

루빅스 큐브를 활용한 다 종류 3차원 객체 증강 시스템

이상준[†], 김수빈^{**}, 황성수^{***}

A Multi 3D Objects Augmentation System Using Rubik's Cube

Sang Jun Lee[†], Soo Bin Kim^{**}, Sung Soo Hwang^{***}

ABSTRACT

Recently, augmented reality technology has received much attention in many fields. This paper presents an augmented reality system using Rubik's Cube which can augment various 3D objects depending on patterns of a Rubik's cube. The system first detects a cube from an image using partitioning clustering and strongly connected graph. Thereafter, the system detects the top side of the cube and finds a proper pattern to determine which object should be augmented. An object corresponding to the pattern is finally augmented according to the camera viewpoint. Experimental results show that the proposed system successfully augments various virtual objects in real time.

Key words: Augmented Reality, Line Clustering, Image Transformation, Viewpoint Estimation

1. 서 론

최근 사용자들에게 실제와 유사한 경험을 제공하는 실감형 콘텐츠 기술이 각광받고 있다. 그 중에서도 증강현실은 현실의 영상 위에 가상의 이미지를 겹쳐 보여주는 기술로서 사용자들에게 가상의 데이터를 실제 영상 위에 시각화 시켜 효과적이고 편리하게 정보를 제공할 수 있는 미래지향적인 기술이다. 오늘날 증강현실에 대한 수요와 관심은 꾸준히 증가하고 있으며 Microsoft사와 Google 을 포함한 여러 기업들이 증강현실 관련 연구 및 제품 개발에 참여하고 있다[1].

현재 증강현실 기술은 게임, 교육, 의료, 건축 등 여러 분야로 사용되고 있다. 대표적으로, Niantic lab

에서 개발한 포켓몬 고[2]와 Vuforia에서 개발한 증강현실 기술 관련 SDK[3]가 있다. 증강현실 기술은 기념품 분야에서도 적용이 가능하며, 카드 인식을 통한 증강현실 기념품이 제안된 바 있다. 증강현실 기술을 기념품 분야에 적용하면 사용자에게 실감 있게 정보를 제공할 수 있으며, 가상의 객체를 제공하기 때문에 물리적 제약이 없다. 또한 제공하는 정보를 업데이트 할 수 있으며, 제공하는 물리적 객체의 형태가 변형이 가능한 경우 형태 혹은 패턴의 변형에 따라 다양한 콘텐츠를 제공할 수 있다는 장점이 있다. 형태 혹은 패턴의 변형이 손쉽게 이루어지는 객체로는 루빅스 큐브가 있다. 루빅스 큐브는 대중적이고 어디서든지 쉽게 구할 수 있으며, 다양한 패턴을 생성할 수 있다. 따라서 큐브의 한 면을 검출하여 해

* Corresponding Author : Sung Soo Hwang, Address: 558, Handong-ro, Namsong-ri, Heunghae-eup, Buk-gu, Pohang-si, Gyeongsangbuk-do, Republic of Korea, TEL : +82-42-260-1684, E-mail : sshwang@handong.edu

Receipt date : July 17, 2017, Approval date : July 28, 2017

[†] School of Computer Science and Engineering, Handong Global University, E-mail : eowjd4@naver.com

^{**} School of Computer Science and Engineering, Handong Global University, E-mail : tnqls0332@gmail.com

^{***} School of Computer Science and Engineering, Handong Global University, E-mail : sshwang@handong.edu

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2016R1D1A3B03934808)

당 면의 패턴을 인식한 뒤 인식한 패턴에 따라 객체를 증강하는 시스템이 개발되면 기념품 분야에서 매우 유용하게 활용될 수 있다.

루빅스 큐브를 활용한 증강현실 시스템의 구현을 위해서는 영상 내에서 큐브를 검출하고, 또 물체를 증강시킬 큐브의 특정한 면을 추적해야 한다. 기존의 증강현실 시스템의 경우 영상 내 특징점 검출을 통해 영상 내에서 가상 객체를 증강할 위치를 계산하였다. 영상 내 특징점은 특이하면서도 변하지 않는 텍스처 패턴을 가지고 있는 객체의 표면에서 다수 검출된다. 그러나 큐브의 패턴은 매우 다양하게 변할 수 있고, 뿐만 아니라 큐브의 여섯 면의 형태가 유사하므로, 기존의 특징점 검출 및 매칭 방법으로는 큐브 검출 및 큐브의 특정한 면을 일관성 있게 추적하기 어렵다.

따라서 본 논문에서는 기존의 특징점 검출 방식이 아닌 큐브 내 직선을 활용한 증강현실 시스템을 제안한다. 제안 시스템에서는 가장 먼저 영상 내 특정한 면에 부착된 마커의 검출을 통해 큐브의 위치를 파악한 뒤, 루빅스 큐브 내 직선 검출 및 분류를 통해 루빅스 큐브의 외곽선을 추출한다. 추출한 외곽선 중 마커와의 위치 관계를 고려하여 큐브의 해당 면을 추출하고 카메라 자세를 추정한다. 이 후 뒷면의 패턴을 분석하여 해당 패턴에 대한 가상 객체가 존재할 경우 이를 증강하도록 하였다. 루빅스 큐브는 휘어지지 않는 물체로서 큐브 내 패턴이 다양하게 변할 수 있음에 반해 그 외형은 변할 수가 없으므로 이를 활용하여 제안 시스템은 영상 내에서 가상 객체를 증강시킬 영역을 정확히 검출하였다. 또한 루빅스 큐브의 외형 검출에 있어서 제안하는 직선 분류 기법을 통해 다수의 직선을 포함하고 있는 루빅스 큐브에서도 큐브의 외형에 해당하는 직선만을 효과적으로 추출하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 증강현실 기술 연구에 대해 소개하며, 3장

에서는 본 논문의 시스템에 대하여 자세히 소개한다. 4장에서는 실험 결과를 보여주고 5장에서 결론을 맺는다.

2. 관련 연구

증강현실 기술을 구현하기 위해서는 사용자가 영상 획득을 위해 사용하는 사용자의 위치 정보 및 방향 정보를 추정하여야 한다. Niantic lab에서 개발한 포켓몬 고의 경우 GPS를 활용하여 사용자의 위치 정보만을 추정하였다. 또한 해당 기술은 GPS의 한계로 인해 실외 환경에서만 정상적으로 동작이 가능하다.

