락 2가 홀로 움직이는 상황이 만들어진다. 이는 한 손가락만 움직이는 상황에 해당하여 의도치 않은 커서 조작 동작이 수행될 수 있다.
- [0054] 이러한 오동작을 해결하기 위해서는 손가락 1을 떼고 일정 시간 동안 커서 조작 동작이 수행되지 않도록 하거나, 최대로 내려놓은 손가락 개수를 세어 특별한 처리를 하거나, 복잡한 유한상태기계(Finite State Diagram)을 이용해 제스처 인식기를 만들어야 하는 문제가 있다.
- [0055] 그러나, 본 발명에 따른 제스처 인식 방법은 단 세 가지의 제스처 정의만으로 모든 동작을 표현할 수 있으며 기존 오동작 문제 또한 자연스럽게 해결할 수 있다. 예를 들어, {손가락 1 내려놓기, 손가락 2 내려놓기, 손가락 1과 2 움직이기, 손가락 1 떼기, 손가락 2 떼기}에서 {손가락 1 떼기, 손가락 2 떼기}의 과정이 {[0,0,2], [0,0,1], [0,0,0]}의 형태로 표현되기 때문에 커서 움직임에 해당하는 {[0,0,0], [1,0,0], [0,0,1]} 제스처로 인식되는 오류가 발생하지 않는다.
- [0056] 도 6 내지 도 7은 도 5를 통해 설명한 제스처를 확장하여 마우스 동작을 수행하는 제스처 키보드의 예시를 도시한 것이다.
- [0057] 도 6에서 터치 단계는 도 4를 통해 설명한 바와 같이 탭 단계, 눌림 단계, 움직임 단계로 3가지 단계로 적용된다. 표현 상의 편의를 위하여 사전 정의된 제스처들은 [A,B,C] 형태의 튜플을 ABC 형태로 단순화 하고, 같은 고유 식별자를 가지는 제스처들을 한데 모아 도 6의 표로 표현하였다. 도 7의 조작 방법은 도 5와 비교하여 매우 복잡한 조작을 수행하는 예시들이다. 이를 본 발명에 따른 방법으로 정의할 경우 19개의 제스처 정의를 이용하여 나타낼 수 있다.
- [0058] 각 제스처에 대한 구체적인 설명은 다음과 같다.
- [0059] 1. Keystroke (701): 탭 동작은 키 입력을 수행한다.
- [0060] - {100,000}, {200,100}, {300,200}: 타이핑과 같은 빠르고 연속적인 탭 동작 중에는 두 손가락 탭이 시간적으로 겹쳐지는 경우가 많다. 따라서, 탭 된 손가락이 떨어질 때 (즉, 탭 단계의 손가락이 감소될 때) 키 입력을 수행한다.
- [0061] 2. Cursor move (702): 한 손가락을 내려놓고 바로 움직이거나, 잠시 손가락을 대고 있는 경우 마우스 커서 조작을 수행한다.
- [0062] - {100,001}, {100,010}, {010,001}: 손가락을 내려놓고 최종 상태가 001이나 010이 된다.
- [0063] 3. Left click, Right click (703): 커서 조작을 하는 경우 (장치 상태가 001, 010에서) 추가로 한 손가락 탭을 할 경우 좌측 버튼 클릭 기능을 실행한다. 두 손가락 탭을 할 경우 우측 버튼 클릭 기능을 실행한다.
- [0064] - {010,110,010}, {001,101,001}: 010, 001 상태에서 추가로 탭 기능을 수행한다.
- [0065] - {001,101,201,101,001}, {010,110,210,110,010}: 010, 001 상태에서 추가로 두 손가락 탭 기능을 수행한다.
- [0066] 4. Left drag, Right drag (704): 마우스 드래그 조작을 위해서, 커서 조작 중 추가적인 손가락을 내려놓은 후 원래 커서를 조작하던 손가락을 움직이면 마우스 드래그 조작이 실행된다.
- [0067] - {101,011}, {110,011}, {110,020,011}: 한 손가락을 내려놓아 눌림 상태가 된 상황에서 또 다른 손가락을 움직이는 조작을 수행한다(최종 상태가 011이 되고, 마우스 좌측 버튼 드래그 동작을 수행한다).
- [0068] - {201,111,021}, {120,021}: 두 손가락을 내려놓고 다른 손가락을 움직이는 조작을 수행한다(최종 상태가 021이 되고, 마우스 우측 버튼 드래그 동작을 수행한다).
- [0069] 5. Left and right release: 드래그 조작이 수행 중에 이 제스처가 인식될 경우 마우스 버튼을 놓는 동작을 수행한다.
- [0070] - {011,001}, {010,000}, {001,000}: 021이나 011 상태에서 000 상태로 갈 수 있는 모든 경로이다.
- [0071] 6. Mouse wheel scroll (705): 마우스 휠을 이용한 스크롤 동작을 수행한다.

