일반논문 (Regular Paper)
방송공학회논문지 제27권 제3호, 2022년 5월 (JBE Vol.27, No.3, May 2022)
https://doi.org/10.5909/JBE.2022.27.3.341
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

다중영상을 이용한 딥러닝 기반 온디바이스 증강현실 시스템

정 태 현[a)], 박 인 규[a)‡]

Deep Learning Based On-Device Augmented Reality System using Multiple Images

Taehyeon Jeong[a)] and In Kyu Park[a)‡]

요 약

본 논문은 온디바이스 환경에서 다중 시점 영상을 입력 받아 객체를 증강하고, 현실 공간에 의한 가려짐을 구현하는 딥러닝 기반의 증강현실 시스템을 제안한다. 이는 세부적으로 카메라 자세 추정, 깊이 추정, 객체 증강 구현의 세 기술적 단계로 나눠지며 각 기법은 온디바이스 환경에서의 최적화를 위해 다양한 모바일 프레임워크를 사용한다. 카메라 자세 추정 단계에서는 많은 계산량을 필요로 하는 특징 추출 알고리즘을 GPU 병렬처리 프레임워크인 OpenCL을 통해 가속하여 사용하며, 깊이 영상 추론 단계에서는 모바일 심층신경망 프레임워크 TensorFlow Lite를 사용하여 가속화된 단안, 다중 영상 기반의 깊이 영상 추론을 수행한다. 마지막으로 모바일 그래픽스 프레임워크 OpenGL ES를 활용해 객체 증강 및 가려짐을 구현한다. 제시하는 증강현실 시스템은 안드로이드 환경에서 GUI를 갖춘 애플리케이션으로 구현되며 모바일과 PC 환경에서의 동작 정확도 및 처리 시간을 평가한다.

Abstract

In this paper, we propose a deep learning based on-device augmented reality (AR) system in which multiple input images are used to implement the correct occlusion in a real environment. The proposed system is composed of three technical steps; camera pose estimation, depth estimation, and object augmentation. Each step employs various mobile frameworks to optimize the processing on the on-device environment. Firstly, in the camera pose estimation stage, the massive computation involved in feature extraction is parallelized using OpenCL which is the GPU parallelization framework. Next, in depth estimation, monocular and multiple image-based depth image inference is accelerated using the mobile deep learning framework, i.e. TensorFlow Lite. Finally, object augmentation and occlusion handling are performed on the OpenGL ES mobile graphics framework. The proposed augmented reality system is implemented as an application in the Android environment. We evaluate the performance of the proposed system in terms of augmentation accuracy and the processing time in the mobile as well as PC environments.

Keyword : Deep learning, on-device, augmented reality

Ⅰ. 서 론

증강현실은 실세계에 3차원 가상 물체를 추가하여 보여주는 기술로 최근 스마트폰, 태블릿 PC와 같이 카메라를 내장한 온디바이스 기기가 대중적으로 보급되며 다양한 응용 연구가 진행되고 있다. 증강현실 구현에는 다양한 컴퓨터비전과 컴퓨터그래픽스 기술이 이용되며, 특히 가상과 실제 세계의 공간적 연속성을 파악하여 증강 객체의 가려짐을 올바르게 구현하는 기술은 증강현실에 필요한 실감성 측면에서 필수적인 요소이다.

이러한 차폐인지 (occlusion-aware) 증강현실 시스템은 일반적으로 특수한 센서를 이용하는 방식과 이용하지 않는 방식으로 나뉜다. 특수한 센서를 활용하는 방식은 카메라 자세를 추정하는 가속도 센서와 3차원 공간 정보를 취득할 수 있는 RGB-D 카메라, LiDAR 센서 등을 활용하여 깊이 정보를 직접 취득한다. 이는 차폐인지 증강현실에 필요한 정보를 쉽게 획득할 수 있는 장점을 가지지만 일반적인 단안 카메라를 가진 디바이스에서는 작동할 수 없다는 단점을 가진다. 이에 반해 영상 기반 방식은 별도의 센서 없이 영상만으로 증강현실 구현이 가능하며, 고전적인 다중뷰 컴퓨터비전 기술로 많은 발전을 이루었으나, 최근 심층신경망과 딥러닝 기술의 발전에 따라 더욱 주목받고 있다. 이 경우 고전적인 스테레오 정합 과정 없이 입력 영상만으로 카메라 자세 추정 및 깊이 영상 취득이 가능하다.

본 논문은 이러한 최신 연구의 흐름에 따라 입력 영상의 카메라 간의 거리가 큰(wide-baseline) 다중뷰 영상 입력 시 작동하는 실용적인 증강현실 시스템을 제안한다. 제안하는 시스템은 카메라 자세 추정, 깊이 영상 추론, 객체 증강 세 가지로 나뉘며, 온디바이스 최적화를 위해 다양한 모바일 프레임워크를 통해 알고리즘을 가속한다. 카메라 자세 추정에는 병렬 처리 GPU 프레임워크 OpenCL[7,10]으로 고속 특징 추출 알고리즘 SIFT[9]를 가속하며, 깊이 추정에는 모바일 신경망 프레임워크 TensorFlow Lite[5]를 사용하여 기존 가중치의 양자화를 통해 깊이 영상 추론 속도를 증가시킨다. 객체 증강 시에는 OpenGL ES[3] 그래픽스 프레임워크를 통해 임베디드 GPU를 활용한 빠른 그래픽 렌더링을 수행한다. 제안하는 논문의 기여점을 요약하면 다음과 같다.

- 카메라 자세 추정, 깊이 추정 과정, 객체 증강이 통합된 완성된 증강현실 시스템을 제안
- 병렬처리 프레임워크인 OpenCL 기반 고속 특징 추출 알고리즘을 통한 다중 뷰 기반 카메라 자세 추정 알고리즘 제안
- 모바일 환경에서 구현된 다양한 깊이 추정 모델의 증강현실 시스템에서의 장단점 분석 및 비교

본 논문의 전체적인 구성은 다음과 같다. 2장에서는 신경망 기반 깊이 영상 추정과 모바일 프레임워크에 대한 기술을 소개하며, 3장에서는 제안하는 온디바이스 증강현실 시스템을 기술한다. 4장에서는 구현한 시스템의 동작과 실험 결과를 기술한다. 추가로, 다양한 깊이 영상 추정 모델을 사용하여 전체 처리 속도를 비교한다. 마지막으로 5장에서 논문의 결론을 맺는다.

