[서식 3]

항공우주종합설계 중간보고서

주제명 (가칭)	OpenCV 를 이용한 ArucoMarker 위치 및 자세추정과 모델을 이용한 검증			
지도교수	소속학과	항공우주공학과	성명	이대우
대표학생	성명	정대현	휴대전화	010-2639-3630
	E-mail	empcik@pusan.ac.kr	학번/학년	201527137 / 4 학년

참여학생 (대표학생 제외)	학과	성명	학번	학년	비고
	항공우주공학과	박유경	201725171	4 학년	

본인을 포함한 팀원 전원은 항공우주종합설계를 성실히 수행하고 중간보고서를 제출합니다.

2021 년 08 월 25 일

신청인(대표학생) (서명)

지도교수 (서명)

항공우주공학과장 귀하

목차

1. 설계목표 및 문제점

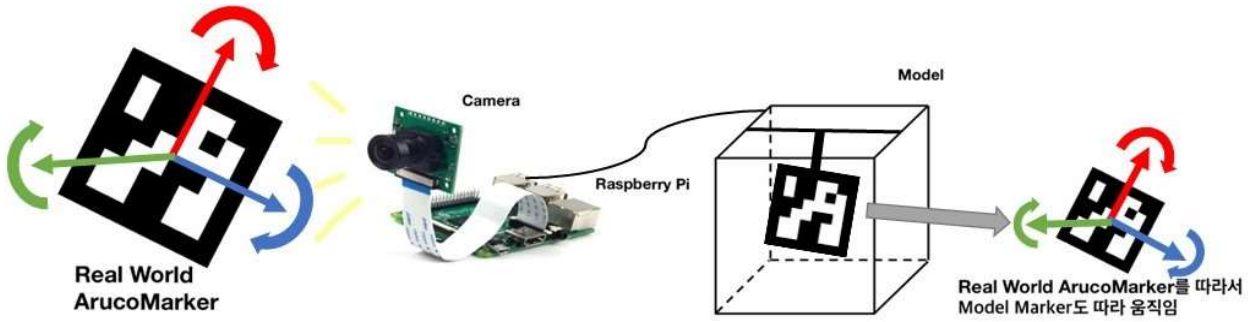

Figure 1 최종 설계 모형 예상도

1.1 설계목표

본 프로젝트의 설계목표는 OpenCV(Open Source Computer Vision)와 ArucoMarker 를 이용한 위치 및 자세 추정과 모델을 이용한 검증입니다. 목표 달성을 위해 컴퓨터 비전에서 대중적으로 사용되는 OpenCV 를 이용하여 Fig. 2 과 같은 마커를 인식하여 해당 마커의 위치 및 자세에 대한정보를 획득합니다. 이 과정에서 식별 마커인 ArucoMarker 에 대한 이해가 필요합니다. 그리고 설계 프로그램을 가동하기 위해서 임베디드 시스템인 라즈베리파이를 사용합니다.

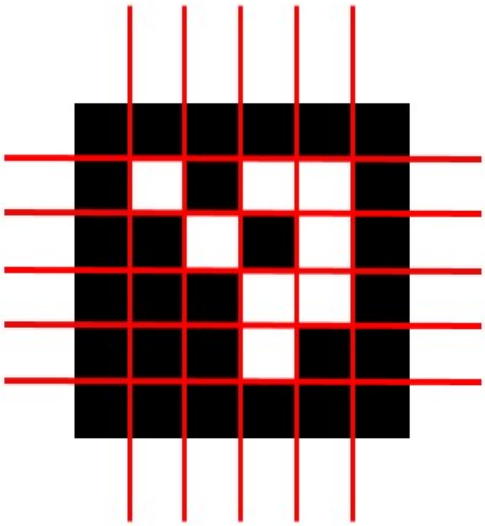

Figure 2 ArucoMarker의 구성

1.2 프로젝트 예상 문제점

1.2.1 단안카메라의 한계점

본 프로젝트는 기존의 거리 측정을 위해 많이 사용하는 방식인 라이다(LIDAR)와 적외선 센서, 스테레오 카메라를 사용하지 않고 단안카메라(Mono Camera)로 프로젝트를 수행하는 것이 목표입니다. 사람의 눈은 두개로 이루어져 있습니다. 그러다 보니 '단안'을 사용하는 경우에 대해서 한 가지 중요한 사실을 놓치고 있는데 우리는 두 개의 눈을 통해서 거리 정보를 획득하고 있습니다. 어떤 특정 물체를 쳐다볼 때, 오른쪽 눈과 왼쪽 눈에 맺히는 상은

서로 다른 곳에 위치하며 이 거리 차이를 통해서 물체와 나의 거리를 얻을 수 있습니다. 컴퓨터 비전에서도 마찬가지로 스테레오 카메라(Stereo Camera)를 이용한 물체의 거리 추정이 가능하나, 연산량이 급격하게 늘어나는 단점을 가지고 있습니다. 따라서 본 프로젝트에서는 연산량이 적은 단안카메라를 사용하기로 했습니다. 하지만 단안카메라를 사용하게 되면 3 차원의 물체가 2 차원의 이미지로 변환하는 과정에서 거리 정보가 지워지게 되는 한계가 있습니다.

