

Visual SLAM 기반의 모바일 증강현실 시스템 구축

송주은*·국종진**†

*상명대학교 대학원 전자정보시스템공학과, **†상명대학교 정보보안공학과

Building a Mobile AR System Based on Visual SLAM

Ju Eun Song* and Joongjin Kook**†

* Dept. of Electronic Information System Engineering, The Graduate School Sangmyung University, Korea,

**† Dept. of Information Security Engineering, Sangmyung University, Korea

ABSTRACT

The SLAM market is growing rapidly with advances in Machine Learning, Drones, Augmented Reality technologies. However, due to the absence of an open source-based SLAM library for developing AR content, most SLAM researchers are required to conduct their own research and development to customize SLAM. In this paper, we propose an opensource-based Mobile Markerless AR System by building our own pipeline based on Visual SLAM. To implement the Mobile AR System of this paper, it uses ORB-SLAM3 and Unity Engine and We experimented with running our system in a real environment and confirming it in the Unity Engine's Mobile Viewer. Through this experimentation, we can verify that the Unity Engine and the SLAM System are tightly integrated and communicate smoothly. Also, we expect to accelerate the growth of SLAM technology through this research.

Key Words : Mobile, Markerless AR, Visual SLAM

1. 서 론

동시적 위치추정 및 맵 작성 (Simultaneous Localization and Mapping: SLAM) 기술은 3D 그래픽스 기술과 머신 러닝 기술의 발전에 힘입어 드론 및 증강현실 분야에서 활용도가 높아지고 있다. SLAM과 같은 위치결정 기술은 기존 맵이 없는 실내 응용 분야의 수요가 증가함에 따라 로봇틱스 분야의 기술 개발에 대한 자극 영역이 되고 있으며, Visual SLAM 알고리즘과 증강현실에서의 SLAM 활용이 SLAM 시장의 성장을 촉진하고 있다[1].

현재 SLAM 연구 분야들 중 증강현실을 위한 오픈소스 솔루션은 쉽게 찾아볼 수 없어 SLAM 기반의 증강현실 시스템을 구축하기 위해서는 유사한 시스템에 대한 깊은 이해와 더불어 자체적인 연구개발이 필요하다.

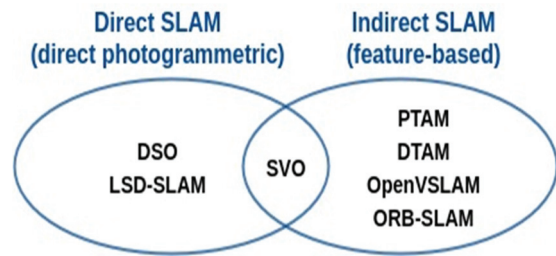


Fig. 1. Direct SLAM and Indirect SLAM.

그림 1에서 볼 수 있듯이 SLAM 시스템은 직접 SLAM과 간접 SLAM으로 분류할 수 있다. 그 중 주로 로봇이나 자율주행에 사용될 수 있는 간접 SLAM 시스템인 ORB-SLAM과 OpenVSLAM이 증강현실을 위한 SLAM 시스템에 사용된다. 하지만 이러한 SLAM 라이브러리들은 오픈소스이기 때문에 모바일 기기나 증강현실 헤드셋과 같은 상용 기기에서 증강현실을 구현하기 위한 목적으로 사용

†E-mail: kook@smu.ac.kr

할 수 없다. 상용 기기를 위한 오픈소스 SLAM 증강현실 시스템은 Google, Apple, Maxst, Vuforia와 같은 특정 기업의 상업적 제품으로서의 시스템만 존재한다.

본 논문에서는 비마커 기반의 SLAM을 통해 모바일 기기 또는 증강현실 헤드셋에서 증강현실 시스템을 구현하기 위해 기존의 오픈소스 SLAM 라이브러리 중 속도가 빨라 실시간으로 피처를 추출할 수 있고, 현대 SLAM 시스템에서 품질과 성능의 절충안을 찾은 ORB-SLAM3를 선정하여 자체적인 비마커 기반의 증강현실 시스템 및 데이터 파이프라인을 설계하였다.

