

깊이 지각 카메라를 활용한 3D 맵 생성

2016010873 박정욱

목차

- 깊이 인식
 - 기술의 종류
 - Intel RealSense SDK 2.0
- Windows 환경에서의 OctoMap 설치

깊이 인식

- 단안 카메라를 이용한 Visual SLAM은 다음과 같은 약점 존재
 - Scale Ambiguity
 - Initial Map Necessity
- ORB-SLAM2는 기존 버전 + **깊이 정보**를 활용하여 성능 향상
 - Stereo, RGB-D 지원

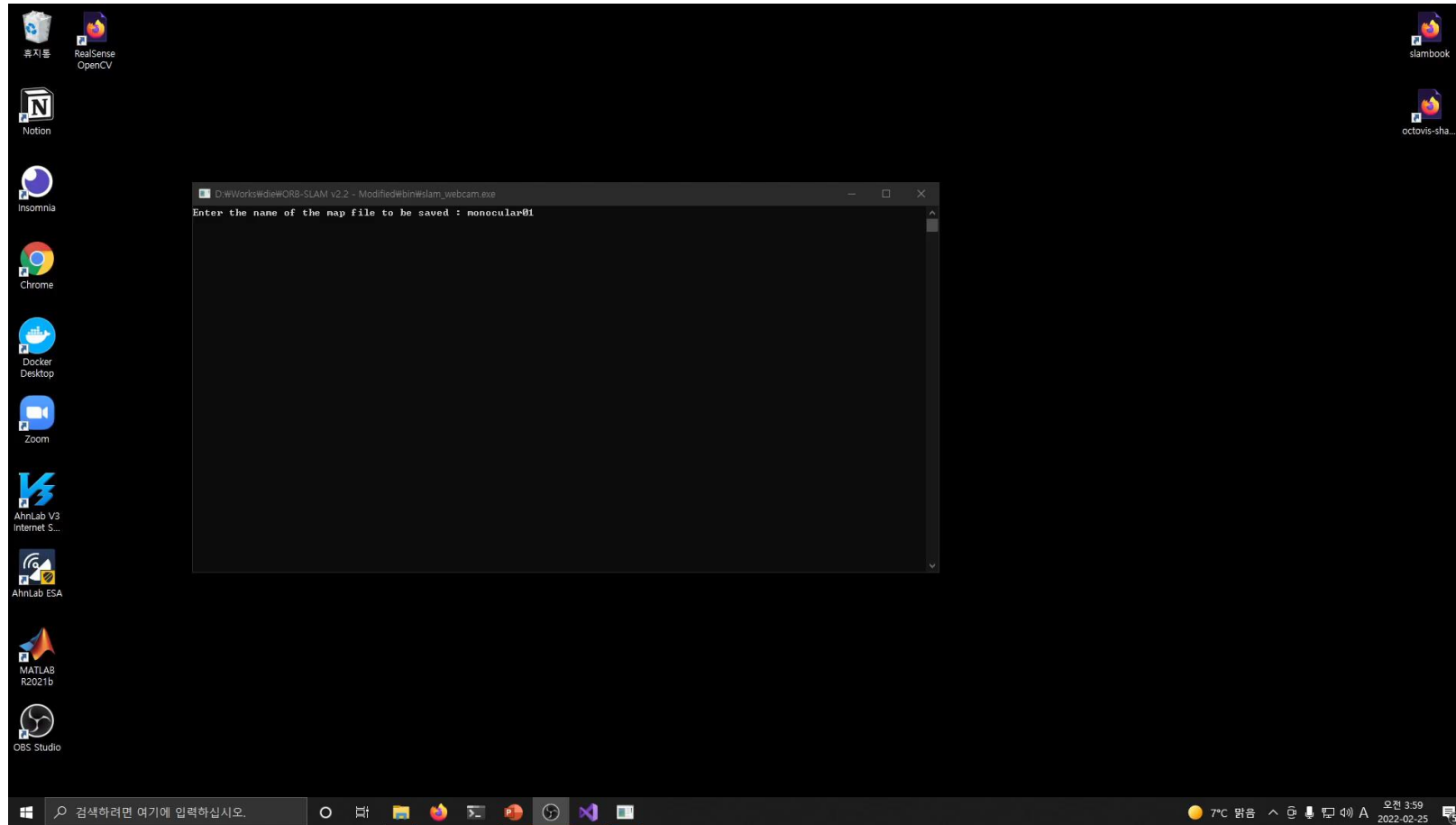
Scale Ambiguity

- ORB-SLAM은 단안 카메라를 이용하여 기기 궤적 추정
 - 이때 사용되는 기술을 통틀어 Visual Odometry로 통칭
- 단안 카메라는 기준이 되는 스케일을 알아낼 수 없음
 - 이는 시간에 따라 누적되는 오차(Scale drift)를 유발
- 때문에 **Scale Factor**를 알기 위해서는 추가 정보 필요
 - Scale Factor: 이동 변환의 실제 길이

