제스처 기반 다중 레이어 드로잉 시스템의 설계 및 구현+

김상준*, 최유주**
*서울미디어대학원대학교 뉴미디어학부
**서울미디어대학원대학교 실감미디어연구소, 교신저자
e-mail:gogo5911@naver.com , yjchoi@smit.ac.kr

Design and Implementation of Gesture-based Drawing System Supporting Multiple Layers

Sang-Joon Kim*, Yoo-Joo Choi**
*Dept of New Media, Seoul Media Institute of Technology
**Immersive Media Lab. Seoul Media Institute of Technology

요 약

본 논문에서는 키넥트(kinect)의 인체 추적 기능을 사용하여 다중 레이어 기능을 제공하는 제스처 인식 기반 드로잉 시스템을 설계 구현하였다. 제안된 드로잉 시스템은 제스처를 통해 자유롭게 그림을 그릴 수 있는 시스템으로 그림을 그리는 붓의 굵기 선택, 지우개를 이용한 그림 지우기, 사물을 이용한 물감색 선택, 템플릿을 이용한 그림그리기 등을 수행하는 다양한 제스처가 정의 되어 있고, 제스처를 이용하여 화면에 자유롭게 그림을 그릴 수 있도록 함으로써 사용자의 몰입감과 흥미를 높일 수 있도록 설계 구현되었다. 제안 시스템은 원하는 템플릿을 선택하고 템플릿을 이용하여 색칠하기를 수행하는 템플릿 기반 드로잉 레이어와 템플릿 없이 자유롭게 그림을 그리는 프리 드로잉 레이어를 지원함으로써 다양한 그림 그리기가 가능하도록 구현되었다.

1. 서론

제스처 인식 기술의 발전으로 사용자와 자연스러운 상호작용이 가능한 다양한 콘텐츠들이 제안되고 있다. 이러한 상호작용 콘텐츠는 게임, 교육 분야에서 많이 사용되었지만 현재는 광고, 전시, 영화, 공연 등에서도 널리 사용되고 있다. 자연스러운 직관적 상호작용을 위해서는 사람의 제스처를 인식하는 시스템이 요구된다. 제스처 인식을 위해 사용되는 대표적 하드웨어로 키넥트(Kinect), 립모션(Leap Motion), 마요 암밴드(Myo Armband) 등이 있으며 SDK에는 오픈포즈(Open Pose)가 있다. 키넥트는 컨트롤러 없이 사용자의 신체를 이용하여 게임과 엔터테인먼트를 경험 할 수 있는 엑스박스 주변기기로 실시간 깊이 정보 뿐만 아니라 RGB영상과 관절을 추적하여 제공하는 단말기이다[1]. 립모션은 1/100mm의 정밀도로 손가락과 기타 물체를 인식할 수 있는 입력장치로 근거리 환경을 타겟으로 개발된 단말기이다. 마요 암밴드는 사용자의 팔에 착용하여 사용하는 형태로 근육센서를 이용하여 움직임을 인식하고 6축 가속센서로 팔의 다양한 움직임을 인식하는 단말기이다. 오픈포즈는 Carnegie Mellon University에서 개발된 라이브러리로 사진 또는 영상에서 실시간으로 사람들의 동작에서 몸, 손, 그리고 얼굴의 형태를 추출할 수 있는 소프트웨어 라이브러리이다[2].

이러한 단말기와 소프트웨어들을 기반으로 사람의 손과 몸을 이용한 다양한 형태의 드로잉 시스템이 제안되어 왔다. 첫 번째는 키넥트를 사용하여 손과 손가락을 추적하고 손의 상태에 따라 모래그림을 그릴 수 있는 콘텐츠이다[3]. 두 번째는 립모션을 이용하여 사람의 손의 위치 X, Y, Z 움직임을 추적하여 3D 공간에 그림을 그리는 콘텐츠이다[4]. 세 번째 방법은 마요 암밴드를 이용하여 사람의 손동작을 인식하고 지정된 도화지에 색칠하여 많이 색칠한 쪽이 이기는 게임 형식의 그림 그리기 콘텐츠이다[5]. 이러한 제스처 드로잉 시스템은 제한된 붓의 색상과 제한된 이미지의 크기로 사용자가 원하는 그림을 그리는 것에 있어 제한이 있다.

