

iOS 기반 실시간 객체 분리 및 듀얼 카메라 합성 개발[☆]

Development of Real-Time Objects Segmentation for Dual-Camera Synthesis in iOS

장 유 진¹
Yoo-jin Jang

김 지 영¹
Ji-yeong Kim

이 주 현²
Ju-hyun Lee

황 준^{1*}
Jun Hwang

요 약

본 논문에서는 모바일 환경에서 실시간으로 전면과 후면 카메라의 객체를 인식하여 객체 픽셀의 영역을 분할하고 이미지 처리를 통해 합성하는 방법을 연구하였다. 이를 위해 Apple사의 iOS에서 제공하는 듀얼 카메라에 DeepLabV3 머신러닝 모델을 적용하여 객체를 분할하였다. 또한 이미지 합성 및 후처리를 위해 Apple사의 코어 이미지와 코어 그래픽 라이브러리를 이용하여 영역의 배경 제거 및 합성 방식을 제안하고 구현하였다. 또한, 이전 연구에 비해 CPU 사용량을 개선하였고 깊이와 DeepLabV3의 처리 속도를 비교하여 처리 결과에 영향을 주는 요소를 분석하였다. 마지막으로 이 두 방식을 활용한 카메라 애플리케이션을 개발하였다.

☞ 주제어 : iOS, Semantic Segmentation, 깊이, 영상처리, 듀얼카메라, 모바일 애플리케이션

ABSTRACT

In this paper, we study how objects from front and back cameras can be recognized in real time in a mobile environment to segment regions of object pixels and synthesize them through image processing. To this work, we applied DeepLabV3 machine learning model to dual cameras provided by Apple's iOS. We also propose methods using Core Image and Core Graphics libraries from Apple for image synthesis and postprocessing. Furthermore, we improved CPU usage than previous works and compared the throughput rates and results of Depth and DeepLabV3. Finally, We also developed a camera application using these two methods.

☞ keyword : iOS, Semantic Segmentation, Depth, Computer Vision, Dual Camera, Mobile Application

1. 서 론

스마트폰 보급률이 높아지고 사람들의 SNS 사용률이 증가하면서 사진 촬영은 일상을 기록하는 수단이 되었다. 또한 오늘날에는 자신을 표현하고 이를 다른 사람과 공유하는 수단으로까지 확장되었다[1, 2]. 특히 SNS를 통해 사진 및 동영상 콘텐츠를 주로 소비하고 스마트폰 사용이 익숙한 미래 소비자층인 Z세대는 스마트폰 카메라를 콘텐츠 생산 및 자기표현의 방식으로 활용한다[3]. 기존 디지털카메라의 장점이었던 고해상도, 야간 모드 등의 다

양한 기술들은 스마트폰 카메라로 이전되어 왔으며 깊이 카메라, 듀얼 카메라, 3D 카메라 등 다양한 카메라 센서들이 스마트폰에 탑재되었다. 이는 스마트폰 사용자가 다양한 촬영 기술들을 스마트폰으로 경험할 수 있도록 촉진시켰다[4]. 카메라 센서 뿐 아니라 스마트폰에 탑재되는 GPU의 발전으로 인해 영상처리, 인공지능 등 많은 연산을 수행하는 기술들이 지원되었다[5]. 수많은 카메라 애플리케이션들은 필터, 편집, 셀피(Selfie) 등 특수한 목적에 맞게 구현되어 애플리케이션 시장에 출현하였다. 특히 사진에서 인위적인 표현보다는 자연스러운 필터와 합성을 이용하여 원본 느낌이 날 수 있도록 시간을 소요하는 것에 부담을 느끼지 않고 자신이 원하는 대로 콘텐츠를 생성하고 재가공함으로써 자기표현의 욕구를 충족시킨다. 사람들의 이러한 욕구를 반영하여 카메라 애플리케이션들은 카메라에서 영상 인식 등 인공지능 기술을 접목하여 사람들에게 양질의 콘텐츠를 제공하고 있다[3, 6]. 본 논문에서는 iOS를 기반으로 하는 모바일 환경의 전면과 후면 카메라에서 객체 인식 및 분할 방법을 비교해보고 전면과 후면 카메라 합성 시 후처리 하는 방식을 제안

1 Dept. of Software Convergence, Seoul Women's University, Seoul, 01797, Korea.

2 Dept. of Communication and Media, Seoul Women's University, Seoul, 01797, Korea.

* Corresponding author (hjun@swu.ac.kr)

[Received 26 February 2021, Reviewed 12 March 2021(R2 28 Apr 2021), Accepted 12 May 2021]

☆ 이 논문은 서울여자대학교 교내연구비의 지원을 받았음 (2021-0058).

