

Unity 기반 손동작 인식

2023-07-05

연구현황

목차

- 개요
- 연구, 개발 현황
- 향후 계획

개요

연구 배경, 기존 연구, 연구 계획 및 일정

배경

- 정밀 인터랙션을 위한 손동작 인식 기술 연구
 - 위탁과제, 한국전자통신연구원
- 기술 구현 및 통합
 - Python에서 SolvePnP 구현
 - Unity에서 MediaPipe 손 인식 구현
 - Unity 환경으로 통합

기존 연구 - solvePnP

cv::solvePnP(

InputArray

objectPoints, (현실)

imagePoints, (마커)

cameraMatrix, (내부파라미터)

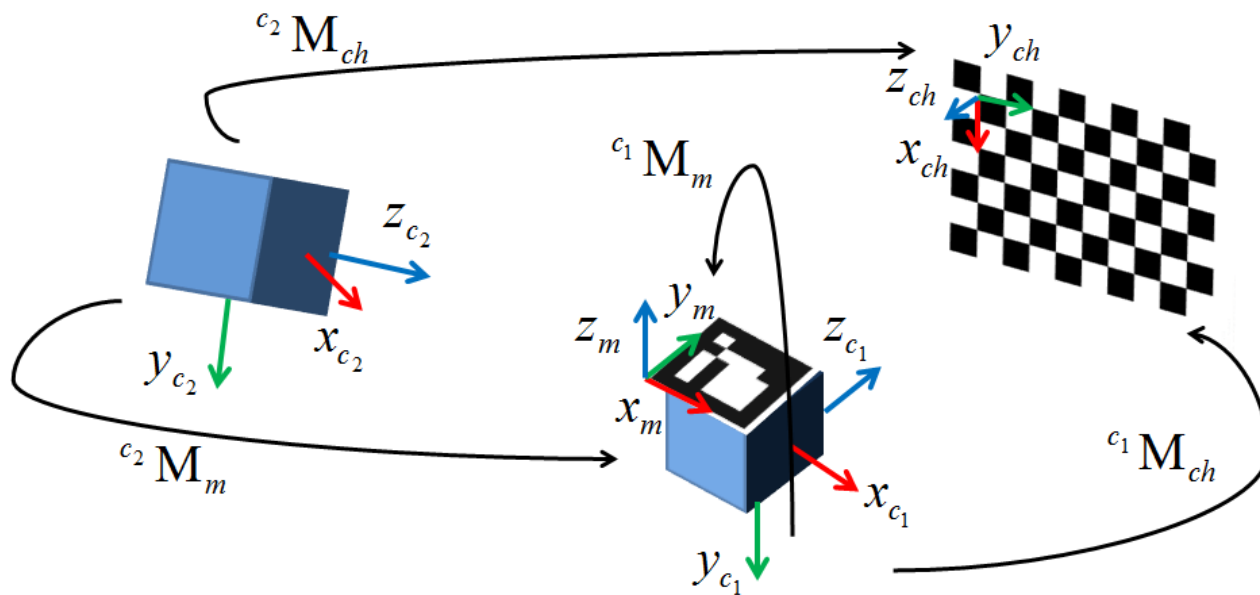
distCoeffs, (왜곡계수)

OutputArray

rvec, (rotation vector)

tvec, (translation vector)

)