1.2.2 실시간성 보장

이번 프로젝트는 영상을 통해 ArucoMarker 정보를 획득하는데 실시간성을 보장하고자 합니다. 이 프로젝트에서 실시간성을 보장한다는 것은 카메라로부터 들어온 입력과 해당 영상을 분석해서 결과를 보여주는 화면의 출력의 속도가 같은 것을 의미합니다. 예를 들어 30Hz 는 1 초당 30 장의 사진이 모니터를 통해 출력되는 것을 의미합니다. 이러한 속성을 가진 영상을 라즈베리파이에 입력했을 때 한 장당 1/30 초로 이내로 처리할 수 있다면 입력과 동일한 시간의 출력을 보장받을 수 있으며 이는 실시간성이 있다고 할 수 있습니다. 하지만 초소형컴퓨터인 라즈베리파이의 성능을 고려했을 때 고해상도, 고주사율의 영상을 사용하는데 제한이 있습니다.

2. 해결방안 및 예산

앞서 계획한 설계목표와 예상되는 문제점을 해결하기 위해서 팀원들과 브레인스토밍을 통해 다양한 해결방안을 생각해 보았습니다. 이를 위해 생각한 해결방법 아이디어는 다음과 같습니다.

2.1 위치 및 자세 추정의 정확도 개선을 위한 방안

자세추정 정확도 개선을 위한 첫번째 아이디어는 단안카메라의 정확한 거리 추정을 위해서 고정된 크기의 ArucoMarker 를 사용할 것입니다. 단안카메라는 위에서 언급하였듯이 3 차원의 물체가 2 차원의 이미지로 변환하는 과정에서 거리 정보가 지워지게 되는 한계가 있습니다. 하지만 고정된 크기의 마커를 사용함으로써 거리 정보를 추정할 수 있습니다. 또한, 카메라 캘리브레이션을 통해 카메라 내부 파라미터를 구하고 영상의 왜곡을 보정하여 정확도를 보다 개선할 것입니다. 마지막으로 흑백사진을 이용하여 자세 추정을 할 것입니다. 본 프로젝트에서 사용할 ArucoMarker 는 흑과 백으로 이루어져 있기 때문에 이 프로젝트를 수행함에 있어 필요한 정보는 컬러가 아닌 흑백입니다. 흑백을 사용하면 이미지의 대비가 명확해지며 마커의 가장자리가 뚜렷하게 구분이 됩니다. 이는 마커의 인식이 보다 잘 되어 정확도가 올라가는 효과를 기대할 수 있습니다.

2.2 실시간성 보장을 위한 방안

실시간성 보장 목표를 달성하기 위해 각 해상도와 프레임에 따라서 여러 차례 실험을 수행할 것입니다. 동영상의 속성은 크게 두가지로 해상도와 프레임입니다. 동영상의 속성이 달라지면 영상의 정보량 크기가 달라진다는 의미입니다. 이는 라즈베리파이가 해당 영상을 재생과 처리할 때 요구하는 자원의 양이 달라진다는 의미입니다. 우리는 목표한 입력 값을 처리할 수 있는 해상도와 주사율을 결정하여 입출력의 실시간성을 보장하는 것입니다. 예를 들어 60 프레임을 기준으로 했을 때 저해상도 영상에서 1 프레임당 1/60 초안에 충분히 처리를 한다면 이는 자원에 여유가 있다는 의미이며, 해상도를 더 키울 수 있다는 것을 알 수 있습니다. 입력하는 영상의 해상도와 프레임을 조금씩 변경해보며 trial and error 시도를 통해 최적의 값을 찾을 것입니다.

2.3 예산, 장비 목록

본 프로젝트에서 사용된 장비, 예산 목록은 아래의 표로 첨부합니다.

No.	품명	수량	단가	상품금액합계 (VAT 별도)	배송비 (VAT 포함)
1	스텐와이어 2.0mm 100M	1	28,000 원	28,000 원	
2	2 중 구멍 커넥팅 로드 64502B 10 개입 [SZH-GNP382]	1	1,500 원	1,500 원	
3	와이어로프 슬리브 M2.5 - 땅콩형 [SZH-WSD005]	10	80 원	800 원	
4	라즈베리파이용 GPIO E3P 확장보드 [SZH-AT037]	1	6,400 원	6,400 원	
5	라즈베리파이 카메라모듈 V2, 8MP (RPI 8MP CAMERA BOARD)	1	33,000 원	33,000 원	
6	[정품] SG90 360 도 디지털 서보모터	4	3,600 원	14,400 원	
7	5V 학생실습용 DC 모터 [SZH-MT001]	3	600 원	1,800 원	
8	2A L298 모터드라이버 모듈 (아두이노 호환) [SZH-EK001]	2	2,000 원	4,000 원	
9	메탈 컴파운드기어 3.85ø 15/55T [SZH-GNP179]	2	1,500 원	3,000 원	
10	RS390 피니언기어 DC 모터 [SZH-GNP208-1]	2	12,800 원	25,600 원	0 원
11	플라스틱 기어 78 개입 세트 [SZH-GNP231]	1	6,900 원	6,900 원	
12	메탈 샤프트 3ø 150mm 10 개입 [SZH-GNP339]	1	2,400 원	2,400 원	
13	소형 DC 기어드모터 [SZH-GNP093]	1	14,000 원	14,000 원	
14	2GT6mm 타이밍 벨트 [SZH-BPM005]	1	4,500 원	4,500 원	
15	[리퍼제품]체인&스프라켓 키트 (Chain and sprocket kits)	1	1,000 원	1,000 원	
16	[리퍼제품] 아타치먼트 체인 (40-1 K1/2-35LINK)	1	1,000 원	1,000 원	
17	7 인치 와이어 스트립퍼 (AWG20~30) [K11710]	1	3,700 원	3,700 원	
18	플라스틱 래크기어 125mm [SZH-GNP291]	6	700 원	4,200 원	
19	[리퍼제품]MSGB048E01	1	3,000 원	3,000 원	
20	비흐름성 순간접착제 AXIA 659 GEL, 20g	1	3,500 원	3,500 원	
21	6F22(9V,FC-1)Be	3	1,200 원	3,600 원	
22	Snap.I-3(for 9V)	3	230 원	690 원	
23	라즈베리파이 4 (4GB) 스타터 키트 MicroSD 용량 선택:16GB	1	103,000 원	103,000 원	2,500 원
소계		49		269,990 원	2,500 원
합계 (VAT 포함)	299,489 원				