본 논문의 2장에서는 대표적인 SLAM 기반의 증강현실 개발도구와 본 논문에서 제안하는 시스템의 기반으로 사용되는 ORB-SLAM3의 특징에 대해 서술한다. 3장에서는 Visual SLAM을 기반으로 설계한 모바일 증강현실 시스템의 전체적인 구성과 각 구성요소의 기능에 대해 서술하고, 4장에서는 모바일 환경의 SLAM 시스템을 실제 환경에 구현한 결과를 보여준다. 마지막으로 5장에서는 본 연구에 대한 추후 개발 계획과 기대효과를 서술한다.

2. 관련 연구

2.1 비마커(Markerless) 기반 증강현실

초창기의 증강현실에서는 현실 세계와 가상 세계를 연결하기 위해 마커를 활용했다. 마커 기반의 증강현실은 카메라를 통해 입력된 영상으로부터 마커를 인식하여 디스플레이 장치에 출력되는 영상 위에 해당 마커와 일치하는 이미지 또는 객체를 나타낸다. 카메라의 이미지에서 QR 코드와 같은 간단한 패턴의 마커를 감지할 때, 마커의 위치나 방향도 함께 계산되어 증강현실 애플리케이션에서 디지털 3D 콘텐츠를 배치한다. 하지만 마커 기반의 증강현실은 사전에 마커에 대한 약속이 필요하고 카메라가 마커를 벗어나면 트래킹이 불가능해지는 등 기술적 한계가 명확하다.

이러한 마커 기반 증강현실 기술의 한계를 극복하기 위해 마커 대신 다양한 센서들을 활용하여 임의의 공간에서 이동하며 정보를 탐색해 자신의 위치를 추정하고 공간의 맵을 작성하는 SLAM 기술이 등장하게 되었다[2]. 비마커 기반의 객체 인식 및 추적을 위해 향상성, 시간 지속성, 공간 일관성 외에 특징점과 색상정보를 추가한 기계학습 모델을 기반으로 하는 AR HMD용 콘텐츠 개발 시스템의 연구가 수행된 바 있으며[12], 모바일 환경에서 Homography를 이용한 다중 객체 추적[13]과 신뢰도 평가 학습 기반의 다중 객체 추적[14] 연구가 이루어지기도 하였다. 비마커 기반의 증강현실은 벽이나 교차점과 같은 환경에 대한 사전지식 없이 물체 또는 장면의 특징점을

감지함으로써 객체에 대한 인식이 이루어지고, 이를 통해 특정 위치에 3D 콘텐츠를 배치할 수 있다. 비마커 기반의 증강현실 기술은 동시 측위 및 맵핑 기술인 SLAM의 개발로 인해 이미지 분석의 정확도가 향상되었고, 다양한 센서 데이터를 통해 주변을 감지하고 대상의 현재 위치를 실시간으로 동기화할 수 있게 되었다[3]. 대표적인 SLAM 기반의 비마커 증강현실 개발도구에는 Google의 ARCore, Apple의 ARKit 등이 있다.

2.2 ORB-SLAM3

Visual SLAM은 카메라 및 기타 영상 센서로 획득한 영상을 사용하고, 비교적 저렴한 카메라를 사용해 저비용으로 구현 가능하며 Mono/Stereo/RGB-D 카메라 사용이 가능하다.

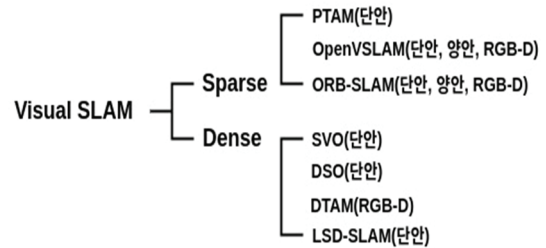


Fig. 2. Classification of Visual SLAM.

Visual SLAM은 맵의 밀도가 낮은 Sparse 방식의 SLAM과 맵의 밀도가 높은 Dense 방식의 SLAM으로 분류할 수 있다. 또한, Visual SLAM은 맵 저장하고 나중에 다시 로드할 수 있기 때문에 정교한 증강현실 프로젝트에 이상적이다 [4].