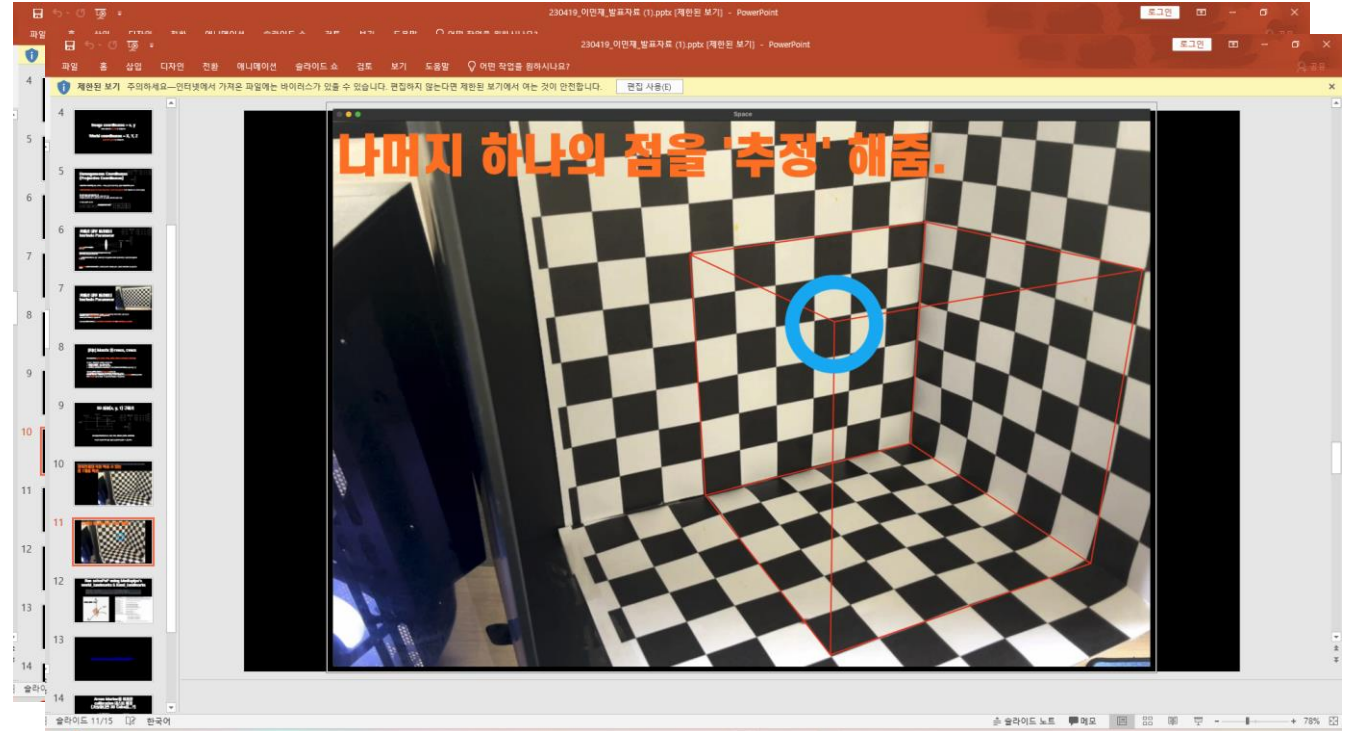


solvePnP 활용

solvePnP

정육면체의 한 점 추정

출처: 이민재 랩미팅 발표자료 (230419)



기존 연구 -MediaPipe

연구, 개발 현황

연구 계획 및 일정

1. Unity 2020.1.0f1 버전 다운그레이드
2. OpenCV(C++)을 Unity(C#)에서 사용할 수 있는 환경 구축
3. solvePnP 체스보드 -> Aruco로 변경
4. Aruco 마커 위에 물체 올리기
5. Python에서 구현한 SolvePnP를 이용한 손 3D 인식을
Unity에서 구현

연구 계획 및 일정

1. Unity 2020.1.0f1 버전 다운그레이드 (완료)
2. OpenCV(C++)을 Unity(C#)에서 사용할 수 있는 환경 구축 (완료)
3. solvePnP 체스보드 -> Aruco로 변경 (진행중)
4. Aruco 마커 위에 물체 올리기
5. Python에서 구현한 SolvePnP를 이용한 손 3D 인식을
Unity에서 구현

개발환경

- Windows 11, 64비트
- OpenCV 4.8.0
- CMake 3.26.4
- Visual Studio 16 2019
- Unity 2020.1.0f1