GPS를 사용하는 방안 외에도 카메라를 통해 획득되는 영상을 분석하여 사용자의 위치 및 방향을 추정하는 방안도 제안되었다. 영상 분석을 통해 사용자의 위치 및 방향을 추정하는 방법은 다음과 같다. 먼저 가상 객체를 증강하려는 위치 혹은 사물의 특징점 정보를 사전에 추출하여 데이터베이스화 한다. 사용자의 실제 사용 시 현재 영상 내에서 특징점을 추출하여, 추출한 특징점과 데이터베이스화 한 특징점 정보와 비교함으로써 사용자의 위치 및 방향 정보를 추정한다.

영상 기반으로 사용자의 위치 및 방향 정보를 추정하기 위해서는 Solve PnP 기술을 활용하는 방안과 호모그래피를 적용하는 방안이 있다[4]. Solve PnP (solve Perspective-n-Problem)[4] 기법은 데이터베이스화 한 특징점들의 측위정보를 획득할 수 있는 경우에 활용 가능하다. 특징점들의 측위정보를 획득할 수 있는 방안으로는 Fig. 1(a)와 같이 사전에 측위가 이루어진 인위적인 마커를 활용하는 방안이 있다. 이때 인위적인 마커는 인접 객체와 매우 달라서 특징점 검출 방식으로 용이하게 검출이 되어야 한다. 특징점들의 측위정보를 획득할 수 있는 또 다른 방안으

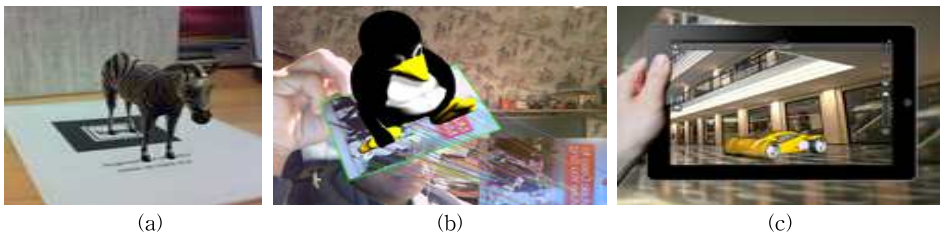


Fig. 1. Examples of image-based augmented reality (a) marker-based augmented reality, (b) 2D markerless augmented reality, (c) 3D markerless augmented reality.

로는 Fig. 1(c)와 같이 인위적인 마커의 사용없이 SfM (Structure From Motion) 기법[5] 혹은 SLAM (Simultaneously Localization And Mapping)기법[6, 7]을 활용하여 물체의 특징점의 3차원 측위를 추정하는 방안이 있다. 특히 SfM혹은 SLAM을 활용하는 경우 사용자의 위치, 방향뿐 아니라 지형 정보까지 추정할 수 있어 보다 실감있는 증강현실 서비스 구현이 가능하다. 단, 가상객체를 증강하려는 위치 혹은 객체에서 다수의 특징점이 추출되어야 한다. 특징점들의 측위 정보를 획득할 수 없는 경우에는 Fig. 1(b)와 같이 영상 간의 호모그래피를 계산하여 위치 및 방향 정보를 추정한다. 그러나 정확한 호모그래피 계산을 위해 기존 기법에서는 다수의 특징점 검출이 필요하였다. 본 논문에서 제안하는 시스템은 영상을 기반으로 사용자의 위치 및 방향을 추정하였다. 또한 측위가 가능한 루빅스 큐브를 활용한 마커 기반 증강현실로서 Solve PnP를 활용하여 사용자의 위치 및 방향을 추정하였다. 마커 기반 증강현실의 경우 특징점 검출을 통해 마커가 용이하게 검출이 되어야 함에 반해 루빅스 큐브는 그렇지 않으므로, 추출한 직선의 분류를 통해 루빅스 큐브에 대해서도 사용자의 위치 및 방향을 안정적으로 추정하였다.

3. 제안하는 3차원 객체 증강 시스템

3.1 제안하는 시스템의 개요

Fig. 2는 본 논문에서 제안하는 시스템의 전체적인 개요를 나타낸다. 제안 시스템에서는 큐브의 특정한 면에 부착된 마커를 특징점 검출 및 원 검출 기반

으로 검출하는 과정을 통해 영상으로부터 큐브의 위치를 파악한다. (제안 시스템에서는 사전에 정의한 특정 면이 윗면일 경우 동작하도록 설계되었으며, 이후 큐브의 특정한 면을 편의상 큐브의 윗면으로 명명한다.) 검출한 영상 내 큐브 영역에 대해서 직선 검출 및 기울기에 따른 직선 분류를 통해 큐브의 외곽선을 검출한다. 검출한 외곽선을 활용하여 큐브의 윗면을 구성하는 꼭지점을 검출하여 큐브의 윗면을 검출한다. 마지막으로 검출한 큐브의 윗면에 대해 호모그래피 변환을 활용하여 패턴을 분석하고, 파악한 패턴에 따라 가상객체를 증강한다.

3.2 영상 내 큐브 영역 검출

영상으로부터 큐브가 어디에 위치하는지 추정하기 위해 큐브 위에 부착된 마커를 검출한다. 큐브 위에 부착된 마커는 내부에 이미지가 붙어있는 원형으로 제한하였고, 이미지 매칭을 통해 마커를 검출했을 경우 마커의 크기가 작아 검출된 특징점의 개수가 불충분한 문제를 해결하기 위해 추정된 마커의 위치로부터 ROI(Region of Interest)를 생성한 뒤, 그 영역 내에서 원을 검출하여 더 정교하게 마커를 추정한다.

