

캐위성을 이용한 사진 촬영 기반
거리 측정 시스템 개발

-원근법의 기초에 근거한 거리 측정 시스템-

2021 년 12월

충북과학고등학교

최성호 손상원 이용호

국문초록

주제어: 캔위성(CanSat), 원근법, 고도측정,

기존 고도 측정을 위해서는 고도센서 또는 GPS 센서를 주로 이용해 왔다. 고도센서의 경우 기압의 변화를 기반으로 고도를 예측하며 GPS 센서는 인공위성과 그 센서의 위치관계를 기반으로 한다. 이 때문에 고도센서는 강풍 등의 요인에 의해 쉽게 오차가 발생하며 기존 기압에 대한 정보가 없는 외계행성에서 사용 불가능하다. GPS 센서 또한 인공위성이 없는 외계행성에서 사용 불가능하다는 한계가 있다. 이를 극복하고자 우리는 멀리 있는 것은 작게, 가까이 있는 것은 크게 보이는 원근법에 기초하는 사진 촬영을 통한 고도 측정 시스템(이하 IBDiMS)을 제안한다. 피사체의 크기를 조절해가며 1m ~ 5m 수평 상황과 5m ~ 15m 수직 상황에서 사전실험을 진행하며 오차를 보정하고 정확도를 높이고자 했다. 최종적으로 캔위성을 발사하여 GPS 센서와 계산한 높이를 비교하여 IBDiMS의 정확성을 보였다. 캔위성 발사 시 약 150m 상공에서 캔위성이 머물며 이미지 데이터를 송신해왔고 이를 계산하여 실시간으로 GPS와 IBDiMS로 측정한 고도를 비교하여 의미 있는 결과를 내놓을 수 있었다.

목차(국문: 바탕, 굵게, 16포인트)

1. 서론(국문: 바탕, 굵게, 12포인트)

1.1 연구 동기(국문: 바탕, 11포인트)

2. 이론적 배경(국문: 바탕, 굵게, 12포인트)

2.1 캔위성

2.2 고도센서

3. 연구 방법 및 절차(국문: 바탕, 굵게, 12포인트)

3.1 거리 계산식의 수학적 유도

3.2 Python으로 계산식 표현

3.3 수평 상황에서의 실험 진행

3.4 이미지 왜곡으로 인해 발생하는 오차 보정

3.5 수직 상황에서의 실험 진행

3.6 실제 캔위성 발사

4. 연구 결과

4.1 3.3에 대한 결과

4.2 3.5에 대한 결과

4.3 실제 캔위성 발사에서 알고리즘의 효용성 검증

5. 결론 및 제언

6. 참고문헌

[여기에 입력]

캔위성을 이용한 사진 촬영 기반 거리 측정 시스템 개발

- 원근법의 기초에 근거한 거리 측정 시스템 -

충북과학고등학교

최성호, 손상원, 이용호

I. 서론(국문: 바탕, 굵게, 14포인트)

1.1 연구 동기(국문: 바탕, 굵게, 11포인트)

기존 고도 측정을 위해선 고도센서 또는 GPS 센서를 사용하며 이 둘은 각각 다음과 같은 이유 때문에 모든 상황에서 믿을 수 있는 정보가 아니라는 단점이 존재한다. 먼저 일반적인 고도센서의 경우 그 고도의 측정을 고도 간 압력차이를 기반으로 하는데 이는 강풍 등의 요인에 의해 방해받거나 특히 대기 정보가 없는 외계 행성에서의 사용이 불가능하다. 또한 GPS 센서의 경우에도 GPS 위성과 그 센서 간의 거리를 계산하는 방식으로 고도를 계산하는데 이 또한 GPS 위성이 존재하지 않는 외계 행성 및 위성 등에서 사용하지 못한다.

이러한 기존 고도 측정 방식의 한계를 느껴 우리는 새로운 방식의 고도 측정 방식을 설계해 보았다. 기타 위성이나 통제가 불가능한 변인을 대상으로 고도를 측정하는 것이 아닌, 새로운 방식의 고도 측정 시스템을 설계했다. 멀리 있는 것은 작게, 가까이 있는 것은 크게 보인다는 원근법의 기초 이론에 근거하였다.

II. 이론적 배경(국문: 바탕, 굵게, 14포인트)

2.1 캔위성(CanSat)

캔위성(Cansat)은, 1998년 스탠퍼드대학의 로버트 트윅(Robert Twiggs) 교수에 의해 제안된 교육용 위성모사 프로그램에서 사용한 위성으로, 위성의 역할을 모사하는 구조계, 전력계, 데이터 처리계, 통신계, 탑재체 등의 기능을 음료수 캔(예: 콜라 캔) 내에 구현한 데서 명명(Cantsatellite)의 유래를 찾을 수 있다. 발사 및 임무운용에 필요한 위성구성요소를 매우 단순하게 구성하여, 고공에서 짧은 시간 동안 위성처럼 운용하는 것이 특징이다. 캔위성은 실제 위성처럼 우주로켓에 실려 지구궤도에 투입되는 것을 목표로 하지는 않는다. 기구(balloon), 드론, 소형 과학로켓 등을 활용하여 상공 수백 미터 고도까지 캔위성을 올려준 후, 캔위성을 분리하고 캔위성에 달린 낙하산을 전개하여 캔위성이 서서히 고공에서 낙하하는 동안 마치 위성처럼 사전에 계획된 일련의 임무를 수행하는 방식을 취한다.¹

