(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0155590
(43) 공개일자 2021년12월23일

(51) 국제특허분류(Int. Cl.)
G06F 3/01 (2006.01) *G06F 3/00* (2006.01)
G06K 9/00 (2006.01) *G06N 20/00* (2019.01)
G09B 5/06 (2006.01)
(52) CPC특허분류
G06F 3/017 (2013.01)
G06F 3/005 (2013.01)
(21) 출원번호 10-2020-0073011
(22) 출원일자 2020년06월16일
심사청구일자 2020년06월16일

(71) 출원인
한국기술교육대학교 산학협력단
충청남도 천안시 동남구 병천면 충절로 1600 (한국기술교육대학교내)
(72) 발명자
성상만
충청남도 천안시 서북구 시청로 39 101-401(불당동, 대동다숲아파트)
이현수
충청남도 천안시 동남구 병천면 충절로 1638 신한아파트 105동 206호
(뒷면에 계속)
(74) 대리인
장수현

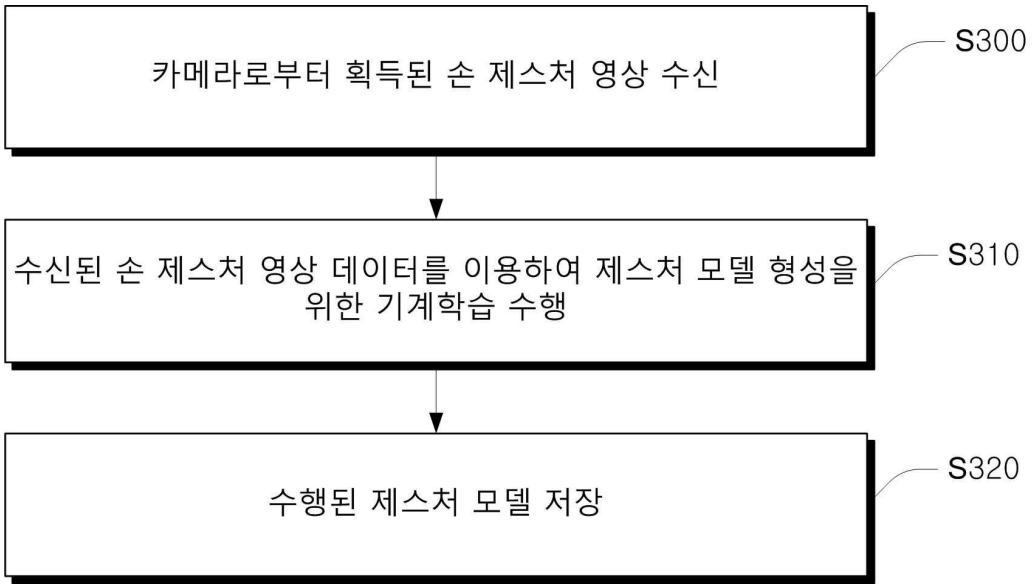
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 **가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델 학습 방법 및 장치**

(57) 요 약

본 발명은 사용자 정의 제스처 모델 학습에 관한 것으로서, 더욱 상세하게는 가상훈련 콘텐츠의 활용시 항상 손에 들고 있어야 하는 컨트롤러 대신 사용자 정의 손 제스처를 학습시켜 사용자에 따른 맞춤형 기능을 가능하게 하는 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델 학습 방법 및 장치에 관한 것이다.

대 표 도 - 도1



(52) CPC특허분류
G06K 9/00335 (2013.01)
G06N 20/00 (2021.08)
G09B 5/065 (2013.01)
(72) 발명자
이원형
서울시 은평구 연서로44길 55, 428동 603호
박정혁
서울시 양천구 신정로 300 신정5차현대아파트 505-701

명 세 서

청구범위

청구항 1

가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델을 학습하는 방법으로서,

(a) 카메라로부터 획득된 손 제스처 영상을 수신하는 단계;

(b) 상기 단계 (a)에서 수신된 손 제스처 영상 데이터를 이용하여 제스처 모델 형성을 위한 기계학습(Machine Learning)을 수행하는 단계; 및

(c) 상기 단계 (b)에서 수행된 제스처 모델을 저장하는 단계

를 포함하는 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델 학습 방법.

청구항 2

청구항 1에 있어서,

상기 손 제스처는 손모양 및 손동작

을 포함하는 것을 특징으로 하는 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델 학습 방법.

청구항 3

청구항 1에 있어서,

상기 단계 (b)의 기계학습은,

ART 알고리즘의 클러스터링 방식으로 이루어지는 것

을 특징으로 하는 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델 학습 방법.

청구항 4

청구항 1에 있어서,

상기 단계 (b)의 기계학습은,

Levenshtein Distance 방식으로 이루어지는 것

을 특징으로 하는 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델 학습 방법.

청구항 5

청구항 1에 있어서,

상기 단계 (c)의 저장은,

패턴별 카테고리로 구분되어 저장되는 것

을 특징으로 하는 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델 학습 방법.

청구항 6

가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델을 학습시키는 장치로서,

적어도 하나의 프로세서; 및

컴퓨터로 실행가능한 명령을 저장하는 적어도 하나의 메모리를 포함하되,

상기 적어도 하나의 메모리에 저장된 상기 컴퓨터로 실행가능한 명령은, 상기 적어도 하나의 프로세서에

의하여,

(a) 카메라로부터 획득된 손 제스처 영상을 수신하는 단계;

(b) 상기 단계 (a)에서 수신된 손 제스처 영상 데이터를 이용하여 제스처 모델 형성을 위한 기계학습(Machine Learning)을 수행하는 단계; 및

(c) 단계 (b)에서 수행된 제스처 모델을 저장하는 단계

가 실행되도록 하는 사용자 정의 제스처 모델 학습 장치.

청구항 7

청구항 5에 있어서,

상기 사용자 정의 제스처 모델 학습 장치는 카메라 모듈과 연결되는 것

을 특징으로 하는 사용자 정의 제스처 모델 학습 방법.

청구항 8

가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델을 학습시키기 위한 컴퓨터 프로그램으로서,

비일시적 저장 매체에 저장되며, 프로세서에 의하여,

(a) 카메라로부터 획득된 손 제스처 영상을 수신하는 단계;

(b) 상기 단계 (a)에서 수신된 손 제스처 영상 데이터를 이용하여 제스처 모델 형성을 위한 기계학습(Machine Learning)을 수행하는 단계; 및

(c) 단계 (b)에서 수행된 제스처 모델을 저장하는 단계

가 실행되도록 하는 명령을 포함하는 사용자 정의 제스처 모델을 학습시키는 컴퓨터 프로그램.

발명의 설명

기 술 분 야

[0001] 본 발명은 사용자 정의 제스처 모델 학습에 관한 것으로서, 더욱 상세하게는 가상훈련 콘텐츠의 활용시 항상 손에 들고 있어야 하는 컨트롤러 대신 사용자 정의 손 제스처를 학습시켜 사용자에 따른 맞춤형 기능을 가능하게 하는 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델 학습 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 가상환경을 통한 콘텐츠들이 증가하면서 가상현실 상에서의 교육 프로그램도 활발히 개발 되고 있다. 특히 가상현실을 이용하면 현실적으로 재현이 어려운 상황들을 실감적으로 체험할 수 있고, 원거리 상황에서도 제약 없이 콘텐츠 제공이 가능하기 때문에, 현실적인 제약을 극복하여 다양한 교육환경 및 콘텐츠를 제공할 수 있다는 면에서 교육 프로그램 응용분야가 급속히 증가하고 있다. 특히, 직업훈련 같은 경우 몰입도 증대와 안전한 상호작용을 통한 훈련 효과 향상이 실제적인 장점으로 부각되고 있다.