본 연구에서는 이러한 제한점을 해결하고자 키넥트의 제스처 인식 기능을 이용하여 사용자의 양손의 움직임과 개폐상태를 인식하여 드로잉에 필요한 기능을 선택하고, 일반 사물을 이용하여 물감색을 선택하도록 함으로써 제한된 색상 없이 사용자가 원하는 물체의 색상을 사용하도록 하였다. 또한, 색칠을 위한 템플릿을 선택하여 템플릿을 기반으로 색칠을 하는 템플릿 기반 드로링 레이어와 오른손의 움직임을 통하여 자유롭게 그림을 그릴 수 있는 자유 드로잉 레이어를 지원함으로써 다양한 그림 그리기가 가능하도록 드로잉 시스템을 설계 구현하였다.

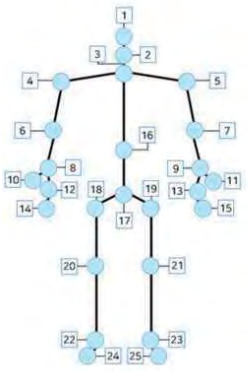
2. 키넥트 센서

+ 본 연구는 한국연구재단 이공학개인기초연구지원사업 (NRF-2017R1D1A1B03035718)에 의하여 수행됨

키넥트는 두 가지 버전이 있으며 표 1에서 보이는 것처럼 지원하는 해상도, 인물 추적 수, 관절의 수 등에 차이가 있다. 키넥트 V1의 경우 사용자의 관절의 수를 추적하기 위해서는 개발자 툴킷(Developer Toolkit)을 사용자가 직접 추가하여야 하지만 키넥트 V2에서는 기본적으로 제공하는 SDK를 이용하여 추적할 수 있다. 또한 키넥트 V1은 여러 응용 프로그램이 동시에 같은 센서에 연결할 수 없지만 키넥트 V2는 키넥트 서비스(Kinect Service)를 추가시키는 것으로 여러 응용 프로그램이 동시에 동일한 센서에서 데이터를 검색 할 수 있도록 할 수 있다. 본 논문에서는 키넥트 V2를 사용하여 제안 시스템을 구현하였다. 그림1은 키넥트 V2를 통하여 감지 할 수 있는 관절을 보여주고 있다.

표 1. Kinect V1과 Kinect V2 센서의 사양비교[6]

		Kinect v1	Kinect v2
색상(Color)	해상도	640x480	1920x1080
	fps	30fps	30fps * 3
심도(Depth)	해상도	320 x 240	512 x 424
	fps	30fps	30fps
인물 자세		2명	6명
관절		20관절	25관절
손의 개폐 상태		△	○
심도의 취득 범위		0.8 ~ 4.0m	0.5 ~ 8.0m
인물의 검출 범위		0.8 ~ 4.0m	0.5 ~ 4.5m



번호	구분	번호	구분
1	Head	14	HandTipLeft
2	Neck	15	HandTipRight
3	SpineShoulder	16	SpinMid
4	ShoulderLeft	17	SpineBase
5	ShoulderRight	18	HipLeft
6	ElbowLeft	19	HipRight
7	ElbowRight	20	KneeLeft
8	WristLeft	21	KneeRight
9	WristRight	22	AnkleLeft
10	ThumbLeft	23	AnkleRight
11	ThumbRigh	24	FootLeft
12	HandLeft	25	FootRight
13	HandRight		