Ⅱ. 관련 연구

증강현실에서 현실 공간의 정보를 인식하여 증강 객체의 가려짐을 구현하는 연구는 크게 RGB-D와 같은 특수한 센서를 이용한 방법과 입력 영상만으로 이를 구현하는 방법으로 나눠진다. 우선 전자의 방식은 Microsoft Kinect[12]와 같은 RGB-D 카메라를 통해 빠른 속도로 깊이 영상을 취득하고 차폐 증강현실을 쉽게 구현할 수 있다. 하지만 이는 이미 촬영된 다중 영상에 대한 깊이 영상 취득이 불가능하며, 전반사 물체 취득의 어려움, 해상도와 동작 거리와 같은 센서 성능의 한계로 실외 영상에 대한 처리가 부족하다. 또한 카메라 자세 추정 시 영상 간 대응점 기반의 계산이 아닌 가속도 센서인 관성 측정 장치(inertial measurement unit,

a) 인하대학교 전기컴퓨터공학과(Inha University, Department of Electrical & Computer Engineering)
‡ Corresponding Author : 박인규(In Kyu Park)
E-mail: pik@inha.ac.kr
Tel: +82-32-860-9190
ORCID:https://orcid.org/0000-0003-4774-7841
※ This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2020-0-01389, Artificial Intelligence Convergence Research Center (Inha University), No.2021-0-02068, Artificial Intelligence Innovation Hub).
· Manuscript received February 22, 2022; Revised April 12, 2022; Accepted April 12, 2022.

IMU)[17]을 통해 이를 취득한다면 실시간 처리에서만 사용이 가능하다.

영상 기반 방식은 별도의 3차원 정보 없이 입력 영상만으로 카메라 자세와 3차원 깊이 정보를 추론한다. 카메라 자세는 일반적으로 고전적으로 사용되는 영상 특징 추출 및 대응점 정보를 통해 point cloud를 구축하는 structure-from-motion (SfM)[11] 기법을 통해 얻어진다. 깊이 정보 추정 연구의 경우, 과거 사진기하학 기반의 스테레오 정합 방식[6]부터 최근 신경망 기반의 방식이 연구되고 있다. 이중 신경망 방식의 경우 입력 영상의 수에 따라 단안[14,18]과 다중 뷰[16,19] 방식으로 나눠진다. 단안 방식의 경우 모바일 환경에서도 실시간 처리가 가능하다는 장점을 가지지만 한 장의 영상만을 입력 받으므로 여러 영상에서의 상대적 거리가 맞지 않는 스케일 문제를 가진다. 다중뷰 처리의 경우 다중 영상과 카메라 자세를 입력 받아 비용 볼륨 계산을 통해 입력 영상에 대한 깊이 영상을 추론한다. 이는 단안 영상과는 다르게 많은 처리 시간이 필요하지만, 실제 거리 기반의 결과를 출력할 수 있다.

모바일 환경은 고성능의 PC 환경에 비해 CPU 처리 시 많은 시간이 필요하다. 그러나 CPU에서의 처리에 비해 모바일 환경에 최적화되어 있는 다양한 프레임워크를 통해 알고리즘을 가속화하고 처리시간을 줄일 수 있다. 우선 병렬 GPU 프레임워크 OpenCL은 다중 플랫폼에 최적화된 GPGPU 프레임워크로 다양한 디바이스에서 GPU 병렬처리를 할 수 있다는 점에서 매우 주목받고 있다. 3차원 그래픽스 프레임워크 OpenGL은 모바일 환경에서 OpenGL ES를 통해 구현된다. 심층신경망 모델을 모바일 환경에서 구동하기 위해서는 TensorFlow Lite 혹은 PyTorch mobile[4]을 통한 모델 경량화 과정이 선행되어야 한다. 이는 기존의 모델을 양자화하여 모델의 성능을 높여 빠른 추론을 가능하게 한다.

Ⅲ. 제안하는 온디바이스 증강현실 시스템

본 장에서는 카메라 내부 인자가 주어진 다중 입력 영상이 입력되었을 때 현실 공간을 인식하여 증강 객체의 가려짐을 표현하는 차폐인식 증강현실 시스템에 대해 자세히 서술한다. 제안하는 시스템은 카메라 자세 추정, 깊이영상