[0003] 다양한 IT 기술의 융합을 통해 구현되는 가상현실 기술은 사용자의 체험영역을 확대하고 물리적인 에너지와 각종 비용을 절감하는 기술로 최근에 많은 주목을 받고 있으며, 글로벌 기업들이 경쟁적으로 기술개발을 추진하고 있다. 그 이유로는 기술변화에 부응하는 첨단기술 장비의 교체수명이 짧아 지속적인 재원 투입에 한계가 있어, 이를 대체할 수 있는 가상훈련 시스템은 매우 효과적이고 높은 수요를 발달시키고 있고, 특히 고위험, 고비용의 산업훈련을 대체하는 장비실습에 대한 체험형, 실습형 가상훈련(Virtual Training) 콘텐츠 요구가 증대하고 있으며, 가상공간에서 인터랙션이 가능한 형태의 학습 시스템은 가상현실 시스템의 실감 상호작용, Natural User Interface 기술과 관련된 신개념의 인터페이스 장치 및 소프트웨어 기술은 향후 다양한 산업, 교육, 의료, 국방, 엔터테인먼트 등 많은 산업분야에서 적용될 것으로 예상되어 시장 확대가 전망되기 때문이다.

[0004] 따라서 가상현실기기의 발달과 더불어 가상현실 상에서의 교육 프로그램의 개발이 활발해지고 있으며, 원거리 상황, 혹은 현실적으로 재현이 어려운 상황 들을 실감적으로 체험하고 실습할 수 있는 가상현실 훈련 콘텐츠는 현실적인 제약을 극복하여 다양한 교육환경 및 콘텐츠를 제공한다는 측면에서 응용분야가 급속히 증가하는 추세

이다. 특히, 직업훈련에 가상현실 기술을 접목할때, 몰입감의 증대와 상호작용을 통한 훈련 효과 및 학습전이 향상이 실제적인 장점으로 부각되고 있다.

[0005] 그러나 가상현실 기반 훈련시 사용자의 장비 착용은 큰 부담으로 작용하고 있으며, 이는 훈련의 실재감과 몰입감을 낮추는 단점이 있다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) KR 10-2003-0065620 A

발명의 내용

해결하려는 과제

[0007] 본 발명은 이와 같은 문제점을 해결하기 위하여 창안된 것으로서, 가상훈련 콘텐츠의 활용 시 항상 손에 들고 있어야 하는 컨트롤러 대신 사용자 정의 손 제스처를 인식하여 사용자에 따른 맞춤형 기능을 가능하게 하여 실재감과 몰입감을 높여 가상훈련 콘텐츠를 동작시키기 위한 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델 학습 방법 및 장치를 제공하는 것을 그 목적으로 한다.

과제의 해결 수단

[0008] 이와 같은 목적을 달성하기 위하여 본 발명에 따른 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델을 학습하는 방법으로서, (a) 카메라로부터 획득된 손 제스처 영상을 수신하는 단계; (b) 상기 단계 (a)에서 수신된 손 제스처 영상 데이터를 이용하여 제스처 모델 형성을 위한 기계학습(Machine Learning)을 수행하는 단계; 및 (c) 단계 (b)에서 수행된 제스처 모델을 저장하는 단계를 포함한다.

[0009] 상기 손 제스처는 손모양 및 손동작을 포함하는 것이다.

[0010] 상기 단계 (b)의 기계학습은, ART 알고리즘의 클러스터링 방식으로 이루어지는 것이다.

[0011] 상기 단계 (b)의 기계학습은, Levenshtein Distance 방식으로 이루어지는 것

[0012] 이다.

[0013] 상기 단계 (c)의 저장은, 패턴별 카테고리로 구분되어 저장되는 것이다.

[0014] 이와 같은 목적을 달성하기 위한 본 발명의 다른 측면은 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델을 학습시키는 장치로서, 적어도 하나의 프로세서; 및 컴퓨터로 실행가능한 명령을 저장하는 적어도 하나의 메모리를 포함하되, 상기 적어도 하나의 메모리에 저장된 상기 컴퓨터로 실행가능한 명령은, 상기 적어도 하나의 프로세서에 의하여, (a) 카메라로부터 획득된 손 제스처 영상을 수신하는 단계; (b) 상기 단계 (a)에서 수신된 손 제스처 영상 데이터를 이용하여 제스처 모델 형성을 위한 기계학습(Machine Learning)을 수행하는 단계; 및 (c) 단계 (b)에서 수행된 제스처 모델을 저장하는 단계가 실행되도록 한다.

[0015] 상기 사용자 정의 제스처 모델 학습 장치는 카메라 모듈과 연결되는 것이다.

[0016] 이와 같은 목적을 달성하기 위한 본 발명의 또 다른 측면은 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델을 학습시키기 위한 컴퓨터 프로그램으로서, 비일시적 저장 매체에 저장되며, 프로세서에 의하여, (a) 카메라로부터 획득된 손 제스처 영상을 수신하는 단계; (b) 상기 단계 (a)에서 수신된 손 제스처 영상 데이터를 이용하여 제스처 모델 형성을 위한 기계학습(Machine Learning)을 수행하는 단계; 및 (c) 단계 (b)에서 수행된 제스처 모델을 저장하는 단계가 실행되도록 하는 명령을 포함한다.

발명의 효과

[0017] 본 발명에 의하면, 가상현실 기반 훈련시 사용자의 장비 착용의 부담을 줄여주는 효과가 있다.

[0018] 또한 가상현실 훈련의 실재감과 몰입감을 높여주는 효과가 있다.

도면의 간단한 설명

[0019] 도 1은 본 발명에 따른 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델의 학습 방법을 보여주는 도면.

도 2 내지 도 3은 도 1에 따른 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델의 학습 방법의 이해를 돕기 위한 도면.

도 4는 본 발명에 따른 사용자 정의 제스처 모델을 이용한 가상훈련 콘텐츠의 동작을 보여주는 순서도.

도 5 내지 도 7은 도 4에 따른 본 발명의 이해를 돕기 위한 도면.

도 8 내지 도 9는 도 4에 따른 사용자 정의 제스처 모델을 이용한 가상훈련 콘텐츠의 동작에서 활용된 마우스 기능과 메뉴 및 버튼을 보여주는 화면.

도 10 내지 도 14는 도 4에 따른 본 발명의 사용자 정의 제스처 모델을 이용한 가상훈련 콘텐츠의 동작을 위하여 사용자가 직접 데이터를 습득하고 학습시킬 수 있는 인터페이스 화면을 보여주는 도면.

도 15는 본 발명에 따른 사용자 정의 제스처 모델을 이용한 가상훈련 콘텐츠의 동작 시스템의 구성을 보여주는 도면.

발명을 실시하기 위한 구체적인 내용

[0020] 이하 첨부된 도면을 참조로 본 발명의 바람직한 실시예를 상세히 설명하기로 한다. 이에 앞서, 본 명세서 및 청구범위에 사용된 용어나 단어는 통상적이거나 사전적인 의미로 한정해서 해석되어서는 아니되며, 발명자는 그 자신의 발명을 가장 최선의 방법으로 설명하기 위해 용어의 개념을 적절하게 정의할 수 있다는 원칙에 입각하여 본 발명의 기술적 사상에 부합하는 의미와 개념으로 해석되어야만 한다. 따라서, 본 명세서에 기재된 실시예와 도면에 도시된 구성은 본 발명의 가장 바람직한 일 실시예에 불과할 뿐이고 본 발명의 기술적 사상을 모두 대변하는 것은 아니므로, 본 출원시점에 있어서 이들을 대체할 수 있는 다양한 균등물과 변형예들이 있을 수 있음을 이해하여야 한다.

[0021] 도 1은 본 발명에 따른 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델의 학습 방법을 보여주는 도면이고, 도 2 내지 도 3은 도 1에 따른 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델의 학습 방법의 이해를 돕기 위한 도면으로, 도 2는 도 1에 따른 기계학습의 이해를 돕기 위한 데이터 궤적에 따른 영역 구분을 보여주는 도면이고, 도 3은 도 1에 따른 기계학습의 이해를 돕기 위한 두 궤적간의 Levenshitein Distance를 설명하기 위한 도면이다.

[0022] 사용자 정의 제스처 모델 학습은 기계학습을 통하여 형성되는데, 이때 기계학습은 ART 알고리즘의 클러스터링(Clustering) 기법을 활용한다. 이는 기본적으로 ART는 비지도 클러스터링으로 데이터가 추가될 때마다 클러스터링이 자동으로 빠르게 이루어지며, 또한 최적화를 위해 반복적인 학습이 필요하지 않기 때문에 적합하다. ART 알고리즘은 데이터의 영역을 안과 밖으로 구분하기 때문에, 데이터 입력이 있을 때마다 기존에 형성된 영역에 새 데이터가 포함되느냐 또는 아니냐 그리고 기존 영역을 확장하느냐로 바로 판단이 가능하다.