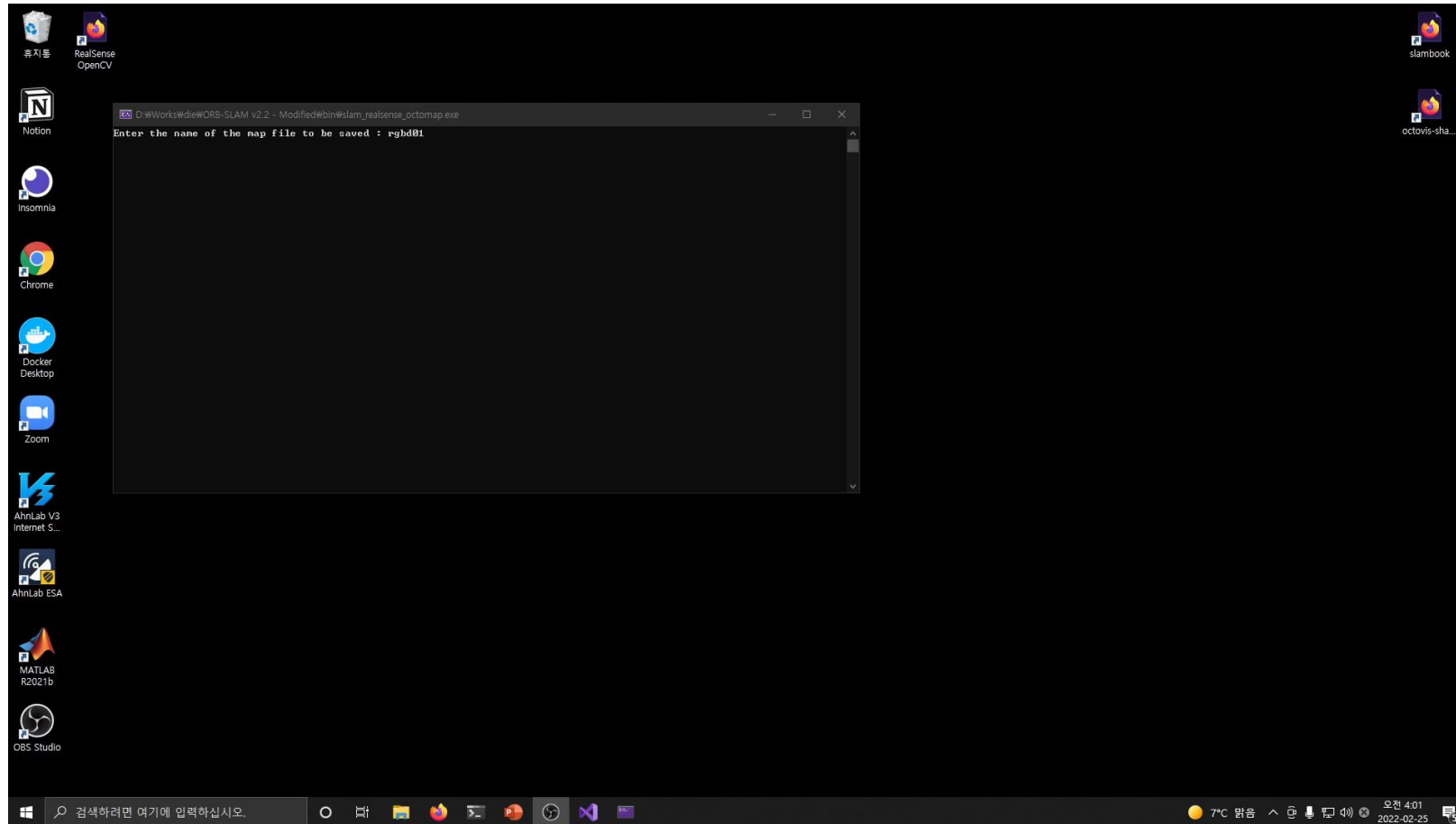
Initial Map Necessity

- Scale Ambiguity에서 파생되는 문제점
- 초기 기동 시 자신의 궤적을 결정하기 위한 정보가 없음
- ORB-SLAM은 자동 초기화 방식을 고안하여 문제를 해결함
 - 깊이 정보가 있다면 **불필요**한 작업

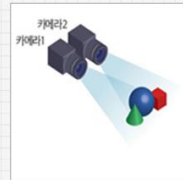
Monocular vs. Depth



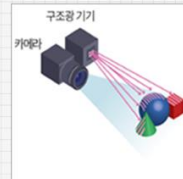
Monocular vs. Depth



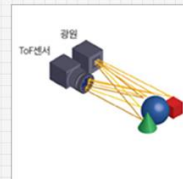
깊이 인식 기술의 종류



1. 스테레오 방식

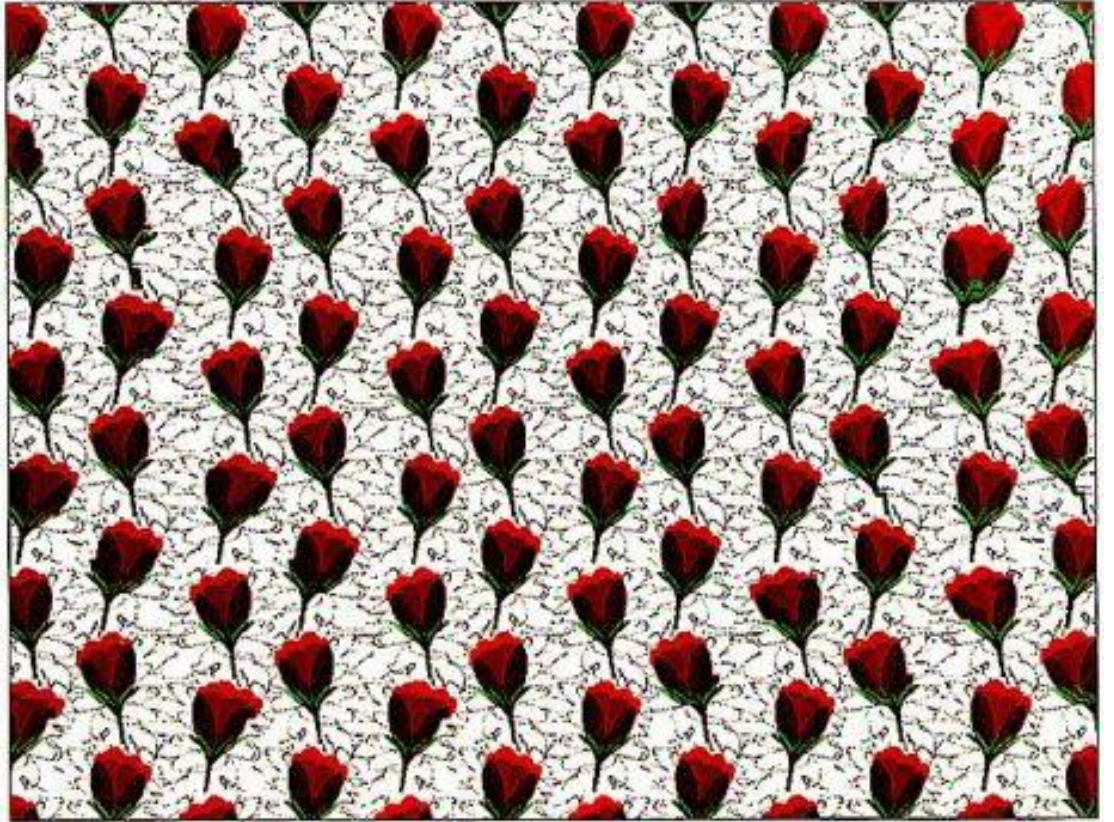
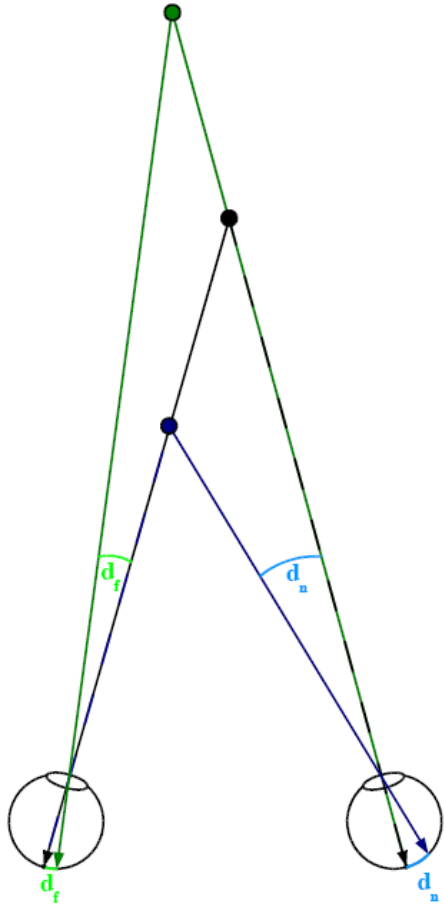