3. 분석된 내용 및 학습해야 할 내용

3.1 ArucoMarker 의 이해

물체의 위치 및 자세 추정은 로봇 위치 탐색, AR 증강현실 등 컴퓨터 비전에서 중요한 부분 중 하나입니다. 물체의 위치 및 자세를 추정하는 데는 크게 두가지로 나누어지는데 이는 마커의 유무입니다. 이 프로젝트에서는 마커를 사용하기로 결정했으며 해당 마커는 Fiducial marker(식별 마커)라고 불리게 됩니다 컴퓨터 비전을 위한 증강현실 식별 마커는 대표적으로 ARToolKit, ARTag, AprilTag, ArUco 등이 존재합니다.

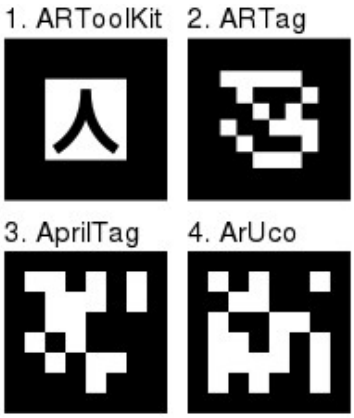

Figure 3 식별마커의 종류

본 프로젝트에선 OpenCV 라이브러리에 내장되어 있는 ArucoMarker 를 식별마커로 사용할 계획입니다. ArucoMarker 란 NxN 크기의 비트와 비트를 둘러싼 테두리로 구성된 정사각형 마커이며 크기에 따라 각기 다른 고유번호를 갖습니다. ArucoMarker 의 검은색 테두리는 이미지에서 빠른 감지를 용이하게 하고 이진 코드화를 통해 식별과 오류 감지 및 수정이 가능합니다. ArucoMarker 만이 가지고 있는 특징은 첫번째로 ArucoMarker 내부거리와 비트 천이 횟수를 고려한 마커 사전과의 간단한 비교만으로 식별자(id)를 구분할 수 있습니다. 둘째, QR 코드에 있는 에러 검출 코드가 존재합니다. 셋째, 마커 식별에 회전이 고려되기 때문에 회전되어 있는 마커도 간단히 구분할 수 있습니다.

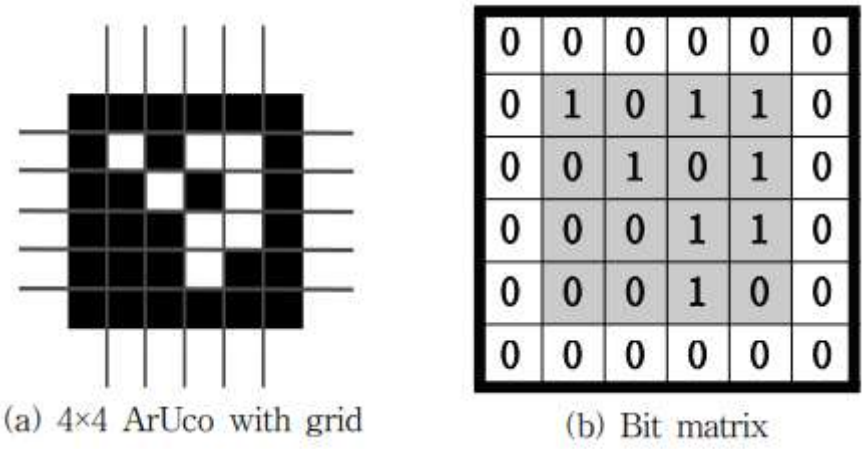

Figure 4 ArucoMarker의 이진분류

마커를 식별하는 과정은 Fig. 4b 에서 보이는 내부 4×4 비트 매트릭스는 좌측 상위부터 오른쪽으로 스캔 되며 Circular shift 연산을 통하여 변수에 저장되고 바이트 리스트로 변환됩니다. 위 예시에서는 (4×4)/8=2 바이트가 필요합니다. 각 바이트를 10 진수로 표현하여 미리 저장되어 있는 Dictionary 와 비교하게 됩니다. 이 과정에서 마커의 식별자구분, 에러검출, 마커의 회전을 알 수 있습니다.[1]

2.1 에서 언급했듯이 단안카메라 자체만으로는 3 차원의 물체가 2 차원의 이미지로 변환하는 과정에서 거리 정보가 지워지게 되는 한계가 있어 거리 정보를 얻을 수 없습니다. 하지만 사전에 설정된 크기가 있다면 거리 정보를 얻을 수 있습니다. 따라서 본 프로젝트에선 15cm x 15cm 크기의 ArucoMarker 선정하여 단안카메라로부터 거리를 추정할 것입니다.

