

HSV 컬러 영역을 이용한 실시간 손동작 검출 증강현실 퍼즐 게임 개발

김민혁, 김영식
한국산업기술대학교 게임공학과
{piti5, kys}@kpu.ac.kr

Development of an Augmented Reality Puzzle Game Detecting Hand Posture Using HSV Color Space in Real Time

Minhyuk Kim, Youngsik Kim
Dept. of Game & Multimedia Engineering, Korea Polytechnic University

요 약

증강현실 기술을 게임에 접목하려는 시도가 많이 이루어지고 있다. 본 논문에서는 실시간으로 스마트 기기의 카메라로부터 손동작 2차원 영상을 획득하고 HSV 컬러 영역을 활용하여 해당 색상의 물체 또는 신체를 실시간으로 검출해, 게임의 요소로 활용하여 즐길 수 있는 2D 퍼즐게임을 개발하였다. 이는 스마트 기기에 컴퓨터 비전을 접목한 형태의 게임으로, 교육 및 엔터테인먼트 환경에서 스마트 기기의 휴대성과 가상현실 체험의 현장감을 접목시킨 새로운 인터랙티브 기술이 될 것이다.

ABSTRACT

There have been many trials for development of games using augmented reality. This paper presents a 2D puzzle game system using HSV color space to detect hand posture from an input image by smart device's cameras in real time. The developed game in this paper brings smart devices and the computer vision closer. Also this game can be a new interactive technology utilizing both the mobility of smart devices and the tangible interactivity of virtual reality in education and entertainment environments.

Keywords : Augmented Reality (증강현실), Smart Devices (스마트 기기), Puzzle Game (퍼즐 게임)

Received: Aug. 07, 2014 Revised: Sept. 24, 2014
Accepted: Oct. 13, 2014
Corresponding Author: Youngsik Kim(Korea Polytechnic Univ.)
E-mail: kys@kpu.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

컴퓨터 기술의 발전으로, 최근 현대 사회는 시간과 장소에 구애받지 않고 고성능의 컴퓨터를 언제 어디서나 편리하게 사용할 수 있는 스마트 기기 시대를 열게 되었다. 최근, 스마트 기기에 대한 관심과 더불어 증강현실에 대한 관심 또한 높아져 가는 추세이다[1,2,3,4,5,6,7,8,9,10].

사용자의 이러한 관심과, 기대를 충족시킬 수 있는 다양한 콘텐츠가 요구되는 가운데, 증강현실과 관련된 연구로 논문 [6]에서는 PC카메라를 이용해 추출된 모션을 활용하여 몰입형 게임을 개발하였고, 동작영상에 대한 성능평가를 하였다. 논문 [7]에서는 위치기반 서비스와 증강현실을 혼합하여, Wi-Fi 신호의 세기로 거리를 측정하여 화성 박물관 내에서의 체험형 관광 서비스를 개발하였고, 마커 기반 증강현실에 대한 성능평가를 하였다. 논문 [8]에서는 모델간의 매칭을 수행하여 손 포즈를 인식한 후, 다음 프레임의 은닉된 손가락의 예측이 가능한 인식법을 개발하였고, 포즈인식에 대한 성능평가를 하였다. 논문 [9]에서는 착용형 컴퓨터를 이용하여, 착용형 입력장치로써 Video see-through HMD를 사용하여, 거리 변형 행렬과 외각점 추출을 통해 손을 검출하였고, 메모 프로그램의 성능을 비교하였다. 논문 [10]에서는 휴대폰 내에서 손동작을 이용한 비전 기반 인터페이스를 구현하여, 회전에 따른 인식률의 성능을 비교하였다. 다만 손의 모양을 완벽히 검사하기 위한 알고리즘을 적용하는 과정에서 실제로 활용하기는 어려운 속도로 동작하였다. (논문[10] P8, 1~4 - SAMSUNG Galaxy S2 1.2GHz dual core processor, 1GB RAM, Android 2.3 OS상에서 최대 5.4fps에서 최소 4.4fps의 속도를 보였으며, 평균적으로 5fps의 속도로 동작했다.) 이에 비해 본 논문에서는 실제 게임을 원활히 즐길 수 있도록 최초 게임 시작 시에 색상 값을 입력받는 알고리즘과, 색상값을 검출하는 알고리즘만을 적용하여, (iPad2 1GHz dual core processor 512MB RAM, iOS7)에서도 30fps

의 속도로 동작할 수 있도록 하여 게임의 진행에 지장을 주지 않았다.