먼저, 데이터베이스에 저장된 마커 이미지를 ORB 특징점 검출기를 통해 특징점을 검출한 후, 영상으로부터 특징점 검출을 통해 얻어진 특징점들과 매칭을 통해 데이터베이스에 저장된 마커의 이미지가 영상상에 어디에 위치하는지 파악한다. Fig. 3의 (a)는 특징점 검출을 통해 마커를 검출하는 모습이다. 마커를 검출하고 나면 Fig. 3의 (b)의 하늘색 원 같이 마커의 위치가 검출되는데 특징점의 개수가 많지 않아 정확

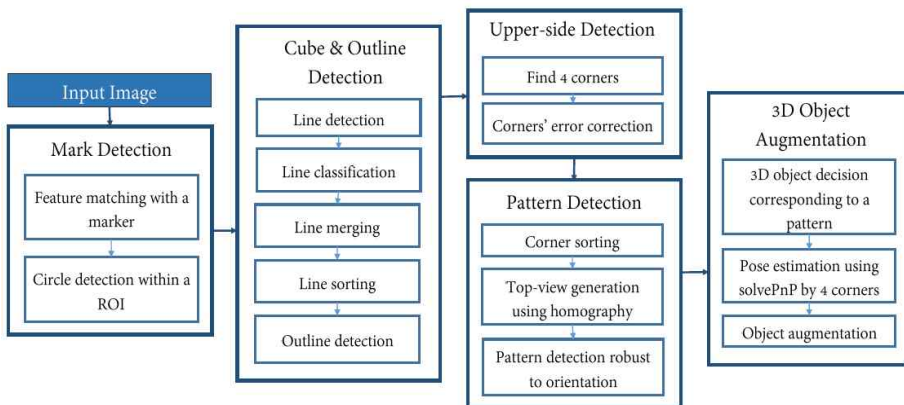


Fig. 2. flowchart for the proposed system.

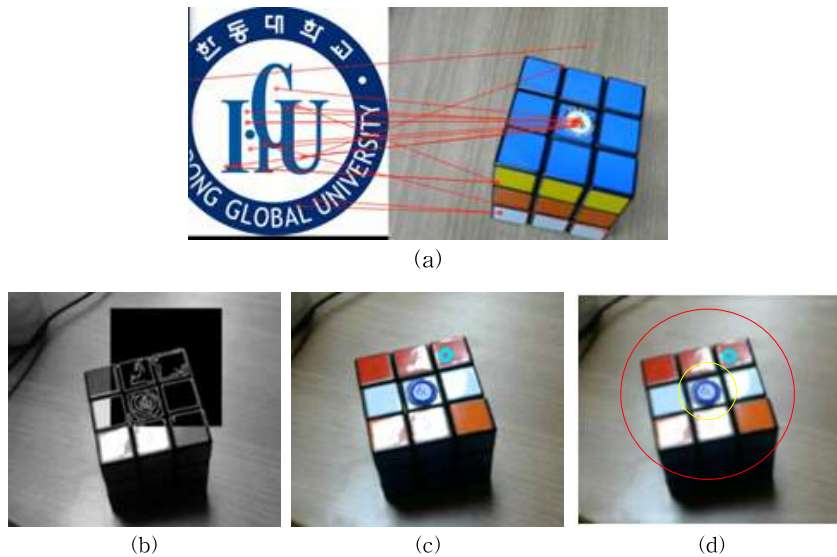


Fig. 3. Marker detection process (a) detection of marker using feature detection (b) ROI setting for circle detection (c) circle detection within ROI (d) ROI setting for cube detection,

한 위치 검출이 어렵다. 그래서 마커의 위치를 추정 한 뒤 ROI을 생성하고 그 영역 내에서 원을 검출한다 [8]. Fig. 3 (b)는 추정된 마커의 위치로부터 ROI를 지정한 그림이며 Fig. 3의 (c)는 ROI영역에서 원을 검출한 그림이다. 원을 검출하면 영상 내에서 큐브의 위치가 어딘지 알기 때문에 원의 중심을 기준으로 일정한 영역 내에서 큐브의 외곽선 검출 모듈을 실행한다. Fig. 3의 (d)는 검출된 원의 위치로부터 큐브의 외곽선을 검출할 영역을 지정해준 그림이며 빨간색 원의 내부로부터 노란색 원의 외부까지 외곽선 검출을 시도한다. 빨간색 원의 반지름은 검출한 원의 반지름의 유클리디안 길이로부터 각각 20배, 5배이다. 노란색 원을 지정해준 이유는, 추후에 직선 검출을 시도할 때 마커 부분까지 직선으로 검출하는 것을 방지하기 위함이다.

3.3 영역 내에서 큐브의 외곽선 검출

검출된 큐브의 영역으로부터 큐브를 검출해야 하며, 큐브는 직선의 형태로 이루어져 있기 때문에 직선 검출을 통해 큐브의 외곽선을 검출할 수 있다. 가장 먼저, 획득된 영상으로부터 색상의 변화가 큰 가장자리 선분(edge)들을 모두 검출한다[9, 10]. 각 선분들을 각도의 크기(degree) d 로 정렬한 뒤 각도의 크기 변화가 D ($D \subset N$)이상 차이 나는 구간의 중간

값을 문턱치 값(threshold) T 들로 지정한 뒤, 문턱치 값으로 직선들의 기울기를 비교하여 직선 1 들을 집합 S 로 분류한다. 문턱치 값은 최대 2개까지 나와야 하는데, 그 이유는 문턱치 값이 최대 2개일 경우에 직선들이 3개의 분류로 나뉘게 되고 이것이 각각 큐브의 가로, 세로, 높이에 해당하는 직선들의 집합이며 영상으로부터 큐브가 세 면이 모두 투사된 경우이다. 문턱치 값이 1개 일 때는 큐브에서 면이 2개밖에 표현되지 않아 세로와 높이 또는 가로와 높이에 해당하는 직선들이 유사한 기울기를 가지는 경우이며 따라서 직선들의 기울기의 집합이 두 가지로 분류 된 경우이다. 그 외에 문턱치 값이 없거나 3개 이상인 경우는 예외 처리한다.

$$T_j = \frac{d_i - d_{i-1}}{2} \text{ if } d_i - d_{i-1} > D, i = 0, 1 \dots N \quad (1)$$

$$l_i \in S_j \text{ if } T_{j-1} \leq d_i < T_j, 0 < j < 3$$

분류된 직선들을 다른 색으로 출력한 결과 Fig. 4 의 (a)와 같이 확인할 수 있다. 0도 부근과 180도 부근의 각도를 갖는 직선들은 사실상 같은 직선의 집합으로 분류해야한다. 따라서 이 직선들을 함께 분류시키기 위해 가장 작은 각도와 가장 큰 각도를 비교하여 D 이상 차이 나지 않으면 첫 번째 직선의 분류구간과 마지막 직선의 분류구간을 하나의 분류 구간으로 지정한다.