[0023] 먼저 도 1에 도시된 바와 같이 사용자 정의 제스처 모델의 학습은 카메라로부터 획득된 손 제스처 영상을 수신한다(S300). 여기서 손 제스처는 손모양 및 손동작을 포함한다.

[0024] 그리고 단계 S300에서 수신된 손 제스처 영상 데이터를 이용하여 제스처 모델 형성을 위한 기계학습을 수행한다(S310). 여기서, 기계학습은 도 2에 도시된 바와 같이, 도 2의 오른쪽(b)에서와 같이 빨간선은 연속적으로 들어오는 입력의 궤적이고, 파란박스는 입력에 따라 자동으로 영역을 구분한 결과이다. 이때 파란 박스 바깥쪽도 가까운 순서에 따라 영역이 구분되며, 거리가 있는 경우 이를 반영하여 영역 소속정도도 계산된다. 손 제스처 영상 데이터가 들어올 때마다 바로 영역을 업데이트시키기 때문에 궤적 입력이 끝나는 파란 박스로 표시된 영역들의 업데이트도 동시에 바로 마무리가 된다.

[0025] 한편 연속된 입력이 한 영역 안에 반복적으로 들어올 때가 있는데, 예를 들면 카메라로부터 초당 최대 60번의 손 제스처 정보가 입력이 되는데, 손동작이 느려 한 영역에서 다른 영역으로 궤적이 옮겨가는데 1초가 걸렸다면, 앞선 영역에서 60번의 손 데이터가 들어온 셈이다. 만약 0.1초가 걸렸다면 6번의 손 데이터가 들어온 셈이 된다. 이렇게 한 영역 안에서 반복적으로 들어오는 입력을 모두 고려한다면, 영역 내 반복되는 손 데이터 샘플의 숫자로 손 움직임의 속도를 고려할 수 있게 된다. 즉, 같은 패턴의 손동작이더라도 빠르게 움직일 때와 느리게 움직일 때를 구분할 수 있다. 그러나 같은 사용자가 동일 제스처를 반복하더라도 속도까지 같게 반복하

는 데에는 어려움이 있기 때문에 영역 내에서 반복되는 손 데이터는 중복처리하지 않는다. 즉 궤적의 영역이 바뀌기 전까지는 손동작의 정보가 동일한 것으로 간주한다.

[0026] 이후 입력된 데이터가 어떤 시퀀스(Sequence)를 갖느냐로 손동작 제스처의 패턴이 결정되는데, 이때 패턴의 결정은 두 시퀀스간의 유사도 비교를 통하여 이루어진다. Edit Distance 라고 불리는 Levenshitein Distance는 두 문자열 간의 차이를 비교하기 위해 몇 번의 수정이 필요한가를 세어서 거리로 정의하고 있다. 예를 들어 "kitten"과 "sitting"의 차이는 총 3번의 수정으로 같은 글자가 될 수 있기에 Levenshitein Distance가 3이 되고, "saturday"와 "sunday"는 3번의 수정으로 같은 글자가 될 수 있기에 Levenshitein Distance가 3이 된다. 이때 문자열을 비교하는 것으로 예를 들었지만, 본 발명에서는 클러스터의 번호 시퀀스를 문자열로 가정하여 비교하면 동일한 방법으로 유사도(거리)를 계산할 수 있다. 다음의 [식 1]은 유사도 값을 계산하는 방법이다.

[0027] [식 1]

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

[0028]

[0029] 그러나 [식 1]의 방법처럼 Recurrsive 하게 계산할 경우, 계산량이 많아지기 때문에 본 발명에서는 Wagner Fischer Algorithm을 이용한다.

[0030] 도 3은 도 1에 따른 기계학습의 이해를 돕기 위한 두 궤적간의 Levenshitein Distance를 설명하기 위한 도면이다.

[0031] 도 3을 참조하면, 도 3의 왼쪽(a)의 그림과 같이 두 개의 다른 패턴(빨간 궤적 1, 2) 입력이 들어왔을 때 바로 갱신되는 클러스터(파란 박스)에 따라 궤적 1, 2는 다음과 같은 차이를 가진다.

[0032] 궤적 1: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

[0033] 궤적 2: (0, 1, 10, 11, 12, 13, 8, 9))

[0034] 두 궤적간의 Levenshitein Distance(Edit Distance)는 6

[0035] 여기서 두 궤적 중 클러스터의 상대적 거리에 대해서는 고려하지 않았기 때문에, 클러스터가 다르다고 거리(distance)를 무조건 1로 카운트하기보다는 클러스터 간의 상대거리를 바탕으로 거리(distance)를 계산하도록 한다. 즉 두 클러스터가 다를 때 오른쪽(b)의 도면과 같이 초록색 화살표로 표시된 거리에 대한 정보를 계산하여 반영하는 것이다.

[0036] 그리고 단계 S310에서 수행된 제스처 모델을 저장하는데(S320), 이때 저장은 패턴별 카테고리로 구분되어 저장되며, 이와 같은 과정은 통하여 가상훈련 콘텐츠의 동작을 위한 사용자 정의 제스처 모델의 학습이 진행된다.

[0037] 도 4는 본 발명에 따른 사용자 정의 제스처 모델을 이용한 가상훈련 콘텐츠의 동작을 보여주는 순서도이며, 도 5 내지 도 7은 도 4에 따른 본 발명의 이해를 돕기 위한 도면이다.

[0038] 먼저 도4를 참조하면 카메라로부터 획득된 손 제스처 영상을 수신한다(S100). 이때 손 제스처는 손모양 및 손동작을 포함하며, 여기서 손 제스처 영상이 입력되면 도 4에 도시된 바와 같이 궤적(빨간색)이 생기고 ART 알고리즘에 따라 그 궤적에 맞는 클러스터(파란색)들이 선택된다. 그리고 선택된 클러스터들의 인덱스들을 시퀀스로 만들어 $C = (c_1,...C_n)$이라고 한다.

[0039] 이어서 단계 S100에서 수신된 손 제스처 영상을 입력으로 하여, 기 학습된 사용자 정의 제스처 모델과 비교한다(S110). 이때 C는 도 6에 도시된 바와 같이 이전에 학습되었던 제스처와 일치하지 않을 수 있으므로 C와 이전에 학습되었던 제스처 패턴들 $P_i = (P_1,...Pm_i)$과 비교한다. 이때 C와 P_i를 비교할 때는 Levenshtein Distance $\mathrm{lev}_{s,t}$를 계산하되 도 7의 Wangner Fischer Algorithm 을 활용한다.

[0040] 도 7의 Wangner Fischer Algorithm 에서는 s가 P_i, t가 C에 매칭된다고 보게 되며, substitution Cost 가 1이

되어야 하는 경우를 개선하여 공간상의 특징을 살려 거리 정보를 추가하도록 한다.

[0041] 이때 도 7의 알고리즘에 의한 Levenshtein Distance의 계산은 다음과 같다.

[0042] 먼저 C_i와 P_j의 거리는 우선, 각 클러스터의 c_i와 p_j의 중심점 z의 위치를 찾으며 이는 다음의 [식 2]를 통해서 계산된다.

[0043] [식 2]

[0044] $$\boldsymbol{z}_k = \frac{\boldsymbol{u}_k + \boldsymbol{v}_k}{2}$$

[0045] 여기서, $\boldsymbol{w}_k = \{\boldsymbol{u}_k, \boldsymbol{v}_k\}$, $\boldsymbol{w}_k$는 클러스터 k의 웨이트 벡터이다.

[0046] 그리고 c_i와 p_j의 거리 $D(c_i, p_j)$는 ART 알고리즘에서 거리 계산식인 다음의 [식 3]을 활용하며, 패턴의 거리에 따른 차이를 줄이기 위해 패턴의 길이 값만큼 으로 나누어준다.

[0047] [식 3]

[0048] $$D(c_i, p_j) = \left|\boldsymbol{z}_{c_i} - \boldsymbol{z}_{p_j}\right| / m_i$$

[0049] 여기서, $|\boldsymbol{q}| = \sum q_i$ 이다.

