


[서식2]

		<h2 style="margin: 0;">항공우주종합설계 수행계획서</h2>			
주제명 (가칭)		OpenCV를 이용한 Aruco 마커의 트랜잭션과 앵글 실시간 추적			
지도교수	소속학과	항공우주공학과	성명	이대우	
대표학생	성명	정대현	휴대전화	010-2639-3630	
	E-mail	empcik@pusan.ac.kr	학번/학년	201527137/4학년	
참여학생 (대표학생 제외)	학과	성명	학번	학년	비고
	항공우주공학과	박유경	201725171	4학년	
<p>본인을 포함한 팀원 전원은 항공우주종합설계를 성실히 수행하고자 수행계획서를 제출합니다.</p> <p style="text-align: center;">2021년 04월 30일</p> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div> <p style="text-align: right;">신청인(대표학생)</p> <p style="text-align: right;">지도교수</p> </div> <div> <p style="text-align: right;">(서명)</p> <p style="text-align: right;">(서명)</p> </div> </div> <p style="margin-top: 30px;">항공우주공학과장 귀하</p>					

목차

1. 과제성립의 배경	3p
1.1 4차산업혁명과 이미지 처리 기술	
1.2 드론과 이미지 처리의 접목	
1.3 드론 착륙의 자동화	
2. 기존 문제점 및 새로운 요구사항	6p
2.1 거리를 측정하기 위한 제한적인 방식	
2.2 정해진 크기의 마커를 이용	
3. 설계목표(개념) 및 해결하고자 하는 문제점	8p
3.1 모노카메라의 정확도 개선	
3.2 작동과정의 실시간성 담보	
4. 현실적 제한조건	9p
4.1.A 탑재 컴퓨터의 성능 제약	
4.1.B 확장성 한계로 인한 센서와 추가 I/O 장치 제한	
4.2 구성원의 라즈베리파이 숙련도 문제	
5. 기대되는 효과	11p
5.1 비용 절감, 적은 리소스 사용과 범용성	
5.2 직관적인 인터페이스 개발	
5.3 마커의 손쉬운 추적	
6. 제시하고자 하는 최종 결과물	12p
6.1 작동모델 스케치	
6.2 목표 결과물	
7. 설계목표(개념) 및 문제해결 수행계획	13p
7.1 개발방식	
7.2 개발일정	

1. 과제성립의 배경

1.1 4차산업혁명과 이미지 처리 기술

4차산업혁명의 정의는 명확하지는 않지만 산업에서는 '산업의 자동화'에 큰 방점을 두고 있습니다. 프로세싱 자동화에 있어서 가장 중요한 것은 대상을 구분하는 능력입니다. 목표하는 대상에 대한 검출과 동작수행을 사람이 수동으로 하기보다, 고도로 설계된 알고리즘으로 자동화를 한다면 그 효율은 일반적인 노동에 비교할 수 없을 정도로 상승하며 작동하는 소유자에게 있어도 손쉽게 효율을 극대화 할 수 있기 때문입니다. 프로세싱의 많은 부분을 차지하는 것은 대상에 대한 관찰과 분류입니다. 예로 포스코는 OpenCV와 딥러닝 기술을 통해 사물의 표면 결함 검출장치¹⁾를 개발하였으며 이는 육안과 수작업으로 동작하였던 상태와는 비교도 할 수 없는 결과를 가져올 수 있게 되었습니다.