그림 1. 키넥트에서 인식할 수 있는 25개 관절

3. 제안 시스템

본 논문에서 제안하는 드로잉 시스템은 사용자의 골격을 추적하고 사용자의 손의 위치와 개폐상태에 따라 여러 가지 기능들을 제어하고 그림을 그릴 수 있도록 하는 것으로 시스템의 주요 모듈은 그림 2와 같이 키넥트, 인터페이스, 기능, 레이어 모듈로 나눠져 있다. 키넥트 모듈은 키넥트에서 제공하는 카메라 이미지와 골격인식 정보를 받아와 저장하는 역할을 한다. 인터페이스 모듈은 사용자가 사용할 수 있는 기능들과 현재 사용하는 기능들을 사용자가 알기 쉽게 화면에 나타내는 역할을 한다. 기능 모듈은 사용자가 화면에 나타난 인터페이스에 상호작용하거나 미리 지정한 행동과 관련된 상호작용을 했을 경우 실질적으로 선택된 위치값과 기능정보를 레이어 모듈에 전달해주는 모듈이다. 레이어 모듈은 기능 모듈에서 계산된 값을 기반으로 화면에 드로잉 결과로 표현 해주는 모듈이다.

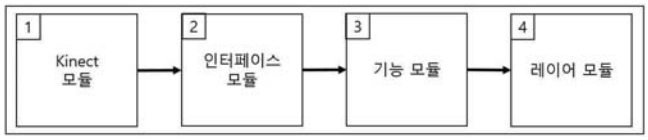


그림 2 s/w 주요 구성 모듈

3.4 제안시스템 처리절차

본 논문의 제안 시스템은 다음과 같이 동작한다. 키넥트 모듈에서 사용자의 골격이 인식되면 사용자의 손의 위치와 개폐상태를 인식한다. 기능 모듈에서는 상단에 있는 메뉴와 오른손의 위치와 개폐상태에 따라 어떤 기능을 사용자에게 전달할지를 검사한다. 상단의 메뉴는 그림 3과 같이 총 5가지 메뉴를 가지고 있다.

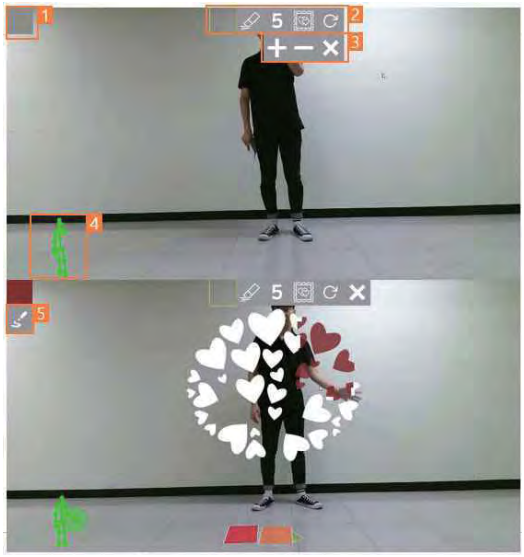


그림 3 드로잉 시스템 인터페이스

첫 번째로 그림 3의 [1]의 영역은 색상 확인 창으로 현재 드로잉하고 있는 색상의 색을 나타낸다. 두 번째로 그림 3의 [2]의 영역은 메인 메뉴 조작기로 드로잉 시스템에서 주로 사용되는 색상인식 영역, 붓과 지우개, 붓과 지우개의 굵기 지정, 템플릿 불러오기, 초기화 기능의 상태를 보여준다. [2]의 영역에서 원하는 기능 영역에 사용자의 손이 위치되어 잠시 머물면 해당 기능이 선택된다. 세 번째로는 그림 3의 [3]의 영역은 서브 메뉴 조작기로 굵기 지정의 +, - 기능과 템플릿의 이미지 선택 기능들과 같은 메인메뉴의 서브 역할의 메뉴를 보여준다. 네 번째로는 그림 3의 [4]는 키넥트가 사용자의 골격을 인식 했는지의 여부를 보여주는 골격 확인창이다. 다섯 번째로 그림 3의

[5]의 영역은 템플릿 레이어 모드에서 지원하는 기능 선택 영역으로 크기 변경, 페인팅, 위치 변경 기능을 선택할 수 있다.