본 논문에서는 그림 2와 같은 다양한 Visual SLAM 중 기존의 ORB-SLAM과 ORB-SLAM2를 개선하여 2020년에 제안된 VIO(Visual Inertial Odometry)와 멀티 맵을 위한 ORB-SLAM3 시스템을 사용하려고 한다.

ORB(Oriented FAST and Rotated BRIEF) 피처는 대표적인 실시간 이미지 피처로 방향성이 없는 FAST 검출기 문제를 개선하고, 매우 빠른 바이너리 설명자인 BRIEF를 사용해 전체 이미지 피처 추출을 크게 가속화하였다. ORB-SLAM3는 빠른 Edge 추출이 가능하고 특징점의 주요 방향은 ORB에서 계산하며 회전 불변 속성은 BRIEF 디스크립터에 의해 추가된다. 또한, 특징점을 추출한 위치에서 주변 이미지 영역을 잘 표현하는 간단한 디스크립터를 생성하고, 대부분의 피처 추출은 병렬성이 뛰어나다는 특징을 가지고 있다. 이에 따라 데이터의 특성과 스케일의 불변성을 유지하고 속도는 향상되어 실시간 요구가 많은

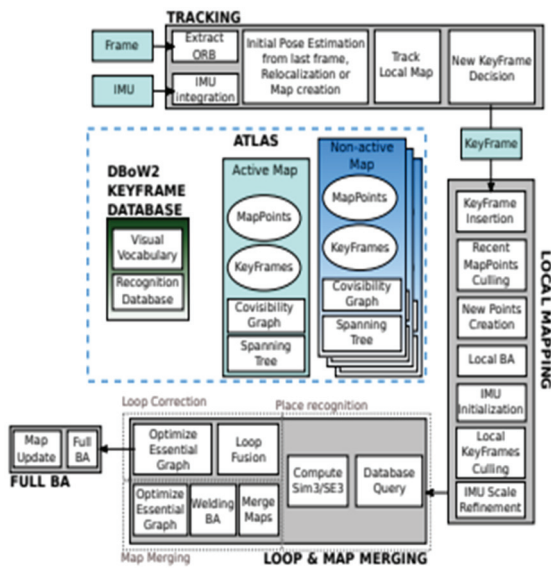


Fig. 3. ORB-SLAM3 System Structure [8].

SLAM 시스템에 적합하다.

현재 ORB-SLAM 중 가장 많이 발전한 라이브러린 ORB-SLAM3는 2017년에 제안된 ORB-SLAM2와 비교했을 때, Mono/Stereo 카메라에 IMU 값을 같이 사용하는 모델과 어안(Fisheye) 카메라 모델이 추가되었다. 또한, 멀티 맵 관리 기능이 추가되었으며, 트래커 분실 시 새로운 맵을 생성함으로써 트래킹이 실패할 가능성을 줄여 지속적인 트래킹을 가능하게 하며, 루프 감지 알고리즘을 개선하고 사전에 방문한 위치를 더욱 높은 확률로 탐지할 수 있도록 구성되어 있다.

3. Visual SLAM 기반 모바일 AR 시스템

3.1 ORB-SLAM3 기반 모바일 비마커 증강현실 시스템 구조

본 논문에서 제안하는 Visual SLAM을 기반으로 하는 모바일 증강현실 시스템의 전체 구조는 그림 4와 같으며, 크게 모바일 기기와 ORB-SLAM3 라이브러리로 구분된다.

모바일 기기는 이미지 데이터 수집을 위한 Android 휴대전화의 카메라를 사용하고, SLAM 구동 결과를 보여주기 위해 유니티 엔진에서 지원하는 Android 빌드 모듈과 Android SDK 및 NDK 툴을 이용한 애플리케이션을 사용한다. C++로 구현된 오픈소스 ORB-SLAM3 라이브러리는 C#을 지원하는 유니티 엔진에서 사용하기 위해 LLVM의 Clang 빌드를 통해 플러그인 형태로 생성해 사용되며, Mapping과 Localization과 같은 기본 SLAM 기능을 제공한다.