분류된 직선들의 중점을 출력해 보면 Fig. 4의 (b)와 같이 많은 직선들이 분포해 있음을 볼 수 있는데, 이 때 파티션 클러스터링(partitional clustering) 기법[11]을 사용하여 중점들의 거리가 유클리디안 거리를 기준으로 K거리 내에 존재할 경우 해당하는 직선들을 클러스터링(clustering)하고 각 군집마다 하나의 직선으로 병합시킨다. 파티션 클러스터링 기법을 통해 병합시킨 결과는 Fig. 4의 (c)와 같다.

Fig. 4의 (c)를 보면 이론상 하나의 직선 위에 있어야 할 직선이 끊어지는 현상을 볼 수 있다. 이를 하나의 직선으로 이어주기 위해 강력연결그래프(Strongly Connected Graph)[12]를 이용한다. 강력연결그래프는 방향성을 가지는 그래프로서 그래프의 노드들 중에서 양쪽 방향으로 경로를 가지는 쌍에 대하여 강력연결그래프로 정의한다. 파티션 클러스터링을 통해 병합된 직선들의 중점들을 양 끝점으로 하여 기울기를 구했을 때, 그 값이 문턱치 값을 통해 분류된 기울기 영역 내에 존재한다면 양 끝점에 대해 경로를 이어준다. 그렇게 모든 가능한 경로를 이어준 후 양쪽 방향으로 이어주는 쌍이 있을 경우 강력연결그래프의 후보군으로 등록한 후 등록된 노드들을 모두 병합시켜준다.

각 직선의 집합 $S_k(k=1,..3)$ 에 포함된 직선 $l_i(i=0,..N)$ 에 대하여 d_i 는 직선 l_i 의 각도의 값을 나타내고 p_i 는 직선 l_i 의 중점의 좌표를 나타낸다. 이 때, 두 직선 $l_i, l_j(i, j=0,..N)$ 에 대하여 중점 p_i, p_j 을 잇는 직선 $L_{i,j}$ 의 기울기 M_i 를 다음과 같이 계산한다.

$$M_i = \frac{p_i \cdot y - p_j \cdot y}{p_i \cdot x - p_j \cdot x} \quad (2)$$

그 후, 기울기 값 $M_{i,j}$ 을 각도 값(degree) $D_{i,j}$ 로 변환한다. 이때, 각 분류 S_k 에 포함된 모든 직선 l_i 에 대하여,

$$d_0 \leq D_{i,j} \leq d_N, (i, j = 0, \dots, N) \quad (3)$$

을 성립할 경우, $D_{i,j}$ 을 생성한 직선 l_i, l_j 를 강력연결 그래프의 후보군으로 등록한다. 강력연결그래프의 후보군으로 등록된 직선들은 서로를 방향 그래프(direct graph)로 가리키고 있다. 후보군을 찾는 작업을 마치고 나서 순환 고리(cycle)를 갖는 모든 강력연결그래프들의 부분 집합 직선들을 하나의 직선으로 병합시켜주면, 그 결과는 Fig. 4의 (d)와 같다. 강력연결그래프를 통해 직선들을 병합하게 되면 다량으로 검출되었던 직선들이 하나의 직선으로 병합되어 검출된다.

최종적으로 병합된 직선들 중에서 회귀선(Regression Line)[13]을 이용하여 처음과 끝 직선을 도출해 낸다. 회귀선은 다량의 데이터의 일련의 경향성을 확인하기 위한 직선을 추론하기 위해 사용하는 방법으로서, 강력연결그래프로 부터 병합된 직선들을 정렬하고 외곽선을 검출하기 위해 사용된다. 각 분류에서 직선들의 중점을 이용하여 회귀선을 추정하고, 유클리디안 거리 기준으로 추정한 회귀선의 y절편으로부터 가장 가까운 거리의 중점을 가지는 직선과 가장 먼 거리의 중점을 가지는 직선을 검출한다. 검출된 직선들은 큐브의 외곽선에 해당한다. Fig. 4의 (e)는 회귀선을 이용하여 각 중점의 정렬 관계를 파악하고 점의 순서를 정렬한 결과이며 Fig. 4의 (f)는 정렬 후 처음과 끝의 직선을 추출하여 외곽선으로 검출한 결과이다.

3.4 큐브의 윗면 검출

앞서 검출한 외곽선 중, 윗면에 해당하는 외곽선은 원의 중점으로부터 가장 가까운 두개의 직선이다. 그러므로 검출한 원의 중점과 가장 가까운 외곽선 2개를 검출한다. Fig. 5의 (a)는 외곽선으로부터 원의

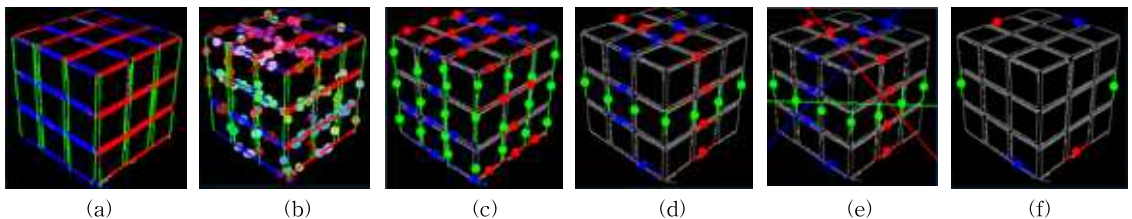


Fig. 4. Cube detection process. (a) line classification by thresholding (b) line merging by partitional clustering (c) line merging result by partitional clustering (d) line merging by using strongly connected graph (e) regression line estimation (f) result of outline detection.