3.2 컴퓨터 비전과 OpenCV 라이브러리 이해

컴퓨터 비전은 최근 4 차 산업 혁명에서 각광받는 기술이며 기계의 시각에 해당하는 부분을 연구합니다. 공학적인 관점에서 컴퓨터 비전은 인간의 시각이 판단하는 일을 자동화하는 것을 목표로 합니다. 과학적인 관점은 이미지에서 정보를 추출하는 인공 시스템 관련 이론에 사용합니다. 우리는 이번 프로젝트를 통해서 ArucoMarker 의 위치 및 자세에 따라 상대적인 위치정보와 회전각도를 추출할 것이며 이는 실제 세계와 카메라의 소통이기 때문에 컴퓨터 비전을 사용합니다.

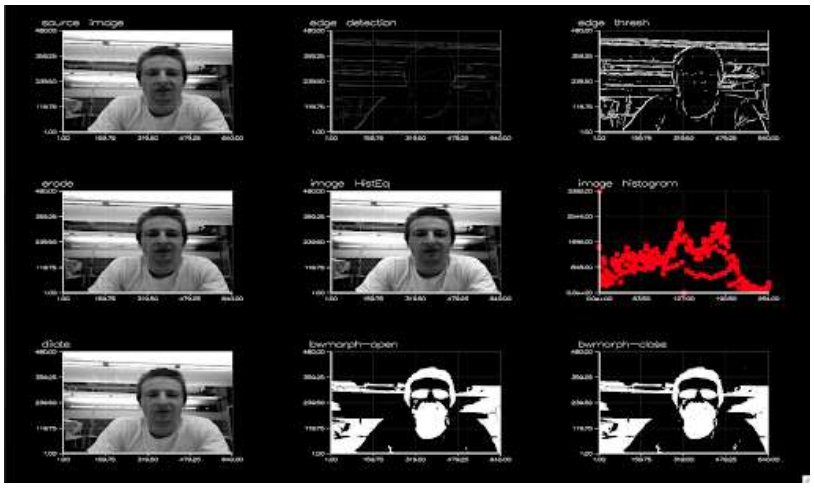

Figure 5 OpenCV를 이용한 이미지 변환

OpenCV 는 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리 중 하나로 크로스플랫폼과 실시간 이미지 프로세싱에 중점을 두었습니다. 크로스플랫폼이란 컴퓨터 운영체제와 프로그래밍 언어가 다르지만 다양한 플랫폼 위에서 동작할 수 있습니다. x86, ARM 과 같은 서로 다른 아키텍쳐를 가진 CPU 부터 C++, Python 다양한 언어에서도 잘 작동합니다. 영상 관련 라이브러리로서 사실상 표준적인 지위를 가지고 있으며, 영상처리 대중화에 기여한 OpenCV 는 다양하고 고난이도의 결과물을 코드 몇 줄로 얻을 수 있다는 큰 이점을 가지고 있습니다. 카메라로 정보를 얻는데 있어 OpenCV 라이브러리를 이용하여 구현할 것이며, 해당 프로젝트는 리눅스 기반의 라즈베리파이와 Python 언어로 구현할 계획입니다.

[1] Optical Flow-Based Marker Tracking Algorithm for Collaboration Between Drone and Ground Vehicle KIPS Tr. Software and Data Eng. Vol.7, No.3 pp.107~112 pISSN: 2287-5905

3.2.1 카메라 캘리브레이션의 이해

카메라 캘리브레이션 (camera calibration)은 영상처리, 컴퓨터 비전 분야에서 번거로운 과정이지만 반드시 필요한 과정 중의 하나입니다. 사람의 눈을 통해서 보는 세상은 3 차원입니다. 하지만 이를 카메라로 찍게 되면 2 차원의 이미지로 변합니다. 이 때, 3 차원의 점들이 이미지 상에서 어디에 맺히는지에 대해서 생각해보면 영상을 찍을 당시의 카메라의 위치 및 방향에 의해 결정됩니다. 하지만 실제 이미지는 사용된 렌즈, 렌즈와 이미지 센서와의 거리, 렌즈와 이미지 센서가 이루는 각 등 카메라 내부의 기구적인 부분에 의해서 큰 영향을 받습니다.

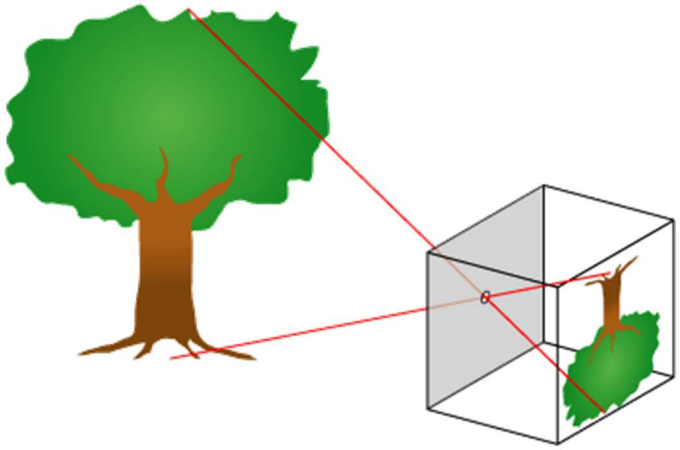

Figure 6 핀홀 카메라에 맺히는 상

3 차원 점들이 영상에 투영된 위치를 구하거나 역으로 영상좌표로부터 3 차원 공간좌표를 복원할 때에는 이러한 내부 요인을 제거하는 과정을 거쳐야 정확한 계산이 가능합니다. 따라서 이러한 내부 요인의 파라미터 값을 구하는 과정을 카메라 캘리브레이션이라고 합니다. 카메라 영상은 3 차원 공간상의 점들을 2 차원 이미지 평면에 투사(perspective projection)함으로써 얻어집니다. 핀홀(pinhole) 카메라 모델에서 이러한 변환 관계는 다음과 같이 모델링 됩니다.