☆ 본 논문은 2020년도 한국인터넷정보학회 추계학술발표대회 우수논문 추천에 따라 확장 및 수정된 논문임.

한다. 본 논문의 구성은 다음과 같다. 2장에서는 본 연구에 사용된 객체 분할 기술과 관련 연구에 대해 설명한다. 3장에서는 객체 분할 및 이미지 후처리 방식을 기술한다. 4장에서는 3장의 구현 결과 및 개선사항을 기술한다. 5장에서는 연구한 내용을 바탕으로 사용자가 커스터마이징을 할 수 있는 카메라 모바일 애플리케이션을 구현한 내용에 대해 기술한다. 6장에서는 결론과 향후 연구 방향에 대해 제시한다. 이 연구를 통해 사용자에게 편집과 촬영을 동시에 할 수 있도록 하는 편리성을 제공하고 합성을 통한 자연스러움을 제공하고자 한다. 뿐만 아니라 사용자들에게 듀얼 카메라를 이용한 콘텐츠를 제공하여 스마트폰에 탑재된 카메라 센서들을 충분히 활용할 수 있도록 한다.

2. 관련 기술 및 연구

2.1 깊이를 이용한 Object Segmentation

깊이를 이용한 분할 중 픽셀 기반의 분할은 픽셀들의 분포를 확인한 후 히스토그램에서 적절한 Threshold를 기준으로 이진화 등의 기법을 통해 픽셀을 분할하는 방법에서 Threshold를 거리의 개념으로 접근하고 이미지의 깊이 정보를 획득하여 분할할 수 있다[7]. 뿐만 아니라 깊이 정보를 얻기 위해 RGB 이미지에서 깊이 정보를 파악하여 특정 영역의 픽셀만 분할하는 방법이 있으며, 두 카메라간의 차이와 투영 물체간의 차이를 이용해 파악한 깊이 정보로 픽셀을 분할하는 방법이 있다[8, 9]. 깊이 정보를 파악하기 위해서 아이폰에서는 전면의 True Depth 카메라와 후면의 Dual 카메라를 이용하여 깊이 정보를 포착한다[10]. TrueDepth 카메라는 3D 센싱 카메라로 Structured light 원리를 이용하여 30,000여개의 점을 투영하고 분석하여 깊이 맵을 만든다[11]. 그림 1은 TrueDepth 카메라에서 포착한 DepthMap에 RGB 값이 부여된 이미지이며, 깊이가 얇을수록 붉은 색을 띄고, 깊이가 깊을수록 푸른색을 띄고 있다. 그림 2에서는 DepthMap를 시각화하여 이미지로 표현하였다. 카메라에서 객체와의 거리가 멀수록 깊이는 깊으며 객체와의 거리가 가까울수록 깊이가 얇다. 이러한 깊이 정보를 통해 이진화 과정을 거쳐 가까운 거리의 객체 분할이 가능하다. 그러나 깊이 분할은 이미지를 입력으로 받는 카메라 기기의 위치나 환경 조건과 전처리에 영향을 받는다[12]. iOS에서는 듀얼 카메라에서의 깊이를 이용하여 객체 분할을 하고 합성이 가능하지만 공식 문서에 따르면 전면과 후면 두 카메라

에서의 깊이를 동시에 호출하는 것은 지원이 되지 않는다[13].