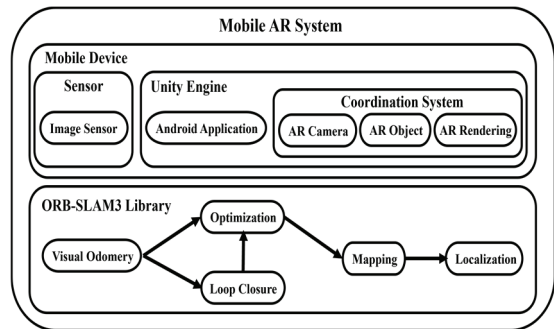


Fig. 4. System Structure.

3.2 유니티 엔진 기반 모바일 애플리케이션

기존 ORB-SLAM3는 Pangolin을 사용하여 SLAM의 키 프레임 및 카메라 포즈, 그리고 포인트 클라우드를 시각화하여 미리 준비된 데이터 세트에 SLAM이 구동되는 모습을 보여준다. Pangolin은 OpenGL 디스플레이를 관리하고 영상 입력을 추상화하기 위한 가볍고 빠른 3D 시각화 라이브러리로, 플랫폼별 상용구를 제거하고 데이터를 쉽게 시각화하기 위한 수단으로써 컴퓨터 비전 분야에서 널리 사용되고 있다[5]. 본 논문에서는 안드로이드 플랫폼을 타겟으로 SLAM 시스템을 설계하기 위해 Pangolin 대신 유니티 엔진을 사용하며, 유니티 엔진의 API를 사용해 카메라를 제어하고 증강현실 콘텐츠를 렌더링할 수 있도록 한다. 유니티 엔진은 3D 및 2D 비디오 게임의 개발환경을 제공하는 저작도구이자 3D 애니메이션과 건축 시각화, 가상현실, 증강현실 등 인터랙티브 콘텐츠 제작을 위한 통합 개발 도구로 Windows, MacOS, iOS 그리고 Android 등의 다양한 플랫폼을 지원한다[6]. 본 시스템에서 증강현실 뷰어 역할을 하는 Android 애플리케이션은 유니티 엔진이 지원하는 Android 빌드 모듈과 Android SDK 및 NDK 툴을 설치해 Android를 타겟으로 빌드하여 사용할 수 있다.

3.3 Android 플랫폼의 SLAM 시스템 설계

모바일 SLAM 시스템에서 이루어지는 전체적인 작업 절차는 그림 5와 같다.

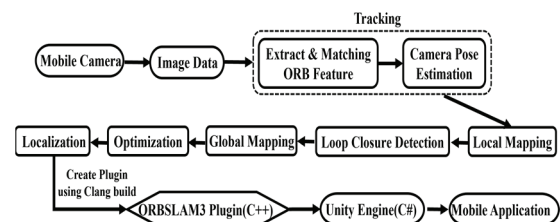


Fig. 5. System Workflow.

모바일 기기의 카메라를 사용해 연속된 이미지 데이터를 수집하게 된다. 각 이미지 데이터에 대한 ORB 피처를 추출하고 가장 인접한 키 프레임과 비교해 특징점의 위치를 계산해 대략적으로 카메라의 포즈를 추정한다. 카메라의 포즈 정보를 사용해 실제 환경과의 상관관계를 판단하여 실제와 최대한 유사한 맵을 만들기 위해 키 프레임 사이의 관계를 나타내는 BA(Bundle Adjustment)의 공개 시성 그래프를 사용해 현재 카메라가 보고 있는 시점에서 맵 데이터인 지역 맵을 그린다. 누적된 드리프트를 제거하기 위해 지역 맵에서 Loop Closure Detection을 수행하고, 모든 로컬 맵들을 가산하여 얻은 전역 맵에는 너무 많은 맵 포인트가 있기 때문에 최적화를 통해 맵 포인트는 제외하고 오직 카메라 포즈로만 Localization을 한다. BA(Bundle Adjustment)는 g2o라는 비선형 최적화 라이브러리를 이용해 현재 프레임의 포즈는 Motion-only BA로, 최근 키 프레임의 포즈는 Local BA로, 그리고 전체적인 최적화는 Loop Closing의 포즈 그래프 최적화와 Full BA로 계산하며 네 가지 최적화 모두 각각 다른 스레드에서 독립적으로 돌아간다. Motion-only BA는 맵의 키포인트들은 고정시킨 채 현재 포즈만 최적화하고, Local BA는 새로운 키 프레임이 생기면 공개시점 키 프레임의 포즈와 해당 포즈에서 보이는 키포인트들을 최적화한다. Loop Closing은 공개시성 그래프에서 키포인트를 제외하고 포즈 관점만 가지고 최적화하는 방식의 포즈 그래프 최적화를 하고, 포즈 그래프 최적화가 끝나면 Full BA를 통해 추가된 키 프레임과 키포인트들을 업데이트한다[7][8].