[0050] 그리고 만약 손 제스처 입력 x의 차원(dimension)이 M이라고 한다면, $lev_{c,pi}$는 0에서 M 사이값을 갖는다.

[0051] 마지막으로, D(.)값을 반영하여 계산된 $lev_{c,pi}$ 값이 가장 작은 패턴을 찾아 다음의 [식 4]에 의하여 결과 값으로 선택한다.

[0052] [식 4]

[0053] $$\text{Gesture number} = \begin{cases} \arg\min_i \text{lev}_{C,P_i} & \text{if } \text{lev}_{C,P_i} \le \Theta \\ \text{None} & \text{otherwise} \end{cases}$$

[0054] 만약 $lev_{c,pi}$ 값이 정해놓은 임계치 $\Theta \in [0, M]$ 값보다 크다면, 아무런 손 제스처 패턴도 선택하지 않는다.

[0055] 이후 단계 S110의 비교에 따른 손 제스처의 인식 결과에 따라 가상훈련 콘텐츠를 구동한다(S120). 여기서 가상훈련 콘텐츠는 예를 들면 배관배선, LED 제작과 관련된 직업훈련 콘텐츠로 대부분의 인터페이스는 마우스 등으로 동작이 가능하지만 본 발명에서는 인식된 손 제스처에 의하여 가상훈련 콘텐츠가 구동된다. 본 발명에서는 일반적으로 버튼 등을 클릭하는 마우스 이벤트 이외에 도 8에 도시된 기능들도 대체 가능하며 도 9에 도시된 콘텐츠의 내부 기능과 연동되는 부분도 구현 가능하다.

[0056] 도 10 내지 도 14는 도 4에 따른 본 발명의 사용자 정의 제스처 모델을 이용한 가상훈련 콘텐츠의 동작을 위하여 사용자가 직접 데이터를 습득하고 학습시킬 수 있는 인터페이스 화면을 보여주는 도면이다.

[0057] 도 10은 기본 버튼 설명을 위한 도면으로, 도 10의 Start Camera, Stop Camera 버튼을 통해서 카메라 센서가 연결만 되어 있으면 켜기, 끄기 동작을 확인할 수 있다. 또한, 본 발명의 사용자 인터페이스를 통해서 학습시킨 제스처를 저장하고자 한다면, Save Model 버튼을 눌러 원하는 위치에 저장시키면 된다. 저장될 파일은 확장자 .gr (Gesture Recognition Model)로 기본 저장된다. 그리고 과거에 학습시켰던 모델을 불러오고자 한다면, Load Model 버튼을 눌러 목표 Directory에서 학습시킨 모델을 불러오면 된다. 학습시킨 모델의 확장자는 .gr (Gesture Recognition Model)만 불러올 수 있다. Load Model 버튼을 누르면 기존 학습 모델은 사라질 수 있기 때문에 경고 메시지가 뜬다. 참고로 손모양 및 손동작 제스처가 함께 한 파일로 저장/불러오기가 가능하다.

[0058] 도 11은 손모양 제스처의 인식 및 등록을 설명하기 위한 도면으로, 도 10의 (1)번 버튼을 눌러 센서를 켜면 (2)번 화면에서와 같이 손을 인식하기 시작한다. 이 때 이미 학습시켜 놓은 모델이 있다면, 손이 화면에 인식된 순간부터는 손모양 제스처 인식이 매 센서 데이터가 들어올 때마다 수행되어 (3)번과 (4)번에 결과로 출력된다. 이 때 손의 순서(3번 First Hand, 4번 Second Hand)는 카메라 이미지 상에서 가장 먼저 보인 손의 순서로, 최대 2개까지 제스처 인식이 동시에 가능하다. 손의 앞뒷면 구분이 쉽지 않기 때문에, 왼손과 오른손의 구분은 이미

지 상에서 좌/우의 위치에 따라 구분 할 수도 있다.

[0059] 그리고 손모양 제스처를 추가로 학습시키고자 한다면, (5)번 버튼을 누르면 기존에 저장되어 있는 제스처 목록들을 확인할 수 있고, 원하는 제스처를 선택하면 된다. 또한, 기존에 없던 제스처를 등록하고 싶을 때에는 (6)번 창에 직접 제스처 이름을 입력하여 사용하면 된다. 학습시킬 제스처를 선택하였다면, 해당 제스처에 해당하는 손모양을 센서 앞에 가져다 놓고 (7)번 Train 버튼을 눌러 학습을 시작시킨다. 이때 학습에 필요한 최소/최대 시간은 정해져 있지 않다. 다만, 같은 손모양이더라도 여러 각도에 따라 센서에서는 다르게 보일 수 있기 때문에, 학습시키는 동안 손을 약간씩 돌려가며 움직여주어 변화에 강인하도록 학습시킨다. 학습이 끝났다고 생각되면 (7)번 버튼을 다시 눌러서 학습을 종료시키면 된다. 만약, 지금까지 학습된 현재 손모양 제스처 모델 전체를 초기화시키고 싶다면, (8)번 Reset 버튼을 누르면 된다.

[0060] 도 12는 손동작 제스처의 인식 및 등록을 설명하기 위한 도면으로, 손동작의 경우는 패턴이 입력되어야 하기 때문에 센서 앞에서 매 프레임마다 학습을 시킬 수 있는 게 아니고, 하나의 제스처 패턴 입력을 수행해야 한다.

[0061] 우선, (1)번 버튼으로 센서를 켠 후, (2)번에서와 같이 손이 화면에 나오면 자동적으로 손동작의 정보가 기록되기 시작한다. 제스처가 인식이 되면 그 결과는 (3)번 창에서 출력이 된다. 현재는 first hand 에 대해서만 모션 인식이 가능하다.

[0062] 제스처가 인식이 되는 타이밍은 앞선 절에서 설명하였듯이, 손이 1초 동안 정지해 있으면 결과가 출력된다. 손이 움직이고 있는지 멈추어있는지는, (4)번 Status 창에서 확인할 수 있으며 손이 움직이고 있을 때에는 moving으로, 멈추었다고 판단되면 ready라고 뜬다.

[0063] 손동작 제스처를 추가로 학습시키고 싶다면, (5)번을 눌러 목록에서 제스처를 선택하거나, 기존에 없던 제스처라면 (6)번에 직접 제스처 이름을 입력한다. 그리고 센서에 손을 위치시키고 (7)번 버튼을 눌러 제스처를 시작하면 된다. 한 번의 제스처가 끝나면 다시 (7)번 버튼을 눌러서 제스처가 끝났음을 확인한다. 만약 추가로 제스처를 학습시키고 싶다면 이 과정을 반복하면 된다.

[0064] 만약 학습시켜 놓았던 제스처 중 하나를 삭제하고 싶다면, (5)번을 눌러 제스처 목록에서 해당 제스처를 선택하고, (8)번 Delete 버튼을 눌러 지우면 된다. 모든 손동작 제스처 정보를 초기화하고 싶다면 (9)번 Reset 버튼을 누르면 된다.

[0065] 도 13은 기능연결을 설명하기 위한 도면으로, 앞서 손모양과 손동작 제스처의 등록이 끝났으면, 실제 콘텐츠와의 연동을 위한 기능과 연결을 시켜주는 작업을 해주어야 한다. 도 13은 앞서 UI 화면에서 좌측 하단을 클로즈업 한 사진이다.