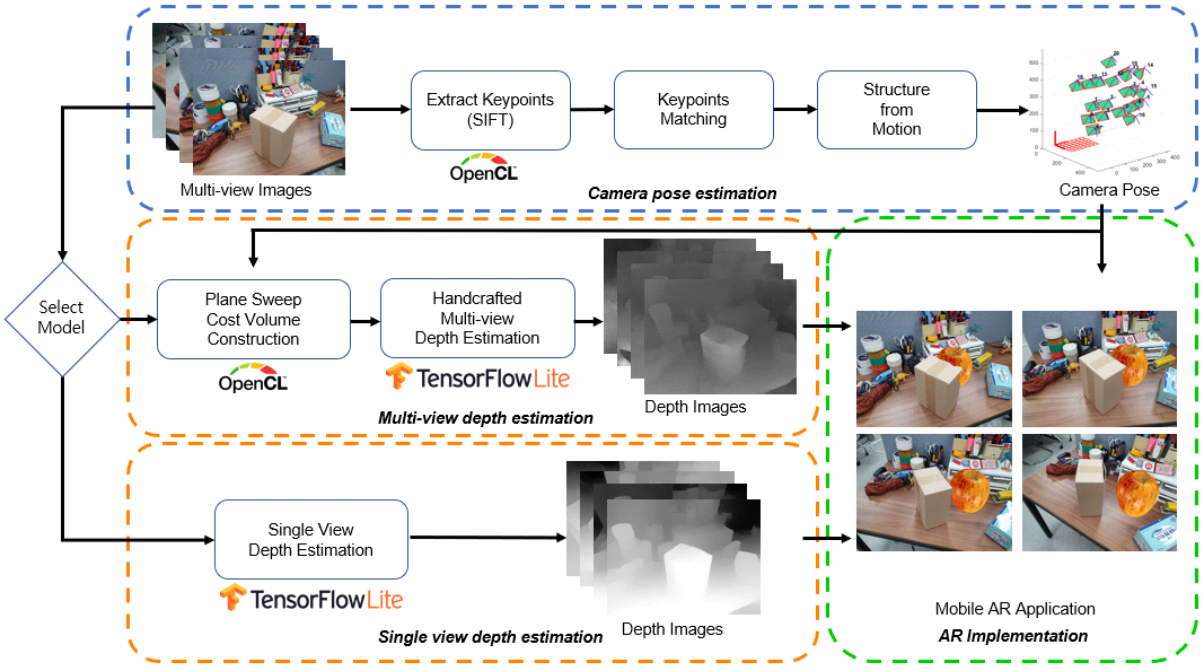

그림 1. 제안하는 공간인식 증강현실 시스템의 전체 파이프라인
Fig. 1. Overview of the proposed AR system

추론, 객체 증강의 세 단계로 나눠지며 제안하는 각 모듈은 임베디드 GPU에 친화적인 모바일 프레임워크인 OpenCL, TensorFlow Lite, OpenGL ES을 사용하여 구현된다. 제안하는 증강현실 시스템의 구조는 그림 1에 도시하였다.

1. 고속 특징 추출 기반 카메라 자세 추정

제안하는 파이프라인은 정확한 객체 증강을 위해 다중 입력 영상 $I_1, ..., I_n$의 카메라 자세 $P_1, ..., P_n$를 추정하고 point cloud를 생성한다. 이는 특징 추출, 이웃 영상간의 대응점 추론, 카메라 자세 추정 및 point cloud 구축으로 나눠진다. 본 시스템은 이웃 카메라 사이의 거리가 큰 (wide-baseline) 다중뷰 영상에서도 증강현실을 구현하고자 하므로 고성능의 특징 추출 알고리즘이 필요하다. 이를 위해 본 논문은 병렬처리 프레임워크 OpenCL을 통해 SIFT 함수를 자체적으로 제작하여 GPU 기반 고속 병렬처리된 SIFT 알고리즘을 사용한다. 이는 Difference of Gaussian (DoG) 기반의 스케일 공간 구축, Non-maximum Suppression (NMS) 기법을 통한 지역적 최대값을 가지는 특징점 검출 과정으로 이뤄진다. 이러한 GPU 기반 OpenCL SIFT 함수를 통해 모든 입력 영상에서 특징점을 검출한다.

다음 과정으로, 이웃한 영상 I_i, I_{i+1}의 특징점에서 K-nearest neighbors 기반의 대응점 추정을 수행하고 대응점의 위치 정보 $M_{i,j+1}$를 저장한다. 이후 취득된 대응 정보를 통해 영상의 카메라 자세를 추정하고 point cloud를 구축한다. 카메라 자세 P_i는 $P_i = K_i[R_i|t_i]$의 3×4 행렬로 표현되며, K_i는 3×3 행렬의 카메라 내부인자를, $t_i \in \mathbb{R}^3$, $R_i \in SO(3)$는 각각 카메라 P_i의 이동, 회전 변환을 의미한다. 또한 첫번째 카메라의 R_1은 단위행렬, t_1는 영행렬로 가정한다.

$$[M_{1,2}^2, 1]^T K_2^T E K_1 [M_{1,2}^1, 1] = 0 \quad (1)$$

$$E = [t_2] \times R_2 \quad (2)$$

처음 두 영상 I_1, I_2에서의 대응점 위치 정보 집합 $M_{1,2}^1$, $M_{1,2}^2$를 이용하여 기하학적인 관계를 추정한다. 대응 정보를 통해 식 (1)을 만족하는 essential matrix E를 계산한다. essential matrix는 식 (2)와 같이 표현이 가능하므로 특이값 분해[13]를 통해 두 영상 사이의 회전, 이동 변환 값을 구할 수 있다. 첫 번째 영상의 카메라 자세는 단위행렬로 가정했으므로 이를 두 번째 영상의 카메라 자세로 사용한다. 이후 triangulation 알고리즘을 통해 3D point cloud를 추정한다. 이는 시점이 다른 여러 영상의 카메라 자세와 대응 정보가 주어질 때 대응점에 해당하는 3차원 점을 구하는 알고리즘으로, 한 영상의 대응점에서 시작하는 직선 ray를 다른 영상으로 투사하며 다른 영상에서의 대응점과 위치 차이를 최소화하는 ray 위 3차원 점을 구한다. 이를 통해 본 논문은 두번째 카메라 자세를 추정하고 처음 두 영상의 대응점에 해당하는 3차원 point cloud를 구축한다.