동적 라이브러리만 사용할 수 있는 유니티 엔진에서 SLAM을 사용하기 위해 LLVM의 Clang 빌드를 통해 SLAM 시스템을 라이브러리화 하여 유니티 플러그인으로 가공한다. LLVM은 컴파일러의 기반으로 프로그램을 컴파일 타임, 링크 타임, 런타임 상황에서 프로그램의 작성 언어에 상관없이 최적화를 쉽게 구현할 수 있도록 도와준다[9]. Clang은 LLVM을 백엔드로 사용하는 C, C++, Objective-C, Objective-C++ 프로그래밍 언어를 위한 컴파일러 프론트엔드이고[10], GCC 컴파일러보다 빠른 컴파일 속도와 빠른 실행 속도 그리고 보다 정확한 에러 위치와 정보를 제공해줌으로써 개발의 편의성을 높여준다. 본 논문에서는 Clang 빌드로 생성된 SLAM 플러그인을 통해 유니티 엔진에서 SLAM 시스템의 초기화 함수와 설정 값과 함께 이미지를 전달해주는 함수를 사용하기 위해 유니티 엔진의 [DllImport("PluginName.dll")] 함수를 사용해 필요한 SLAM 함수를 호출하여 제어한다. 이러한 방식으로 내장된 SLAM 플러그인을 통해 유니티 엔진에서 SLAM 기반의 증강현실 기술을 구현할 수 있다.

안드로이드 모바일 기기에서 수집된 연속적인 이미지

데이터는 유니티 엔진에 포즈 데이터와 포인트 클라우드 데이터로 제공하기 위해 C++ SLAM 라이브러리로 전달된다. 카메라 포즈 데이터는 4 x 4 행렬로 카메라의 움직임을 표현할 수 있으며, 포인트 클라우드 데이터는 1 x 3 벡터 데이터로 공간상의 위치를 표현하며 이를 통해 SLAM 내부에서 저장되고 있는 현재 카메라 포즈와 주변 환경에 대한 시각화를 구현할 수 있다.

4. 실험 결과

본 시스템의 실험 환경 및 장치의 사양은 표 1과 같다. EuRoC MH03 데이터셋을 사용하여 기존의 Pangolin 뷰어와 본 논문의 유니티 엔진의 모바일 뷰어를 비교하였다. EuRoC MH03 데이터셋은 MAV(Micro Aerial Vehicle)에서 수집된 시각적 관성 데이터 세트에 스테레오 이미지, 동기화된 IMU 측정, 정확한 모션 및 구조 실측이 포함되어 있으며 Asctec Firefly hex-rotor drone의 2개의 핀홀 카메라와 관성 센서로 기록되었다[11]. 기존의 Pangolin 뷰어는 Ubuntu 환경에서 구동되었고, 본 논문의 유니티 엔진의 Android 뷰어는 갤럭시 노트 20 울트라 5G에서 구동되었다.