[0066] (1)번 버튼을 누르면 사용자가 연동시킬 수 있는 기능들의 리스트가 기본으로 제공된다. 앞선 내용에서 설명했듯이 마우스와 관련된 기능들과 콘텐츠 내에서 사용될 수 있는 특정 메뉴들이 보이며, 선택이 가능하다. (1)번에서 연동시킬 기능을 선택하면, (6)번과 (9)번에서 현재 연결되어 있는 손모양과 손동작을 확인할 수 있다. 손모양과 손동작이 다르게 학습되었기 때문에, 여기서도 별개로 추가해주도록 구성하였다. 만약 여기서 더 추가하고 싶은 손모양이나 손동작이 있는 경우 (2)번을 눌러 아직 아무 기능에도 연결되어 있지 않은 손모양을 선택하고, 손동작을 추가하고자 한다면 (3)번을 눌러 목록을 열고 선택하면 된다. 손모양의 경우는 (2)번 목록에서 제스처 선택 후 (4)번 화살표 버튼을 눌러 (1)에서 선택된 기능에 연결이 가능하고, 만약 빼고 싶은 것이 있다면, (6)번 리스트에서 빼고자 하는 손모양을 선택한 후 (5)번 화살표 버튼을 누르면 된다. 마찬가지로 손동작의 경우는 (3)번 목록에서 제스처 선택 후 (7)번 화살표 버튼을 눌러 (1)에서 선택된 기능에 연결이 가능하고, 만약 빼고 싶은 것이 있다면, (9)번 리스트에서 선택 후 (8)번 화살표 버튼을 누르면 된다.

[0067] 도 15는 본 발명에 따른 사용자 정의 제스처 모델을 이용한 가상훈련 콘텐츠의 동작 시스템의 구성을 보여주는 도면으로 도 15를 참조하여서는 가상훈련 콘텐츠 동작 장치(20)의 구성요소를 중심으로 간략히 정리하여 설명하기로 한다.

[0068] 도 15에 도시된 바와 같이 가상훈련 콘텐츠 동작 장치(20)는 프로세서(110), 프로그램과 데이터를 저장하는 비휘발성 저장부(120), 실행 중인 프로그램들을 저장하는 휘발성 메모리(130), 다른 기기와 통신을 수행하기 위한 통신부(140), 이들 장치 사이의 내부 통신 통로인 버스 등으로 이루어져 있다. 실행 중인 프로그램으로는, 장치 드라이버, 운영체계(Operating System), 및 다양한 어플리케이션이 있을 수 있다. 도시되지는 않았지만, 전력제공부를 포함한다.

[0069] 가상훈련 콘텐츠 동작 장치는(20)는, 카메라(30)로부터, 영상 데이터 수신 드라이버(150)에 의해 손 제스처 영

상 데이터를 전달받아 이를 가상훈련 콘텐츠 동작 어플리케이션에 있는 사용자 정의 제스처 모델에 입력하고, 사용자 정의 제스처 모델의 출력에 의해 가상훈련 콘텐츠가 동작되도록 하며 이를 모니터를 통하여 출력한다.

[0070] 또한 통신부(140)는 도 1를 참조하여 설명한 바와 같은 사용자 정의 제스처 모델 학습 장치(10)와의 데이터 송수신 역할을 수행하는데, 즉 가상훈련 콘텐츠 동작 장치는 통신부(140)를 통하여, 카메라(30)에서 수신한 영상 데이터를 사용자 정의 제스처 모델 학습 장치로 송신하고, 사용자 정의 제스처 모델 학습 장치로부터(10) 학습된 사용자 정의 제스처 모델을 수신한다.

[0071] 이상과 같이, 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 이것에 의해 한정되지 않으며 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 본 발명의 기술사상과 아래에 기재될 특허청구범위의 균등범위 내에서 다양한 수정 및 변형이 가능함은 물론이다.

부호의 설명

[0072] 10: 사용자 정의 제스처 모델 학습 장치

20: 가상훈련 콘텐츠 동작 장치

30: 카메라

40: 모니터

110: 모니터

120: 저장부

130: 메모리

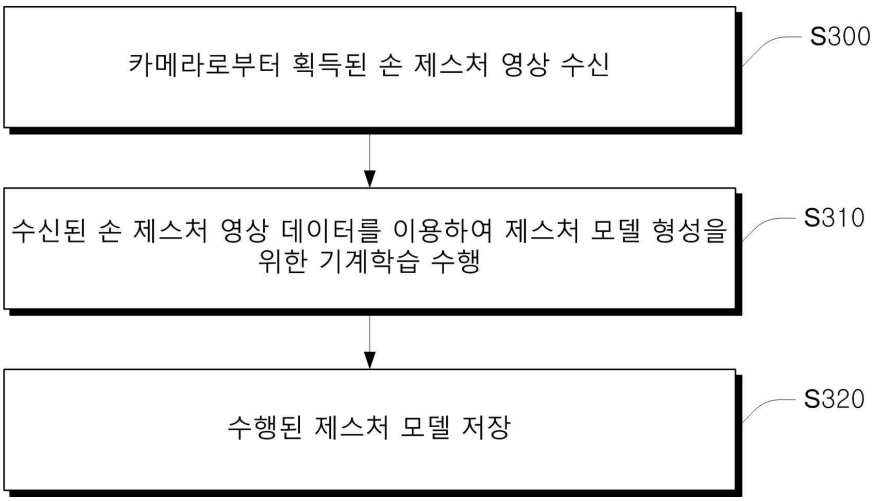
140: 통신부

150: 영상데이터 수신 드라이버

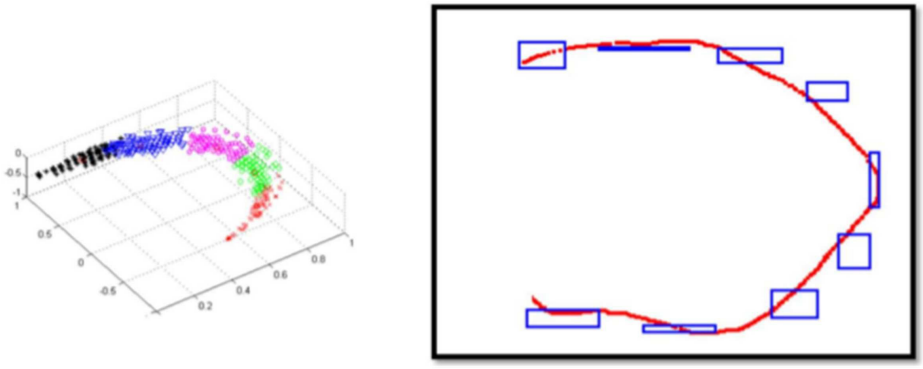
160: UX/UI 출력 드라이버

도면

도면1

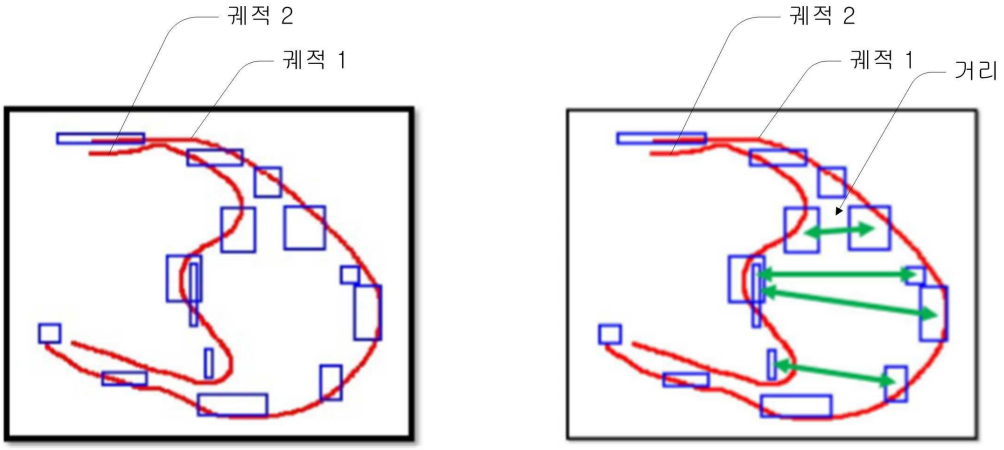


도면2



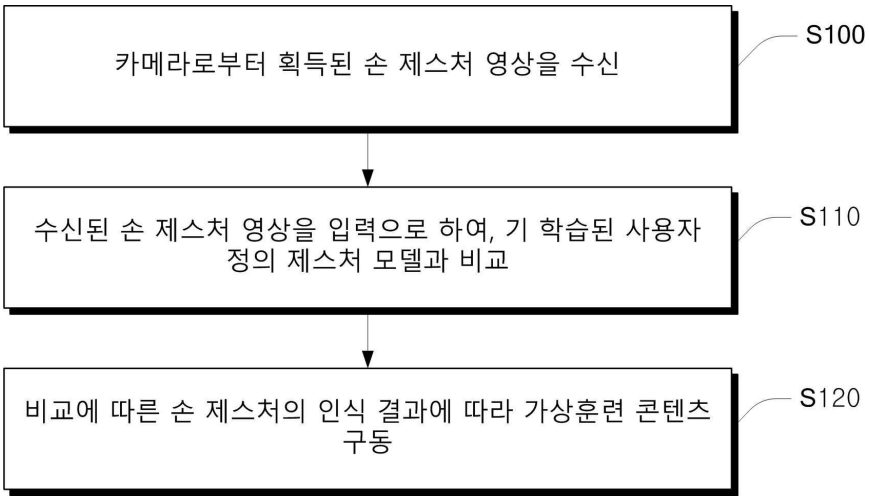
(a) (b)