2.2 합성곱층을 이용한 Object Segmentation

합성곱층을 이용한 분할은 이미지의 특징과 이미지의 공간 정보를 유지할 수 있기 때문에 해당 영역의 픽셀을 클래스로 구분하여 의미론적인 영역을 찾을 수 있다[14]. DeepLabV3은 ResNet을 기반으로 하는 대표적인 합성곱층 기반의 객체 분할 네트워크로 DeepLab1, 2의 성능을 개선한 모델이다[15]. 현재 Tensorflow에서 Pascal VOC 데이터셋을 이용하여 학습된 model 소스코드가 제공되고 있으며 21개의 클래스로 픽셀의 분류가 가능하다[16]. 픽셀의 의미론적인 영역과 위치를 인식할 수 있기 때문에 물체가 이동하더라도 픽셀의 클래스를 구분한다.

본 논문에서는 DeepLabV3 모델을 이용하여 전면과 후면 두 카메라에서 의미론적으로 동시에 실시간으로 객체 분할을 하는 방법을 연구하고 후처리 기법을 제안한다.

DepthMap



(그림 1) TrueDepthCamera 깊이맵 RGB화

(Figure 1) DepthMap of TrueDepthCamera in RGB(10)



(그림 2) TrueDepthCamera 깊이맵 시각화

(Figure 2) Visualizing DepthMap of TrueDepthCamera

3. 듀얼카메라에서의 객체 분할 및 합성 기법 설계

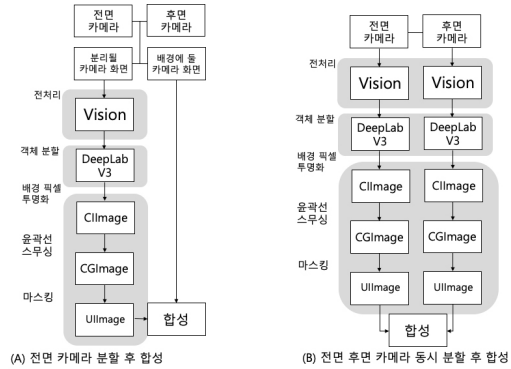
본 연구에서는 2절에 서술된 기술을 활용하여 카메라에서 실시간으로 사람이 분할된 픽셀 영역에 접근하고 코어이미지와 코어그래픽스 라이브러리를 이용한 후처리를 통해 배경을 투명화하고 합성하였다. 카메라는 아이폰에서 지원되는 `AVCaptureMultiCamSession`을 통해 전면과 후면 카메라를 동시에 입력 받는다. 카메라의 미디어 데이터 픽셀에 접근할 수 있도록 `Video Sample Buffer`로부터 `CVPixelBuffer`를 받아온다. 픽셀이 캡처될 때마다 실시간으로 새로운 `CVPixelBuffer`가 호출되고 이를 코어 이미지, 코어 그래픽 라이브러리를 통해 픽셀 이미지의 필터 적용 및 마스크 등 다양한 처리를 진행할 수 있다.

3.1 DeepLabV3 기반 객체 분할 및 합성 설계

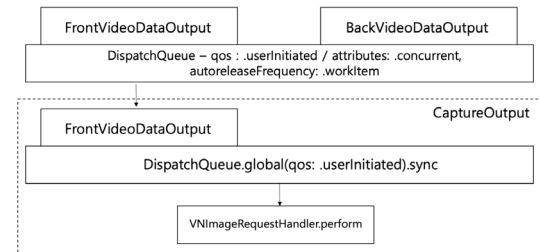
듀얼 카메라에 객체 분할 및 합성을 적용하는 세부 절차는 아래와 같다. 그림 3(A)는 전면카메라에서 객체를 분할하고 후면카메라와 합성하여 `UIImage`로 표현하기까지 `CImage`, `CGImage`, `UIImage`로의 변환 과정이다. 그림 3(B)는 전면과 후면에서 동시에 객체를 인식하고 합성하는 과정을 나타낸다.