본 논문은 스마트 공간에서 사용자의 손동작에 기반하는 인터페이스를 통해 플레이하는 퍼즐형 게임을 개발한다. 본 논문에서 개발한 게임은 손동작이 그대로 게임 퍼즐에 일부로 적용되기 때문에 게임의 몰입감을 증가시킬 수 있다. 본 논문에서 개발한 Finger-Rails 게임은 IOS의 Sprite Kit과, Sprite Kit Physics 엔진, OpenCV[13]를 활용하여 사용자의 손과 물체의 위치를 입력받아 게임의 장애물로 활용하는 게임이다. 사용자는 직접 자신의 신체부위(손) 등을 이용하거나, 특정 물체를 이용해 장애물을 게임 내에 투영할 수 있기 때문에, 몰입도가 높고 성취도를 자극시키는 게임이다. 또한 간단하고 쉬운 게임방식을 통해, 남녀노소 누구나 즐길 수 있도록 기획하였다. 또한 본 논문에서는 손동작 인터페이스를 단순한 배경과 복잡한 배경에서 실험을 통하여 측정하여 각각 92.0%와 67.7%의 인식률을 얻었다.

본 논문은 다음과 같이 구성된다. 2장에서는 전체 알고리즘에 대해 설명하고, 3장에서는 이를 바탕으로 “Finger Rails” 게임 시스템 구현 및 결과에 대해서 다룬다. 4장에서는 손 인터페이스 인식률을 실험을 통해 평가하였고 5장에서 결론을 맺는다.

2. 알고리즘 구현

2.1 전체 동작 흐름

본 논문에서 스마트 기기와 증강현실 손동작 인터페이스를 활용한 게임의 전체 동작 알고리즘은 다음과 같다. 첫 번째로 스마트기기에 부착된 카메라로부터 얻어진 영상을 통해 사용자의 손을 특정 위치에 입력받아 손의 색상의 샘플을 얻어낸다. 두 번째로는 부착된 카메라로부터 얻어진 영상을 HSV 컬러 공간으로 변환 후, 색상 샘플의 이중 임계값 처리를 통하여 영상 내에서 해당 색상 샘플

플 범위 내의 값을 찾아낸다. 세 번째로는 두 번째에서 검출한 영상에서 색상이 급격히 변하는 경계 값 검사를 통해 샘플 색상의 가장자리 좌표를 얻고, 만들어진 가장자리 경계 중 가장 큰 값을 선택하여 가시적으로 표시한다. 마지막으로 이를 응용한 어플리케이션으로써 “Finger Rails” 는 색상 샘플을 이용해 손, 또는 물체를 검출하여 해당 물체의 가장자리 경계선을 만들고, 해당 경계선을 장애물로 판별하여 길을 만들어 공을 옮겨가는 게임이다. [Fig. 1]은 전체 동작의 흐름을 나타낸다.

Pseudo Algorithm : Overall Operational Algorithm

STEP1

Obtain sample color pixels through the camera of smart devices with OpenCV;

STEP2

Transform the color space of input data to HSV color space;

foreach (RecA in Screen Space)

foreach (PixelA in RecA)

SumC += Color of PixelA;

end foreach

Avg. Color of RecA=SumC/Num. of Pixels;

SumA += Avg. Color of RecA;

end foreach

Avg. Color of Space=SumA/Num. of Rectangles;

STEP3

Find contours using two border;

STEP4

Eliminate noise;

STEP5

Construct collision obstacles;

STEP6

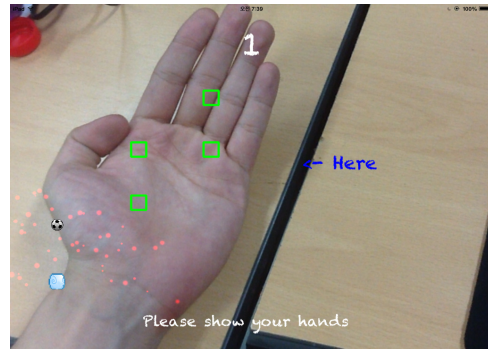
Start game “Finger Rails”;