도면3

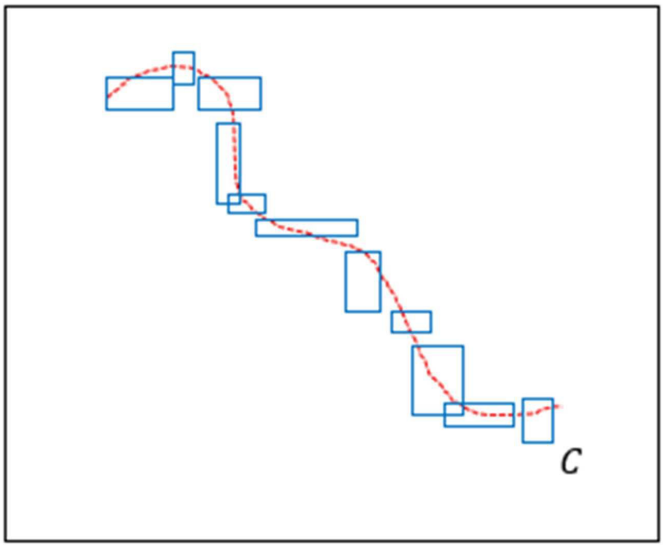


(a) (b)

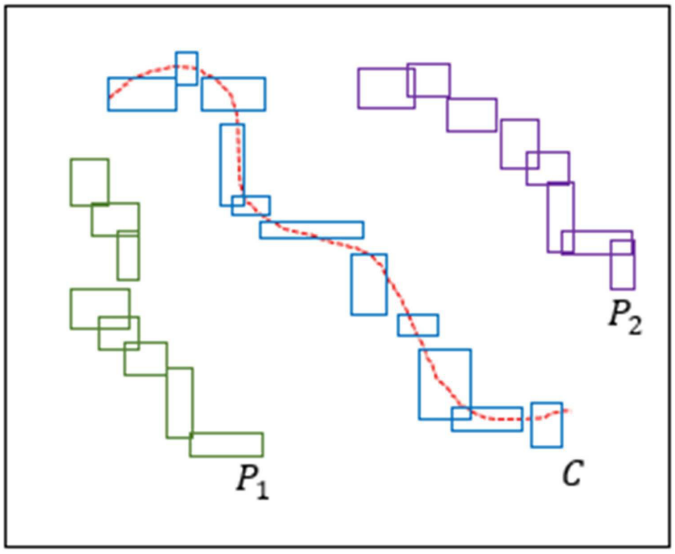
도면4



도면5



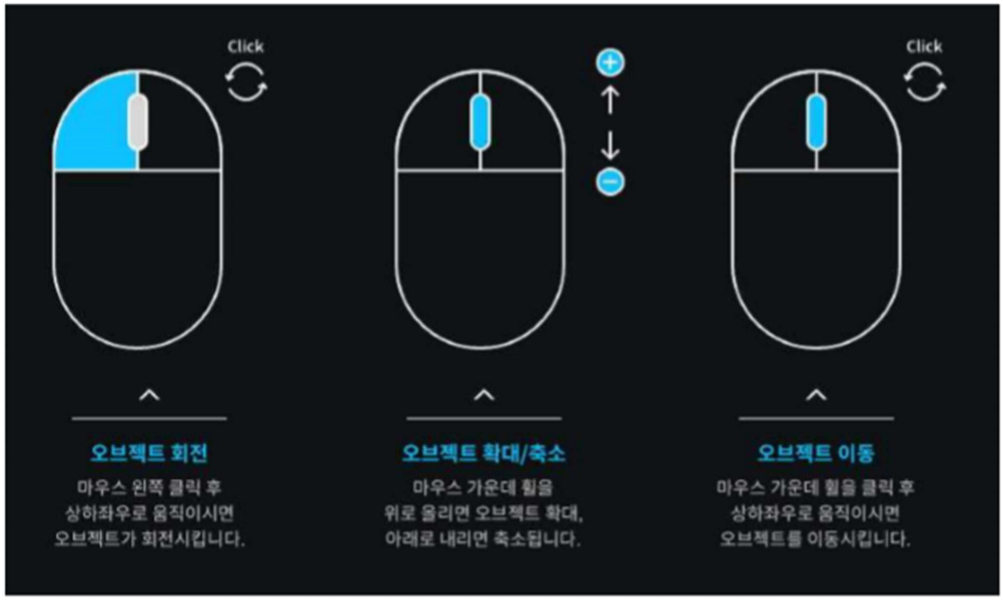
도면6



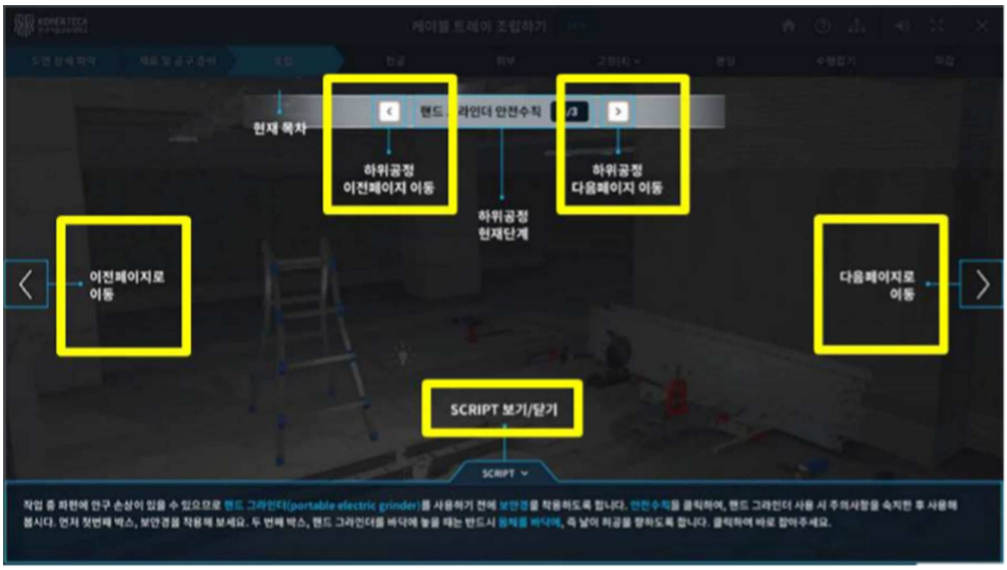
도면7

```
function LevenshteinDistance(char s[1..m], char t[1..n]):
  // for all i and j, d[i,j] will hold the Levenshtein distance between
  // the first i characters of s and the first j characters of t
  // note that d has (m+1)*(n+1) values
  declare int d[0..m, 0..n]

  set each element in d to zero

  // source prefixes can be transformed into empty string by
  // dropping all characters
  for i from 1 to m:
      d[i, 0] := i

  // target prefixes can be reached from empty source prefix
  // by inserting every character
  for j from 1 to n:
      d[0, j] := j

  for j from 1 to n:
      for i from 1 to m:
          if s[i] = t[j]:
            substitutionCost := 0
          else:
            substitutionCost := 1

          d[i, j] := minimum(d[i-1, j] + 1,                  // deletion
                             d[i, j-1] + 1,                  // insertion
                             d[i-1, j-1] + substitutionCost)  // substitution

  return d[m, n]
```