Unity 2020.0.1f1 버전 다운그레이드

협력 업체의 개발환경으로 변경
2021.3.26f1 → 2020.0.1f1

Canvas

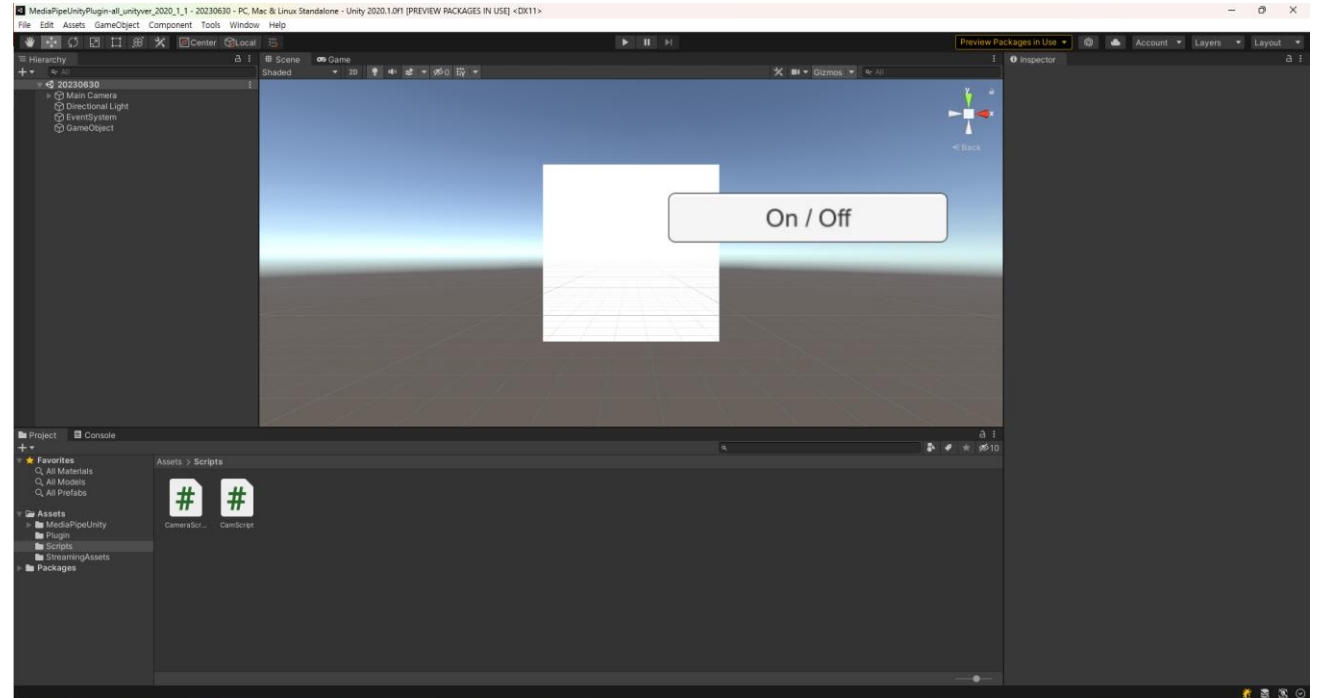
RawImage
StartButton

Scripts

CamScript.cs

Plugin

OpenCV2DLL.dll
opencv_...dll

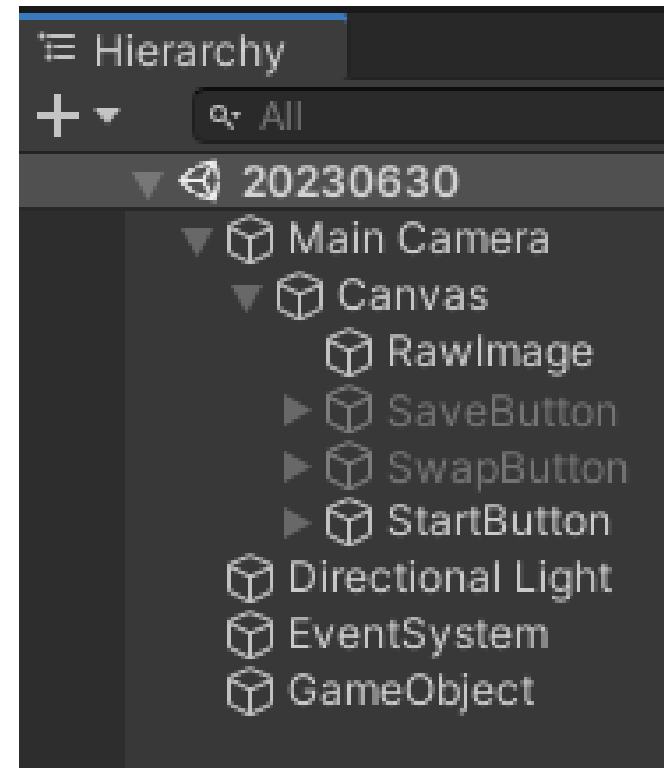


Hierarchy

Canvas

RawImage

StartButton

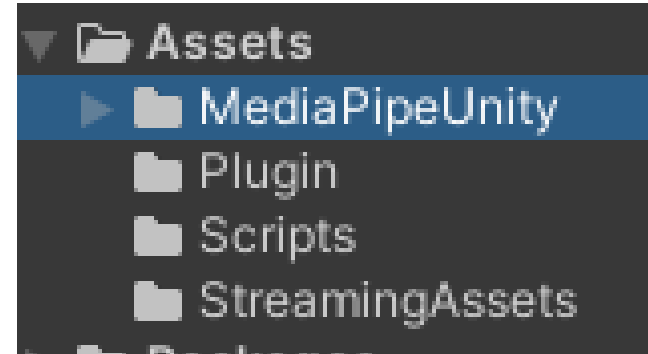


Assets

MediaPipeUnity

Plugin

Scripts



Plugin

OpenCV2DLL(다음 단원에서 소개)을
Unity Assets의 Plugin에 넣음.