[Fig. 1] The overall operational algorithm

2.2 샘플 색상 입력

샘플 색상 입력이란 카메라의 영상을 출력 후 사용자로부터 카메라 특정 위치의 색상을 입력받아 손의 인식을 위한 샘플 색상을 저장해 두는 과정을 말한다. 손을 인식할 때 기존의 알고리즘에 의

해 고정된 값으로 인식을 하게 되면, 개개인의 손의 색상 차이를 반영하지 못할 수 있다. 하지만 샘플 색상을 수집하여 색상의 평균을 낸다면, 사용자에게 맞게 손 인식이 가능해진다. 또한 손대신 물체의 색을 인식시킨다면, 물체의 인식도 가능해진다. 따라서 샘플 색상을 입력받는 방식을 선택하게 되었다.



[Fig. 2] Sample color input

[Fig. 2]에서 네 개의 초록색 사각형의 픽셀을 읽어온 뒤, BGR 컬러 영역에서 HSV 컬러 영역으로 변환을 한다. 입력 영상의 컬러 성분은 BGR인데, 각각의 성분들에 휘도 성분이 포함되어있기 때문에 조명에 따라 특정치의 값이 매우 현저하게 변환된다. 따라서 (eq. 1)에 따라 BGR 컬러 영역에서 HSV 컬러 영역으로 바꾼다.

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \end{cases}$$

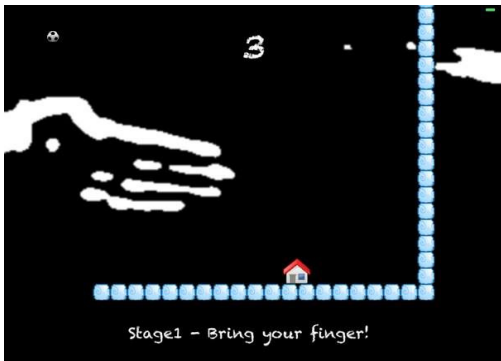
(eq. 1)

휘도 성분을 포함하지 않은 색조 성분인 H 값과 채도 성분인 S 값이 추출된다. 이들 값은 조명의 영향을 덜 받는다[11]. 이들 네 구간의 값의 평균을 내어, 이중 임계값을 적용한 후, 교집합을 구

해 손 영상에서 배경을 분리한다.

2.3 샘플 색상을 이용한 물체 검출

2.2절에서 입력받은 색상을 토대로, 카메라에서 입력받은 영상에서 픽셀 데이터를 스캔하여 해당 색상을 검출한다. 색상을 검출한 이미지에서, [12]의 이진화상에서의 색상 값에 따른 경계 검출 알고리즘을 사용하여, 이미지의 픽셀 데이터를 스캔하여 검출된 색상 이미지의 경계 값을 찾는다. ([Fig. 3]).

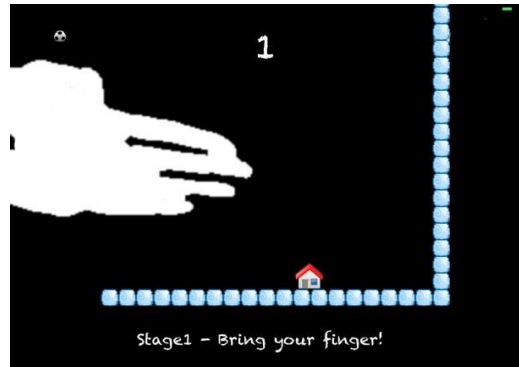


[Fig. 3] Find contours using two border following algorithms

2.4 잡음 제거

검출된 경계 영상을 살펴보면, 원하지 않는 경계와 잡음 화소가 포함되어 있다. 이런 경우, 게임 내에서 사용자가 원하지 않는 결과가 발생할 가능성이 높아진다. 경계값을 구한 영상에서, 사용자가 인식한 손이나 물체외의 인식을 방지하기 위해, 경계값의 좌표값 개수를 검사하여 가장 큰 경계만을 남기고 모두 제거한다.

또한, 블러 필터링과 다운샘플링을 이용하여 경계의 잡음을 제거한다. [Fig. 4]는 이미지의 잡음 제거 과정을 거친 결과이다.