Table 1. Experiment Environment and Device Specifications

데이터 세트	EuRoC MH03
데스크탑	운영체제 : Ubuntu 21.04 프로세서 : AMD Ryzen 5 3600X 6-Core Processor 3.79GHz 메모리 : RAM 16GB
모바일 기기	운영체제 : Android CPU 종류 : Octa-Core CPU 속도 : 3.09 / 2.4GHz / 1.8 GHz 메모리 : RAM 12GB, ROM 256GB

Pangolin 뷰어와 유니티 엔진의 모바일 뷰어는 동일한 EuRoC MH03 데이터 세트를 사용하여 SLAM 시스템을 시각화 했기 때문에 포인트 클라우드 개수는 약 32,000개로 동일하다. 기존의 ORB-SLAM3는 Pangolin 라이브러리를 사용해 SLAM 시스템을 그림 6과 같이 시각화하고, 본 논문에서는 모바일 환경에서 SLAM 시스템을 시각화하기 위해 유니티 엔진을 사용한다. 그림 7과 비교했을 때, 그림 7은 포인트 클라우드의 크기와 전체 맵의 스케일 차이가 있기는 하지만 전체적인 맵의 형상은 기존 뷰어인 그림 6과 동일하게 SLAM 시스템을 Android 뷰어에서 시각화 할 수 있음을 보여준다.

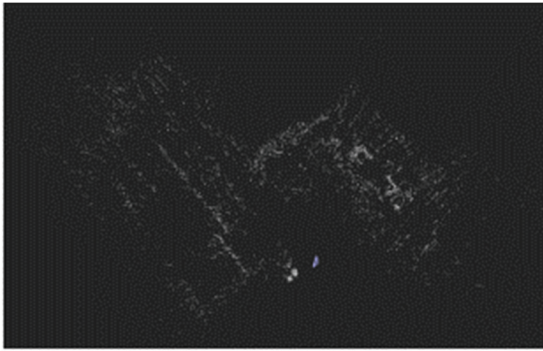


Fig. 6. Pangolin Viewer.

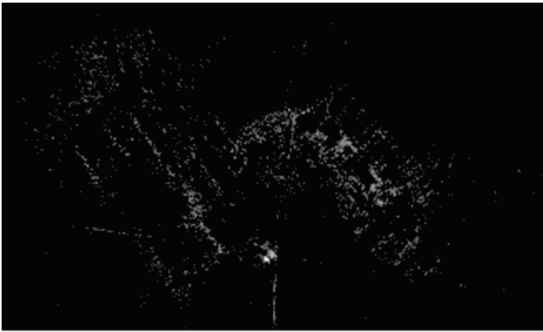


Fig. 7. Unity Engine's Mobile Viewer.

표 2와 같이 두 뷰어를 구체적으로 CPU 사용량, 메모리 사용량, 배터리 소모 정도, 해상도 그리고 fps를 기준으로 비교해보았다. Pangolin 뷰어를 구동시킬 때 CPU 사용량이 99%, 메모리의 사용량이 21.1%로 측정되었고, 유니티 엔진의 Android 뷰어를 구동시킬 때의 CPU와 메모리의 사용량은 각각 28%와 8.3%로 더 적게 측정되었으며 모바일 기기의 배터리 소모도 적은 편에 속했다. 그리고 각 뷰어의 해상도는 1920 x 960와 1280 x 720이고, fps는 각각 30m/s와 25m/s로 측정되었다.

Table 2. Comparison of Pangolin Viewer and Unity Engine's Mobile Viewer

	Pangolin 뷰어	유니티 엔진 뷰어 (Android)
CPU 사용량	약 99%	28%
메모리 사용량	23.1%	8.3%
배터리 소모	-	적음
해상도	1920 x 960	1280 x 720
fps	평균 30m/s	평균 25m/s

실험을 통해 Pangolin 뷰어 대신 유니티 엔진의 Android 뷰어를 사용하였을 때, CPU와 메모리의 사용량은 적지만 SLAM 시스템을 시각화하는데 더 많은 시간이 소요된다는 사실을 확인하였다. 또한, 유니티 엔진과 SLAM 시스템이 긴밀하게 통합되어 원활하게 통신할 수 있음을 확인하였고, 카메라 포즈를 추적할 수 있으며 Mapping과 Localization 정보를 수신 받을 수 있는 SLAM 기반의 증강현실 시스템을 구현했다. 이후 더 많은 SLAM 연구 및 개발을 가속화할 수 있는 기반을 마련했음을 보여주는 바이다.