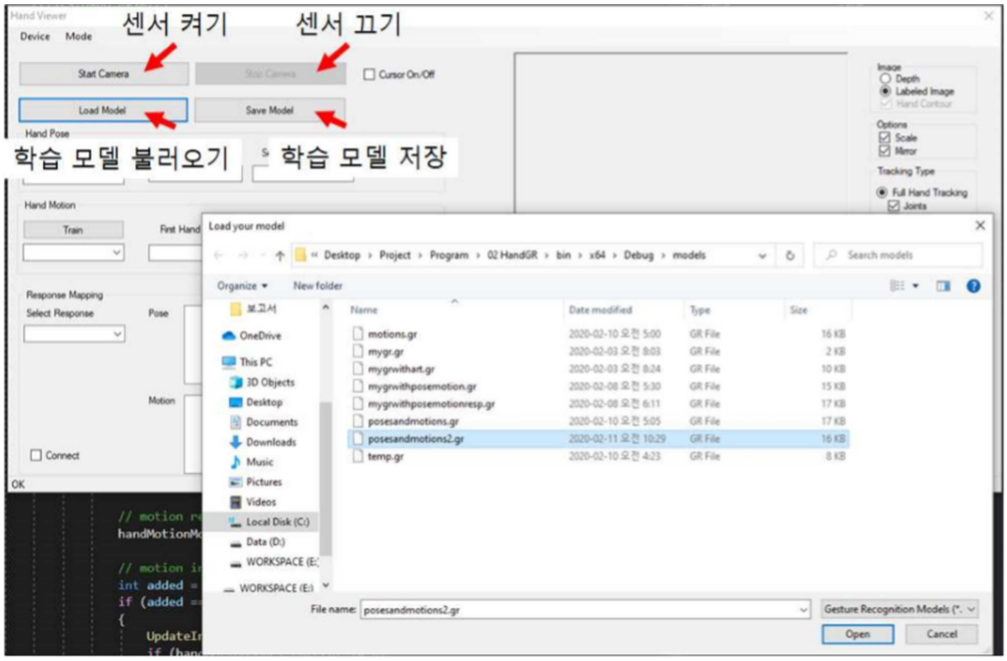
도면8



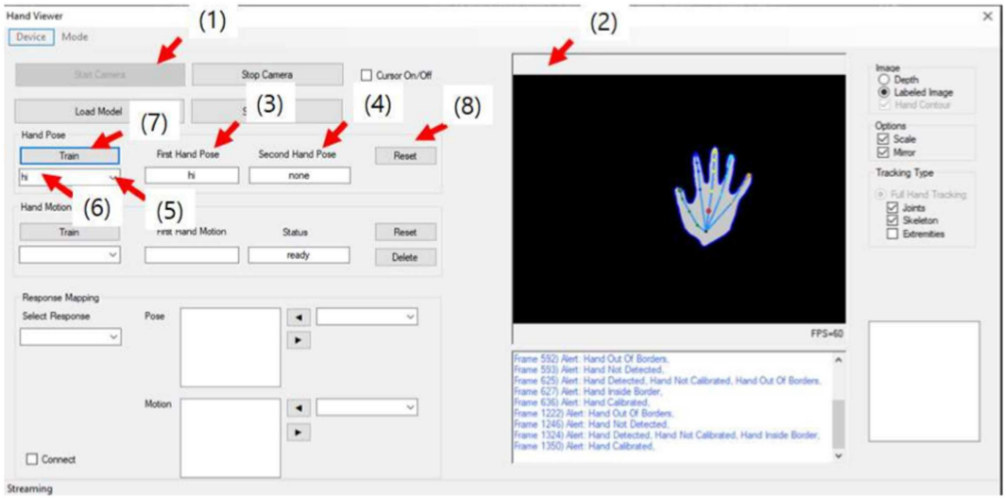
도면9


현재 목차
하위공정
이전페이지 이동
하위공정
다음페이지 이동
하위공정
현재단계
이전페이지로
이동
다음페이지로
이동
SCRIPT 보기/닫기
SCRIPT

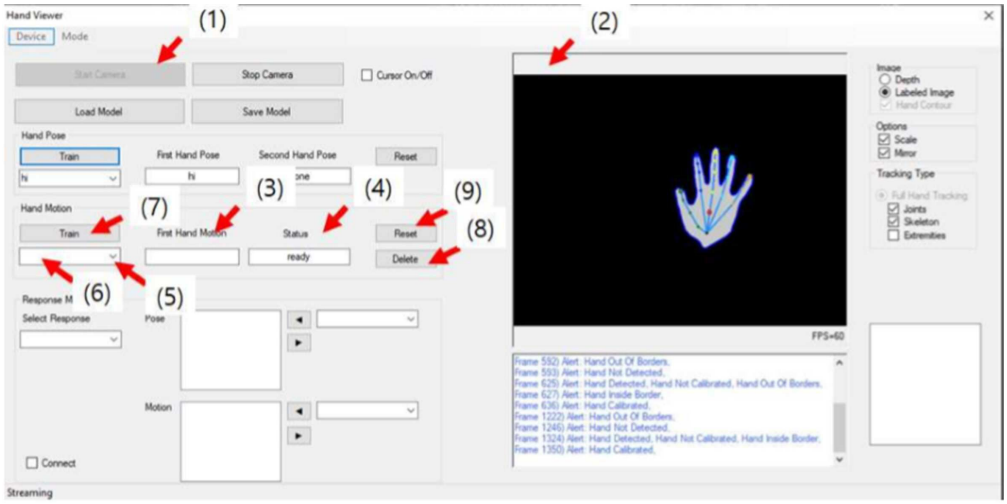
도면10


센서 켜기
센서 끄기
학습 모델 불러오기
학습 모델 저장
Start Camera
Stop Camera
Load Model
Save Model
Load your model
File name: posesandmotions2.gr
Open
Cancel

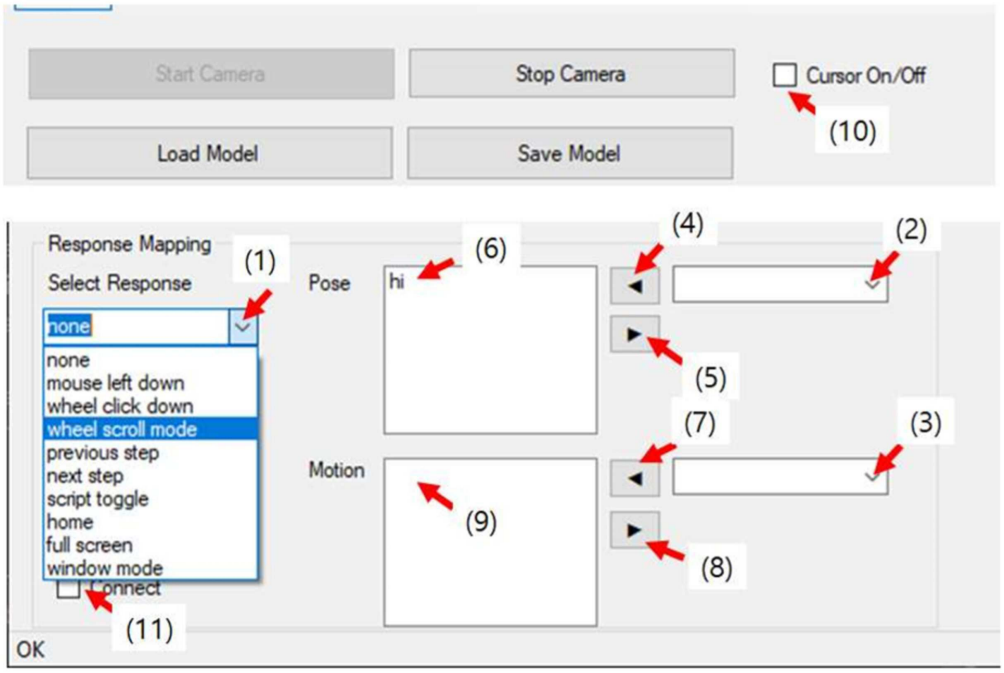
도면11


Hand Viewer
Device
Mode
(1)
(2)
Start Camera
Stop Camera
Cursor On/Off
Load Model
(3)
(4)
(8)
(7)
Hand Pose
Train
First Hand Pose
Second Hand Pose
Reset
hi
none
Hand Motion
(6)
(5)
Train
Status
Reset
ready
Delete
Response Mapping
Select Response
Pose
Motion
Connect
Streaming
Image
Depth
Labeled Image
Hand Contour
Options
Scale
Mirror
Tracking Type
Full Hand Tracking
Joints
Skeleton
Extremities
FPS=60