$$[\blacksquare(x@y@1)] = [\blacksquare(f\_x@0@0)\blacksquare(0@f\_y\ @0)\blacksquare(c\_x@c\_y@1)][\blacksquare(r\_11@r\_21@r\_31\ )\blacksquare(r\_12@r\_22@r\_32\ )\blacksquare$$

$$A[R|t][\blacksquare(X@Y@\blacksquare(Z@1))] \text{ 식 1)}$$

(X,Y,Z)는 월드 좌표계(world coordinate system) 상의 3D 점의 좌표, [R|t]는 월드 좌표계를 카메라 좌표계로 변환시키기 위한 회전/이동변환 행렬이며 A 는 intrinsic camera matrix 입니다. 식 1)에서 [R|t]를 카메라 외부 파라미터(extrinsic parameter), A 를 내부 파라미터(intrinsic parameter)라고 부릅니다. 그리고 A 와 [R|t]를 합쳐서 camera matrix 또는 projection matrix 라 부릅니다.

따라서 정확도를 최대한으로 높이기 위해서 카메라의 외부 파라미터와 카메라 자체의 내부적인 파라미터를 변환행렬을 사용합니다. 여기서 위에서도 언급했듯이 카메라의 외부 파라미터는 카메라 좌표계와 월드 좌표계 사이의 변환 관계를 설명하는 파라미터로서, 두 좌표계 사이의 회전 및 평행이동 변환으로 표현됩니다. 따라서 카메라 외부 파라미터는 카메라 고유의 파라미터가 아니기 때문에 카메라를 어떤 위치 어떤 방향으로 설치했는지에 따라 달라지고 또한 월드 좌표계를 어떻게 정의했는가에 따라 달라집니다. 내부 파라미터는

카메라의 초점 거리(focal length: fx, fy), 주점(principal point: cx, cy), 비대칭 계수(skew coefficient: skew_c)가 있습니다.

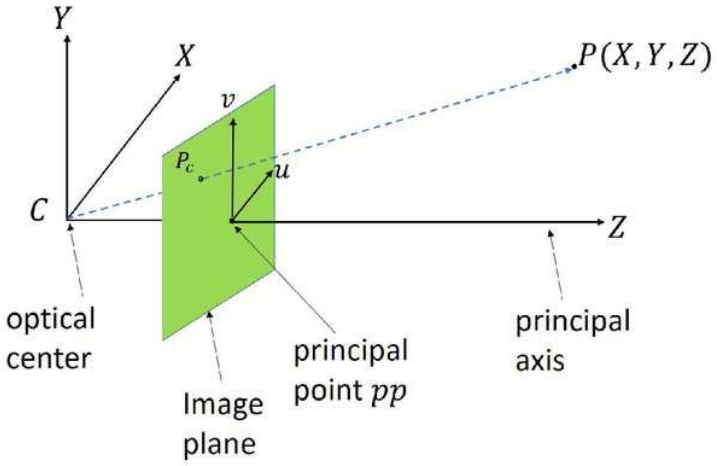

Figure 7 주점(principal point)과 주축(principal axis)

초점거리란 렌즈의 중심점과 초점(피사체)간의 거리를 일컫는 사진 용어로, 모든 렌즈(카메라)는 초점거리를 가지고 있습니다. 주점이란 사진의 중심으로, 투영 중심에서 투영 면에 내린 수선이 투영 면과 만나는 점. 입사동(入射瞳)의 중심을 통하여 사진 면에 직각으로 들어온 광선이 사진 면에 닿은 점을 말합니다.

여기서 비대칭 계수는 고려하지 않았습니다. 요즘 카메라들은 이러한 skew 에러가 거의 없기 때문에 카메라 모델에서 보통 비대칭 수까지는 고려하지 않습니다 즉, skew_c=0 입니다. 이러한 내부 요인의 파라미터 값을 구하는 과정은 카메라 자체의 고유 값을 획득하는 방안입니다. 카메라 보정 모듈은 입력 영상에 대한 전처리 과정입니다. 입력 영상을 흑백영상으로 변환하고 영상 내에 마커 패턴을 검출합니다. 그리고 검출된 패턴의 모서리의 3 차원 위치 정보와 영상 내 2 차원 정보를 쌍으로 얻은 후 OpenCV 에 내장되어 있는 cal-librateCamera 함수의 입력으로 사용하여 영상에 적용할 카메라 내부 행렬과 왜곡계수를 얻습니다. 여기서 왜곡계수는 radial distortion 계수(k1, k2) tangential distortion 계수 (p1, p2)를 의미합니다. 방사왜곡(radial distortion) 볼록렌즈의 굴절률에 의한 것으로서 영상의 왜곡 정도가 중심에서의 거리에 의해 결정되는 왜곡입니다.