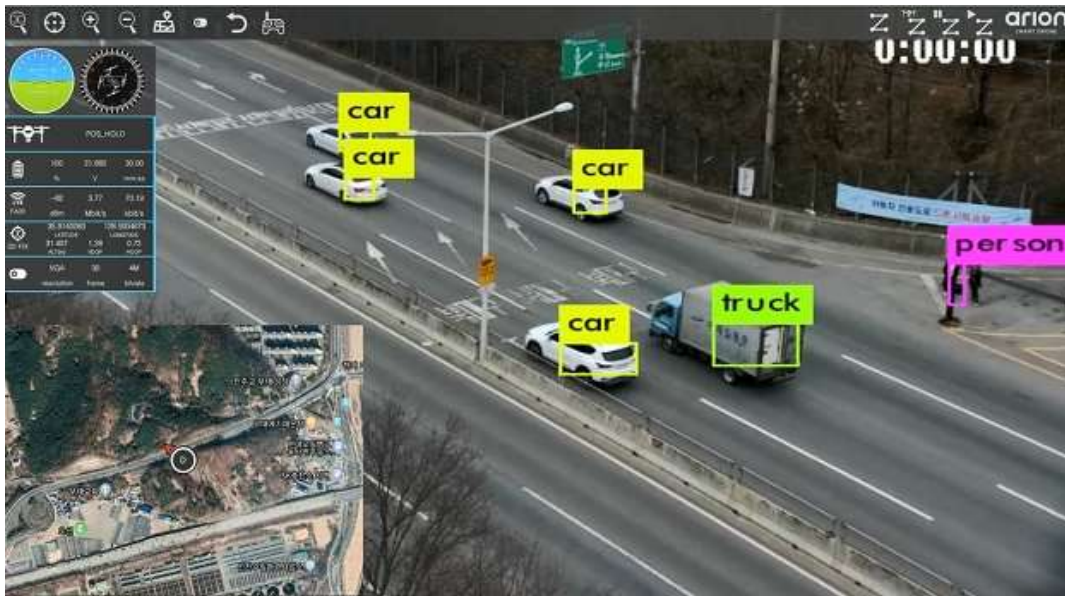


사진1) 드론 순찰 실시간 시연 화면(대구 시설공단)

1.2 드론과 이미지 처리의 접목

최근 크게 각광 받고 있는 산업은 빅데이터, 머신러닝과 항공산업으로 이 둘을 접목한다면 여러 가지 일을 해볼 수 있습니다. 대구시설공단에서는 대구 스마트시티사업 일환으로 딥러닝 기반의 실시간 드론 시범 순찰을 진행하였으며, 기존의 고정식 카메라와 사람이 직접 이동하며 단속하는 이동식 카메라와 달리 LTE통신망을 이용한 드론으로 언제 어디서나 장소와 시간에 구애 받지 않는 시스템을 운영할 수 있게 되었습니다. 단순히 육안으로 구분할 수 있는 사고와 과속 차량 단속 뿐 아니라 낙하물 확인, 적재불량 차량 단속, 도로 파손, 도로 시설물 확인 등 보다 다양한 분야의 처리를 자동화 할 수 있으며 훈련하는 데이터와 알고리즘에 따라서 무수히 많은 일들을 예상 해 볼 수 있습니다.

1) 딥러닝 기반 표면 결함 검출장치 및 방법 <https://patents.google.com/patent/KR101863196B1/ko>

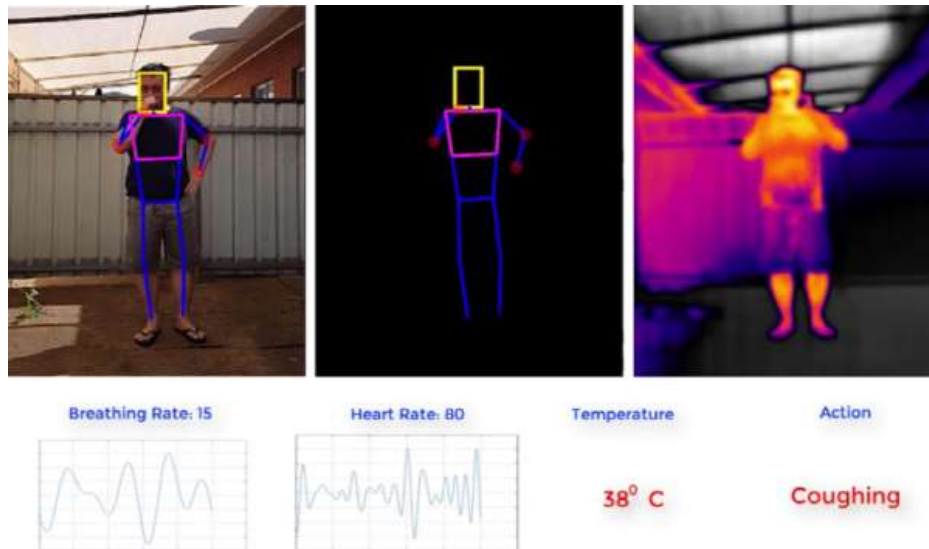


사진2) 생체 정보 원격지에서 확인 (드래간플라이)

COVID-19 감염병이 장기화되고 있는 가운데 캐나다 드론 전문 업체인 '드래간플라이(Draganfly)'는 딥러닝 업체인 '바이탈 인텔리전스'와 협력하여 카메라 네트워크, 드론, 건강 관련 데이터 서비스 등을 활용해 감염병의 조기 발견 및 확산 등 공중 보건 상황을 모니터링할 수 있는 프로젝트를 추진한다고 발표했습니다. 이렇듯 항공산업과 이미지처리 분석을 결합한 사례를 통해 해당 산업군은 시장성과 발전 가능성이 전도유망하며 창의성을 가진다면 여러 가지 일을 해 볼 수 있습니다.