두 번째 대응부터는 essential matrix를 통해 카메라 자세를 추정하지 않는다. 단순히 이웃한 영상의 대응 정보를 통해 카메라 자세를 추정한다면 영상의 수가 증가할수록 미세한 오류가 누적될 수 있기 때문이다. 그러므로 논문은 solvePnP[20]

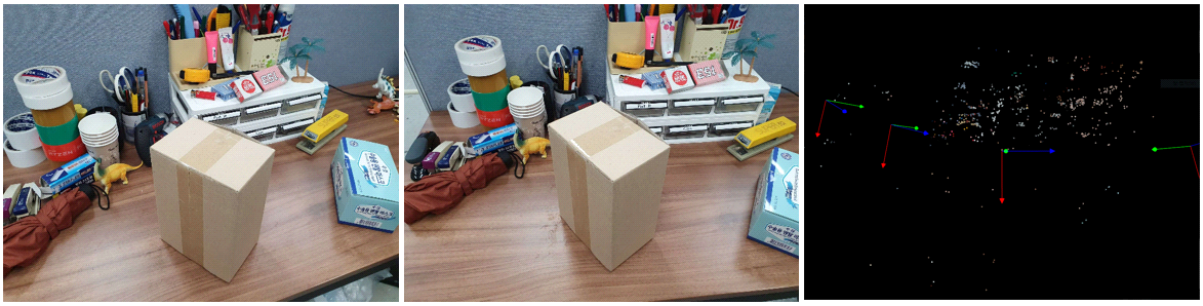

그림 2 입력 다중 영상(좌)에서 취득한 카메라 자세 및 point cloud(우)
Fig. 2. Camera pose and point cloud (right) acquired from multi-view images (left)

알고리즘으로 3차원 point cloud와 이웃 영상의 2차원 특징점 대응 정보를 통해 현재 영상 I_i의 카메라 자세 P_i를 추정한다. 알고리즘은 $\mathrm{M}_{i-2,i-1}^{i-1}$, $\mathrm{M}_{i-1,i}^{i-1}$의 중복된 대응점에 해당하는 m개의 3D point cloud $X_j=(X_j,Y_j,Z_j)^T$와 $\mathrm{M}_{i-1,i}^{i}$의 2D 대응점 $x_j=(x_j,y_j)^T$에서 식 (5)의 비용 함수 C를 최소화하는 회전변환 R_i과 이동변환 t_i를 추정한다.

$$(X_j^{'}, Y_j^{'}, Z_j^{'}) = K_i(\mathrm{R}_i|\mathrm{t}_i)[\mathrm{X}_j, 1] \tag{3}$$

$$\mathrm{x}_j^{'} = (X_j^{'}/Z_j^{'}, Y_j^{'}/Z_j^{'}) \tag{4}$$

$$C = \sum_{j=1}^{m}|\mathrm{x}_j^{'} - \mathrm{x}_j|^2 \tag{5}$$

카메라 자세 추정 이후에는 첫 이웃 영상에서의 처리와 같이 triangulation 알고리즘으로 $\mathrm{M}_{i-1,i}^{i-1}$ 중, $\mathrm{M}_{i-2,i-1}^{i-1}$에 포함되지 않은 대응점에 대해 point cloud를 추가 구축하고 기존의 point cloud와 융합한다. 이를 통해 그림 2와 같이 모든 영상의 카메라 자세 취득과 3차원 point cloud 구축이 가능하다.

2. 고밀도 깊이 영상 추론

본 논문은 증강 객체의 자연스러운 가려짐을 구현하기 위해 모든 화소가 적절한 깊이 값을 가지는 고밀도 깊이 영상을 사용한다. 이는 고전 기하학 기반의 스테레오 매칭, 혹은 신경망 기반의 깊이 추정 네트워크로 획득할 수 있다. 하지만 고전 스테레오 매칭 방식은 카메라 성능에 따라 심각한 노이즈를 출력하며, 복잡한 알고리즘으로 인해 많은 처리 시간이 필요하다. 따라서 본 논문에서는 신경망 기반의 추론 방식을 통해 깊이 영상을 취득한다.

신경망 기반의 깊이 영상 추론 모델은 단일 영상만을 입력 받는 단안 기반과 다중 영상, 카메라 자세를 함께 입력받는 다중 영상 기반의 방식으로 나눠진다. 단안 추론 방식의 경우 수행 속도의 이점이 있지만 한 영상만으로 깊이 영상을 추론하므로 영상 간 거리 일관성이 유지되지 않는다는 단점이 있다. 본 논문은 이를 방지하기 위하여 point cloud 기반으로 학습된 LeReS[18] 네트워크를 사용하여 단안 기반 깊이 영상을 추론한다. 하지만 해당 모델 또한 깊이 스케일 문제가 명확히 해결되지 않으므로 시스템에 사용자가 직접 거리 스케일을 조절하는 기능을 추가한다.

다중 영상 기반 추론의 경우 현재 영상 외에도 다른 영상과 카메라 자세 정보를 입력 받으므로 단안 방식과는 다르게 영상 간 거리 일관성이 유지된 깊이 영상 추론이 가능하다. 따라서 물체와의 거리가 다른 일반적인 다중 뷰 환경에서도 가려짐을 구현할 수 있다. 본 논문에서는 온디바이스 최적화를 위해 OpenCL로 가속화된 Plane-sweeping[8,15] 알고리즘으로 비용 볼륨을 직접 계산하는 모델인 Handcrafted-MVSNet[19]을 사용한다. 이를 통해 거리 일관성이 유지된 깊이 영상 획득이 가능하다. 하지만 해당 방법의 경우 카메라 자세를 통한 비용 볼륨 기반 처리 과정이 포함되기 때문에 많은 처리 시간이 요구되므로 디바이스 성능에 따라 깊이 추정을 위한 입력 다중 영상의 개수를 선택해야 한다.

제안하는 모든 신경망 모델은 모바일 딥러닝 프레임워크인 TensorFlow Lite를 통해 구동한다. 이는 네트워크 가중치의 양자화를 통해 고속 추론이 가능하도록 모델을 변경하며 OpenCL 기반의 모바일 GPU 가속화 과정을 통해 파이프라인의 전체 속도를 증가시킨다.

3. 증강현실 어플리케이션 구현

입력 영상, 카메라 자세, 깊이 영상은 모바일 프레임워크 OpenGL ES를 통해 차폐인지 증강현실을 구현한다. 입력 영상과 깊이 영상은 고속 GPU 처리를 위하여 미리 GL_TEXTURE 형태로 저장한다. 입력 영상은 GL 파이프라인의 fragment buffer로 입력되어 후면 영상을 생성하며 깊이 영상은 depth buffer에 입력되어 영상의 공간성을 표현한다. 객체의 경우 정확한 초기 증강을 위해 구축한 point cloud의 위치에 증강한다. 사용자는 그림 3의 GUI를 통해 객체의 이동, 시점 영상 변경 등의 작업을 통해 물체의 위치를 변경할 수 있다. 이때 모바일 프레임워크 OpenGL ES의 경우 GL 파이프라인 직접 접근이 가능한 2.0 이상의 버전을 사용하여 구현한다.