CamScript

Unity RawImage와 Button 액션 정의

```
public void StartStopCameraClicked()
{
    if (wcTex != null)
    {
        StopCamera();
    }
    else
    {
        // tex = new Texture2D(wcTex.width, wcTex.height, TextureFormat.RGBA32, false, false);
        // rawImage.texture = tex;

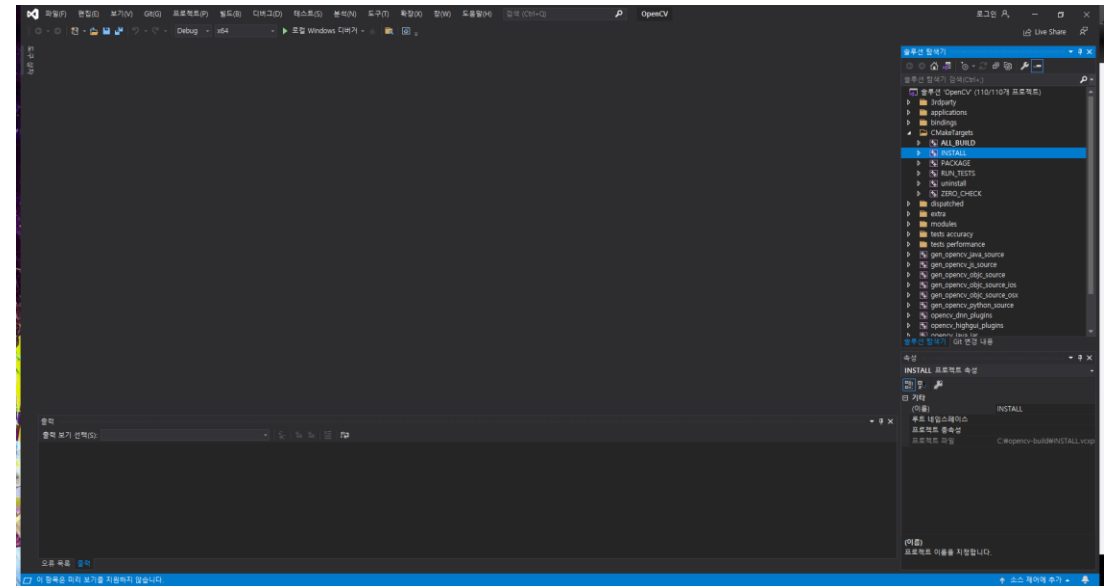
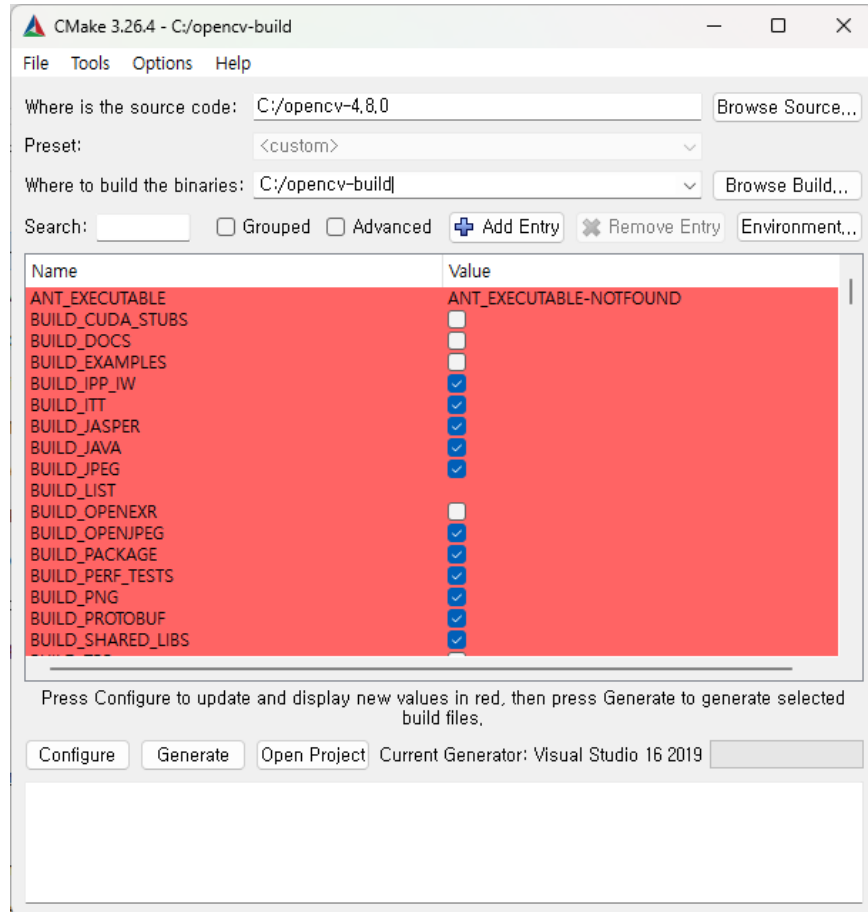
        WebCamDevice device = WebCamTexture.devices[currentCamIndex];