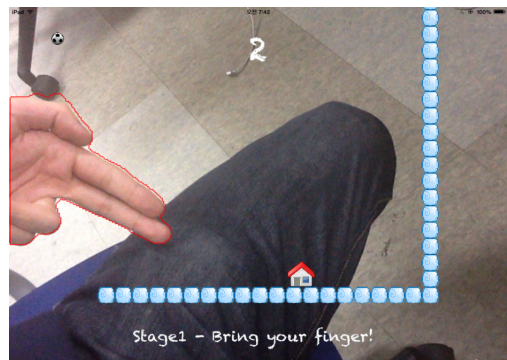


[Fig. 4] Noise elimination

2.5 충돌체 생성

검출된 경계 정보의 위치값을 받아, 해당하는 위치를 잇는 선분을 그어, 경계면으로 이루어진 충돌체를 만든다([Fig. 5]).

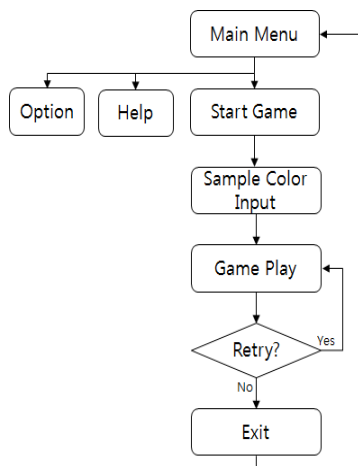
OpenCV[13]의 findContours함수를 이용하여 생성한 경계를 SpriteKit에서 인식할 수 있도록, 점의 벡터리스트를 하나씩 꺼내어 CGMutablePathRef구조체 자료형을 선언하고, CGPathAddLineToPoint() 함수로 자료형을 채운다. 그 후 SpriteKit에서 지원하는 SKShapeNode구조체 자료형을 선언하고, 자료형의 path변수와 물리엔진에 위에서 만든 CGMutablePathRef구조체 자료형을 입력한다.



[Fig. 5] Construction of collision obstacles

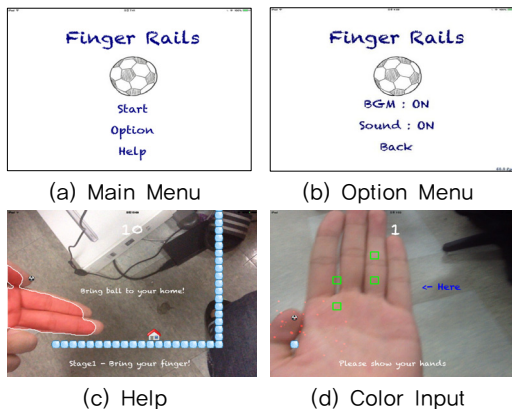
3. Finger-Rails 게임 구현

이 연구에서는 터치 입력으로 UI를 조작하여 스마트폰 게임을 즐기는 것이 아니라, 스마트폰 카메라로부터 촬영되는 영상에서 손의 경계를 추출하여 게임의 일부로 활용하였다. 본 시스템의 입출력 장치로써 Ipad2를 사용하였으며, 영상을 실시간으로 편집하면서 게임을 원활히 진행하기 위하여 640x480 pixel의 해상도로 영상을 입력받았다.



[Fig. 6] Game operation flow

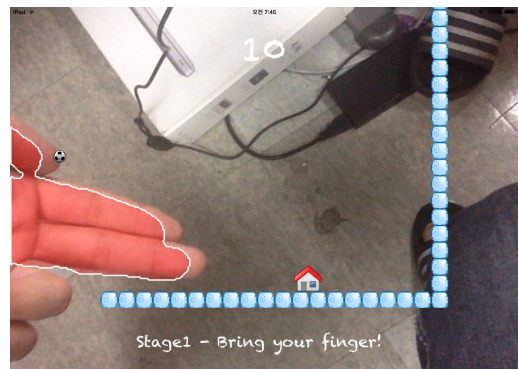
[Fig. 7](a) 메인메뉴에서 터치를 통해 메뉴를 선택할 수 있다.