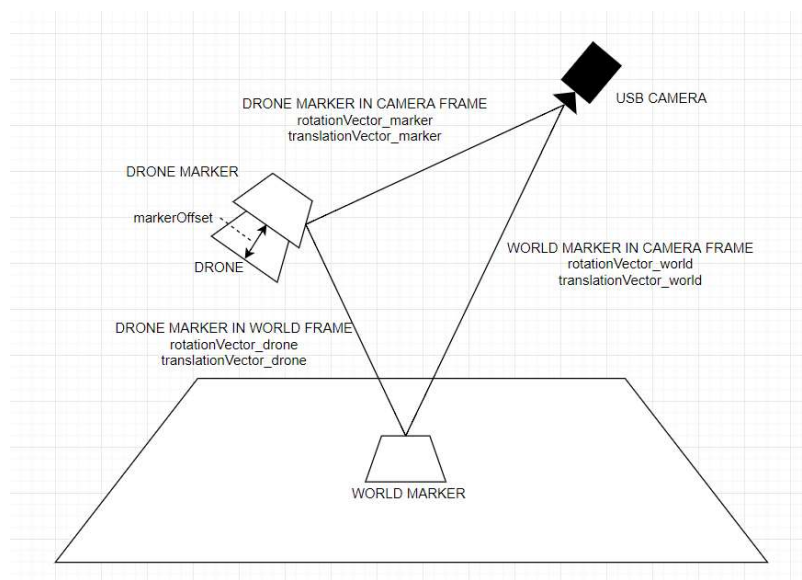


사진3) Calculating pose of Aruco markers wrt a world frame

1.3 드론 착륙의 자동화

순찰, 배달 또는 군사적 목적의 어떤 종류의 드론이던 마지막 동작은 착륙입니다. 가장 보편적으로 드론의 위치를 검출하기 위해서 사용하는 것은 GPS가 있으나 제한사항이 존재합니다. 위치를 찾아가거나 사물을 추적하는데 GPS에 무조건적으로 의존할 경우 위성에서 신호를 받는 과정에서 전리층의 영향, 천체력, 위성 시계, 전파경로, 대류권 영향을 받거나 터널과 같은 외부환경과 차단된 곳과 같이 외부 요소와 불확실성에 따른 자연적인 오차를 피할 수 없게 됩니다. 만약 현장에서 원하는 이미지를 실시간으로 추적할 수 있다면 GPS를 사용할 수 없는 환경이나 또는 GPS를 보조하여 cm단위의 오차로 임무를 수행 할 수 있습니다. 이번 프로젝트의 목표는 드론이 특정한 마커를 검출하고 마커의 위치와 로테이션을 실시간으로 추적하고 이를 추출하는 것에 있습니다.

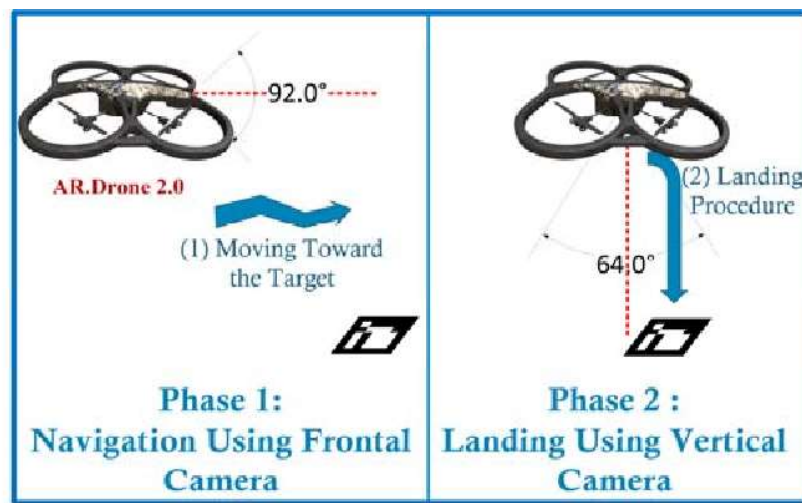


Fig. 1. Various stages of the drone's landing procedure

사진4) 마커의 위치에 따른 드론 제어

포즈 추정은 로봇 탐색, AR(증강현실) 많은 컴퓨터 비전 애플리케이션에서 중요하게 여기는 부분입니다. 해당 프로세스는 실제 환경의 지점과 2D이미지 투영 사이의 대응 관계를 찾는데 기반 합니다. 하지만 일반적으로 모든 것을 처음부터 하기는 어려운 과정이므로 더 쉽게 만들기 위해서 합성 또는 기준 마커를 사용하게 되는데 이 과정을 Rafael Muñoz와 Sergio Garrido가 개발 한 정사각형 기준 마커 감지 라이브러 리인 Aruco Marker²⁾ 기반으로 제작할 계획입니다. Aruco Marker는 넓은 검은 색 테두리와 식별자(id)를 결정하는 내부 이진 매트릭스로 구성된 합성 사각형 마커입니다. 검은 색 테두리는 이미지에서 빠른 감지를 용의하게 하고 이진 코드를 통해 오류 감지 및 수정 기술을 식별 하고 적용 할 수 있습니다.