2. 구조광 방식

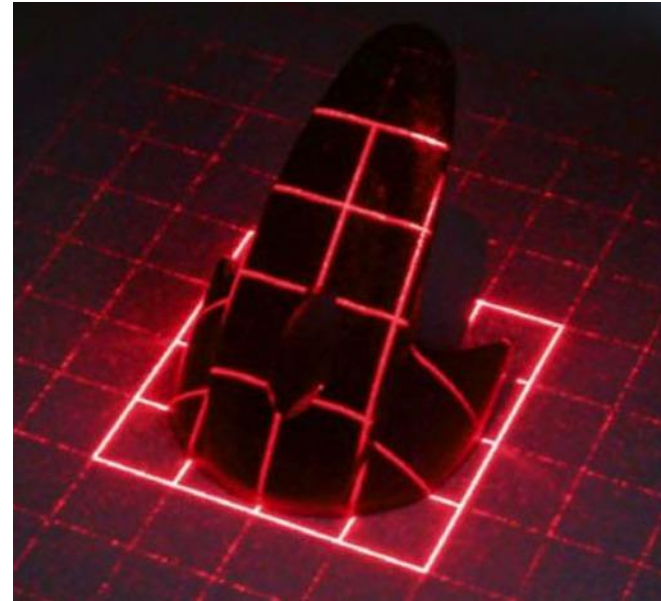
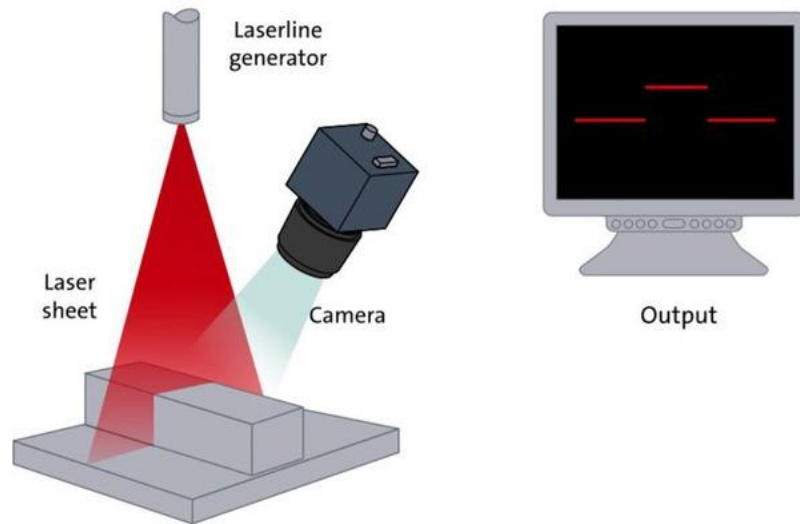