특정 메뉴를 제외한 모든 메뉴는 주먹을 쥔 오른손이 메뉴 속에 들어가 있을 때 동작하며 단일 메뉴가 아닌 경우 2초 뒤에 기능이 변경되거나 동작한다. 예를 들어 붓의 굵기를 줄이고 싶을 경우 - 메뉴에 손을 위치한다. 손의 위치가 계속해서 - 메뉴에 있을 경우 2초마다 붓의 굵기가 줄어든다. 색상인식 영역은 오른손의 개폐상태가 가위일 때 영상의 지정된 영역에서 매 프레임마다 픽셀의 정보를 가져와 RGB값의 평균을 구하고 평균값을 붓의 색상으로 변경하여 사용한다. 사용자는 원하는 색을 가진 실제 사물을 색상 인식 영역에 위치시키고 오른손을 가위로 만들어 원하는 색을 현재색으로 정의할 수 있다. 이로써 그림을 그릴 때에 주어진 팔렛트 내의 물감색으로 제한을 받지 않고 사용자가 원하는 색을 제한 없이 자유롭게 사용할 수 있다.

지우개와 붓의 선택의 경우 주먹을 쥔 오른손이 메뉴에 들어가 있을 때 현재의 상태가 붓이면 지우개로 변경되고 현재의 상태가 지우개면 붓으로 변경된다. 붓의 경우 일반적인 방법과 같은 방법인 RGB컬러의 색을 칠해 그림을 그린다. 제스쳐를 이용하여 그려지는 모든 그림은 OpenGL의 FBO(Frame Buffer Object)에 RGBA 포맷으로 저장되며 카메라 영상 앞에 FBO로 정의된 텍스쳐를 렌더링 함으로 카메라 영상과 렌더링 그림이 함께 보여 지도록 한다. 지우개의 경우 일반적인 바탕화면색의 색을 가져와 덧칠하는 방식으로는 구현이 불가능하여 다음과 같은 방법을 사용한다. RGB 컬러 중 임의의 특정 색(RGB(255,0,0))을 지정하고 오른손의 위치에 붓의 크기에 맞는 원을 그리고 미리 지정한 특정 색으로 색을 칠한다. 원의 그림은 FBO에 RGBA포맷으로 저장된다. 이때 FBO의 모든 픽셀에서 지정된 특정 색을 찾아 알파 값을 0으로 만들어 투명 픽셀로 정의하여 카메라 영상과 함께 렌더링 함으로 지우개에 의해 그려진 픽셀에는 카메라 영상 픽셀이 보여 지도록 함으로 지우개 효과를 가지도록 한다. 만약 붓의 색으로 지정된 특정 색이 들어 왔을 경우에는 특정 색이 아닌 다른 값(RGB(254,0,0))으로 변경한다.

붓의 굵기의 경우 주먹을 쥔 오른손이 메뉴에 들어가면 메인 메뉴 조작기 하단에 서브 메뉴 조작기가 나타난다. 서브 메뉴 조작기도 메인 메뉴 조작기와 마찬가지로 주먹을 쥔 오른손이 메뉴에 들어가면 작동한다. 템플릿 메뉴의 경우 주먹을 쥔 오른손이 메뉴에 들어가면 하단에 서브 메뉴 조작기가 나타난다. 서브 메뉴 조작기에는 템플릿으로 불러올 수 있는 이미지들이 나타난다. 메인 메뉴 조작기와 같은 방법으로 주먹을 쥔 오른손이 이미지 메뉴에 들어가면 이미지 메뉴에 해당하는 이미지가 그림 영역 오른쪽 상단에 나타난다. 여기서 나타나는 이미지는 그림 4에서 보이는 것처럼 영상 상단에 위치한 FBO 보다 상단에 있는 FBO에 저장되며 현재까지 그려진 그림에는 영향을 주지 않는다.