        if (!device.Equals(null))
        {
            int width = 1920;
            int height = 1080;

            Debug.Log("start");
            Debug.Log(rawImage);
            wcTex = new WebCamTexture(device.name, width, height, 24);
            rawImage.texture = wcTex;
            wcTex.Play();
            startStopText.text = "Stop";
        }
    }
}
```

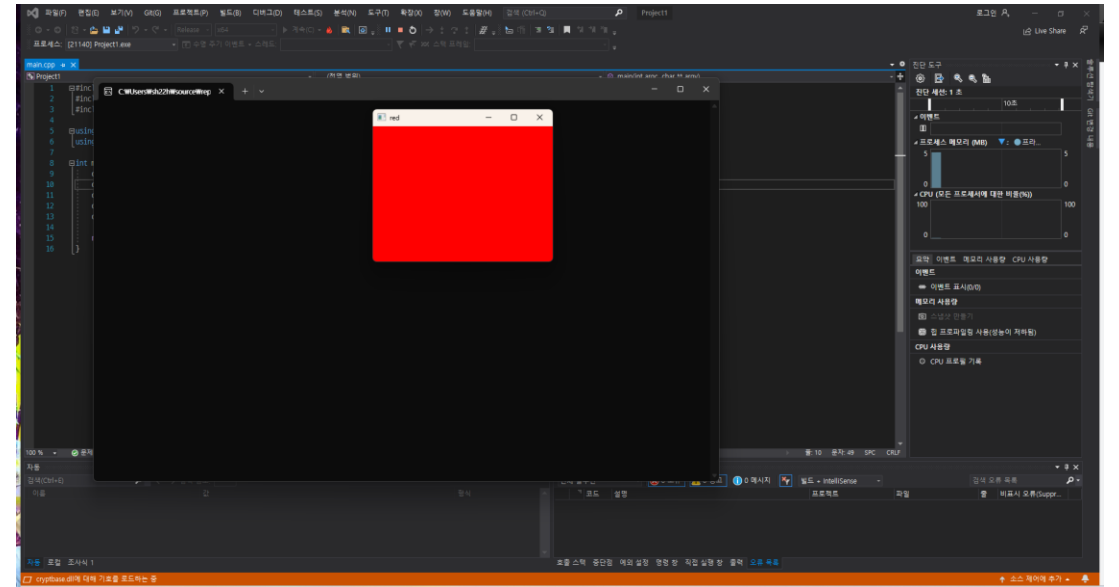
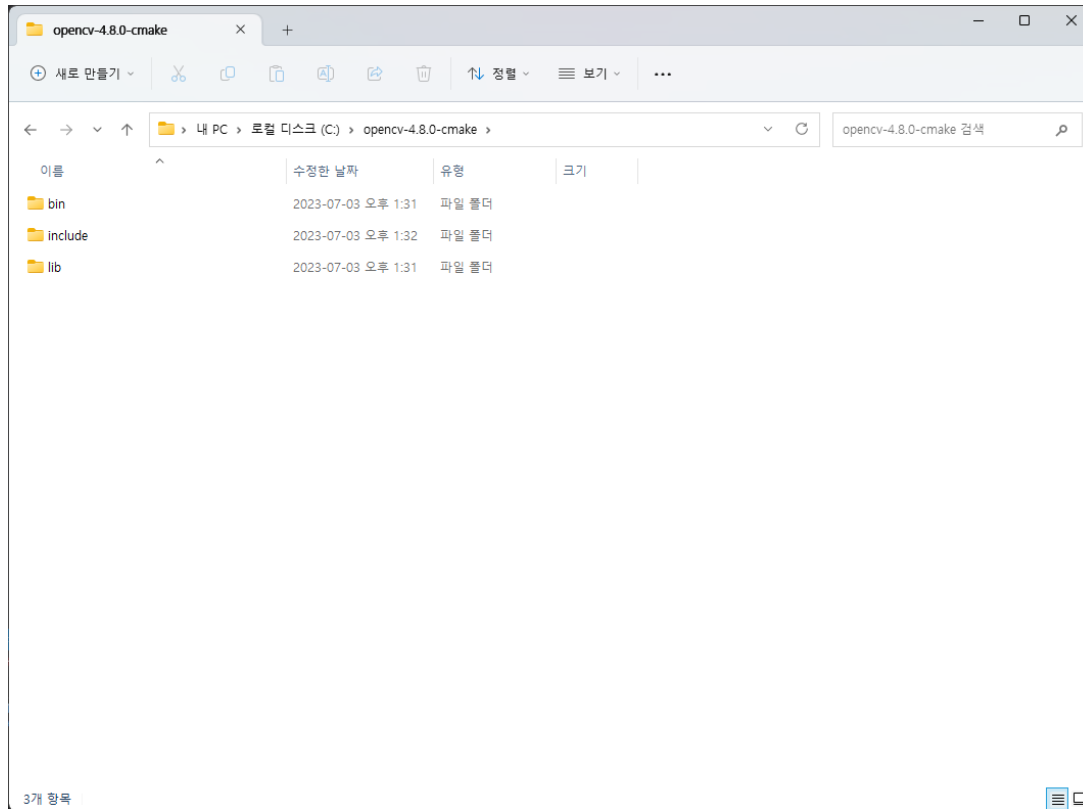