3. ToF 방식

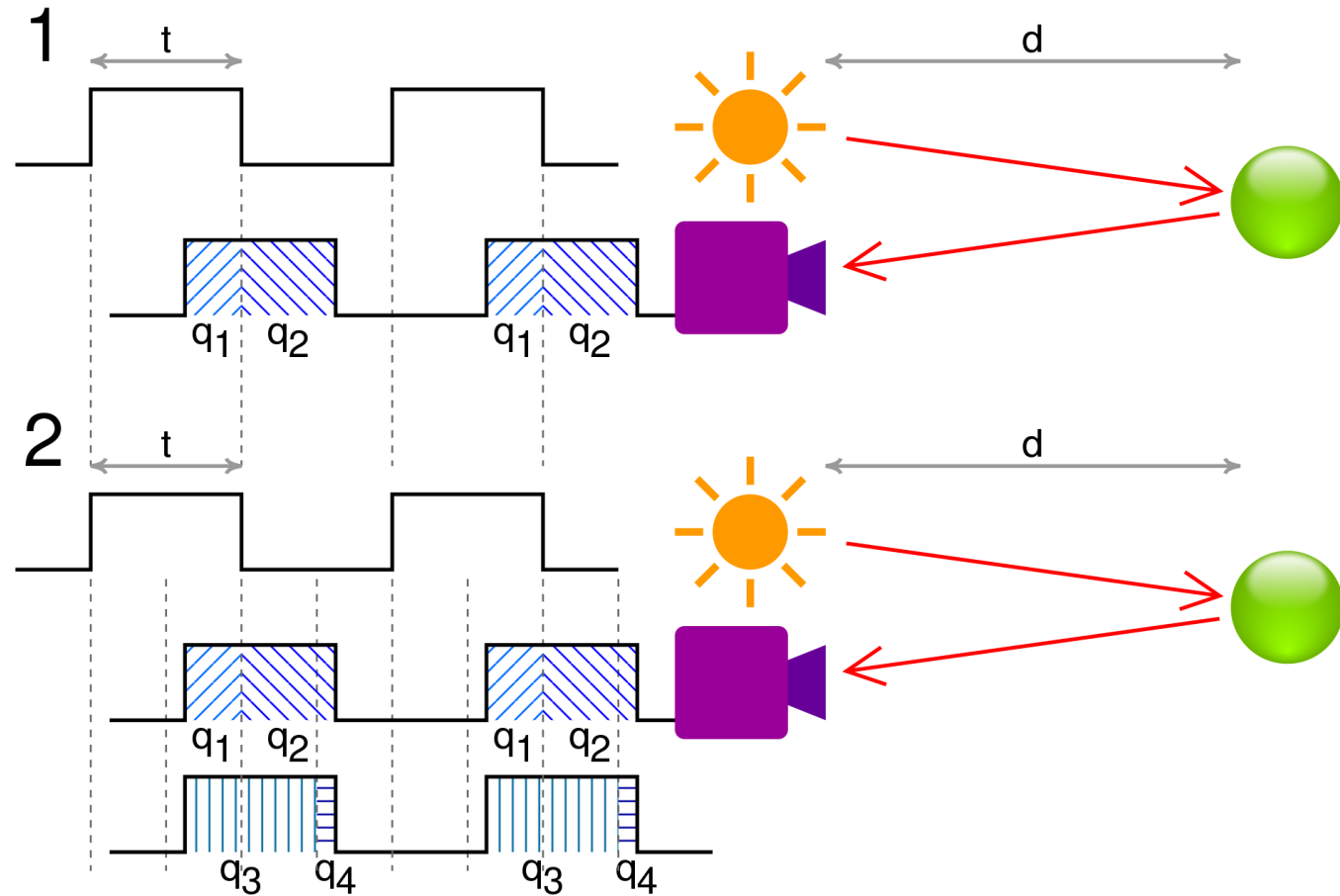
Stereo vision



Structured light

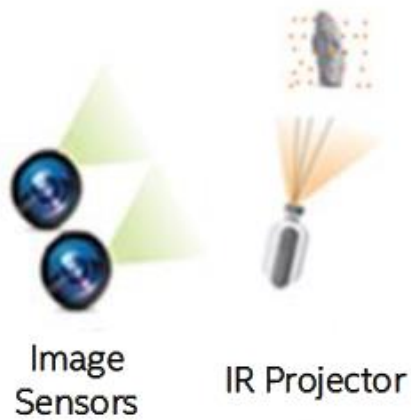


Time of Flight



Intel RealSense Depth Camera D435i

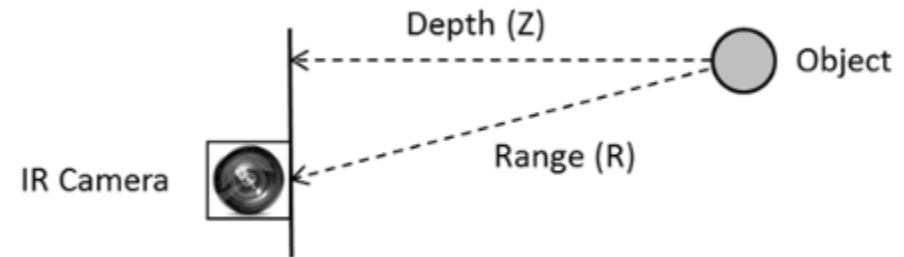
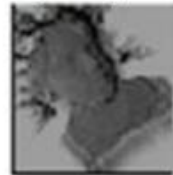
1) Capture