최종적으로 제안하는 증강현실 시스템은 Android Studio[1]를 통해 그림 3과 같이 구현한다. 이는 시스템에 사용할 특징 추출 알고리즘, 추론 모델을 선택할 수 있으며, 객체 증강 시에는 GUI 조작을 통해 객체 회전 및 이동, 시점 변경, 깊이 스케일 조절 등이 가능하다.

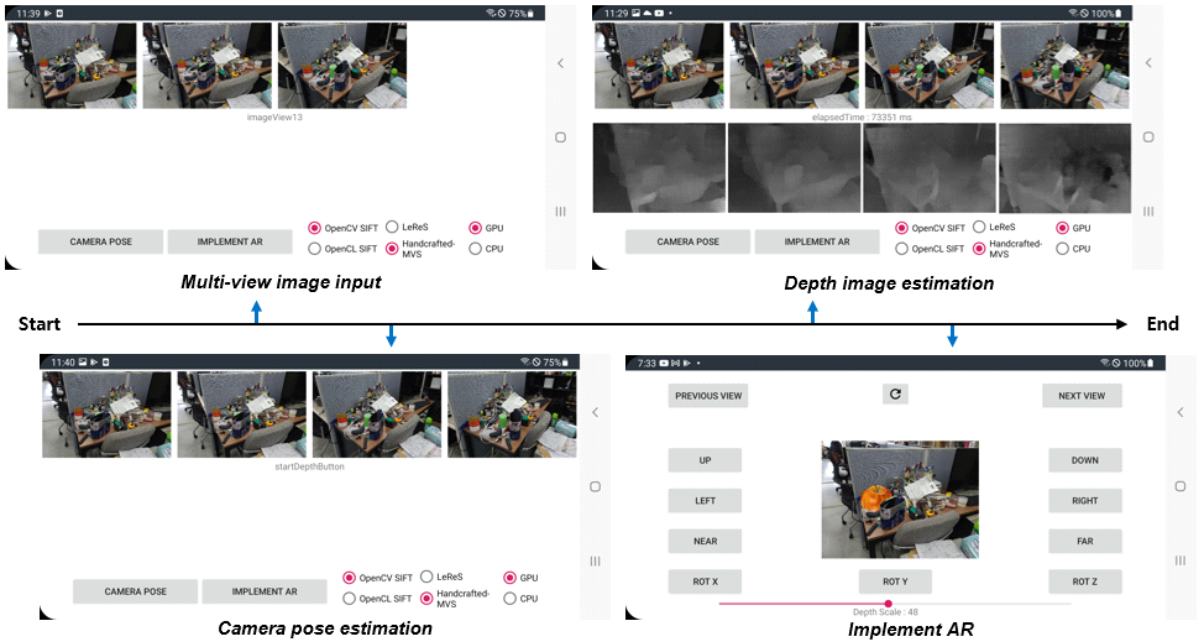

그림 3 공간인식 증강현실 시스템의 GUI
Fig. 3. GUI of spatial-aware AR system

Ⅳ. 실험 결과

1. 특징 추출 알고리즘의 처리 속도 비교

본 장에서는 자체 제작한 GPU 기반 OpenCL SIFT 함수의 가속 정도와 성능을 평가한다. 표 1은 여러 플랫폼에서 1280x720의 HD 해상도 영상의 특징 검출 속도와 개수를 측정한 결과이다. 측정 속도와 가속 배율은 GPU에서 수행되는 OpenCL 기반 SIFT 함수와 CPU에서 수행되는 동일한 알고리즘의 SIFT 함수 간 성능을 비교한다. 모든 플랫폼에서 CPU 기반 함수에 비해 GPU 기반의 OpenCL SIFT가 성능 감소 없이 더욱 빠른 속도로 구동되었으며, 논문이 목표하는 모바일 환경의 경우 약 4배 가속 배율을 기록한다. 이때 각 플랫폼은 GPU 플로팅 계산 속도가 모두 다르므로

표 1. 고속 특징 추출 알고리즘 SIFT의 처리 시간 비교 (단위: 초)
Table 1. Comparison of processing time of SIFT (in seconds)

PC	Intel i7-7700	NVIDIA GTX 1080 Ti	Speed Improvement
Processing Time	1.21	0.113	10.7x
No. of Keypoints	908	1165	-
RK3399	Cortex-A72(2C) + Cortex-A53(4C)	Mali-T860 MP4	Speed Improvement
Processing Time	2.29	1.759	1.30x
No. of Keypoints	908	1259	-
Galaxy NOTE8	2.3GHz Mongoose-M2(4C) + 1.7 GHz Cortex-A53(4C)	Mali-G71 MP20	Speed Improvement
Processing Time	1.68	0.38	4.45x
No. of Keypoints	903	973	-

가속 배율의 차이를 가진다.