2) S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. 2014. "Automatic generation and detection of highly reliable fiducial markers under occlusion". Pattern Recogn. 47, 6 (June 2014), 2280-2292. DOI=10.1016/j.patcog.2014.01.005

2. 기존 문제점 및 새로운 요구사항

2.1 거리를 측정을 위한 제한적인 방식

거리측정을 위해서 사용하는 대표적인 장비는 라이다(Light Detection And Ranging, LIDAR)로 레이저 펄스를 쏘고 반사되어 돌아오는 시간을 측정하여 반사체의 위치를 측정하는 장비입니다. 항공, 위성 시스템은 물론 지형측량, 스피드건, 자율주행차량에 주로 탑재를 하며 광범위하게 쓰이나 단점으로는 가격이 비싸며 무게가 증가하게 됩니다.



사진5) 라이다를 탑재한 드론

그 다음으로 많이 쓰이는 거리 측정 방식은 스테레오 카메라를 이용한 방법입니다. 일반적인 모노 카메라로 찍은 사진은 3차원(3D) 물체의 반사광을 2차원(2D) 필름 또는 2차원 센서에 투영시킨 것이기에 이 과정에서 거리 정보가 사라지게 됩니다. 이 문제를 해결하기 위해 두 대의 카메라를 이용하여 한 대의 카메라에서 얻지 못하는 거리 정보를 얻을 수 있는 장점으로 좌측 카메라 영상에서의 점과 우측 카메라 영상에서의 점이 서로 다른 곳에 위치하게 되며, 이와 같은 각 영상에서의 차이로부터 거리 정보 획득하는 방법이 있습니다. 하지만 이를 사용하기 위해서 두 카메라간의 거리가 충분해야하며 특징점을 검출하기 위해서는 낮은 해상도임에도 많은 연산이 필요하다는 단점이 있습니다.

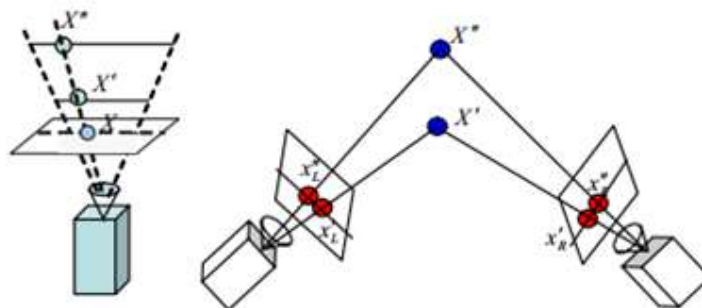


사진6) 스테레오 카메라를 이용한 위치 검출

2.2 정해진 크기의 마커를 이용

고가의 장비와 여러 장치 없이 모노카메라로 마커 위치를 획득하기 위해서 사전에 정해진 크기의 마커를 이용하고자 합니다. 특정거리에서의 픽셀 크기를 획득할 수 있다면 나중에 마커의 크기를 변경하더라도 코드 한 줄을 수정하는 것으로 대응 할 수 있을 것으로 기대합니다.

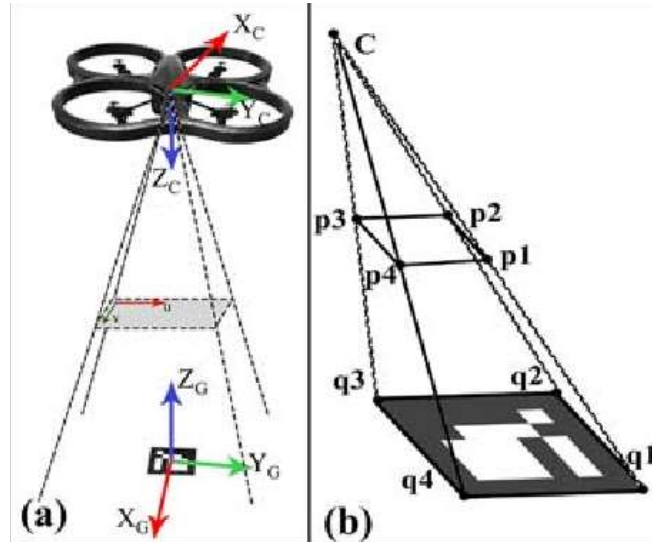


사진7) 모노카메라를 이용한 마커 위치 획득

모노카메라의 거리 추정문제는 기존 차량 충돌방지센서에서 활발하게 진행되는 분야로 도로의 폭은 법정기준을 따르고 어디에서나 같다는 것에 착안하여 거리를 추정하는 방식이 널리 사용되고 있습니다. 마찬가지로 일정한 크기가 담보되는 기준을 가지고 있다면 마찬가지로 방법으로 접근해나갈 수 있을 것으로 기대합니다.