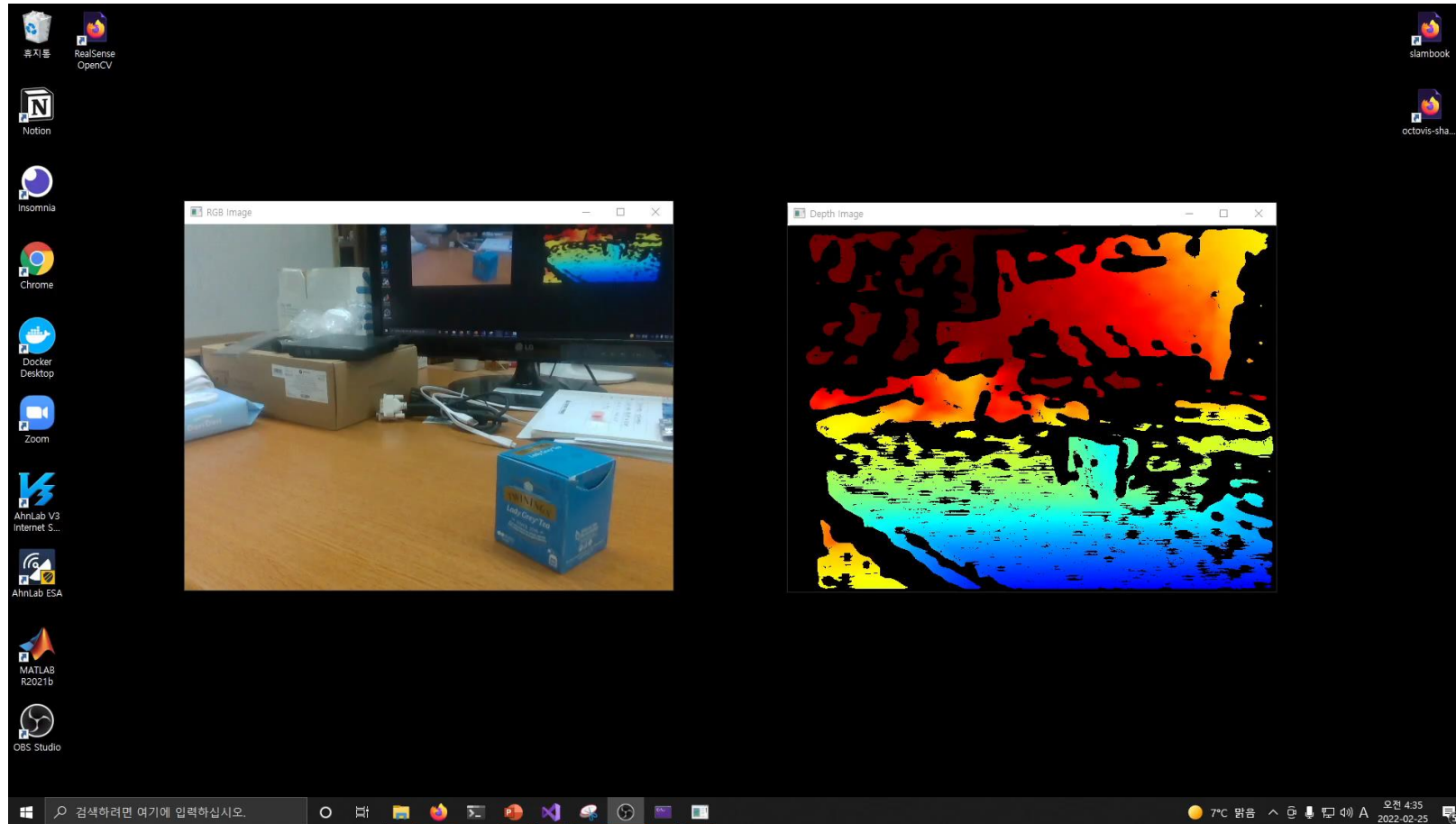
2) Search



3) Depth



Intel RealSense SDK 2.0



Intel RealSense SDK 2.0

```
// rs2 pipeline
pipeline pipe;
rs2::config cfg;
cfg.enable_stream(RS2_STREAM_COLOR, 640, 480);
cfg.enable_stream(RS2_STREAM_DEPTH, 640, 480);

auto profile = pipe.start(cfg);
```

```
frameset data = pipe.wait_for_frames();
frame color_frame = data.get_color_frame();
frame depth = data.get_depth_frame();

const int w = color_frame.as<video_frame>().get_width();
const int h = color_frame.as<video_frame>().get_height();

cv::Mat imgRGB(cv::Size(w, h), CV_8UC3, (void*)color_frame.get_data(), cv::Mat::AUTO_STEP);
cv::Mat imgDepth(cv::Size(w, h), CV_16UC1, (void*)depth.get_data(), cv::Mat::AUTO_STEP);

double timestamp = color_frame.get_timestamp();

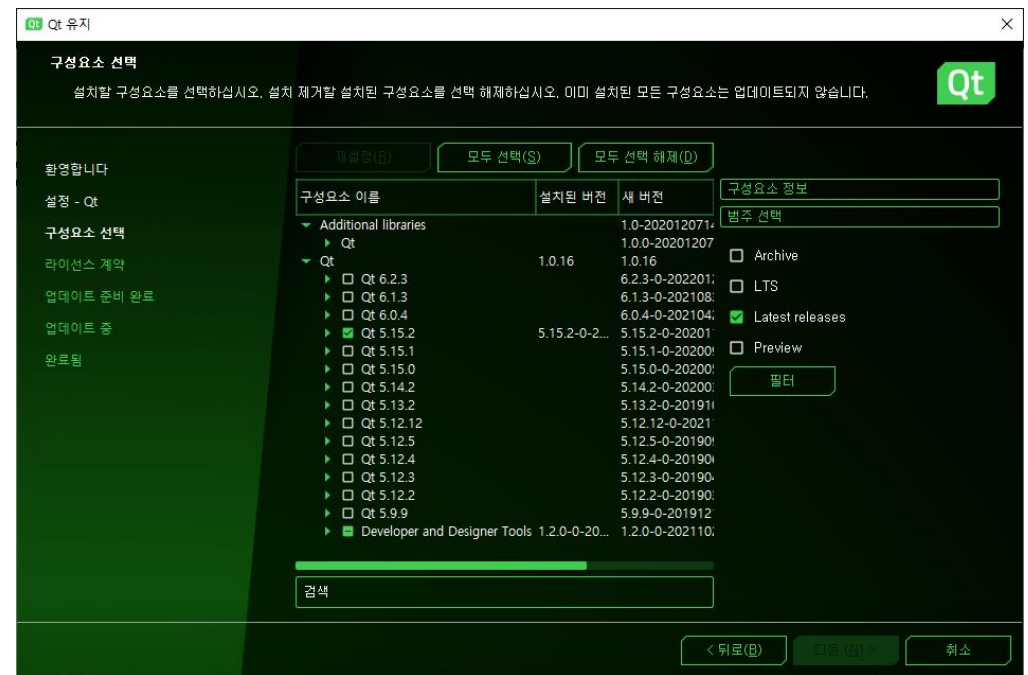
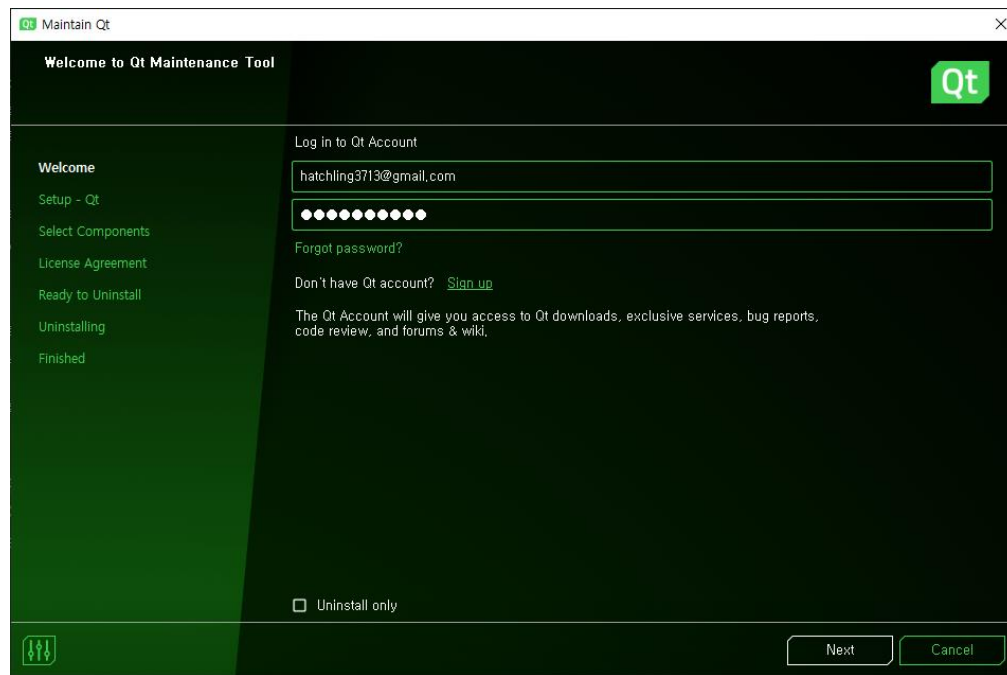
vTimestamps.push_back(timestamp);

SLAM.TrackRGBD(imgRGB, imgDepth, timestamp);
```