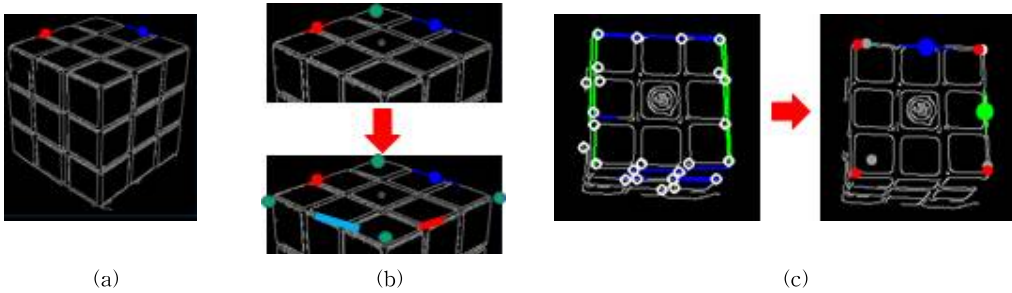


Fig. 5. Upper side detection, (a)two outlines detection for upper side detection (b)Estimation of vertices by parallel transference (c) vertex correction by corner point detection.

중점과 가장 가까운 두개의 외곽선을 추출한 결과이다.

검출된 두 외곽선을 통해 교점을 계산하면 Fig. 5의 (b)의 위의 사진과 같이 윗면의 꼭지점 하나를 추정할 수 있다. 또한 교점으로부터 획득했던 원의 중점과 반대되는 점을 또 다른 꼭지점으로 획득할 수 있으며, 이 꼭지점을 교차하면서 두 외곽선과 평행한 직선을 계산한 뒤 기존의 두 외곽선 간의 교점을 계산하면 Fig. 5의 (b)의 아래의 그림과 같이 윗면에 대한 4개의 꼭지점을 추정할 수 있다. 하지만 이 방식은 평행이동에 대한 꼭지점 추정이며 원근감이 적용되어있는 실제 영상에서는 적절하지 못한다. 따라서 영상 상에서 실제 코너 특징점을 검출한 뒤, 추정된 꼭지점들로부터 가장 가까운 코너 특징점들을 꼭지점으로 대체한다. 그 결과 Fig. 5의 (c)의 오른쪽 그림처럼 보다 정확한 꼭지점 4개를 예측할 수 있다. Fig. 5에서의 회색 점은 평행이동을 통해 추정된 꼭지점들을 나타내며 빨간 점은 코너점을 이용해 보정 후의 꼭지점들을 나타낸다.

꼭지점을 추정한 후, Fig. 6와 같은 오류들을 발견할 수 있다. Fig. 6의 (a)는 검출한 코너 특징점이 충

분하지 못하여 정확한 꼭지점으로 검출되지 못한 경우이며, (b)는 직선 검출당시 직선이 충분히 검출되지 않아 잘못된 외곽선을 검출한 경우이다. 이 두 경우의 에러가 합쳐져 다른 여러 오 검출의 결과를 양산해 낸다. 이 두 경우의 에러를 보정하기 위해 Fig. 6의 (c)와 (d) 같은 방법을 사용한다. Fig. 6의 (c)는 Fig. 6의 (a) 문제를 해결하기 위해 코너 특징점을 사용하기 이전의 계산된 꼭지점을 사용한다. 두 번째 에러의 경우 Fig. 6의 (d)와 같이 추정된 윗면의 선분의 비율을 비교하여 그 비율이 맞도록 꼭지점들을 확장 시켜준다. 윗면의 각 선분을 $u_i(i=1..4)$ 라고 하자. 검출한 원의 중점의 x, y 좌표를 각각 c_x, c_y 라고 하고, u_i 가 포함되는 직선의 방정식이 $a_i x + b_i y = d_i$ 라고 할 때, 중점 c 로부터 각 선분 u_i 로 내린 수선의 발의 길이 $n_i(i=1..4)$ 는 수식 (4)를 통해 구할 수 있다.

$$n_i = \frac{|a_i c_x + b_i c_y + d_i|}{\sqrt{a_i^2 + b_i^2}} \tag{4}$$

이 때, $n_i(i=1..4)$ 중 가장 길이가 긴 값을 N이라고 하자. 가장 짧은 수선의 발의 길이와 N와의 차이가 옅셋(offset) 거리 값 이상의 차이가 날 경우 N보다

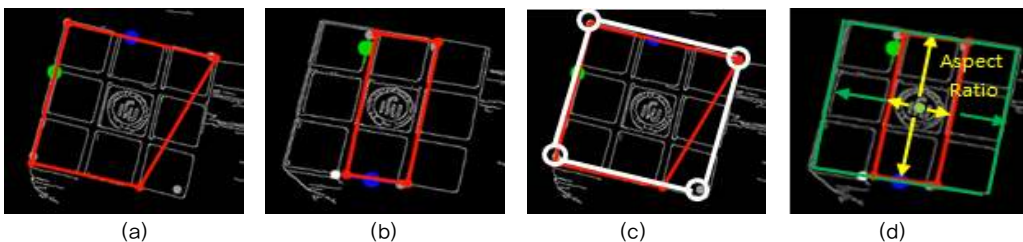


Fig. 6. Problems and Solutions of vertex detection (a) problem of vertex detection caused by lack of corner features. (b) problem of vertex detection caused by inaccurate line detection. (c) solution of vertex detection by using parallel points detected at previous frames (d) solution of vertex detection by expanding lines to match aspect ratio.

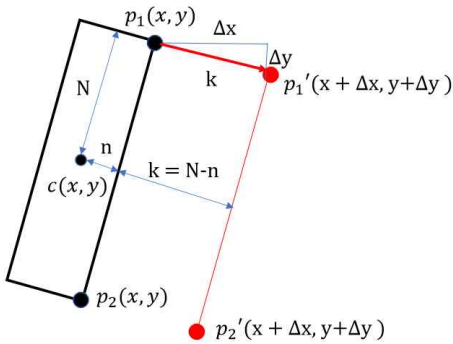


Fig. 7. figure for understanding of a point extension,

짧은 수선의 발을 갖는 직선들을 N길이만큼 확장시켜준다. 여기서 꼭지점 p_1 와 p_2 가 이루는 직선 u 의 수선의 발을 n 이라고 하자. n 은 N 보다 짧으며 $k = N - n$ 은 n 이 확장되어야 할 거리이다. 먼저, 직선이 거리 k 만큼 확장한다는 것은 직선을 이루는 꼭지점 p_1 와 p_2 가 점 c 로부터 거리 k 만큼 수직으로 떨어진다는 것을 의미한다. 이를 계산하기 위해 꼭지점 p_1 와 p_2 가 이루는 기울기 m 으로 부터 직교 값인 $-\frac{1}{m}$ 을 구한다. 점 p_1, p_2 가 점 c 로부터 수선으로 k 만큼 더가기 위해 직교좌표계에서 피타고라스 정의에 의해 (5)로 표현하고, (5)와 (6)을 연산하면 (7)의 결과를 얻을 수 있다.

$$\Delta x^2 + \Delta y^2 = k \tag{5}$$

$$\frac{\Delta y}{\Delta x} = A, A = -\frac{1}{m} \tag{6}$$

$$\Delta x = \pm \sqrt{\frac{k^2}{1+A^2}}, \Delta y = \Delta x * A \tag{7}$$

따라서, 점 p_1, p_2 로 부터 $\Delta x, \Delta y$ 만큼 이동하면 점 c 로부터 수선의 발을 따라 거리 k 만큼 확장한 p_1', p_2' 으로 이동할 수 있다. Fig. 7은 위의 식의 이해를 돕기 위한 그림이다.