OpenCV2DLL

OpenCV(C++)을 Unity(C#)에서 사용할 수 있는 환경 구축

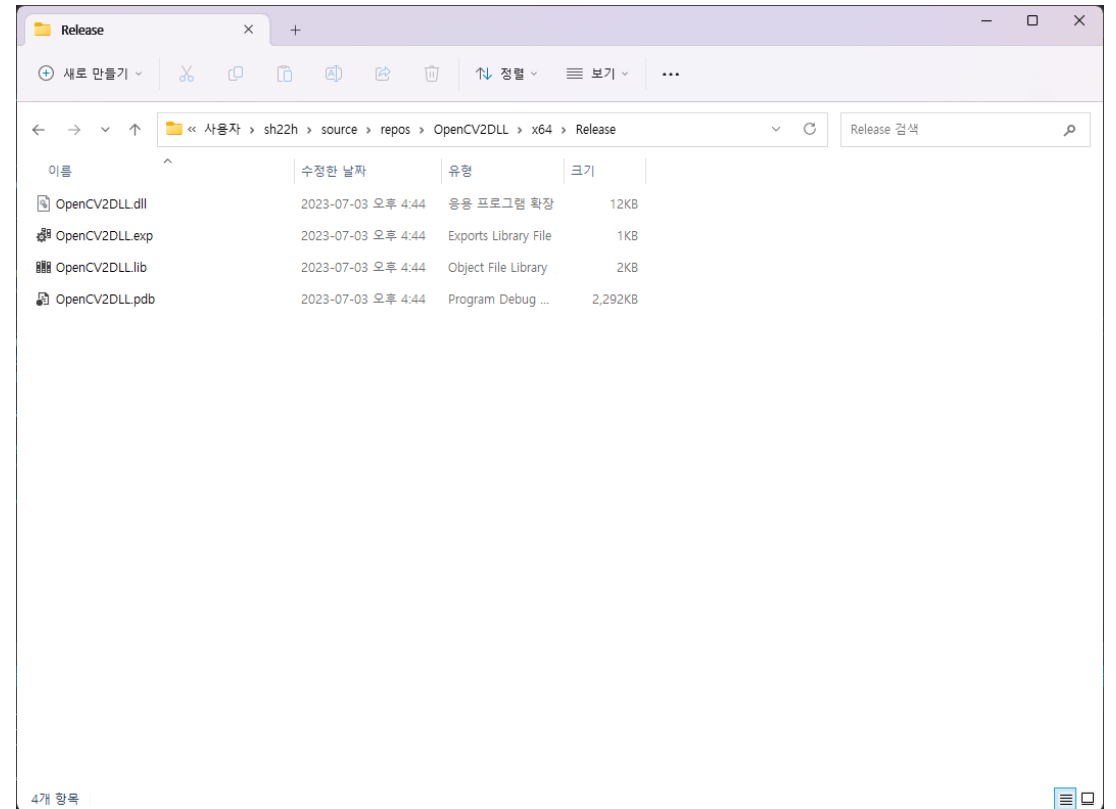
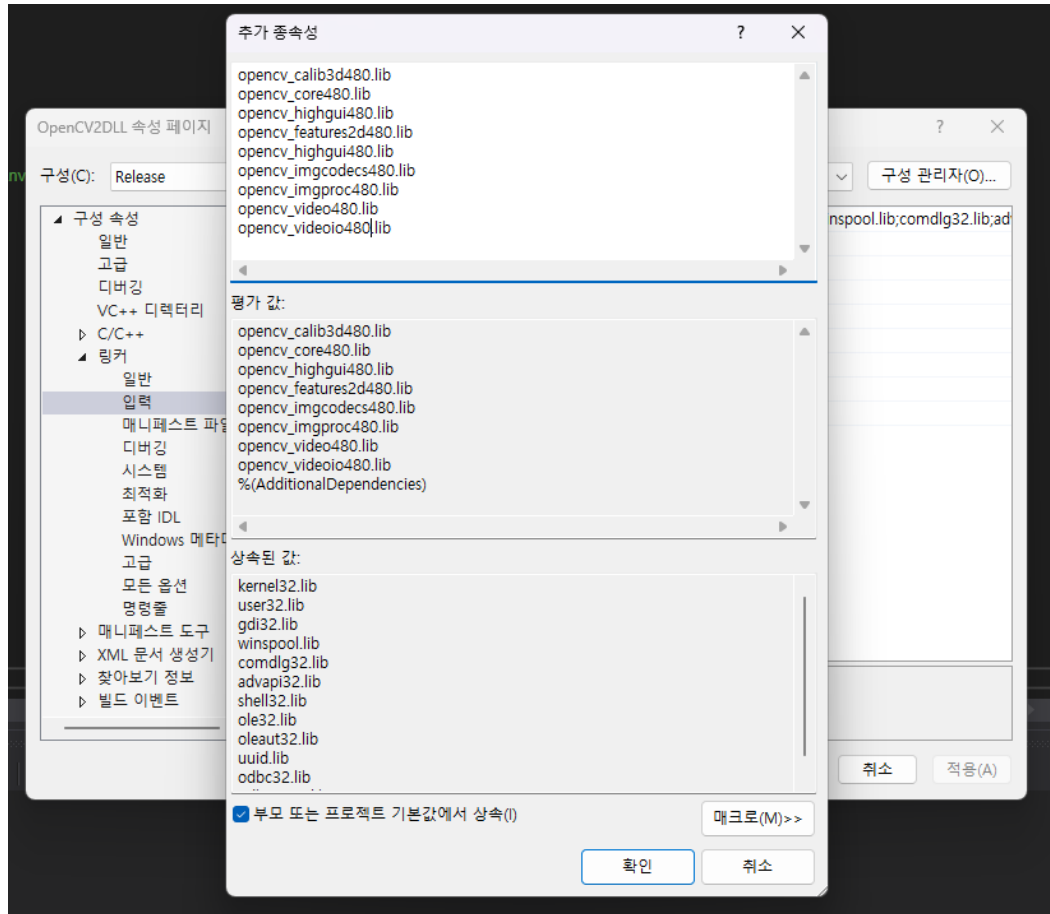
CMake Build



CMake Build



OpenCV2DLL



OpenCV2DLL

ar.h

DLL 파일을 Unity에서 사용할 수 있게 내보냄

```
#pragma once
#include <opencv2/opencv.hpp>

#ifdef MY_AR_EXPORTS
#define MY_AR_API __declspec(dllexport)
#else
#define MY_AR_API __declspec(dllimport)
#endif

struct Color32
{
    uchar red;
    uchar green;
    uchar blue;
    uchar alpha;
};

extern "C" MY_AR_API void ProcessImage(Color32 * *rawImage, int width, int height);
extern "C" MY_AR_API int DetectMarkers(Color32 * *rawImage, int width, int height);
extern "C" MY_AR_API int DetectMarkerPoses(Color32 * *rawImage, int width, int height, PoseContainer & container);
extern "C" MY_AR_API void Hello();
```

OpenCV2DLL

Ar.cpp

rvec, tvec 저장하는 구조체 정의

Unity에서 사용

```
struct Pose // rvec, tvec 저장하는 구조체
{
    int id;
    double x;
    double y;
    double z;
    double rx;
    double ry;
    double rz;
};

struct PoseContainer // rvec, tvec 저장하는 구조체
{
    Pose poses[C_SHARP_MAX_OBJECTS];
};
```