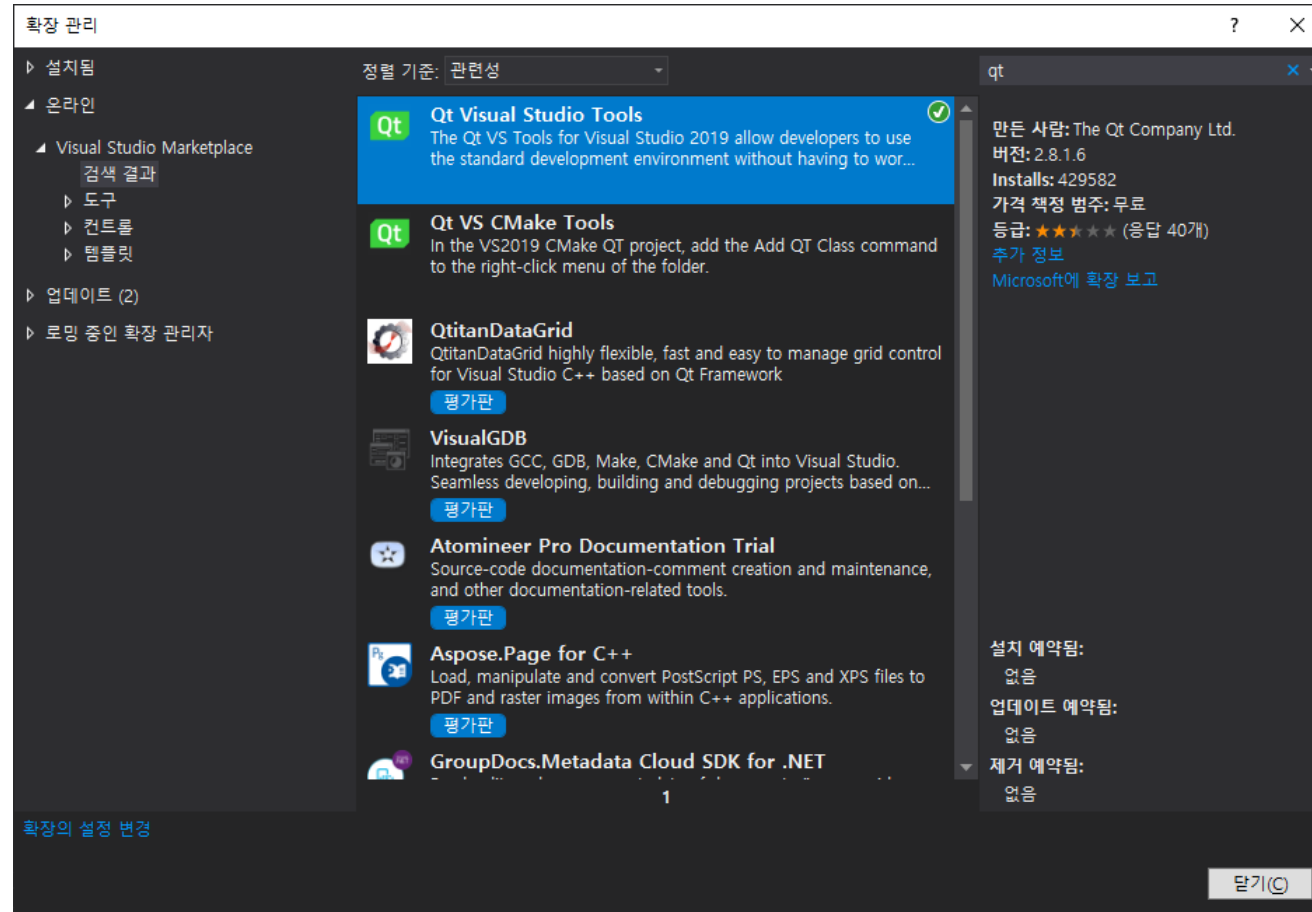
Windows 환경에서의 OctoMap 설치

- QT5 Framework 설치
- libQGLViewer 빌드
- OctoMap/OctoVis 빌드

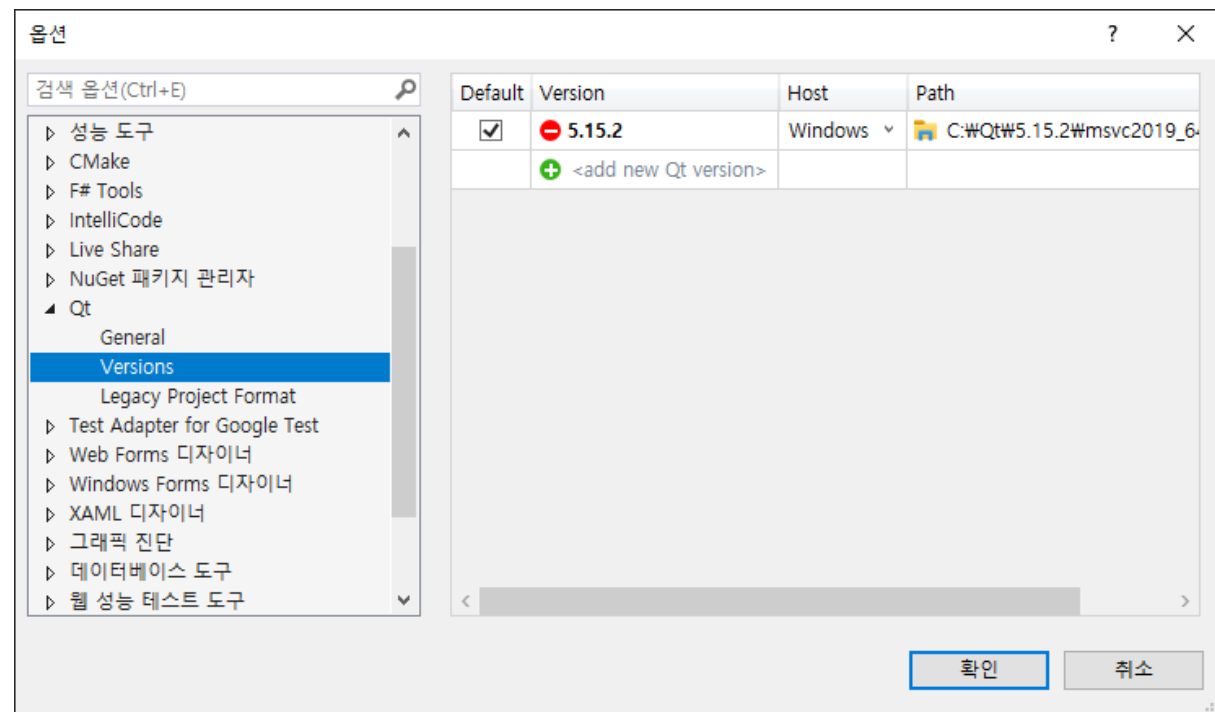
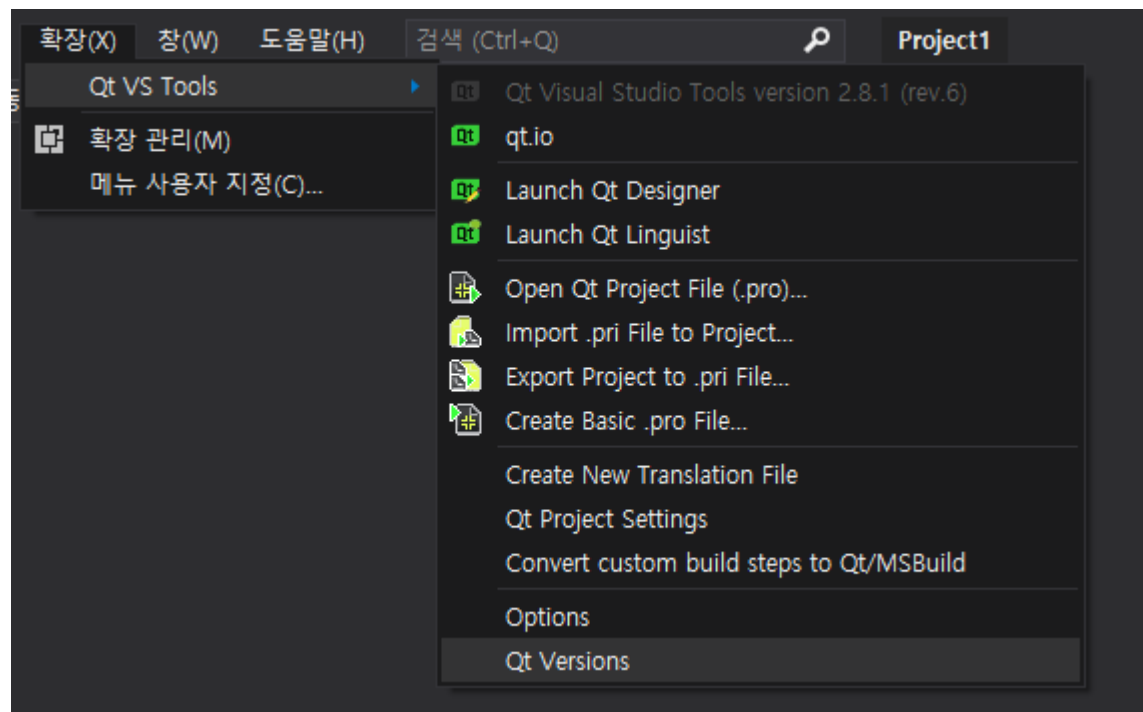
QT5 Framework