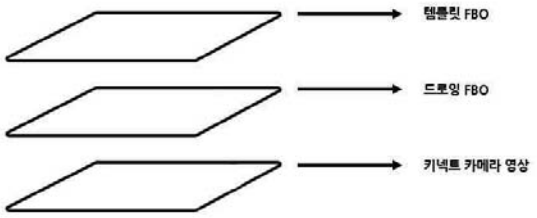


그림 4 드로잉 시스템 레이어

그림 5. 두 손의 거리에 따른 템플릿 이미지 크기 제어

이렇게 FBO에 저장된 템플릿 이미지는 크기를 크게 하는 것과 위치를 옮기는 것, 템플릿 이미지에 색을 칠하는 것 3가지의 독립 메뉴가 있다. 3가지 메뉴를 사용하려면 템플릿 메뉴를 변경해야 하는데 템플릿 메뉴는 템플릿 모드에서 오른손의 개폐상태가 주먹 왼손의 개폐상태가 가위일 경우 템플릿 메뉴가 변경된다. 변경되는 메뉴는 메뉴 확인 창에서 확인이 가능하다. 템플릿 이미지를 크게 만들기 위해서는 그림 5 처럼 템플릿 메뉴를 크기 모드로 변경하고 오른손 왼손의 개폐상태를 주먹을 쥔 후 크기를 크게 하고 싶다면 두 손의 거리를 멀리하고 크기를 작게 하고 싶다면 두 손의 거리를 좁게 하여 템플릿 이미지의 크기를 조정한다.

템플릿 이미지의 위치를 이동시키기 위해서는 그림 6 처럼 오른손 왼손의 개폐상태를 주먹으로 하고 템플릿 이미지 영역 안으로 들어가 원하는 위치로 옮기면 된다. 템플릿 이미지의 페인팅의 경우는 드로잉 FBO와 마찬가지로 일반적인 방법과 같은 방법인 RGB컬러의 색을 칠해 그림을 그린다. 하지만 드로잉 FBO와 달리 템플릿 FBO에서는 템플릿 이미지가 아닌 곳에는 색을 칠하지 않아야

하기 때문에 다음과 같은 방법으로 그림을 그린다. 먼저 오른손의 위치를 찾고 오른손의 위치를 중점으로 붓의 크기 값을 이용하여 붓이 그려질 위치의 픽셀들을 찾는다. 그리고 위치한 픽셀들에서 알파 값이 255인 픽셀들을 찾아 현재 붓의 색을 채워 놓는다. 만약 그렇지 않으면 무시하여 템플릿 이미지에만 그림을 그릴 수 있도록 한다.. 그리고 템플릿 모드에 들어오면 메인 메뉴 조작기에 추가적으로 닫기 메뉴가 생성된다. 주먹을 쥔 오른손이 닫기 메뉴 속에 들어가면 템플릿 모드가 종료되고 현재 템플릿 FBO가 드로잉 FBO에 합쳐진다. 초기화 메뉴는 주먹을 쥔 오른손의 위치가 메뉴 속에 들어가 있을 때 렌더링 되는 드로잉 FBO를 클리어(Clear)시켜 그렸던 그림을 모두 사라지게 한다.