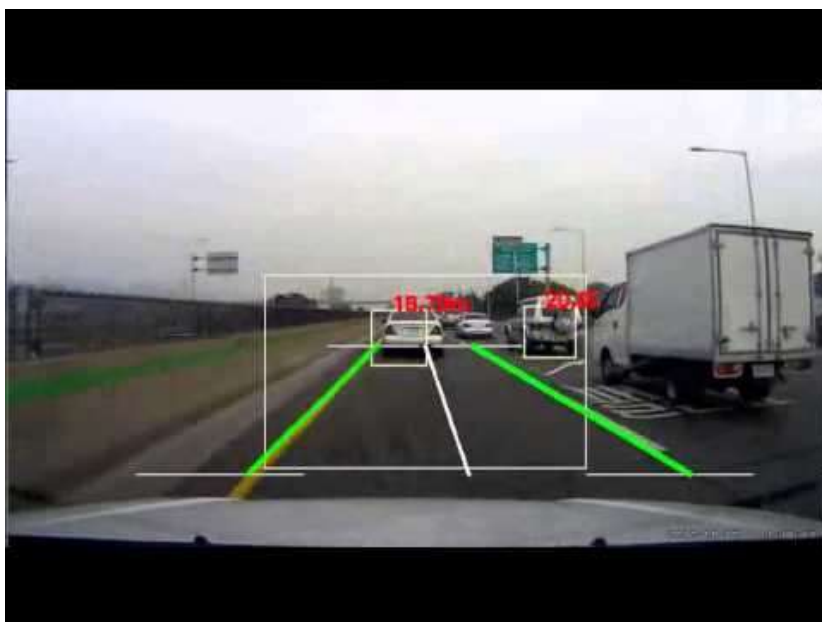


사진8) 도로 길이를 기준으로 하여 거리 추정

3. 설계목표(개념) 및 해결하고자 하는 문제점

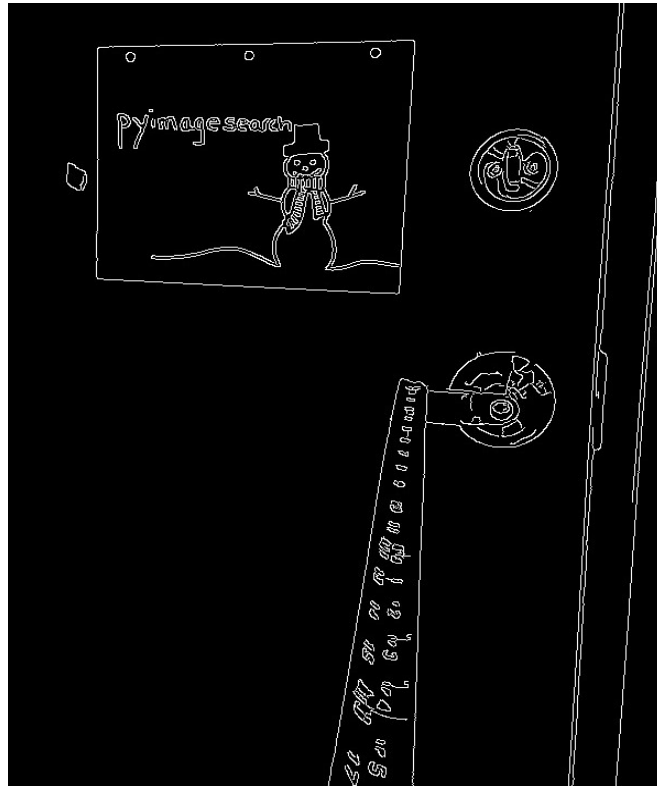


사진9) 가장자리 감지를 적용하여 마커를 찾는 과정

3.1 모노카메라의 정확도 개선

기존의 거리 측정을 위해 많이 사용하는 방식인 라이다와 적외선센서, 스테레오 카메라를 사용하지 않고 일정한 크기의 마커와 모노카메라로 수행하는 것이 목표였으나, 한계점은 분명히 존재합니다. 사전에 정해진 크기의 마커라고 해도 특징점을 제대로 검출하지 못하거나, 픽셀크기의 왜곡이 발생할 경우 거리를 정확하게 구하지 못할 가능성을 배제할 수 없으며 실제 상황에서 거리를 잘 못 구한다는 것은 사고로 직결되는 문제이기에 정확도의 개선에 많은 노력이 들어갈 것으로 예상합니다.