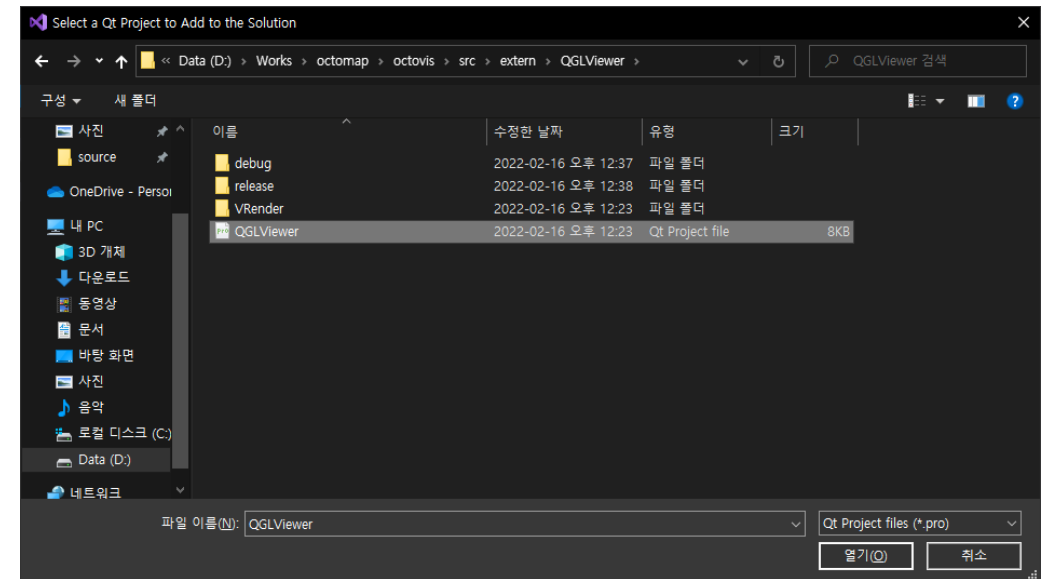
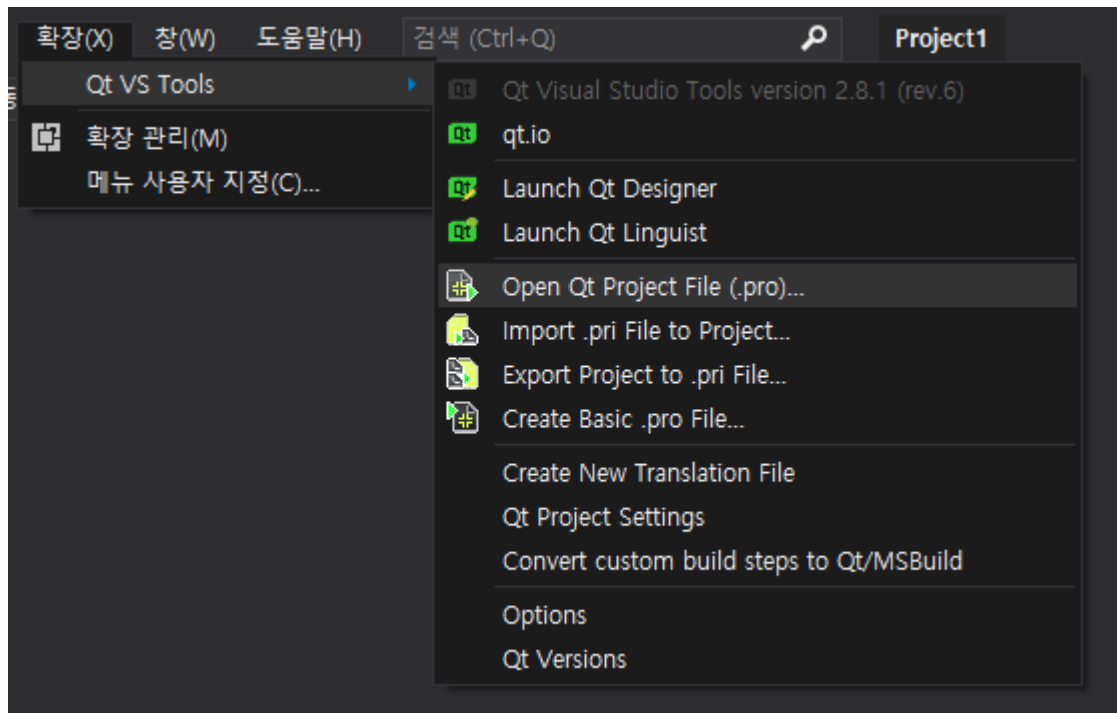
libQGLViewer