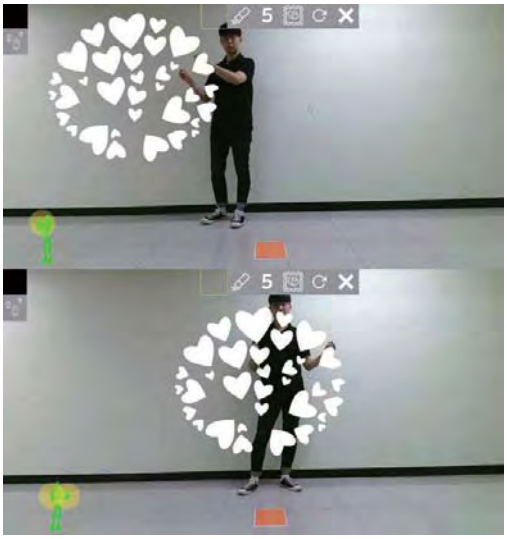


그림 6 양손을 이용한 템플릿 이미지의 이동

4. 구현 결과 및 분석

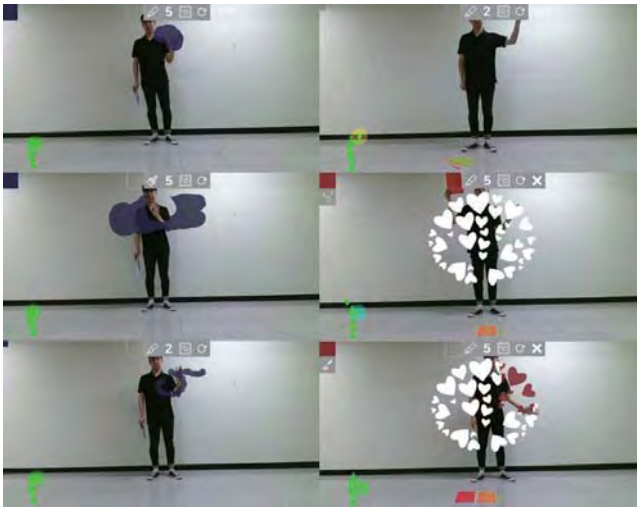


그림 7 드로잉 시스템을 이용한 그림그리기

본 논문에서 제작된 드로잉 시스템을 이용하여 그림 7과 같이 조작하여 그림을 그릴 수 있다. 그림 7의 좌측 상단의 모습은 아무기능이 들어가지 않은 상태에서 드로잉 하는 모습이다. 우측 상단의 모습은 초기화 메뉴를 이용하여 그림을 지우는 모습이다. 중간 좌측 사진은 지우개를 이용하여 그림을 지우는 모습이다. 중간 우측 사진은 사용자가 원하는 색상을 입력하는 모습이다. 하단 좌측 사진은 폰트 굵기를 줄이고 그림을 그리는 모습이다. 하단 우측 사진은 템플릿 이미지에 그림을 그리는 모습이다. 그림 8의 경우 드로잉 시스템을 이용해 그린 그림을 보여주고 있다. 사용자는 자신의 손짓과 개폐상태를 이용하여 자신이 원하는 그림을 그릴 수 있고 템플릿을 이용하여 정교한 이미지까지 그릴 수 있었다.

그림 8 다중 레이어 드로잉 시스템을 이용해 그린 그림 예

5. 결론 및 향후연구

본 논문에서는 사용자와 상호작용이 가능한 하드웨어 키넥트를 사용하여 자유롭게 색감을 선택하고 그림을 그릴 수 있는 드로잉 시스템을 설계 구현하여 사용자의 몰입감과 흥미를 높였다. 향후 연구로 사용자가 원하는 이미지를 직접 추가하는 기능과 사용자가 색을 인식하는 방법을 개선하고 여러 사람들이 그림을 그릴 수 있는 기능을 추가할 예정이다.

참고문헌

[1] Wikipedia Kinect [Internet] https://ko.wikipedia.org/wiki/%ED%82%A4%EB%84%A5%ED%8A%B8

[2] Github OpenPose [Internet] https://github.com/CMU-Perceptual-Computing-Lab/openpose

[3] Kai-Min Chen , Sai-Keung Wong “Interactive Sand Art Drawing Using Kinect”, VINCI '14 Proceedings of the 7th International Symposium on Visual Information Communication and Interaction, Pages78

[4] Leap Motion Orion: Pinch Draw Module [Internet] https://www.youtube.com/watch?v=4LVVpl9tCNE

[5] Painting Simulator... with Myo armbands [Internet] https://www.youtube.com/watch?v=tbjhanrqkT_0

[6] A thorough comparison between Kinect v1 and Kinect v2 [Internet] https://www.buildinsider.net/small/kinectv2cpp/01