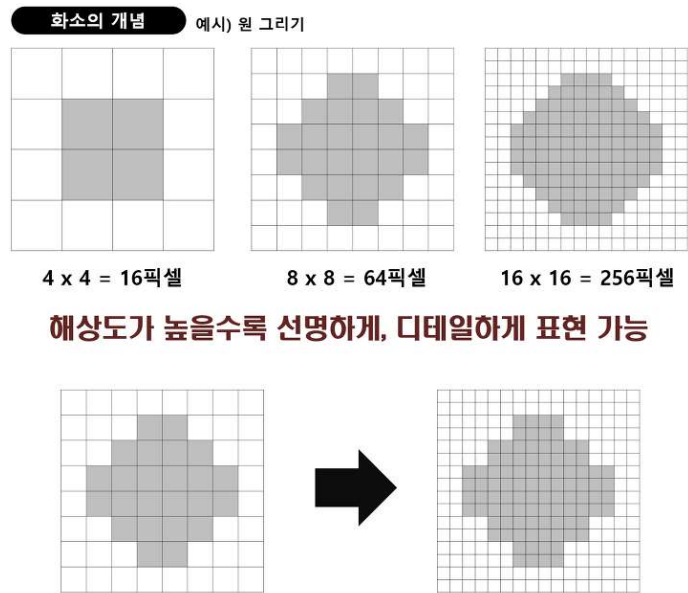


사진10) 화질에 따른 처리량 예시

3.2 작동 과정의 실시간성 보장

이번 프로젝트에서는 마커의 실시간 추적 까지만 연장선으로 마커를 검출하고 드론의 자동 착륙하는 것을 염두에 두고 있습니다. 그렇기에 모든 동작은 지연시간 없이 최대한 실시간으로 이루어져야 한다는 조건이 들어가게 됩니다. 만약에 고해상도의 카메라를 사용한다면 민감도를 높여 특징점을 검출하는데 이득을 볼 수 있겠지만 그만큼 처리량이 많아져 지연이 있을 수 있으며 반대로 저해상도로 처리를 한다면 처리 과정에서의 지연은 줄어들지 몰라도 민감도에 영향을 줄 수 있습니다. 해당 부분을 고려하여 민감도를 만족하며 무시해도 좋을만한 지연시간을 구축하는 문제점이 있습니다.

4. 현실적 제한조건

4.1.A 탑재 컴퓨터의 성능 제약

필요하다면 해당 시스템을 드론에 탑재할 수 있어야 한다는 점을 들어 사용하는 컴퓨터는 초소형/초저가 컴퓨터인 라즈베리파이4 Model B³⁾를 이용할 계획으로 일반적인 고성능 x86-64 CPU가 아닌 ARM기반의 Cortex-A72 CPU가 들어간 브로드컴 BCM2711 SoC를 사용하게 됩니다. 시스템의 동작이 실시간으로 이루어져야 한다는 것을 감안하면 시스템의 부하는 최대한 줄이고 처리 과정은 간결해야 할 것입니다. 그렇기에 다른 센서를 이용하거나 스테레오 카메라로 입체감과 거리를 구현하는 대신 모노 카메라로 전체 과정을 수행할 계획입니다.

3) https://elinux.org/RPi_HardwareHistory

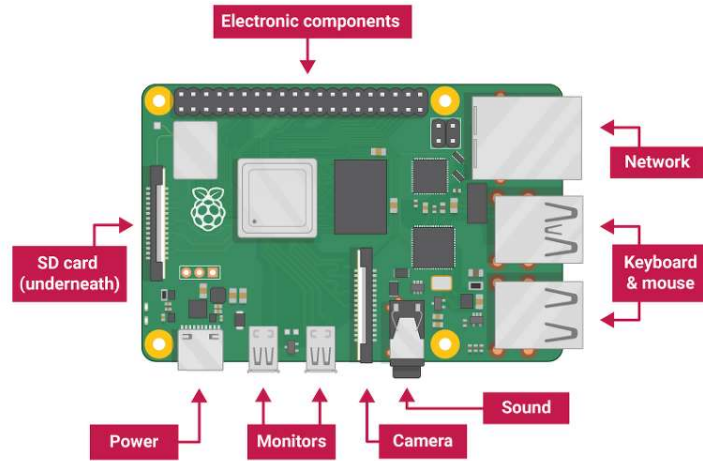


사진11) 라즈베리파이 구성도

4.1.B 확장성 한계로 인한 센서와 추가 I/O 장치 제한

라즈베리파이라는 초소형미니PC의 특성상 많은 센서와 장치를 연결할 수 없다는 단점이 따라오게 됩니다. 모노카메라 이외에 다른 보조적인 장비를 달아 개발의 편의성을 향상하는 것은 어려우며 추후 개발과정에서 설계를 변경하는 것도 제한적인 범위 내에서 이루어지게 됩니다.