- 1) `Vision`을 이용하여 분할하려는 카메라 화면의 이미지가 모델의 입력 크기(513X513)에 맞게 전처리 한다.
- 2) `DeepLabV3` 모델에서 나온 `MLMultiArray`의 각 인덱스에 접근하여 픽셀들을 비교하고 바이트 형태로 변환한다.
- 3) 배경(Label : 0)이라고 인식한 클래스의 픽셀과 같은 위치의 마스크 픽셀의 값을 255, 배경이 아닌 픽셀과 같은 위치의 마스크 픽셀은 0으로 바꾸고 `CImage`에서 `GaussianBlur` 필터와 `CGammaAdjust` 필터를 이용하여 마스크 픽셀값이 255에서 0으로 바뀌는 (그 반대도 해당) 경계선에서 블러 처리와 밝기를 조절하여 마스크 픽셀의 윤곽선을 보정하여 샤프한 경계선을 부드럽게 처리한다.
- 4) `CGImage`에서 카메라 화면의 픽셀과 마스크 픽셀과의 마스크 과정을 통해 카메라 화면의 픽셀을 (255 - 마스크 픽셀값)으로 바꾸어 투명화 처리한다.
- 5) 배경이 제거된 카메라의 `UIImage`와 배경에 배치할 카메라의 `UIImage`를 겹쳐서 합성한다.

이전 연구에서는 전면 카메라의 카메라 픽셀과 마스크 픽셀의 매칭 타이밍이 0.3초 정도 지연이 있었으며, 전면 카메라의 CPU 사용량이 500% 대를 유지했다. 해당 문제를 `GCD(Grand Central Dispatch)`를 이용하여 매칭 타이밍을 맞추고 CPU 사용량을 100%대로 낮추었다. 그림 4은 설계한 `GCD` 구조이다.



(그림 3) 듀얼카메라와 DeepLabV3 합성 흐름도 (Figure 3) Flow Diagram of Synthesizing Dual Camera and DeepLabV3



(그림 4) DispatchQueue 세팅 (Figure 4) DispatchQueue Setting

4. 실험환경 및 결과

4.1 실험 환경

본 연구에서는 듀얼카메라가 지원되는 A12 이상의 프로세서인 iPhone XS와 iPhone 12 Pro를 실험 기기로 설정하였다. 깊이 카메라의 지원 OS 버전은 iOS 12.0, 듀얼 카메라의 지원 OS 버전은 iOS 13.0 이상이다. 전면카메라는 `TrueDepthCamera`, 후면카메라는 `WideAngleCamera`로 설정하였다.

4.2 객체 분할 및 합성 구현 결과 분석

그림 5는 DeepLabV3 기반 객체 분할 합성 구현 결과이다. 시각적인 비교 분석을 위해 거울을 바라보며 구현한 프로그램을 테스트 하였다. 그림 5(B)는 전면 카메라의 경계선이 뚜렷하게 분할이 되었으며, 그림 5(C)에서도 전면 카메라와 후면카메라가 실시간으로 동시에 분할이 되었다. 그림 6은 깊이 기반 객체 분할 합성 구현 결과 이미지이다. 깊이를 분할 할 때 듀얼카메라 세션에 Apple사에서 제공하는 DepthDataOutput을 연결하고 나온 깊이맵을 0.0과 1.0 사이로 정규화했다[17]. 이후 깊이의 Threshold를 실시간으로 설정할 수 있도록 한 후 이진화하여 후처리 후 합성하였다. Threshold를 최대로 하였음에도 불구하고 그림 6(A)에서는 벽과 가까이 있을 때에는 객체와 벽의 깊이값이 비슷하여 뚜렷하게 구분이 되지 않는 것을 확인할 수 있다. 그림 6(B)의 경우는 그림 6(A)보다 객체 분할이 잘 되었으나 그림 5(B)보다는 객체의 테두리가 매끄럽지 않게 분할이 된 것을 확인할 수 있다. 또한 그림 6에서 iOS의 카메라 환경 문제로 불가능했던 전면 후면 동시 분할 합성을 그림 5 환경에서는 가능한 것을 확인할 수 있다.