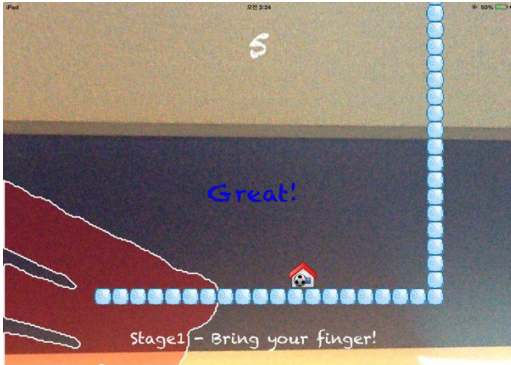
[Fig. 7] Game menus

[Fig. 7](b) 옵션메뉴에서는 배경음과 효과음의 재생여부를 결정할 수 있다. [Fig. 7](c) 도움말 메뉴에서는 게임의 플레이 방법을 알 수 있다. [Fig. 7](d)는 메인 메뉴에서, 시작 버튼을 누르게 되면 샘플 색상을 입력받을 수 있는 화면이 나타난다.

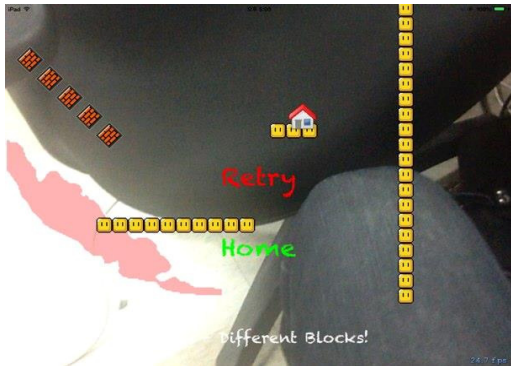
샘플 색상을 입력받고 타이머가 완료되면, 게임 화면으로 넘어가게 된다. [Fig. 8]은 게임의 화면으로, 시간내에 사용자가 공을 집에 어떻게 넣을지 생각하여 손의 포즈를 정하게 된다. 포즈를 정하는 타이머가 완료되면 손의 경계가 충돌체로 변하게 되고, 그림과 같이 추출된 경계와 게임 그래픽의 공이 충돌할 경우 공이 튀기게 되어 이동한다. 시간 내에 공을 집으로 넣으면 게임에서 승리하게 되고([Fig. 9]), 다음 스테이지로 진행할 수 있다. 화면의 밖으로 공이 빠져나가거나 타이머가 완료되면, 게임은 실패로 끝나게 되고 재시작 메뉴([Fig. 10])가 나타나게 된다. 스테이지가 변화할 때 마다, 벽돌의 마찰력이나 탄성력을 변화시켜 사용자가 다양한 재미를 느낄 수 있도록 구성한다([Fig. 11]).