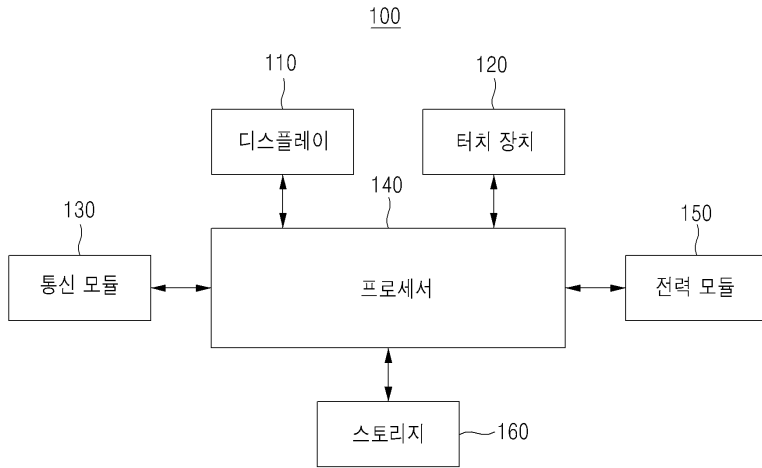
- [0072] - {200,101,002} : 두 손가락을 내려놓고 바로 드래그 하는 조작이다.
- [0073] 본 발명에 따른 제스처 인식 기술의 장점은 제스처가 바뀌어도 장치의 기본 구조와 제스처 인식 장치를 변경하지 않은 채 최소한의 변경으로 새로운 제스처를 사용할 수 있다는 점이다. 실제로, 도 5의 장치를 도 6과 도 7의 장치로 업데이트 하는 경우 전체 구조는 그대로 재사용할 수 있으며, 제스처 정의부와 제스처 실행부만을 교체하면 이와 같은 복잡한 조작을 구현할 수 있게 된다.
- [0074] 본 발명에 따른 제스처 인식 방법은 도 1 내지 도 7을 설명한 상세 내용을 바탕으로 둘 이상의 동작들을 포함할 수 있다.
- [0075] 이처럼 본 발명의 실시예들에 따르면, 본 발명에서 제안하는 제스처 정의는 짧은 길이의 숫자를 나열하는 것으로 특정 제스처를 명확하게 정의할 수 있다는 장점이 있다. 또한, 제스처의 인식에 있어 직관적이고 단순한 방법을 사용하기 때문에 개발자뿐만 아니라 일반 사용자들도 쉽게 임의의 제스처를 추가하고 삭제하는 것이 용이하다. 또한, 제스처의 인식 방법이 복잡하지 않고 명료하기 때문에 제스처 인식 정확도가 높다는 장점이 있다. 따라서, 본 발명의 효과로 멀티 터치 스크린 휴대폰이나 태블릿 장치, 멀티 터치가 부가된 터치 패드나 키보드, 테이블탑 장치 등 다양한 멀티 터치 장치에 쉽게 제스처 인식 기능을 부가할 수 있으며 이를 통해 사용자가 멀티 터치 제스처를 쉽게 추가하거나 편집할 수 있어 그 활용도가 높다.
- [0076] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 프로세서, 컨트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 어플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(configuration)도 가능하다.
- [0077] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [0078] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.
- [0079] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될

수 있다.