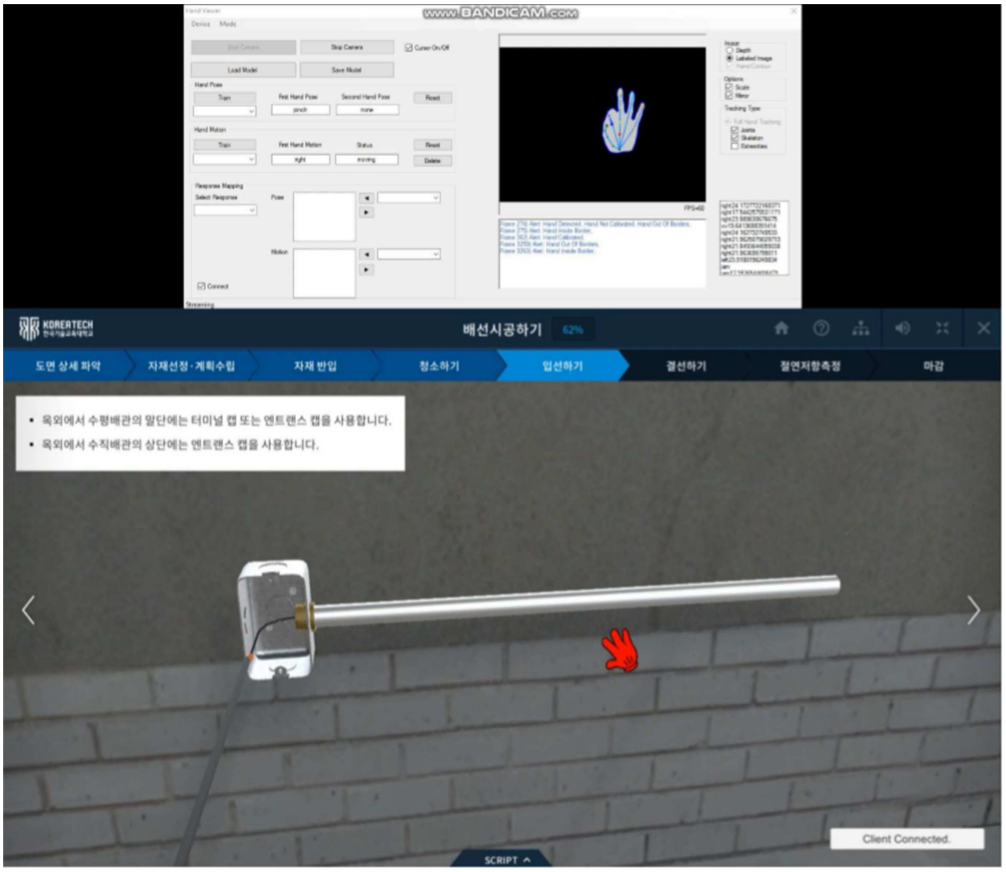
도면12


Hand Viewer
Device
Mode
(1)
(2)
Start Camera
Stop Camera
Cursor On/Off
Load Model
Save Model
(3)
(4)
(9)
(7)
(8)
Hand Pose
Train
First Hand Pose
Second Hand Pose
Reset
hi
Hand Motion
(6)
(5)
Train
Status
Reset
ready
Delete
Select Response
Pose
Motion
Connect
Streaming
Image
Depth
Labeled Image
Options
Scale
Mirror
Tracking Type
Joints
Skeleton
Extremities
FPS=60

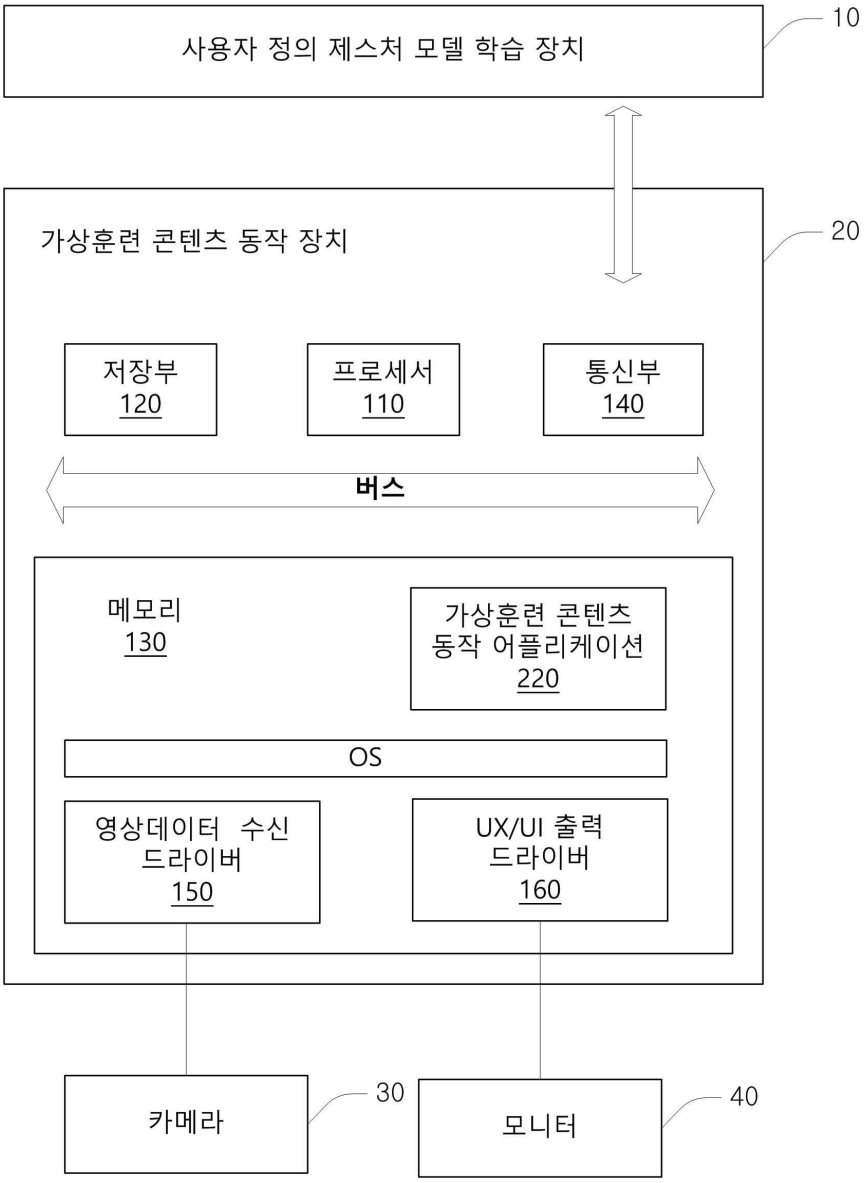
도면13


Start Camera
Stop Camera
Cursor On/Off
(10)
Load Model
Save Model
Response Mapping
Select Response
(1)
none
none
mouse left down
wheel click down
wheel scroll mode
previous step
next step
script toggle
home
full screen
window mode
Connect
(11)
Pose
(6)
hi
(4)
(2)
(5)
Motion
(9)
(7)
(3)
(8)
OK

도면14


www.BANDICAM.com
KOREATECH
배선시공하기
62%
도면 상세 파악
자재선정·계획수립
자재 반입
정소하기
입선하기
결선하기
절연저항측정
마감
• 목외에서 수평배관의 말단에는 터미널 캡 또는 엔트랜스 캡을 사용합니다.
• 목외에서 수직배관의 상단에는 엔트랜스 캡을 사용합니다.
Client Connected.
SCRIPT

도면15


사용자 정의 제스처 모델 학습 장치
10
가상훈련 콘텐츠 동작 장치
20
저장부
120
프로세서
110
통신부
140
버스
메모리
130
가상훈련 콘텐츠
동작 어플리케이션
220
OS
영상데이터 수신
드라이버
150
UX/UI 출력
드라이버
160
카메라
30
모니터
40