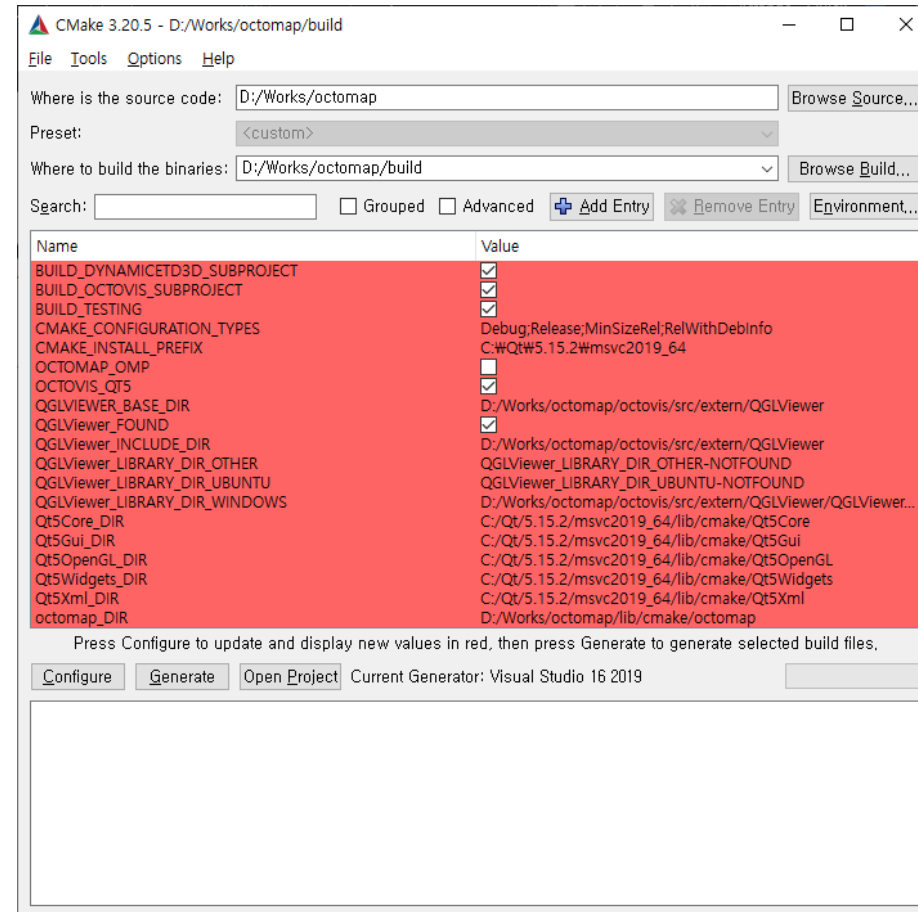
libQGLViewer



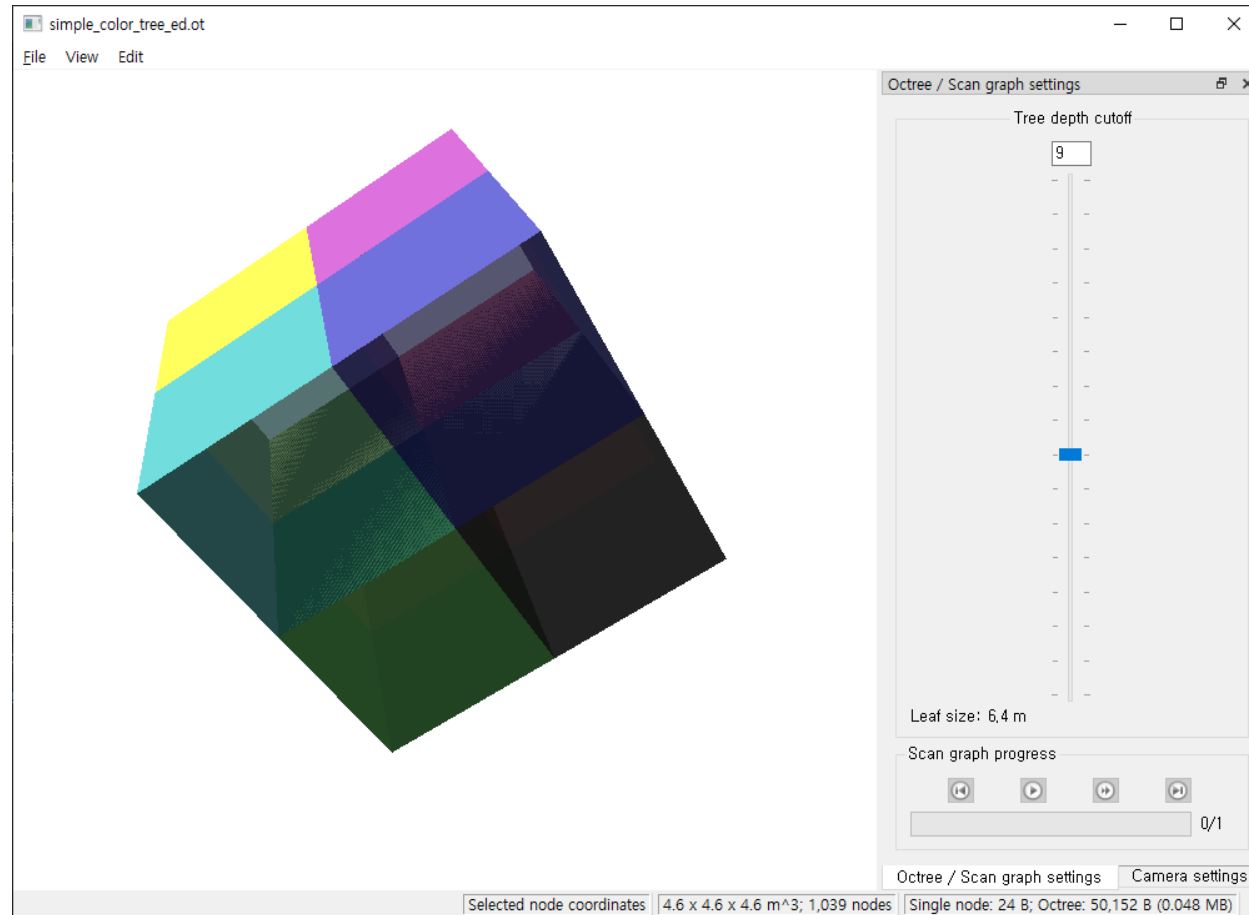
libQGLViewer



OctoMap/OctoVis



OctoMap/OctoVis



남은 작업

1. 카메라 내부 파라미터 접근 - 완료

```
// Camera intrinsics
double ppx = intrinsics.ppx; // 카메라 주점(principal point) x좌표
double ppy = intrinsics.ppy; // 카메라 주점(principal point) y좌표
double fx = intrinsics.fx; // 초점거리(focal length)의 가로 방향 배수
double fy = intrinsics.fy; // 초점거리(focal length)의 세로 방향 배수
double depth_scale = sensor.get_depth_scale(); // 깊이 스케일(scale factor)
```

2. 파라미터와 카메라 Pose를 이용해 점구름(Point Cloud) 계산

3. 계산한 점구름으로 OcTree 생성 및 시각화