4.2 구성원의 라즈베리파이 숙련도 문제

구현하고자하는 프로젝트의 사용 환경과 목표를 검토해보았을 때 가장 적합한 환경을 라즈베리파이 PC를 이용한 것으로 결정하였으나, 프로젝트에 참여하는 인원 모두 라즈베리파이를 다룬 경험이 없습니다. 일반적으로 이용하는 윈도우나 맥OS대신 라즈베리파이에서 구동하는 OS는 라즈베리 재단에서 제공하는 라즈베리 파이 OS(이하 라즈비안)를 사용하거나, 커스텀 리눅스를 사용해야 합니다. 처음부터 라즈베리파이에서 시작하는 것이 아니라 윈도우와 맥 환경에서 개발한 프로젝트를 라즈베리파이로 이식하는 것으로 계획을 세워두었기에 이식하는 과정에서 예기치 못한 오류나 제한 사항이 있을 것으로 예상됩니다. 이 부분에서 개발 기간의 조정이 필요해 보입니다.

5. 기대되는 효과

5.1 비용 절감, 적은 리소스 사용과 범용성

일반적으로 자세 제어를 위해서는 자이로스코프나 라이다와 같은 여러 센서와 부가적인 장비가 요구되었으나 OpenCv 라이브러리를 이용한다면 비교적 저렴한 웹캠과 간단한 마커 간단한 장치 만으로도 드론 전체 자세제어와 상대위치 추적이 가능하며 이는 설계 도구와 프로세스 비용 절감의 효과를 기대할 수 있습니다.

사용하는 컴퓨터는 라즈베리파이와 개발 언어는 범용적인 파이썬으로 이루어져있기 때문에 한번 개발이 끝난 것이라면 추후 다른 것을 연계해서 코드 이식성을 담보할 수 있으며 향후 다른 프로젝트 연계에 있어서도 손쉬운 것으로 예상합니다.

5.2 보다 직관적인 인터페이스 개발

기존의 드론의 정밀한 제어를 위해서 드론 조종기와 조종 스틱을 엄지와 검지로 잡는 핀치 컨트롤(Pinch Control), 또는 엄지로 꺾 누르는 엄지 컨트롤(Thumb Control)등 여러 가지 방법으로 널리 쓰이고 있으나, 각 성향에 맞는 조정방법과 장단점이 있습니다. 허나 이런 기본적인 지식이 없는 초보자들에게 있어 시중에 유통되는 드론 조종기는 접근성이 떨어지고 단기간에 숙련이 어려운 점이 있습니다. 이번 마커추적 시스템을 이용하면 범용적인 측면에서 쓰일 것으로는 힘들어보이나 제한된 환경에서 정밀한 드론 조종을 시도해볼 수 있을 것으로 예상합니다.

5.3 마커의 손쉬운 추적

이번 프로젝트의 실질적인 목표라고 할 수 있는 마커의 손쉬운 추적입니다. 해당 프로젝트가 기대하는 수준을 만족한다면 카메라(또는 드론)의 위치와 마커의 상대위치 그리고 마커의 포즈까지 한 프로세스에 실시간 추적이 가능하며 향후 드론이 마커를 감지하고 착륙하는 과정에서 유용하게 쓰이는 알고리즘이 될 것으로 기대하고 있습니다.

6. 제시하고자 하는 최종 결과물

6.1 작동모델 스케치, 목표 결과물

해당 프로젝트를 문장으로 표현한다면 OpenCv와 'Aruco Marker'을 이용한 마커의 실시간 자세 추적입니다. 라즈베리파이에 연결한 카메라(웹캠)으로 Aruco Marker를 검출하며 화면으로 실시간 전송을 하게 되며 해당 시스템의 스케치는 다음과 같습니다.