마지막으로, 보다 안정적인 자세 보정을 위해 세 프레임동안 검출된 각각의 꼭지점들의 평균 값을 최종 꼭지점으로 사용한다.

3.5 윗면의 패턴 인식

큐브 윗면에 대한 패턴을 찾기 위해서는 먼저 영상 상에 투영 변환(projective transform)[14]에 의해 마름모, 혹은 평행사변형 모양으로 왜곡된 윗면을 정사각형으로 펼쳐줄 필요가 있다. 이 때, 투영 변환 왜곡을 보정한 큐브의 윗면을 탐부라고 명명한다. 윗면의 각 꼭지점과 펼치고자 하는 정사각형의 네 꼭지점은 서로 대응되어야 한다. 때문에 큐브 윗면의 꼭지점을 찾는 과정에서 어떤 가장자리 선분(edge)을 검출 하는지에 따라 해당 꼭지점이 무작위의 순서로 계산되므로, 이를 다시 좌상단을 기준으로 시계방향으로 재배열하는 과정이 필요하다. 재배열을 위해서는 얻어진 네 꼭지점을 영상의 원점을 기준으로 유클리디안 거리 상 가장 가까운 순서대로 정렬한다. 그렇게 되면 윗면의 좌상단과 우하단은 항상 처음과 마지막에 저장되게 되는데, 그 후에 우상단과 좌하단의 x 좌표를 비교하여 우상단의 점이 항상 두 번째 순서로 저장되게 한다. 마지막으로 좌하단과 우하단의 순서를 바꾸어 저장하면 큐브 윗면의 네 꼭지점을 좌상단을 기준으로 시계방향으로 저장할 수 있다. 이해를 돕기 위해 Fig. 8을 보면 그림과 같이 꼭지점의 순서가 정렬 됨을 확인할 수 있다.

큐브 윗면을 정사각형으로 변환시킨 후, 윗면의 패턴을 인식해야한다. 큐브의 면은 다시 작은 9개의 정사각형으로 이루어져있기 때문에 윗면을 9칸으로 분할한다. 변환된 윗면을 세로로 3등분 가로로 3등분하여 총 9개의 칸으로 나누어준 뒤, 각 칸의 중점에 대한 픽셀 값을 저장한다. 이때, 마커가 포함된 정가운데의 칸은 제외한다. 그 후, 각 픽셀 주위로 약

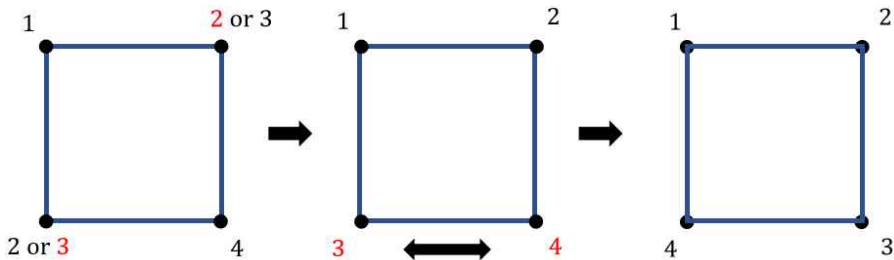


Fig. 8. points rearrangement for clock-wise order.

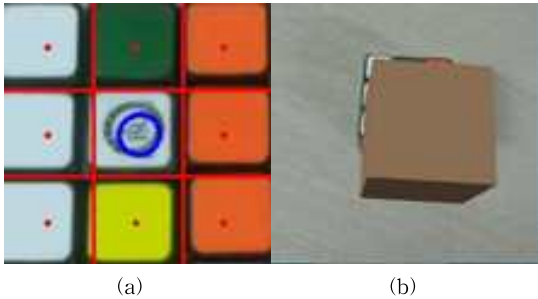


Fig. 9. Pattern recognition process. (a) Rectified cube (b) 3D object augmentation

25개의 픽셀에 대한 Red, Green, Blue 값의 평균을 구하여 해당 색이 빨강, 파랑, 주황, 노랑, 초록, 흰색 중 어떤 색에 속하는지 미리 지정해 놓은 문턱치 값을 통해 구별한다. 마커가 포함된 가운데 칸을 제외한 총 8칸에 대해 검출한 색 정보를 일련의 순서로 저장하여 미리 저장해둔 패턴들 중 어떤 패턴과 일치하는지 비교한다. Fig. 9(a)은 답류를 9칸으로 나누고 색을 검출하는 과정을 나타낸다.

큐브의 패턴을 인식함에 있어서 한 가지 더 고려하여야 하는 것이 큐브의 방향성이다. 즉, 같은 패턴임에도 불구하고 큐브를 0° , 90° , 180° , 270° 회전시켜 다른 방향성을 갖게 되었을 때, 색의 배열이 다르게 저장되기 때문에 다른 패턴으로 인식하는 것을 지양하고 같은 패턴으로 인식하는 과정이 필요하다. 이를 위해 하나의 패턴에 대해 4가지 각도에 대한 경우를 모두 저장하여 현재 영상에서 큐브가 어떤 방향을 갖는지 판별하고, 검출된 패턴의 일련의 순서 비교를 통해 존재하는 패턴이 있을 경우 네 꼭지점의 순서를 방향성이 없는 패턴(0°)을 기준으로 순서를 정렬해 준다. 그 결과 패턴이 어떠한 회전에도 강인한 객체 증강이 가능하다. 마지막으로 패턴 인식과정을 통해 어떠한 객체를 증강할 것인지에 대해 결정한다.