2. 다양한 신경망에 따른 성능 비교

본 장에서는 단안, 다중 기반 깊이 추정 모델의 결과와 생성한 증강현실을 정성적으로 평가하며, 모델의 처리 시간을 비교한다. 구현한 증강현실의 정성적 결과는 그림 4와 같다. 단안 기반 추론 모델의 한 영상 내 정보만으로 깊이 영상을 취득하므로 (b)와 같이 영상 간 상대적 거리가 다르다면 일관된 공간 정보 취득이 불가능하다. 따라서 고정된 3차원 공간에서 영상에 따라 카메라 자세를 변경하는 증강현실에서는 (c)와 같이 가려짐 구현에 있어 좋은 성능을 보여주지 못한다. 상자의 경우 모든 영상에서 가장 앞에 있다고 판단하여 가려짐이 잘 구현되지만, 마지막 그림과 같이

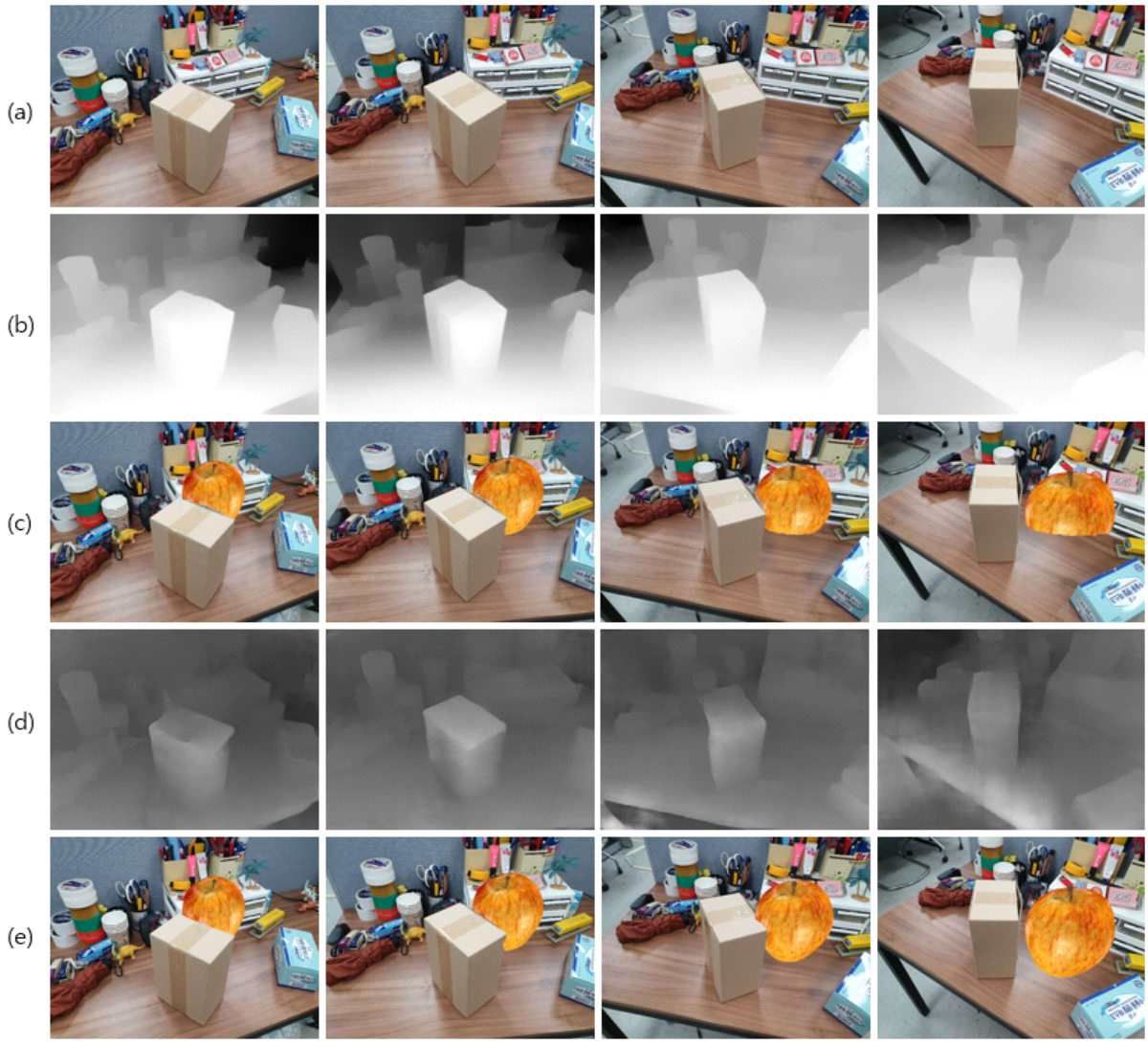

그림 4. 다중영상에서의 공간인식 증강현실 결과 비교 (a) 입력 다중뷰 영상, (b) LeReS[18] 깊이영상 추론 결과, (c) LeReS[18] 증강현실 결과, (d) Handcrafted-MVSNet[19] 깊이영상 추론 결과, (e) Handcrafted-MVSNet[19] 증강현실 결과

Fig. 4. Comparison of augmented reality results in multiple images (a) Input multi-view image, (b) LeReS[18] depth image results, (c) LeReS[18] AR results, (d) Handcrafted-MVSNet[19] depth image results, (e) Handcrafted-MVSNet[19] AR results

영상 간 깊이 영상의 스케일 차이로 인하여 의도치 않은 가려짐이 표현될 수 있다. 이에 따라 본 어플리케이션은 단안 깊이 영상에 대해 수동으로 깊이 스케일을 조절할 수 있는 기능을 추가한다. 다중 영상 기반 모델의 경우 깊이 영상 취득을 위해 많은 시간이 필요하지만, 비교적 상대적 거리가 유지된 결과를 취득할 수 있다. 해당 모델의 경우 깊이 영상 추론을 위해 다른 영상의 정보를 함께 사용하므로 상대적 거리가 유지된 깊이 영상 추론이 가능하다. (e)의 결과는 (c)와는 다르게 별도의 깊이 스케일 변경 없이도 정상적인 객체의 차폐인지가 구현된다. 그림 5는 제안하는 어플리케이션의 다양한 활용 예시이다.