OpenCV2DLL

DetectMarkerPoses

웹캠 입력받아 Aruco 마커를 찾고, Aruco
의 위치 반환하는 코드

Python OpenCV Aruco solvePnP
코드 들어가야함.

```
int DetectMarkerPoses(Color32** rawImage, int width, int height, PoseContainer& container) {
    std::vector<int> ids;
    std::vector<std::vector<cv::Point2f>> corners;

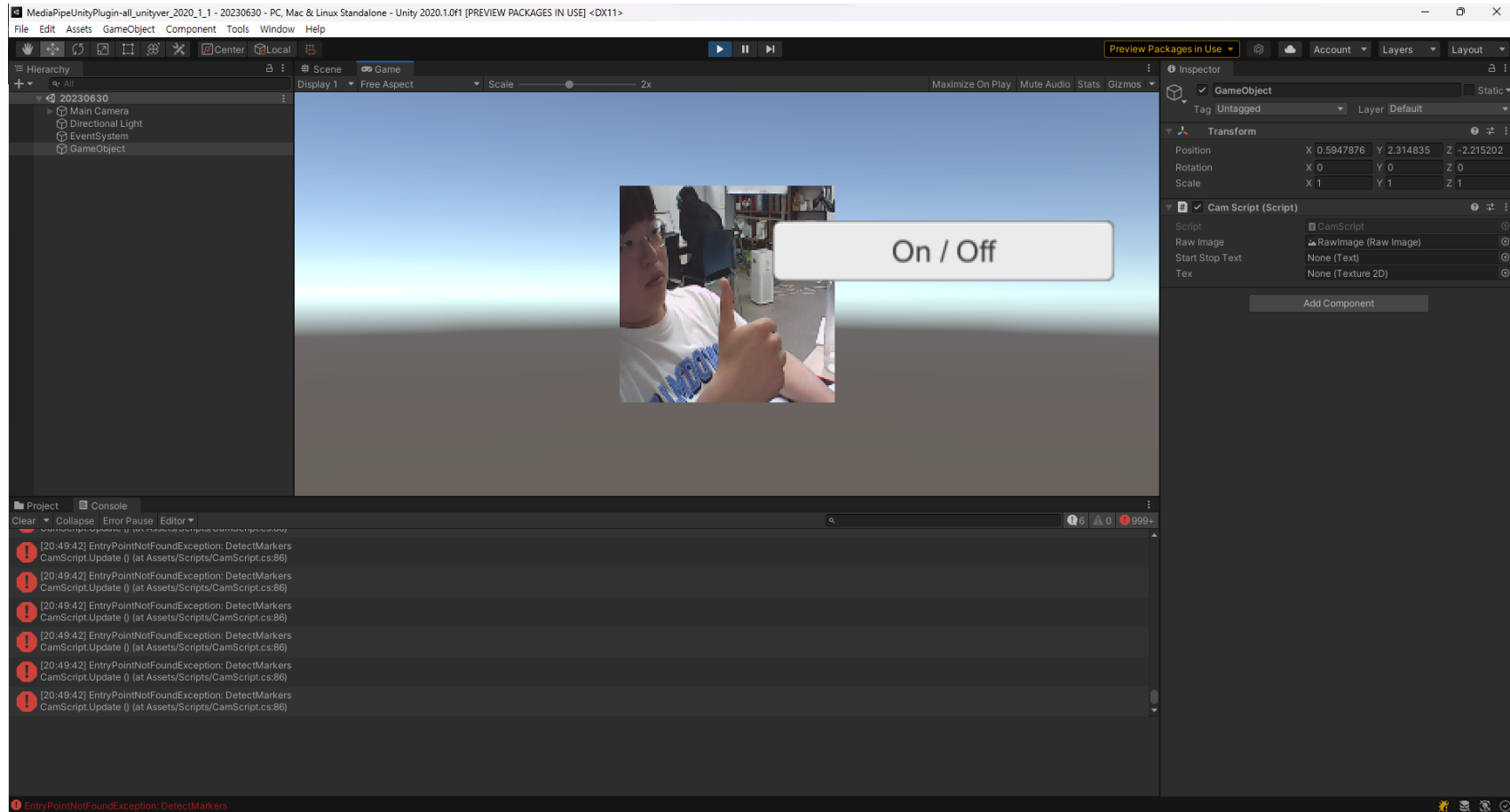
    // create an opencv object sharing the same data space
    cv::Mat image(height, width, CV_8UC4, *rawImage);
    cv::Mat imageCopy;
    image.copyTo(imageCopy);
    flip(imageCopy, imageCopy, 0);
    cv::cvtColor(imageCopy, imageCopy, cv::COLOR_RGBA2BGR);

    // opencv 4.8.0에서는 사용하지 않는 메소드.
    // cv::aruco::detectMarkers(imageCopy, dictionary, corners, ids);
    // 아래 코드로 변경
    cv::aruco::DetectorParameters detectorParams = cv::aruco::DetectorParameters();
    cv::aruco::Dictionary dictionary = cv::aruco::getPredefinedDictionary(cv::aruco::DICT_7X7_1000);
    cv::aruco::ArucoDetector detector(dictionary, detectorParams);
    detector.detectMarkers(imageCopy, corners, ids);

    // if at least one marker detected
    if (ids.size() > 0) {
        std::vector<cv::Vec3d> rvecs, tvecs;
        // opencv 4.8.0에서는 사용하지 않는 메소드.
        // cv::aruco::estimatePoseSingleMarkers(corners, 0.05, cameraMatrix, distCoeffs, rvecs, tvecs);
        // solvePnP로 대체할 것
        //
        // PyOpenCV에서 개발한 내용 들어가야함.

        // draw axis for each marker
        for (int i = 0; i < ids.size(); i++) {
        }
    }
    return ids.size();
}
```

실행 화면



향후 계획 및 일정

1. Unity 2020.1.0f1 버전 다운그레이드 (완료)
2. OpenCV(C++)을 Unity(C#)에서 사용할 수 있는 환경 구축 (완료)
3. solvePnP 체스보드 -> Aruco로 변경
-> 민재, 상현과 협업하여 코드 통합
-> 7. 7. 까지
4. Aruco 마커 위에 물체 올리기
-> 7. 10. 까지
5. Python에서 구현한 SolvePnP를 이용한
손 3D 인식을 Unity에서 구현
-> ...