3.6 객체 증강

획득된 영상에 물체를 자연스럽게 증강시키기 위해 해당 영상을 획득할 당시의 카메라의 자세를 알아야한다. 이 카메라 자세에는 카메라가 영상을 찍고 있을 당시의 회전과 이동값에 대한 정보를 내포하고 있는데, 이 정보를 알면 카메라의 자세를 추정할 수 있다. 이는 SolvePnP라는 문제를 해결함으로써 카메라의 회전과 이동 값을 추정할 수 있는데, 영상

상에서 구해진 일련의 점들과 3차원 공간상에서 그와 대응되는 점들 간의 관계 정의를 통해 2차원 영상에 투영된 3차원 점의 관계를 도출해내어 2차원 영상의 회전과 이동 값을 유추해낸다.



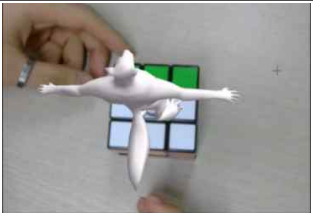


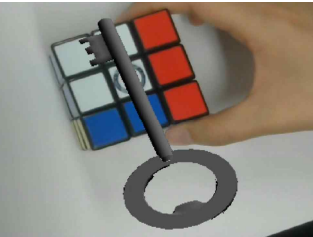


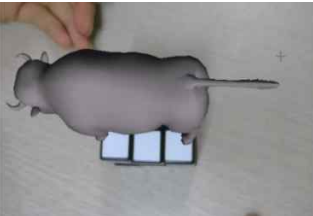

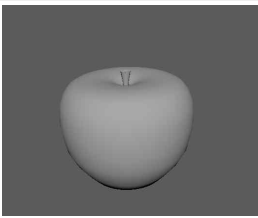
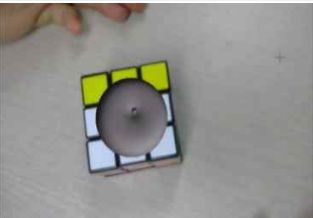


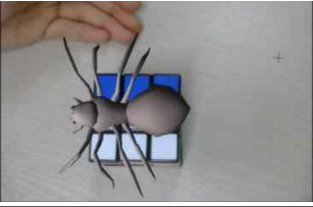
가장 먼저, 카메라 캘리브레이션(camera calibration)과정을 통해 카메라의 내부 파라미터(Intrinsic) 정보를 추정하여 왜곡을 제거한다. 그 후 영상으로부터 유추한 꼭지점 4개에 대한 좌표를 원점으로부터 가장 가까운 점을 처음으로 하여 시계방향순서로 3차원 공간상의 좌표 4개(x, y 평면에 대하여 (0,0,0), (0,5,0), (5, 5, 0), (5, 0, 0))와 매칭시킨다. 그렇게 되면 큐브의 윗면에 해당하는 좌표를 3차원 공간상의 좌표와 2차원 영상상의 좌표와의 매칭(matching)을 통해 영상을 획득한 당시의 카메라 자세를 추정할 수 있으며 그 회전과 이동 값을 3D 객체를 영상위에 렌더링(rendering)할 당시 역으로 곱해주면 마치 그 위치에서 물체를 보고 있는 것과 같은 자세로 물체 렌더링이 가능해진다. 물체를 렌더링하는 것은 OpenGL library를 사용하였으며 OpenGL상에서 렌더링된 객체를 추후에 영상위에 합성해준다. 그 결과 Fig. 9의 (b)와 같이 영상을 획득할 당시의 카메라 자세의 정보가 계산되어 렌더링된 객체가 영상위에 합성된다.

4. 실험결과

본 시스템은 visual studio 2013에서 C++언어와 OpenCV 3.0, OpenGL을 사용하여 개발되었으며 i5 64bit CPU와 8G RAM, intel HD Graphics 520의 노트북 환경에서 개발되었다. 파라미터 값으로는 마커 검출을 위한 ROI 크기로 100×100 pixel을 잡았으며, 직선 분류를 위한 D 값은 10° , 클러스터링을 이용한 병합을 위한 k 거리로는 5, 윗면 검출에서 꼭지점 확장을 위한 offset값은 20을 주었다. 실험은 webcam을 이용하여 녹화된 물체 증강 영상의 1000프레임을 검사하여 패턴마다 다른 물체가 증강되는지의 여부와 함께 큐브의 기울임 정도에 따라서 물체 증강의 정확도 및 속도를 확인하였다. Table 1은 실험에 사용한 루빅스 큐브, 가상 객체, 그리고 가상 객체가 증강된 결과를 나타낸다.

패턴에 대하여 물체가 다양한 각도에서 신뢰할 수 있고 안정적인 증강이 되는지 확인하기 위해 증강률과 정확도를 계산한다. 증강률은 물체가 화면에 증강이 되는가에 대한 신뢰정도이며 정확도는 물체가 정

Table 1. The Result of real-time AR from each of Object corresponding patterns

Index	Pattern	Test object	Result of real-time AR
1			
2			
3			
4			
5			

확한 자세로 증강이 되는지에 대한 안정정도에 해당한다. 우리는 큐브를 45도 이상 기울이면 마커인 원을 검출할 수 없어 윗면 찾기가 어려울 것으로 제한 조건을 주었으며, 그 결과 Table 2에 나타나듯이 증강률은 모두 100%를 보였고, 정확도는 탐류에서 100%, 나머지는 70%를 나타내었다.

패턴에 따라 다양한 물체를 증강이 이루어지는지 실험하기 위해 제안 시스템은 5개의 패턴을 저장하였으며 5개의 패턴에 대하여 각각 다른 객체를 지정

하여 객체가 정확히 증강하는 것을 확인할 수 있었다. 또한 저장한 패턴이 아닌 패턴이 들어오거나 큐브가 없을 경우 물체를 증강하지 않으며 사용자의 손이 영상에 있거나 다른 물체가 있더라도 큐브를 정상적으로 검출하여 객체를 증강하였다.