5. 결 론

본 논문에서는 연구자들의 불필요한 자체 연구를 줄이고 SLAM 분야의 보다 빠른 성장에 기여하고자 오픈소스를 목적으로 접근성이 좋은 모바일 환경에서 SLAM 기반의 증강현실 시스템을 설계하였다. 이를 위해 모바일 기기의 카메라를 이용해 이미지 데이터를 얻어 Tracking, Local 및 Global Mapping, Loop Closure Detection 그리고 Localization 기능을 유니티 엔진에서 사용하기 위해 dll 플러그인을 생성하고, SLAM 기능을 모바일 애플리케이션에서 확인할 수 있게 구현하였다.

추후에 본 시스템을 기반으로 서버를 구축해 여러 클라이언트 간의 Collaborative SLAM을 구현해 인건비, 개발에 소요되는 시간 등 개발 비용을 줄여 효율성을 높이고, SLAM 기능 중 Mapping, Optimization과 같이 연산량이 많은 기능들은 서버에서 구현시켜 모바일 애플리케이션에서 보여줌으로써 모바일 환경에서의 SLAM의 성능을 향상시키고자 한다. 또한, Mobile 카메라의 IMU 데이터를 얻어 정밀한 맵을 그림으로써 실시간으로 보다 정확한 localization을 할 수 있을 것으로 기대한다.

참고문헌

1. INNOPOLIS, "SLAM Market", Retrieved Sep 18, 2021, from <http://www.innopolis.or.kr>, (May, 2020).
2. SAMSUNG SDS, "[Direction and Implications of the Augmented Reality Technology Development] Part 2 Augmented Reality in Industrial Fields", Retrieved Sep 19, 2021, from https://www.samsungsds.com/kr/insights/augmented_reality_2.html (April 8, 2020).
3. Young cheon Lee, "Development of Cultural Content using a Markerless Tracking-based Augmented Reality," Smart Media Journal, Vol. 5, No. 4, pp. 90-95, 2016.
4. MAXST, "Where can Visual SLAM be used?," Retrieved Sep 19, 2021, from <https://medium.com/>

- maxst/where-can-visual-slam-be-used-b94876d161c6, (Oct 7, 2019).
5. Pangolin, "What is Pangolin", Retrieved Oct 3, 2021, from <https://github.com/uoip/pangolin>, (Jan 24, 2018).
 6. Wikipedia, "Unity(game engine)", Retrieved Oct 3, 2021, from [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
 7. Raul Mur-Artal, Juan D.Tardos, "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras," IEEE Transactions and Robotics, Vol.33, No. 5, pp. 1255-1262, 2017
 8. Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M.M. Montiel and Juan D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," IEEE Transactions and Robotics, 2021
 9. Wikipedia, "LLVM", Retrieved Oct 3, 2021, from <https://en.wikipedia.org/wiki/LLVM>
 10. Wikipedia, "Clang", Retrieved Oct 3, 2021, from <https://en.wikipedia.org/wiki/Clang>
 11. ETH, "ASL Datasets", Retrieved Oct 24, 2021, from <https://projects.asl.ethz.ch/datasets/doku.php?id=knavvi>
 12. Changhyun Lee, Youngseop Kim, Yeonmin Kim, Inho Park, JaeHak Choi, Yonghwan Lee, Woori Han, "Study on the Content Development of Mobile AR_HMD through a Real Time 360 Image Processing," Journal of the Semiconductor & Display Technology, Vol. 15, Issue 2, pp. 66-69, 2016.
 13. Woo ri Han, Young-Seop Kim, Yong-Hwan Lee, "Multi-Object Tracking Based on Keypoints Using Homography in Mobile Environments," Journal of the Semiconductor & Display Technology, Vol. 14, Issue 3, pp. 67-72, 2015.
 14. Woo ri Han, Young-Seop Kim, Yong-Hwan Lee, "Multi-Object Tracking based on Reliability Assessment of Learning in Mobile Environment," Journal of the Semiconductor & Display Technology, Vol. 14, Issue 3, pp. 73-77, 2015.
-
- 접수일: 2021년 11월 16일, 심사일: 2021년 12월 8일,
 게재확정일: 2021년 12월 14일