(그림 5) 듀얼카메라와 DeepLabV3를 통한 분리 결과
(Figure 5) The Results of Segmentation using Dual Camera and DeepLabV3

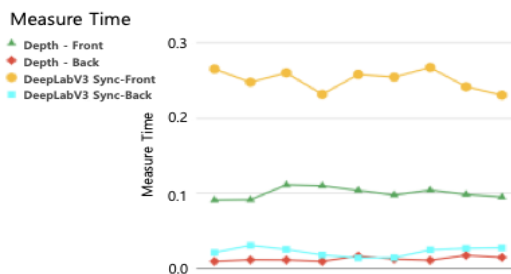
그림 7는 전면 카메라에서 객체 인식 및 배경 분리까지의 처리 시간을 측정하고 도표화 하였다. 그림 7를 보면 깊이가 DeepLabV3보다 약 0.1 ~ 0.2초 정도 처리시간이 적게 걸리는 것을 확인할 수 있다. 반면 후면 카메라의 경우 전면카메라와는 비동기적으로 데이터를 받고 있기 때문에 처리 속도가 차이가 없는 것을 확인할 수 있다.

그림 8는 마스크를 착용 유무와 카메라와 얼굴과의 거리 두 요소를 기준으로 깊이와 DeepLabV3의 인식 및 분리 결과를 나타낸다. 이 실험에서는 후처리 과정 중 윤곽

선 보정을 하지 않았다. 깊이의 Threshold는 0.7로 설정하였다. 그림 8(A)의 경우 DeepLabV3는 마스크를 착용한 사람을 잘 인식 하지 못하나 깊이는 잘 인식하는 것을 알 수 있다. 그림 8(B)의 경우 마스크를 착용하지 않았을 때 깊이에 비해 DeepLabV3가 사람 영역의 픽셀을 더 미세하게 잡아내고 있다. 그림 8(C)에서는 DeepLabV3는 그림 8(A)와 마찬가지로 마스크 착용 시 사람의 얼굴에서 마스크 부분 및 옷 영역은 잘 구분하지 못하나, 머리카락 영역은 세밀하게 구분하고 있다. 깊이의 경우, 거리가 가까운 얼굴 주위의 깊이가 변화하며 머리영역 주변의 픽셀을 크게 잡아내고 있다. 그림 8(D)의 경우 얼굴이 가까워졌을 때 깊이가 그림 8(C)와 비슷하게 요동을 치고 있는 것을 확인할 수 있다. 정확도 측면에서 DeepLabV3는 머리카락 부분을 비교해보았을 때 깊이보다 세밀하게 분리하는 것을 알 수 있다. 그러나 마스크나 옷 등이 분리에 영향을 주고 있다. 깊이는 이 요소들이 영향을 끼치지 않지만 Threshold의 값에 따라 인식할 수 있는 거리와 범위가 달라지기 때문에 이상적인 Threshold를 선정해야 한다.



(그림 6) 듀얼카메라와 Depth를 통한 분리 결과
(Figure 6) The Results of Segmentation using Dual Camera and Depth

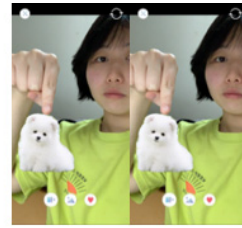


(그림 7) DeepLabV3와 깊이의 측정 시간 그래프
(Figure 7) Graph of Time measuring Depth and DeepLabV3

(좌) Depth - Threshold 0.7 / (우) DeepLabV3



(그림 8) 깊이와 DeepLabV3의 사람 인식 및 분리 비교
(Figure 8) Comparing Depth and DeepLabV3 for Person Segmentation



(A) 스티커 카메라



(B) 듀얼 카메라



(C) 배경 카메라

(그림 9) 구현 애플리케이션 실행 이미지
(Figure 9) Images of Application Implementation

5. 합성 기반 애플리케이션 구현

3장에서 제안한 객체 분리 및 합성 방식을 이용하여 콘텐츠를 만들 수 있는 카메라 모바일 애플리케이션을 구현하였다. 그림 9는 애플리케이션 실행 화면이다. 애플리케이션은 그림 9(A) 스티커 카메라, 그림 9(B) 듀얼 카메라, 그림 9(C) 배경 카메라 총 3가지 카메라로 구성되어 있다. 스티커 카메라는 싱글카메라 상태에서 촬영 혹은 앨범에서 갖고 온 이미지를 DeepLabV3 모델에 입력으로 하여 배경 제거를 한 후 스티커 방식으로 실시간으로 카메라에 붙여 합성한다. 배경이 분리된 화면은 제스처 인식을 통해 회전, 확대 및 축소, 이동이 가능하다. 듀얼 카메라의 경우, 전면과 후면 전환을 통해 분리하려는 카메라 화면에 따라 모델이 받아들이는 픽셀의 입력값이 달라지도록 변경하였다. 사진 촬영을 통해 해당 화면을 촬영할 수 있으며 싱글카메라와 마찬가지로 제스처 인식을 통해 배경이 분리된 화면의 뷰 변환이 가능하다. 배경 카메라는 싱글 카메라에서 깊이를 이용하여 객체를 인식하고 배경에는 고정이미지 뿐만 아니라 프레임별로 움직이는 동영상 느낌이 나게끔 배경 이미지를 사용자가 추가할 수 있게 하였다.