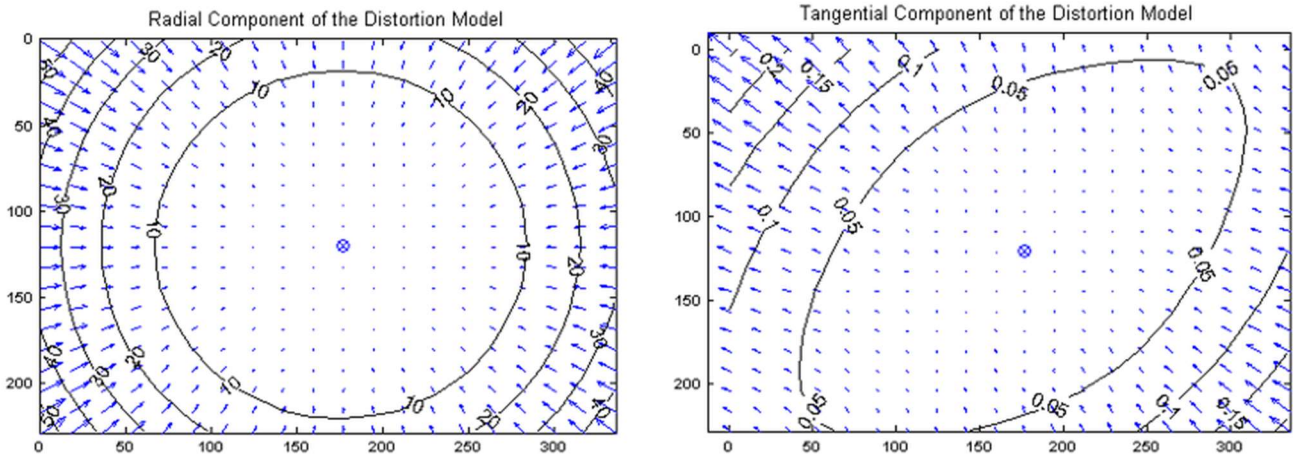

Figure 8 방사왜곡(radial distortion) 예시(좌)

Figure 9 접선왜곡(tangential distortion) 예시(우)

이렇게 얻은 카메라 내부행렬과 왜곡 계수를 이용하여 입력으로 들어오는 영상의 왜곡을 보정하여 본 프로젝트의 정확도를 보다 개선할 수 있습니다. 본 프로젝트에서 수행한 캘리브레이션 결과는 아래와 같습니다.

$$s\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}914.426678 & 0 & 640\\ 0 & 914.426678 & 360\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11}r_{12}r_{13}t_1\\ r_{21}r_{22}r_{23}t_2\\ r_{31}r_{32}r_{33}t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}=A[R|t]\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$ 식 2)

- Focal length: fx = 914.426678, fy= 914.426678
- Principal point: cx = 640.000000, cy = 360.000000
- Distortion: k1 = -0.006462, k2 = 0.852628, p1 = -0.003371, p2= -0.003302
  (k1, k2: radial distortion 계수, p1, p2: tangential distortion 계수)

캘리브레이션이 끝난 다음 ArucoMarker를 측정한 결과 직선거리와 회전각도의 값은 다음과 같습니다.