6. 서비스 개발
-> ...
7. 개발 및 연구과정 문서화
-> ...
8. 논문 작성
-> ...

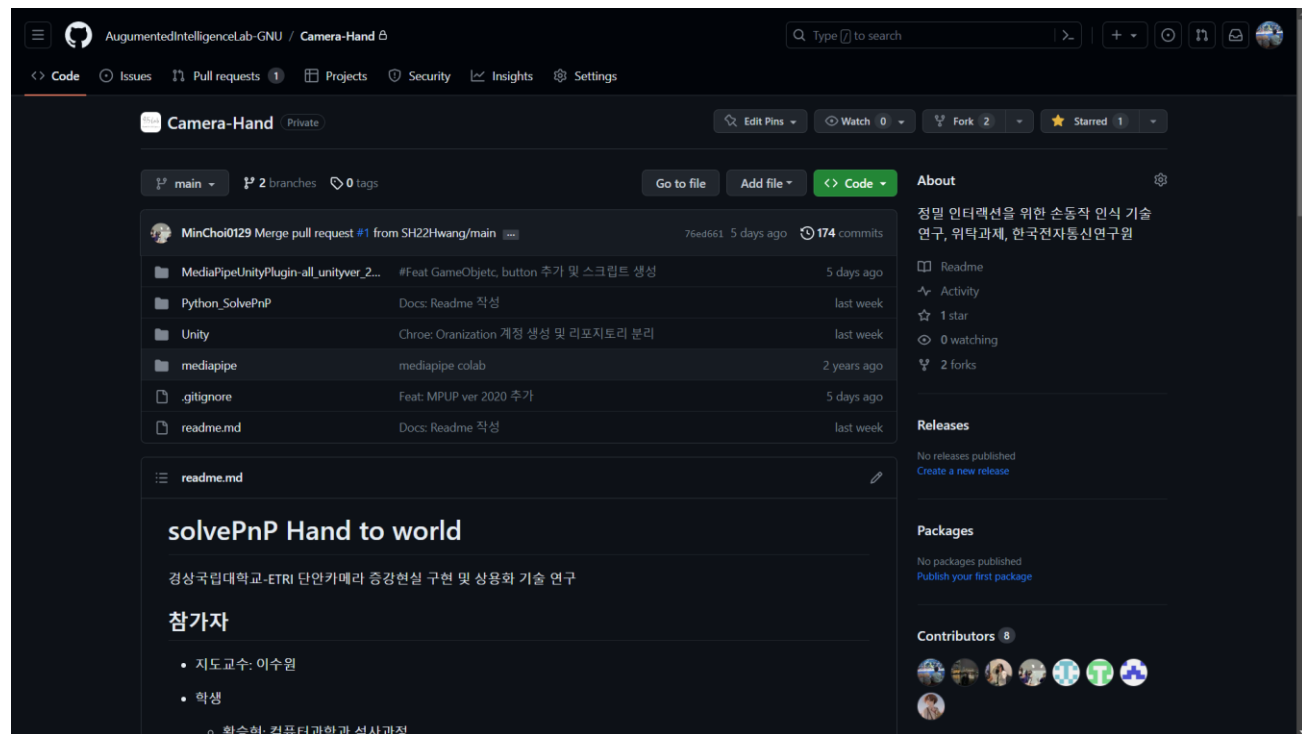
협업

경상국립대학교 증강지능연구실
organization 계정 repository

<https://github.com/AugumentedIntelligenceLab-GNU/Camera-Hand>

참가자:

김상현, 이민재, 황승현



참고자료

https://www.zhihu.com/column/c_1480111039178420224

칼럼 제목: Unity: 처음부터 AR에 참여

칼럼 설명: 간단한 AR 프로젝트를 통해 메타버스로 안내합니다.

작성자: 程法说

참고자료에서 바뀐 점

- DLL 파일 추출 방법 변경
- OpenCV 4.5.3 -> OpenCV 4.8.0
- Aruco 관련 일부 코드 변경

知乎

专栏
Unity: 从零开始搞AR

切换模式

写文章

Unity: 从零开始搞AR

通过一个简单的AR项目，带你进入元宇宙

程法说 · 9 篇内容 · ...

【Unity: 从零开始搞AR|07】将3D物体显示在ArUco上



万事具备，只欠东风。通过前面的介绍，ArUco的位姿已经可以获取到，同时Unity中的Prefab也了解了创建方法，本节，也是本系列的最后一节，将把上一节创建的Cube实时地显示在ArUco出现的位置... [阅读全文](#) ✓

赞同 4

8 条评论

收藏

【Unity: 从零开始搞AR|06】创建3D物体并在Canvas之前显示



经过前几节的铺垫，我们已经可以获取摄像头中的视频以及Marker的位姿，后续的工作就主要集中在3D虚拟物体的显示。因此，本节将介绍简单的3D物体在Unity中的创建及显示。主要有：新建一个Cube上... [阅读全文](#) ✓

赞同

添加评论

收藏

【Unity: 从零开始搞AR|05】OpenCV识别Aruco位姿（下）



本节主要解决的问题是Marker位姿估计及在Unity脚本中接收从dll返回的位姿结果。同时，本节也可看做是C#调用Dll的P/Invoke教程。Marker位姿估计 我们可以从opencv提供的接口中... [阅读全文](#) ✓

赞同 1

添加评论

收藏

【Unity: 从零开始搞AR|04】使用OpenCV识别ArUco位姿（Marker-Based