6. 결론 및 향후계획

본 논문에서는 iOS에서 제공되는 듀얼 카메라와 DeepLabV3을 이용하여 카메라 화면에서의 객체 분리를 연구하였다. 또한, 이전 연구에서의 픽셀 매칭 타이밍 개선 및 픽셀 운곽선 보정과정을 거쳤다. 연구 결과, 기존 iOS에서는 전면과 후면 카메라에서 동시에 깊이를 이용한 분할은 기기의 지원 문제로 불가능하였고 Threshold에 영향을 쉽게 받았으나 DeepLabV3 모델을 통해 전면과 후면카메라에서 동시에 분할이 가능하였다. 위 연구 내용을 구현한 애플리케이션으로 실제 상황에서 비교한 결과, 듀얼카메라에서는 배경을 제외한 객체를 잘 인식하지만 사람이 마스크를 착용하고 있거나 모델에 학습되었던 데이터 클래스에 없는 객체일 경우 배경으로 인식하는 등 원래 모델의 학습되었던 데이터에 따라 픽셀의 구분이 불안정해진다는 한계가 있었다. 그러므로 다양한 상황에서 자연스럽게 분리가 적용이 될 수 있는 모델의 성능에 대한 연구가 필요하다. 또한 모바일에서의 모델의 예측 속도와 영상 후처리 시 지연시간도 0.1초 이내로 감소시킬 필요가 있다. 향후 연구에서는 GPU를 이용한 듀얼카메라의 픽셀 고속 연산 수행 속도 및 연구를 통한 성능 개선할 수 있는 방법을 연구하고자 한다.

참고문헌(Reference)

