ARuco Marker Detection (with OpenCV)

3개의 세부 프로젝트(ARuco\_Calibration, ARuco\_Detection, ARuco\_3DRecon)로 구성됨

# ARuco\_Calibration

기능

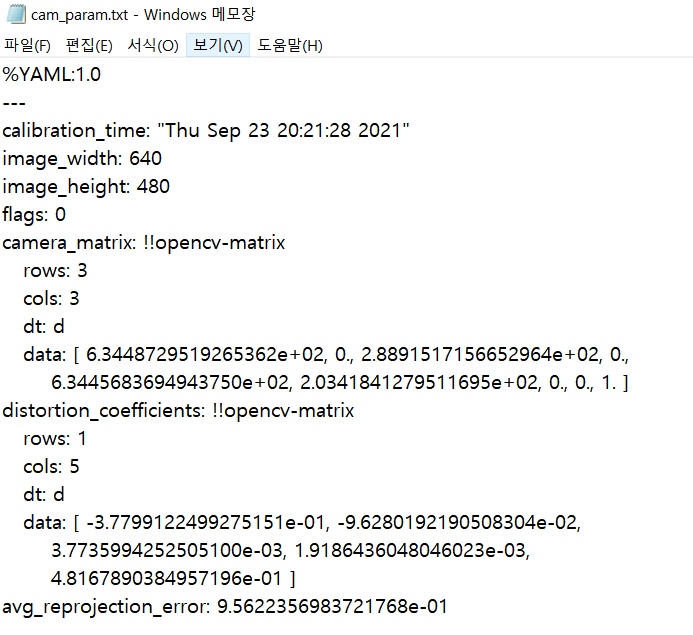
* ARuco marker 와 checkerboard 를 이용해 Camera Calibration 수행
* Camera Intrinsic (focal length, principle point) 와 Distortion coefficient 를 찾음

동작과정

* Project “ARuco\_Calibration.exe” 실행
* 입력 인수 :

|  |  |
| --- | --- |
| 주요 구분자 | 의미 |
| --w | Number of squares in X direction |
| --h | Number of squares in Y direction |
| --sl | Square side length (in milimeters) |
| --ml | Marker side length (in milimeters) |
| --d | ARuco marker dictionary |
| --v | Input from video file |
| --ci | Camera id if input doesnt come from video |
| --sc | Show detected chessboard corners after calibration |
| (empty) | Output text file |

* 명령인수 예시 :
  + (촬영된 동영상을 입력으로 주는 경우) ARuco\_Calibration.exe --w=5 --h=7 --sl=36 --ml=18 --d=10 --v=../dataset/ARuco\_test.mp4 --sc=true "../dataset/cam\_param.txt"
  + (웹캠을 통해 실시간으로 촬영하는 경우) ARuco\_Calibration.exe --w=5 --h=7 --sl=36 --ml=18 --d=10 --ci=1 --sc=true "../dataset/cam\_param.txt"
* 출력 데이터 :
  + Text file : Camera intrinsic matrix 와 Distortion coefficient 등의 정보를 포함한 text 파일 출력
* 결과 영상 예시: (text 파일 캡쳐)



* Reference Source : OpenCV\_contrib-4.5.3 의 aruco modules 의 샘플 코드 사용

# ARuco\_Detection

기능

* 웹캠을 통해 캡쳐되는 ARuco marker 의 마커 정보와 3차원 공간 정보를 찾음
* 모든 ARuco marker 의 인덱스와 3D 좌표축을 표시
* 특정 ARuco maker 위치에 고정된 3D 정육면체 표시
* 3D reconstruction 을 위한 일부 장면을 캡처 (ARuco\_3DRecon 의 input data 로 활용)

동작과정

* Project “ARuco\_Detection.exe” 실행
* 입력 인수 :

|  |  |
| --- | --- |
| 주요 구분자 | 의미 |
| --d | ARuco marker dictionary |
| --c | Camera parameter file |
| --ci | Camera id if input doesnt come from video |
| --ov | Output video file |
| --o | Output folder of save captured frames |

* 명령인수 예시:
  + (웹캠을 사용하는 경우) ARuco\_Detection.exe --c=../dataset/cam\_param.txt --d=10 -ci=1 --ov=../dataset/AR\_detection.avi --o=../dataset/pics
* 출력 데이터 :
  + avi 파일: 3D axis 와 3D cube 를 그린 결과 영상을 출력
  + png (또는 jpg) 파일 셋: 동영상의 일부 프레임에 대해 캡쳐 후 png (또는 jpg)으로 저장 🡪 해당 데이터 셋은 3D reconstruction 을 위해 사용됨
* 결과 영상 예시: (viewer 의 일부 화면 캡쳐)

|  |  |
| --- | --- |
| 3D Axis | 3D Cube |
|  |  |
|  |  |

* Reference Source : OpenCV\_contrib-4.5.3 의 aruco modules 의 샘플 코드 참조

# ARuco\_3DRecon

기능

* 다시점 정지 영상으로부터 Camera Pose Estimation 수행 (camera pose xyz 파일 생성)
* 다시점 정지 영상으로부터 Space Point Cloud 생성 (3d object xyz 파일 생성)
* 생성된 3D point file (camera pose, object) 을 읽고 OpenCV viewer 로 디스플레이

동작과정

* Project “ARuco\_3DRecon.exe” 실행
* 입력 인수:

|  |  |
| --- | --- |
| 주요 구분자 | 의미 |
| --i | Input folder to get image set |
| --o | Output file to save 3d points (xyz file) |
| --c | Camera intrinsic parameter file |

* 명령인수 예시:
  + (캡쳐된 사진을 입력으로 받는 경우) ARuco\_3DRecon.exe --i=../dataset/pics -c=../dataset/cam\_param.txt -o=../dataset/sfm\_point.xyz
* View control 방법
  + 상하좌우 이동 : 4개의 방향키 (→, ←, ↑, ↓) 이용
  + 줌인 : ‘Page in’ 키 이용
  + 줌아웃 : ‘Page out’ 키 이용
* 출력 데이터:
  + Camera pose xyz file : 입력 카메라의 위치를 3D point 로 저장
  + Object xyz file : 3차원 공간 상의 structure 를 3D point 로 저장
* 결과 영상 예시: (viewer 의 일부 화면 캡쳐)

|  |  |
| --- | --- |
| Camera position | 3D structure |
|  |  |

* Reference Source : <https://github.com/sunglok/3dv_tutorial> 소스 코드 참조