표 1) ArucoMarker 회전각도 기술통계 요약

	pitch	yaw	roll
mean	6.860151	-0.24269	-6.10117
std	0.373937	0.32947	0.089732

표 2) ArucoMarker 직선거리 기술통계 요약

	x	y	z
mean	-0.03755	0.079574	0.478185
std	0.00018	0.000271	0.000624

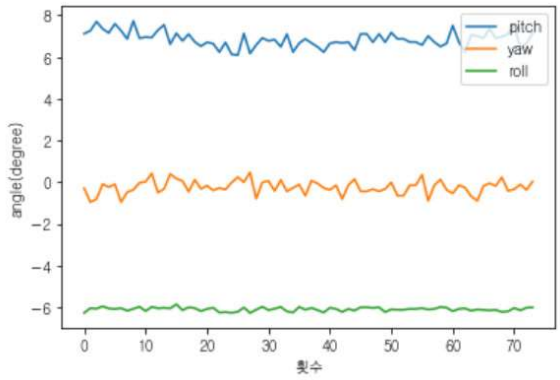

Figure 10 회전각의 변화량 그래프

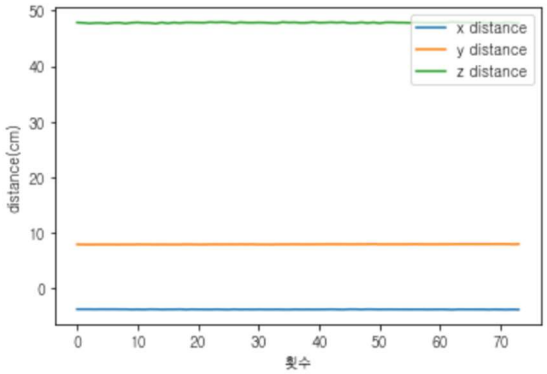

Figure 11 직선거리의 변화량 그래프

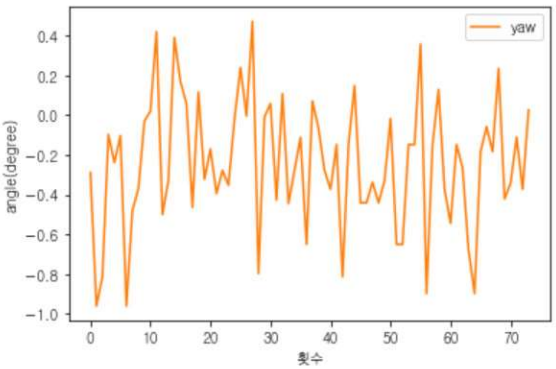

Figure 12 yaw 회전각의 변화량 그래프

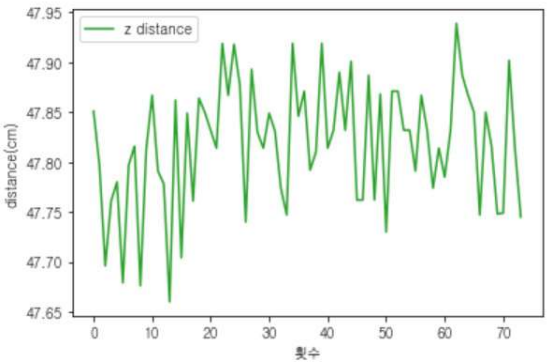

Figure 13 z 축 직선거리 변화량 그래프

이번 측정에서 가장 변화량이 큰 각도는 Pitch였으며 제일 작은 변화량은 Roll로 각도의 범위는 2도와 0.4도, 표준편차는0.3739, 0.0897을 기록했습니다. ArucoMarker을 측정할 때 바닥에 대고 찍었지만 이를 고정하기 위해 손으로 들고 찍는 과정에서 Human factor 에러를 고려하지 못해 발생한 것으로 예상합니다. 다음 실험에서는 이를 고려하여 마커를 정확히 고정할 계획입니다. 직선거리의 변화량은 x, y, z축 모두 0.5cm 내외이며 표준편차도 0.0001~0.0002로 매우 낮으며 안정적으로 일정한 값을 보여주었습니다. 나머지 자세한 데이터셋은 설명서에 첨부되어 있습니다.

3.3 임베디드 시스템과 라즈베리파이 이해

프로젝트를 수행하는데 고성능 컴퓨터를 사용한다면 좋겠지만 비용과 공간적인 문제를 피할 수 없습니다. 그렇기에 해당 목적만을 수행하는 단일 수행체계 컴퓨터, 임베디드 시스템을 구축할 계획입니다. 해당 시스템을 구현하는데 성공한다면 이는 드론이나 다른 기기에 자유롭게 이식할 수 있는 범용성을 얻을 수 있습니다. 우리는 이번 프로젝트를 위해서 영국의 라즈베리 파이(Raspberry Pi) 재단에서 만든 초소형/초저가의 컴퓨터를 이용할 계획입니다. 라즈베리파이는 리눅스 기반 개발 보드이며 기본적으로 오픈소스로 이루어져있습니다. 이는 소스 코드와 설계도가 공개되어있으며, 무제한적인 접근이 가능하고 개인이 자유롭게 수정과 배포가 가능하기에 많은 사람들의 참여와 후기를 참고할 수 있습니다. 프로젝트를 하는데 있어 어려운 점이나 도전사항이 있더라도 오픈소스 포럼을 통해 폭넓은 의견교환과 토론이 가능하며 정보를 얻기 용이하기에 선택했습니다. 이번 프로젝트를 완성한다면 해당 코드를 가지고 다른 곳에도 적용을 해볼 수 있을 것이라 기대하며, 이를 위해 라즈베리파이에 대한 환경 구축과 운용에 대한 이해가 필요합니다.