표 2는 CPU, GPU 환경에서 깊이 영상 추론에 필요한

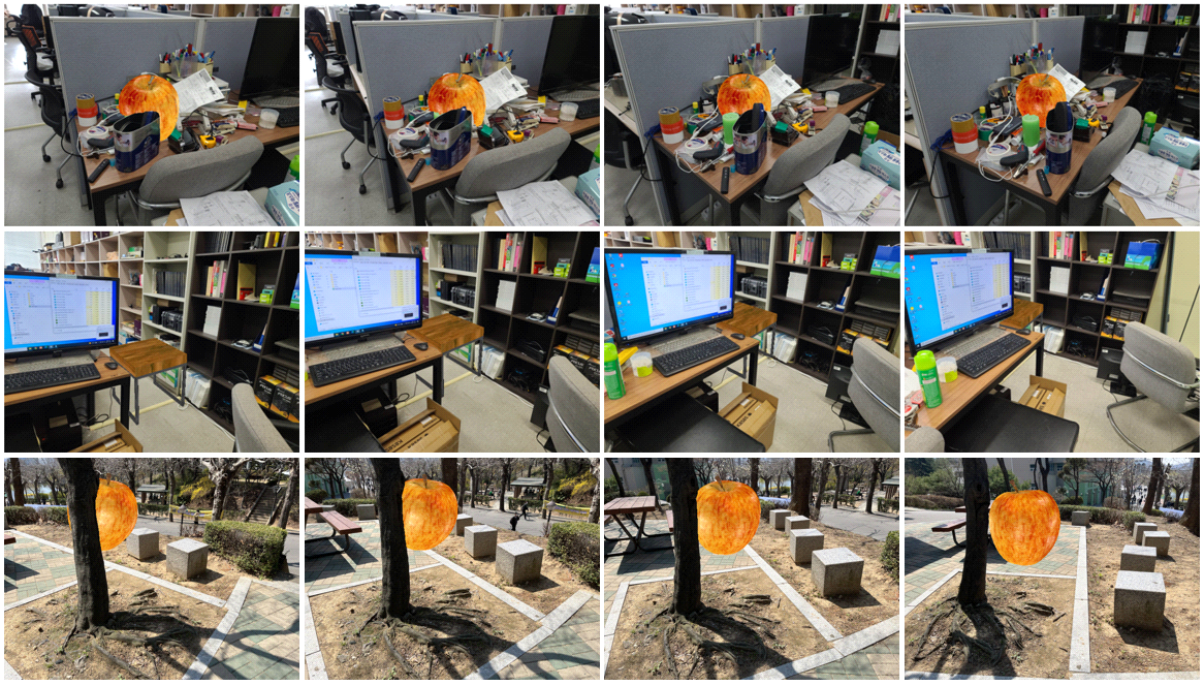

그림 5. 차폐인지 증강현실 시스템의 다양한 예시
Fig. 5. Results of occlusion-aware AR system

표 2. 깊이 영상 추정 네트워크의 처리시간 (단위: 초)
Table 2. Depth estimation processing time (in seconds)

PC	Intel i9-9900KF	NVIDIA RTX 2080 Ti	Speed Improvement
LeReS[18]	4.147	0.018	230.4x
Handcrafted-MVSNet[19]	21.615	0.009	2401.7x
Cost Volume Generation	-	2.357	-
Galaxy S21 Ultra	2.9 GHz Cortex-X1(1C) + 2.8 GHz Cortex-A78(3C) + 2.2 GHz Cortex-A55(4C)	Mali-G78 MP14	Speed Improvement
LeReS[18]	4.483	0.981	4.6x
Handcrafted-MVSNet[19]	166.389	3.723	44.7x
Cost Volume Generation	-	30.153	-

표 3. 깊이 영상 추정 네트워크의 파라미터 개수와 입력 크기
Table 3. Number of parameters and input size of depth estimation network

Algorithm	Parameters	Input Size
LeReS[18]	52,126,529	640 x 480 x 3
Handcrafted-MVSNet[19]	33,898,500	640 x 480 x 67

처리 시간을 측정한다. 모든 모델은 CNN 구조 네트워크로, GPU 병렬 처리를 통한 추론 가속화가 가능하다. 이때 GPU 사용 시, 다중뷰 기반 모델은 단안 기반 모델에 비해 약 10배의 가속 배율을 가진다. 이는 표 3의 모델 파라미터의 개수와 입력 크기 차이에 의한 결과이다. 다중뷰 기반 모델의 경우 3채널 영상 이외에 64채널의 비용 볼륨을 함께 입력받는다. CPU 구동 시, 67채널의 직렬 처리로 인해 다중뷰 기반 모델은 많은 처리 시간을 요구한다. 이와 다르게 GPU 구동 시에는 병렬 연산으로 파라미터의 개수가 적은 다중뷰 기반 모델이 더욱 높은 가속 배율을 기록한다.

다중뷰 기반 모델의 경우 네트워크 추론 이전에 다수의 참조 영상으로부터 Plane-sweeping 알고리즘을 통한 비용 볼륨 생성 과정이 선행되어야 한다. 표 2의 비용 볼륨 생성(Cost Volume Generation)은 하나의 비용 볼륨 생성 처리 시간으로, 참조 영상의 개수를 조절하여 전체 처리 시간 조절이 가능하다.

3. 분석 및 한계

제안하는 어플리케이션은 신경망 기반의 깊이 추정 모델로 공간 정보를 추출하므로 네트워크에 따라 차폐인지 성능이 크게 변한다. LeReS와 같은 단안 기반의 모델의 경우 영상 간 상대적 깊이 스케일이 일관되지 않으며, Handcrafted-MVSNet의 경우 주로 일반적인 사무실 환경에서 모델 학습이 진행되어 실외 영상에 대해서는 낮은 성능을 보인다. 따라서 깊이 추정 네트워크 변경에 따라 본 시스템의 성능 향상이 가능하다.

OpenCL SIFT 함수는 OpenCV[2] 라이브러리 내 SIFT 함수에 비해 빠른 작동이 가능하지만 적은 특징점을 검출한다. 본 파이프라인은 카메라 자세 추정을 위해 소수의 특징점만을 요구하므로 일반적인 영상에 대해서는 OpenCL SIFT 함수로 정상 구동이 가능하다. 하지만 텍스처가 없는 등 특징점 추출이 어려운 영상에서는 충분한 특징점 추출이 수행되지 않아 카메라 자세 추정이 불가할 수 있다. 따라서 이와 같은 영상에서는 OpenCV SIFT 함수를 사용해 특징 검출을 수행하여 안정적으로 프로그램을 구동할 수 있다.