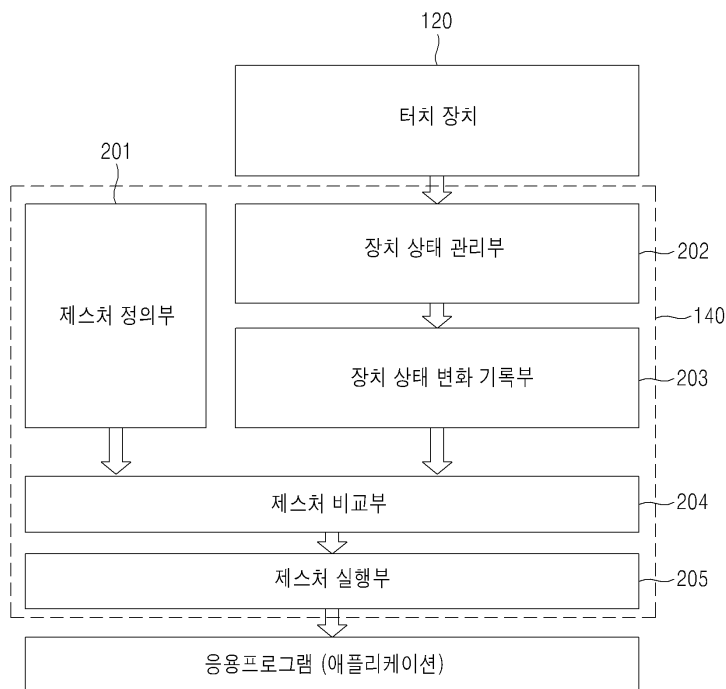
[0080] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

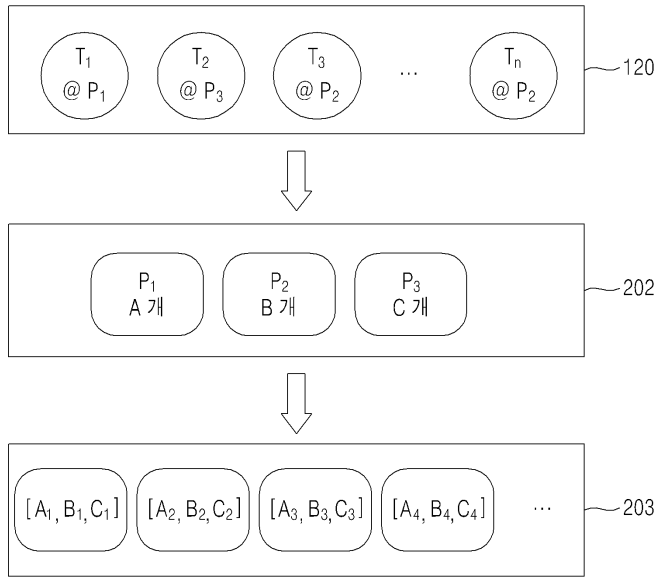
도면1



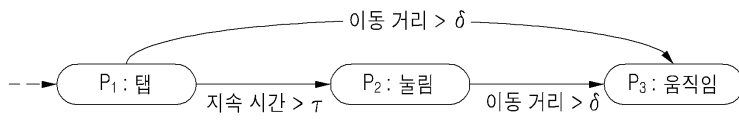
도면2



도면3



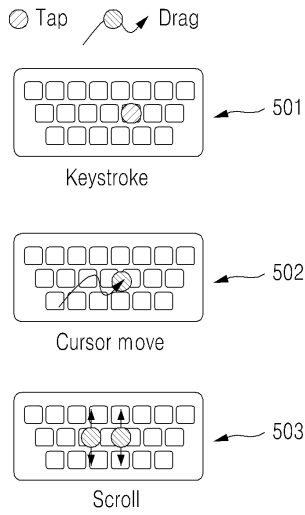
도면4



도면5

제스처 정의부

{ [0,0,0], [1,0,0], [0,0,0] } => 탭  
 { [0,0,0], [1,0,0], [0,0,1] } => 드래그  
 { [0,0,0], [1,0,0], [2,0,0], [1,0,1], [0,0,2] } => 스크롤



도면6

제스처 정의	수행할 기능
{100,000},{200,100},{300,200}	Keystroke
{100,001},{100,010},{010,001}	Mouse cursor move
{010,110,010},{001,101,001}	Mouse left-button click
{001,101,201,101,001},{010,110,210,110,010}	Mouse right-button click
{101,011},{110,011},{110,020,011}	Mouse left-button press and drag
{201,111,021},{120,021}	Mouse right-button press and drag
{011,001},{010,000},{001,000}	Mouse left and right release
{200,101,002}	Mouse-wheel scroll

도면7