- [1] JS. Oh, JM. Kim, “A Study on the Users’ Attitudes of Photo Editing Application - Focus on SNOW Photo Rectification App”, *Journal of Communication Design*, Vol. 65, No. 10, pp.129-138, 2018.
<http://kiss.kstudy.com/thesis/thesis-view.asp?key=3650108>
- [2] HS. Hwang, “Why Social Comparison on Instagram Matters: Its impact on Depression”, *KSII Transactions on Internet and Information Systems*, Vol.13, No.3, pp.1626-1638, 2019.
<http://dx.doi.org/10.3837/tiis.2019.03.029>
- [3] MJ. Kang, CW. Lee, DY. Lee, YH. Lee, “Identifying Characteristics and Types of Generation Z according to the Behavior of Smartphone Camera Use”, *Archives of Design Research*, Vol. 33, No. 3, pp. 155-175, 2020.
<https://doi.org/10.15187/adr.2020.08.33.3.155>
- [4] HS. Choi, “YouTube, Netflix’s Camera Trend, ‘Video-oriented Change’”, 2020 KISA Report, Vol. 8, No. 8, 2020, pp. 58-62, 2020.
<https://www.kisa.or.kr/synap/doc.html?fn=202008310922085620.pdf&rs=/synapfile/>
- [5] IG. Park, HY. Choi, “Using GPUs for Image Processing in Smartphones”, *Information and Communications Magazine*, Vol. 29, No. 4, pp. 46-51, 2012.
<http://image.inha.ac.kr/paper2/KICS12Park.pdf>
- [6] YJ. Tak, “Smartphone Cameras are now in the age of Artificial Intelligence”, Samsung Display NewsRoom, 2019.
<https://news.samsungdisplay.com/19674>.
- [7] T.H. Dinh, M. T. Pham, M. D. Phung, D. M. Nguyen, V. M. Hoang and Q. V. Tran, “Image segmentation based on histogram of depth and an application in driver distraction detection”, 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), pp. 969-974, 2014.
<https://doi.org/10.1109/ICARCV.2014.7064437>
- [8] J. J. H. López, A. L. Q. Olvera, J. L. L. Ramírez, F. J. R. Butanda, M. A. I. Manzano, D. L. A. Ojeda, “Detecting Objects using Color and Depth Segmentation with Kinect sensor,” *Procedia Technology*, Vol.3, pp.196-204, 2012.
<https://doi.org/10.1016/j.protcy.2012.03.021>
- [9] Y. Wang et al. “Anytime Stereo Image Depth Estimation on Mobile Devices”, 2019 International Conference on Robotics and Automation (ICRA), pp. 5893-5900, 2019.
<https://doi.org/10.1109/ICRA.2019.8794003>
- [10] Apple, “Creating Photo and Video Effects Using Depth”, WWDC 2018, 2018.
<https://developer.apple.com/videos/play/wwdc2018/503>.
- [11] Apple, “About a leading-edge technology of applied to the Face ID”, Apple Support.
<https://support.apple.com/ko-kr/HT208108>.
- [12] CO. Lee, H. Song, BH. Choi, YS. Ho, “Multi-view Generation using High Resolution Stereoscopic Cameras and a Low Resolution Time-of-Flight Camera”, *The Journal of Korean Institute of Communications and Information Sciences(JKICS)*, Vol. 37, No. 4, pp.239-249, 2012.
<https://doi.org/10.7840/KICS.2012.37A.4.239>
- [13] Apple, “Advances in Camera Capture and Photo Segmentation - Introducing Multi-Camera Capture for iOS”, WWDC 2019, 2019.
<https://developer.apple.com/videos/play/wwdc2019/249/>.
- [14] J. Long, E. Shelhamer, T. Darrell, “Fully Convolutional Networks for Semantic Segmentation”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.3431-3440, 2015.
<https://doi.org/10.1109/CVPR.2015.7298965>
- [15] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 4, p.834-848, 2018.
<https://doi.org/10.1109/TPAMI.2017.2699184>
- [16] LC. Chen, G. Papandreou, F. Schroff, H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation”, eprint arXiv, 2017.
<https://arxiv.org/pdf/1706.05587.pdf>
- [17] “Enhancing Live Video by Leveraging TrueDepth Camera Data”, Apple Documentation.
https://developer.apple.com/documentation/avfoundation/cameras_and_media_capture/enhancing_live_video_by_leveraging_truedepth_camera_data.

● 저 자 소개 ●



장 유 진(Yoojin Jang)

2016년~현재 서울여자대학교 소프트웨어융합학과 재학 중
관심분야 : 영상처리, 컴퓨터비전
E-mail : wkddbgyjmail@gmail.com



김 지 영(Jiyeong Kim)

2021년 서울여자대학교 소프트웨어융합학과(공학사)
2021년~현재 이화여자대학교 대학원 인공지능 · 소프트웨어학부 석사과정
관심분야 : 컴퓨터비전, etc.
E-mail : wldud8946@gmail.com



이 주 현(Juhyun Lee)

2020년 서울여자대학교 디지털영상학과(문학사)
2020년 서울여자대학교 시각디자인학과(예술사)
2021년~현재 서울시립대학교 디자인전문대학원 시각디자인학과 석사과정
관심분야 : 그래픽디자인, etc.
E-mail : sallyjuhyun@naver.com



황 준(Jun Hwang)

1987년 중앙대학교 대학원 컴퓨터공학과 졸업(석사)
1991년 중앙대학교 대학원 컴퓨터공학과 졸업(박사)
1992년~현재 서울여자대학교 소프트웨어융합학과 교수
1998년~1999년 Senior Research Professor, Department of Computer Science, Illinois Institute of Technology
2013년~2014년 Research Professor, Department of Electrical & Computer Engineering, Georgia Institute of Technology
관심분야 : 임베디드 시스템, 드론 소프트웨어
E-mail : hjun@swu.ac.kr