본 시스템의 속도는 Table 3에 나타나있듯이 증강하는 객체의 크기에 따라 차이가 있으나 그 차이가 크지 않으며, 평균 0.131sec의 속도를 나타내었다. 즉, 제안 시스템은 평균 10fps 정도로 가상 객체를

Table 2. Augmentation rate and accuracy from each of views

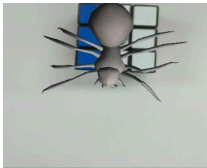
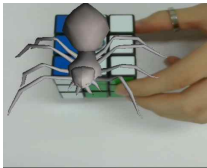
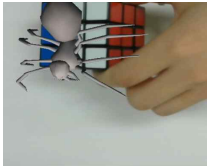
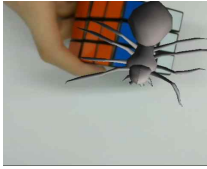
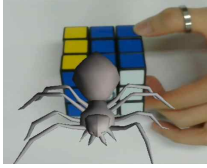
View	figure	Augmentation ratio	accuracy
Top view		100%	100%
Up 45°		100%	70%
Left 45°		100%	70%
Right 45°		100%	70%
Down 45°		100%	70%

Table 3. Computational time of the proposed system according to each test object

Test object	1	2	3	4	5
File Size	50KB	147KB	54KB	107KB	7KB
System Speed	0.134sec	0.134sec	0.126sec	0.134sec	0.126sec

증강하며 따라서 보다 실감 있는 증강현실 시스템을 구해서는 계산량에 대한 개선이 이루어져야 할 것으로 판단된다. 그러나 특징점 검출 기반 방식으로 카메라 위치 추정이 전혀 이루어지지 않은 점에 비해 제안 기법은 일관성을 가지고 카메라 위치 추정이 이루어진다는 장점이 있으므로 직선이 다수 존재하면서 특징점 검출이 어려운 물체에 대한 증강현실 시스템 구현에 제안 시스템이 유용하게 활용될 수 있을 것으로 판단된다.

5. 결 론

본 논문은 루빅스 큐브를 이용하여 3차원 다중 객체를 증강할 수 있는 기념품을 제안하였다. 제안 시스템에서는 입력 영상에서 큐브의 윗면을 추출하여 윗면의 색깔 패턴을 인식한 뒤 패턴에 맞는 3D 객체를 결정하여 카메라 위치에 따라 객체를 증강한다. 제안하는 시스템은 윗면에 대한 수많은 패턴에 대해 수많은 종류의 객체를 동적으로 증강시킬 수 있어 3D 기념품 산업에 기여할 수 있다. 또한 제안하는

시스템은 AR을 접목시킨 새로운 아케이드 완구에도 적용이 가능하며, 큐브가 아니더라도 많은 직선을 내포하고 있는 형태의 물건을 검출하고 증강하는데 유용할 것으로 기대된다. 향후 연구방향으로는 급격한 조명의 변화와 각도 변화에도 강인한 증강현실 시스템을 개발할 예정이다.

REFERENCE

- [1] Apple, Google, Microsoft-Who is Winning the Augmented Reality War?, [http:// www.ar-intelligence.info/2017/06/apple-google-microsoft-who-is-winning-the-augmented-reality-wars/](http://www.ar-intelligence.info/2017/06/apple-google-microsoft-who-is-winning-the-augmented-reality-wars/) (accessed July, 3, 2017).
- [2] Pokémon Go Catches Five New World Records, <http://www.guinnessworldrecords.com/news/2016/8/pokemon-go-catches-five-world-records-439327> (accessed July, 3, 2017).
- [3] Unity with Native Tango Support and Vuforia Integration Coming Later This Year, <https://augmented.reality.news/news/unity-with-native-tango-support-vuforia-integration-coming-later-year-0177435/> (accessed July, 3, 2017).
- [4] The Perspective-Three-Point Principle, <http://iplimage.com/blog/p3p-perspective-point-overview/> (accessed May, 28, 2017).
- [5] N. David, "Preemptive RANSAC for Live Structure and Motion Estimation," *Machine Vision and Applications*, Vol. 16, Issue 5, pp. 321-329, 2005.
- [6] M. Raul, J.M.M. Montiel, and J.D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, Vol. 31, Issue 5, pp. 1147-1163, 2015.
- [7] Y. Lee, and Y. Seo, "Vision-Based SLAM in Augmented/Mixed Reality," *Journal of Korea Multimedia Society*, Vol. 13, Issue 3, pp. 12-20, 2009.
- [8] J. Illingworth and J. Kittler, "The Adaptive Hough Transform," *IEEE Transactions on Pattern an Analysis and Machine Intelligence*, Vol. 9, Issue 5, pp. 690-698, 1987.
- [9] C. John, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern an Analysis and Machine Intelligence*, Vol. 8, Issue 6, pp. 679-698, 1986.
- [10] P.V.C Hough, *Method and Means for Recognizing Complex Patterns*, United States, 1962.
- [11] D. Kim, *C++ API OpenCV Programming: Digital Image Processing by OpenCV*, Korea, 2015, Kame, Gyeonggi-do.
- [12] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms: Second Edition*, The MIT Press and McGraw-Hill, Cambridge, Massachusetts London, 2001.
- [13] D.A. Freedman, *Statistical Models: Theory and Practice*, Cambridge University Press, England, 2009.
- [14] R.C, Gonzalez, and R.E. Woods, *Digital Image Processing: Third Edition*, Pearson Education International, New Jersey, 2010.



이 상 준

2017년 한동대학교 전산전자공학부 학사
2017년 ~ 현재 한동대학교 정보통신공학부 학, 석사 통합과정
관심분야: VR, AR, 그래픽스, 패턴인식, 영상처리, 머신러닝



김 수 빈

2018년 2월 한동대학교 전산전자공학부 학사 졸업 예정
관심분야: 영상처리, 증강현실



황 성 수

2002년 한동대학교 입학
2008년 한동대학교 졸업(전산전자 복수전공)
2008년 한국과학기술원 전기 및 전자공학과 입학
2010년 석사학위 취득
2015년 박사학위 취득
2015년 - 현재 한동대학교 전산전자공학부 조교수
관심 분야: 컴퓨터 비전을 활용한 자율 주행 이동체 개발 및 증강현실 콘텐츠 개발, 3차원 데이터 생성 및 부호화 및 장애인을 위한 인터페이스 개발