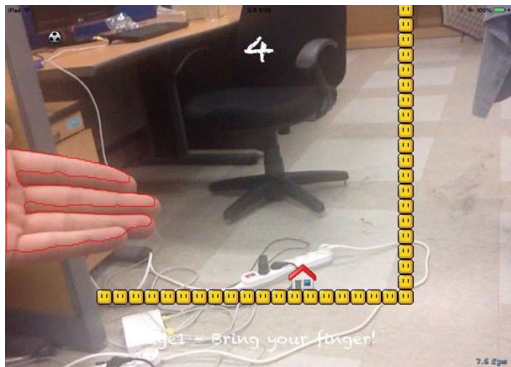
[Fig. 8] Game screen shot



[Fig. 9] Win screen shot



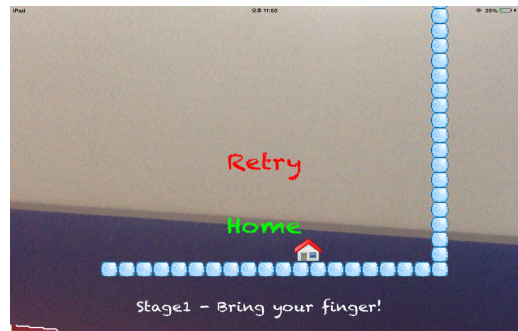
[Fig. 10] Retry screen shot



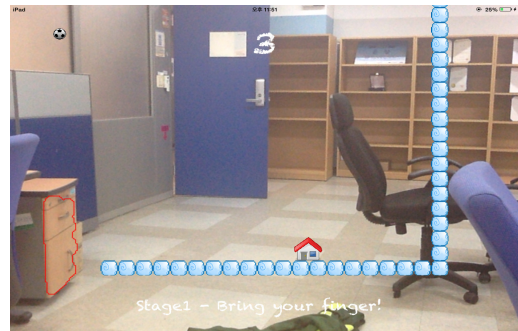
[Fig. 11] Game stage example

4. 손 인터페이스 인식을 실험

손 동작 인터페이스의 인식을 단순한 배경과 복잡한 배경에서 각각 100회씩 3회 실험을 통해 측정하였다. [Fig. 12]는 실험에 사용한 단순한 배경과 복잡한 배경을 보여준다. [Fig. 13]은 실험을 통하여 측정한 손 동작 인터페이스의 인식이 단순한 배경과 복잡한 배경에서 각각 92.0%와 67.7%이다. 실험결과 복잡한 배경에서는 배경 색상과 복잡한 물체로 인하여 인식이 떨어지는 것을 알 수 있다. 단순한 배경에서의 게임 수행에는 큰 문제가 없음을 알 수 있다. 복잡한 배경에서는 손 색상과 배경 색상이 확연히 차이 나지 않으면 인식이 떨어지므로 색상 대비가 큰 배경을 선택해야 함을 알 수 있다. 게임의 전체 동작 속도는 20~40FPS(frame per second)으로 평균 30FPS로 동작에 문제가 없었다.

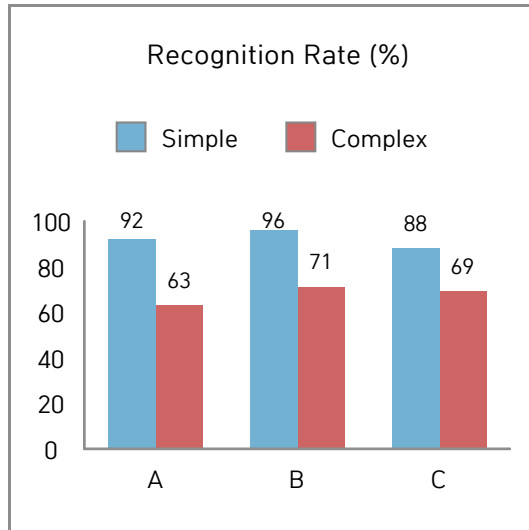


(a) Simple background



(b) Complex background

[Fig. 12] Backgrounds for recognition experiments



[Fig. 13] Experimental results for recognition

5. 결 론

이 연구는 iPad에서 영상을 획득하여 영상으로부터 실제 사용자가 입력한 색상을 바탕으로 해당 색상의 물체 또는 신체를 검출해, 게임의 요소로 활용하여 즐길 수 있는 몰입형 2D게임을 제작하였다. 게임의 입력으로 스마트 기기에서의 단순한 터치만을 사용하여 게임을 즐기는 것이 아니라, 사람이 직접 몰입하여 게임을 즐길 수 있도록 실시간으로 iPad의 카메라로부터 2차원 영상을 획득하고 샘플 픽셀의 색상을 획득하여 토대로 실제 영상의 물체를 게임의 요소로 활용하였다. 기존의 연구 [14]에서는, 색상에 기반한 피부색을 자동으로 탐지하는 알고리즘을 사용하여 정확하게 피부를 탐지할 수 있었지만, 해당 알고리즘의 연산량이 많은 관계로 스마트 기기에서 사용하기는 부적합했다. 이에 따라서 본 연구에서는, 게임에 필요한 최소한의 알고리즘만을 사용하여 원활히 게임을 즐길 수 있도록 코드를 설계하였다. 또한 사용자가 원할 때에는, 본인의 신체가 아닌, 다른 색상을 통하여서도 게임을 즐길 수 있도록 하였다.