V. 결 론

본 논문은 모바일 환경에서 입력 다중 영상으로부터 카메라 자세 및 깊이 영상을 추출하고 공간에 의한 가려짐을 구현하는 증강현실 시스템을 제안한다. 이는 고전 컴퓨터 기하학 기반의 방식과 심층신경망 기반의 방식을 융합하여 제작하였으며 다양한 모바일 프레임워크를 사용하여 온디바이스 친화적인 통합 파이프라인을 구축한다. 카메라 자세 추정을 위해서는 OpenCL을 통해 가속화된 SIFT를 통해 임베디드 GPU 기반의 병렬 가속화를 진행한다. 깊이 영상 취득에는 신경망 기반의 단안, 다중 모델을 사용하여 각각의 장단점을 정리하여 제시한다. 또한 본 연구는 통합 과정 외에 각 기법에서 자세한 구현 방법을 다루므로 제안하는 애플리케이션 외에 다양한 연구로의 확장이 가능하다. 특히 고속 깊이 추정 모델, 카메라 추정 모델을 통한다면 실시간 차폐인지 증강현실 애플리케이션을 구현할 수 있을 것으로 보인다.

참 고 문 헌 (References)

[1] Android Studio, https://android.com/ (accessed Feb. 2, 2022)
[2] OpenCV, https://opencv.org/ (accessed Feb. 2, 2022)
[3] OpenGL ES, https://www.khronos.org/opengles/ (accessed Feb. 2, 2022)
[4] PyTorch Mobile, https://pytorch.org/mobile/ (accessed Feb. 2, 2022).
[5] TensorFlow Lite, https://www.tensorflow.org/lite/ (accessed Feb. 2, 2022)
[6] A. Ivan and I. K. Park, “A flexible and configurable GPGPU stereo matching framework,” Multimedia Tools and Applications, vol. 79, no. 25, pp. 18367-18386, 2020.
doi: https://doi.org/10.1007/s11042-020-08756-2
[7] A. Munshi, B. Gaster, T. G. Mattson, and D. Ginsburg, OpenCL programming guide, Pearson Education, 2011.
[8] D. Gallup, J. M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, “Real-Time Plane-Sweeping Stereo with Multiple Sweeping

Directions," Proc. IEEE Conference on Computer Vision and Pattern Recognition, June 2007.
doi: https://doi.org/10.1109/cvpr.2007.383245

[9] D. G. Lowe, "Object recognition from local scale-invariant features," Proc. IEEE International Conference on Computer Vision, September 1999.
doi: https://doi.org/10.1109/iccv.1999.790410

[10] J. E. Stone, D. Gohara, and G. Shi, "OpenCL: A parallel programming standard for heterogeneous computing systems," Computing in Science & Engineering, vol. 12, no. 3, pp. 66-72, 2010.
doi: https://doi.org/10.1109/mcse.2010.69

[11] J. L. Schonberger and J. -M. Frahm, "Structure-from-motion revisited," Proc. IEEE Computer Vision and Pattern Recognition, June 2016.
doi: https://doi.org/10.1109/cvpr.2016.445

[12] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, and A. W. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," Proc. IEEE International Symposium on Mixed and Augmented Reality, October 2011.
doi: https://doi.org/10.1109/ismar.2011.6092378

[13] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, Cambridge University Press, 2003.

[14] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer" IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 44, no. 3, pp. 1623-1637, March 2020.
doi: https://doi.org/10.1109/tpami.2020.3019967

[15] R. T. Collins, "A space-sweep approach to true multi-image matching," Proc. IEEE Conference on Computer Vision and Pattern Recognition, June 1996.
doi: https://doi.org/10.1109/cvpr.1996.517097

[16] S. H. Im, H. G. Jeon, S. Lin, and I. S. Kweon, "DPSNet: End-to-end deep plane sweep stereo," Proc. International Conference on Learning Representations, May 2019.

[17] V. Garro, G. Pintore, F. Ganovelli, E. Gobbetti, and R. Scopigno, "Fast metric acquisition with mobile devices," Proc. Vision, Modeling and Visualization, pp. 29 – 36, 2016.

[18] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen, "Learning to recover 3D scene shape from a single image," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2021.
doi: https://doi.org/10.1109/cvpr46437.2021.00027

[19] Y. B. Jeon and I. K. Park, "Deep neural network for handcrafted cost-based multi-view stereo," Proc. International Workshop on Advanced Imaging Technology, January 2021.
doi: https://doi.org/10.1117/12.2591008

[20] Z. Zhang, "A flexible new technique for camera calibration," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330 – 1334, 2000.
doi: https://doi.org/10.1109/34.888718

저 자 소 개

정 태 현

- 2021년 2월 : 인하대학교 정보통신공학과 학사
- 2021년 3월 ~ 현재 : 인하대학교 전기컴퓨터공학과 석사과정
- ORCID : https://orcid.org/0000-0003-0865-5661
- 주관심분야 : 컴퓨터비전 및 그래픽스, deep learning, GPGPU

박 인 규

- 1995년 2월 : 서울대학교 제어계측공학과 학사
- 1995년 2월 : 서울대학교 제어계측공학과 석사
- 2001년 8월 : 서울대학교 전기컴퓨터공학부 박사
- 2001년 9월 ~ 2004년 2월 : 삼성종합기술원 전문연구원
- 2007년 1월 ~ 2008년 2월 : Mitsubishi Electric Research Laboratories 방문연구원
- 2014년 9월 ~ 2015년 8월 : MIT Media Lab 방문부교수
- 2018년 7월 ~ 2019년 6월 : University of California, San Diego (UCSD) 방문학자
- 2004년 3월 ~ 현재 : 인하대학교 정보통신공학과 교수
- 2020년 4월 ~ 현재 : 인하대학교 인공지능융합연구센터 센터장
- ORCID : https://orcid.org/0000-0003-4774-7841
- 주관심분야 : 컴퓨터비전 및 그래픽스, deep learning, GPGPU