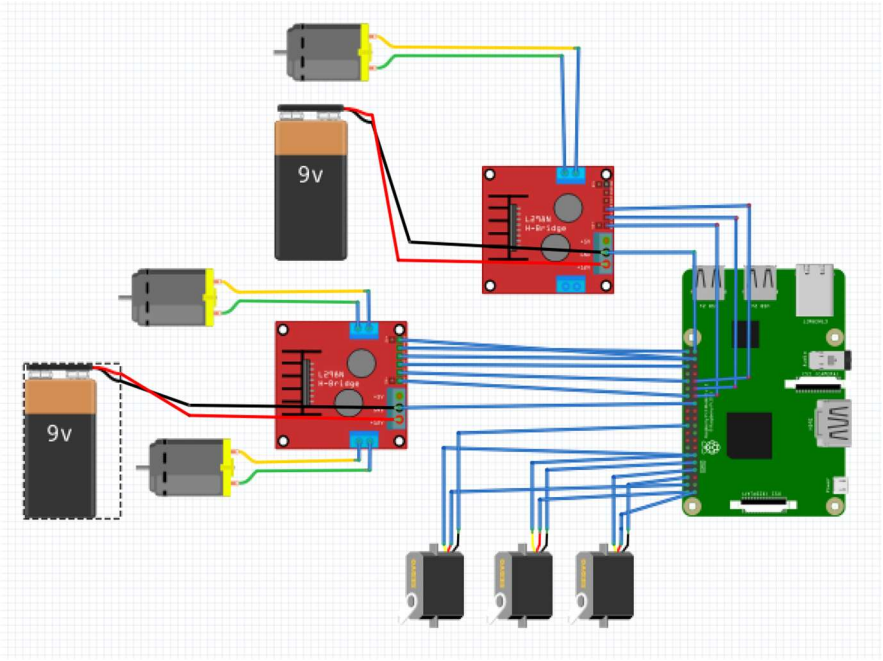

Figure 14 라즈베리파이 모터 연결도

3.3 서보모터와 DC 모터 동작을 위한 GPIO, 시리얼 통신 이해

라즈베리파이에 서보모터 3 개와 DC 모터 3 개를 연결하여 직선운동과 회전운동을 하는 모델 구현할 계획입니다. 이러한 모터를 제어하기 위한 통신으로 라즈베리파이에 내장된 GPIO 와 시리얼통신을 이용할 것입니다. GPIO 는 General Purpose Input/Output 의 줄임말로, 마이크프로세서가 주변장치와 통신을 위해 범용적으로 입력 또는 출력할 수 있는 기능입니다. 각 포트당 3 개의 I/O 레지스터(DDRx, PORTx, PINx)를 가지게 되며, DDRx 는 입출력을 설정하는 레지스터이며 PORT 의 입출력을 결정하게 됩니다. PORTx 는 데이터의 출력을 담당하는 레지스터로 DDRx 가 출력으로 설정되어 있고 PORTx 에 1 을 주게 되면 해당 포트는 HIGH 상태가 됩니다. PINx 는 포트 입력 레지스터로 DDRx 가 입력으로 설정되어있고 PINx 를 읽으면 해당 포트의 상태가 HIGH/LOW 상태인지 읽을 수 있습니다. 라즈베리파이에서는 UART(Universal asynchronous receiver&transmitter) 비동기 시리얼 통신을 위한 단자가 있습니다. 시리얼 통신(Serial Bus, 시리얼 버스)은 연속적으로 비트 단위로 데이터를 전송하는 과정을 말합니다. 이를 위해서 RX(데이터 수신), TX(데이터 송신) 단자가 준비되어 있으며 라즈베리파이에서 카메라 모듈의 정보를 얻기 위해 시리얼 통신을 하며 획득한 정보를 가지고 모터 제어를 위해 값을 전달하게 됩니다. 전체 모델 동작을 위해서 GPIO 와 시리얼 통신을 충분히 이해해 효과적인 모델제어를 수행하는 것이 목표입니다.

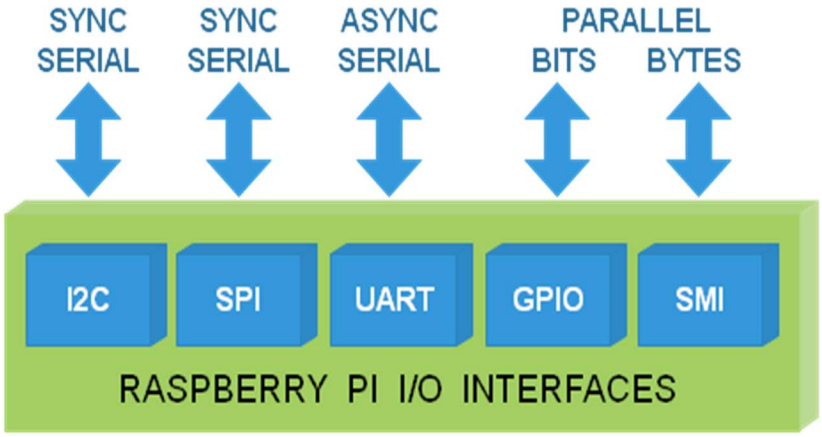

Figure 15 라즈베리파이에 포함된 병렬 I/O 인터페이스

4. 담당자

4.1 문제 해결을 위한 목록

설계목표/문제점	해결방안/예산	분석된 내용 / 학습해야할 내용	담당자
ArucoMarker 의 정확한 검출 / 단안카메라의 정확도	마커의 크기 고정과 카메라 켈리브레이션 / 0 원	ArucoMarker 의 특성 / 단안카메라의 특징과, 카메라 켈리브레이션	정대현
ArucoMarker 추적의 실시간성 / 라즈베리파이의 하드웨어 한계	해상도와 주사율 조절, 흑백 사진 사용 / 0 원	라즈베리파이의 성능과, 효과적인 마커 검출 방법 / 마커 검출을 위한 엣지 탐색 방법	정대현
물체의 6 가지 운동을 표현 / 라즈베리파이 GPIO 보드의 개수 제한	GPIO 확장보드 구입 / 6,400 원	물체 운동을 표현을 위한 6 가지 표현 / 물체의 운동 이해	박유경
카메라 켈리브레이션을 통한 거리 측정 / 인식률 저하	카메라의 초점 수정과 내외부 파라미터 수정 / 0 원	3 차원의 물체를 2 차원으로 이동하는 과정에서 왜곡 현상 / 카메라 켈리브레이션을 위한 보정 과정 학습	정대현
SG-90 서보모터를 이용한 각도 표현 / 라즈베리파이의 PWM 제어 문제	제어를 위한 pigpio 라이브러리 사용 / 0 원	SG-90 서보모터는 각도가 아니라 지정된 위치로 이동 / pigpio 라이브러리의 작동 원리	박유경
DC 모터를 이용한 트랜슬레이션 표현 / DC 모터를 라즈베리파이에 직접 연결 불가능	DC 모터 드라이브와 9V 건전지 추가 구매 / 6,860 원	DC 모터를 연결하기 위해서는 모터드라이브가 필요 / DC 모터 드라이브 회로도 이해	박유경