스마트 기기에 컴퓨터 비전을 접목한 형태의 게임으로, 본 연구의 결과물이 현재의 스마트 시대에 서 가상현실 환경으로 발전하는 교두보가 될 것이며, 특히 교육 및 엔터테인먼트 환경에서 스마트 기기의 휴대성과 가상현실 체험의 현장감을 접목시킨 새로운 인터랙티브 기술의 핵심이 될 것이다. 그러나 PC에 비해 스마트 기기의 성능이 좋지 못한 관계로 실시간 영상 처리와 물리엔진을 적용시키는 데에 제한이 있었고, 해상도를 줄여야만 했다. 또한, 색상값을 기준으로 영상을 판별하다 보니 인식을 시킨 물체보다 배경에 샘플 색상이 더 많은 경우에는, 제대로 인식을 하지 못하고 배경을 인식하게 되는 문제점이 있었다. 마지막으로 IOS의 입출력방식과 OpenCV의 입출력방식에 차이가 있어서, 매 프레임마다 서로 호환될 수 있도록 변환을 해주는 점도 게임의 프레임에 영향을 미치는 요소가 되었다.

REFERENCES

- [1] Jong-Yeol Yang, Sang-Kyung Lee, Dong-Wuk Kyoung, Kee-Chul Jung, "Gesture-based Table Tennis Game in AR Environment", Journal of Korea Game Society, Vol. 5, No. 3, pp.3~10, 2005
- [2] Jong-Hyun Yoon, Jong-Seung Park, "Augmented Reality Game Interface Using Hand Gestures Tracking", Journal of Korea Game Society, Vol. 6, No. 2, pp.3~12, 2006.
- [3] Jong-Seung Park, Young-Jun Jeon, "Design and Implementation of Motion-based Interaction in AR Game", Journal of Korea Game Society, Vol. 9, No. 5, pp.105~116, 2009.
- [4] Trien Van Do, Jeong-Gyu Lee, "3DARModeler: a 3D Modeling System in Augmented Reality Environment", Journal of Korea Game Society, Vol. 9, No. 5, pp.127~136, 2009.
- [5] Yo-Seop Yun, Tae-Young Kim, "A Boundary-based Marker Binary Coding Method for Augmented Reality Games", Journal of Korea Game Society, Vol. 10, No. 4, pp.63~72, 2010.

- [6] Mi-Young Song. “Immersive Game Development Using Extracted Motion from PC Camera as Games Control Input,” Journal of The Korean Society for Computer Game, No. 19, 2009 December.
- [7] Sunho Lee, Woosik Lee, Namgi Kim, Junchul Chun, “A Design and Implementation of Ubiquitous Museum(U-Seum) Using Location Based Service and Augmented Reality,” Journal of Korean Society for Internet Information 2012. Aug: 13(4): 63-71
- [8] Min-Young Na, Jae-In Choi, Tae-Young Kim, “A Real-time Hand Pose Recognition Method with Hidden Finger Prediction,” Journal of Korea Game Society, Vol. 12, No. 5, pp.79~88, 2012
- [9] Taejin Ha, Woontack Woo, “Video see-through HMD based Hand Interface for Augmented Reality,” HCI2006 pp169~174, 2006
- [10] Junyeong Choi, Hanhoon Park, Jungsik Park, Jong-Il Park, Implementation of Hand-Gesture-Based Augmented Reality Interface on Mobile Phone, Journal of Broadcasting Engineering, Vol. 16, No. 6, pp.941~950, 2011
- [11] G. Bradski. Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal, 1998.
- [12] Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46 (1985)
- [13] OpenCV 2.4.2 for IOS : <http://sourceforge.net/projects/opencvlibrary/files/opencv-ios/2.4.2/opencv2.framework.zip/download>
- [14] Vladimir Vezhnevets, “A Survey on Pixel-Based Skin Color Detection Techniques” Graphics and Media Laboratory Faculty of Computational Mathematics and Cybernetics Moscow State University, 2003



김민혁 (Kim, Minhyuk)

한국산업기술대학교 게임공학과 전공 재학 중

관심분야 : 게임프로그래밍, iOS프로그래밍, 증강현실



김영식 (Youngsik Kim)

1993년 연세대학교 컴퓨터과학과 학사
1995년 연세대학교 컴퓨터과학과 석사
1999년 연세대학교 컴퓨터과학과 박사
1999년-2005년 삼성전자 System LSI 책임연구원
2013년 University of Pittsburgh 방문교수
2005년-현재 한국산업기술대학교 부교수

관심분야 : 게임기구조, 컴퓨터구조, 3차원 그래픽가속기, 임베디드 시스템 등