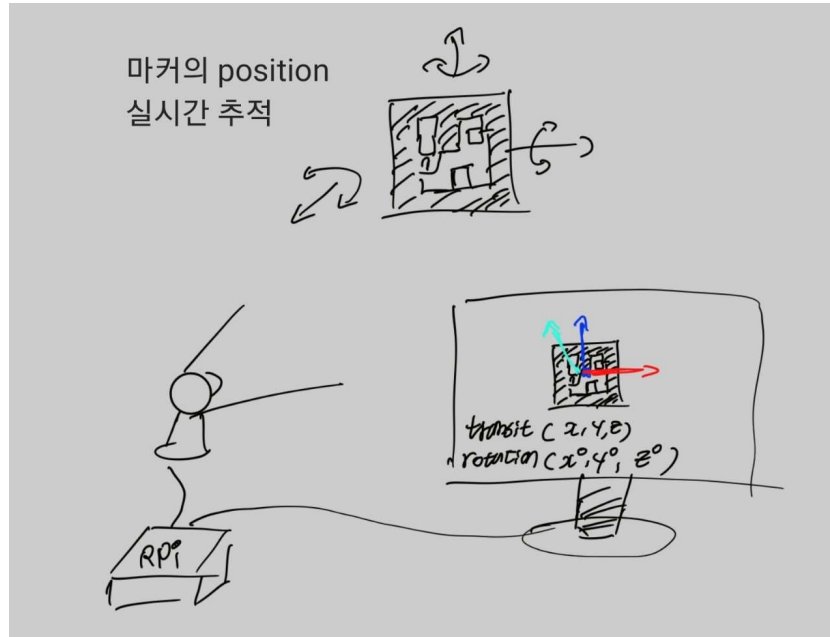


사진12) 작동 스케치.1 마커 - 웹캠 - 화면 출력

그렇게 얻은 3축과 3좌표 이렇게 6가지의 값을 가지고 3개의 모터와 3개의 서보모터에 해당 값을 전달하여 모형비행기의 자세 제어를 실시간으로 구현하고자 합니다.

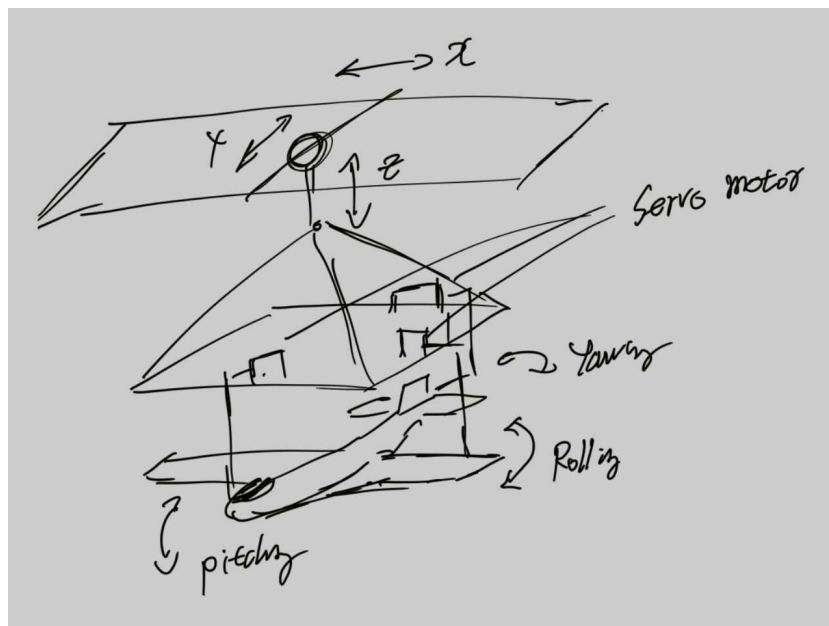


사진13) 작동 스케치.2 마커 포지션 기반 모형비행기 자세제어

7. 설계목표(개념) 및 문제해결 수행계획

7.1 개발 방식

개발방식은 프로토타이핑 모델을 적용하여 최대한 빠르게 시스템을 구현한 다음 계속해서 수정하는 과정을 도입할 예정입니다. 개발 일정에 따라서 최대한 움직이겠지만 가능하다면 첫 모델을 빠르게 만들 것입니다.

7.2 개발 일정

4월	가상환경 구축, 프로젝트 계획
4월 30일 설계과제 수행계획서 제출	
5월	Aruco Maker 트랜잭션 추적 설계
1주차 : Aruco Marker의 x, z 좌표 추적 구현 2~3주차 : Aruco Marker의 y 좌표(마커 거리) 추적 구현 4주차 : x, y, z 좌표 추적 민감도 검토	
6월	Aruco Maker 로테이션 추적 설계
1주차 : Aruco Marker의 3축 draw 구현 2~3주차 : Aruco Marker의 3축 각도 실시간 추적 구현 4주차 : x, y, z 3축 각도 추적 민감도 검토	
7월	라즈베리파이 이식
1주차 : 설계 코드 라즈베리파이 호환성 검토 2~3주차 : 서보모터, 모터 구동 연계 확인 4주차 : 시스템 호환성 점검	
8월	중간보고서 작성, 시스템 모델 구축
1주차 : 실제 모델 구축 2~4주차 : 동작 모델링 테스트 8월 25일 설계과제 중간보고서 제출	
9월~11월	시스템 개선, 최종 보고서 작성
최종 보고서 작성 11월 22일 설계과제 최종보고서 제출 11월 26일 설계과제 최종 발표 평가	