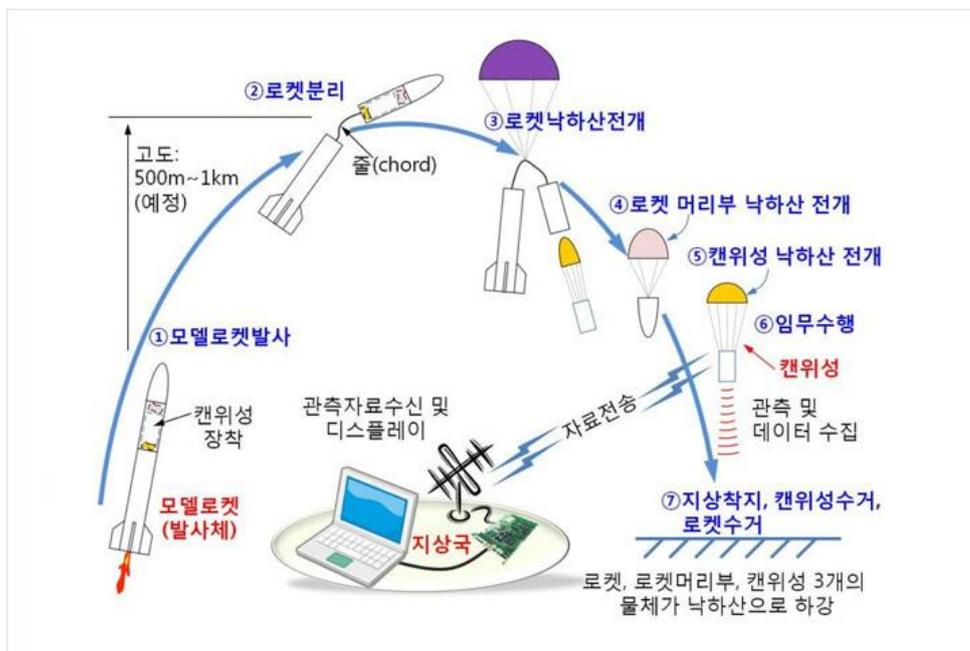


Fig 1 모델로켓을 이용한 캔위성 발사 및 운용 개념도

2.2 고도센서

BMP280으로 대표되는 고도센서들의 특징은 고도측정의 수단으로 기압의 변화를 이용한다는 데 있다. 이러한 방법은 일반적인 상황에서는 잘 들어 맞지만, 빠른 속도로 낙하하거나 강풍을 만나는

¹ http://cansat.kaist.ac.kr/04_01.php

등의 환경에서는 오차가 발생할 확률이 매우 높아진다. 더불어 지구와는 대기 조성 및 환경이 상이한 외계 행성의 경우 사용할 수 없다는 단점 또한 존재한다.

III. 연구 방법 및 절차

3.1 거리 계산식의 수학적 유도

멀리 있는 것은 크게, 가까이 있는 것은 작게. 먼저 이 정성적인 말을 정량적으로, 수식으로 표현하고자 했다. 먼저 본 실험에 활용할 사용할 카메라 즉 캔위성에 탑재된 카메라의 화각을 측정하였고 $\omega_0 = \frac{A_0}{r_0^2}$ 피사체의 입체각을 다음과 같이 계산하였다 $\phi = \frac{A}{h^2}$. 이에 피사체와 전체 사진의 비율을 $\alpha = \frac{\phi}{\theta_0} = \frac{A}{h^2\omega_0}$ 로 나타내어 결국에 캔위성의 고도, 피사체와 카메라 사이의 거리를 $h = \sqrt{\frac{A}{\alpha\theta_0}}$ 라 나타낼 수 있게 된다.

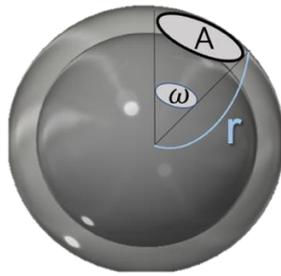


Fig 2 시스템 설계 이해를 돕기 위한 사진

3.2 Python으로 계산식 표현

Python으로 위 수학적식을 구현하여 코드를 만들었다. 이미지 속 피사체와 배경과의 구분은 색상을 기준으로 하였다.

```
img_color= cv2.imread(pic_file)
img_except = cv2.imread(pic_file)
img_blur = cv2.bilateralFilter(img_color, 20, 100, 100)
# print('shape: ', img_color.shape)
h, w = img_color.shape[:2]
totalPixel = h*w

img_hsv = cv2.cvtColor(img_blur, cv2.COLOR_BGR2HSV)
lower_blue1 = (e-ranges, s-ranges, v-ranges) # hsv 이미지에서 바이너리 이미지로 생성, 적당한 값 30
upper_blue1 = (e+ranges, 255, 255)
print(lower_blue1, upper_blue1)
#ret, binary = cv2.threshold(img_gray, 220, 255, cv2.THRESH_BINARY)
img_mask = cv2.inRange(img_hsv, lower_blue1, upper_blue1)
print(img_mask.shape, img_color.shape)

img_res = cv2.bitwise_and(img_color, img_color, mask = img_mask)

contours, _ = cv2.findContours(img_mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

print(len(contours))

for i in range(len(contours)):
    cnt = contours[i]

    if area < cv2.contourArea(cnt):
        area = cv2.contourArea(cnt)
        index = cnt

try:
    epsilon1 = 0.01 * cv2.arcLength(index, True) # 외곽선 길이
    approx1 = cv2.approxPolyDP(index, epsilon1, True)

    cv2.drawContours(img_color, [approx1], 0, (255, 255, 0), 2)
```

[여기에 입력]

3.3 수평 상황에서의 실험 진행

알고리즘의 효용을 알아보기 위해 실험실 환경에서 벽면에 피사체를 부착시킨 채 캔위성의 카메라를 수평으로 움직이며 수식의 효용을 검증하였다. 지상국 프로그램은 구현되어 있는 상태였으며 실시간으로 이미지를 통해 거리가 산출됨을 검증할 수 있었다.

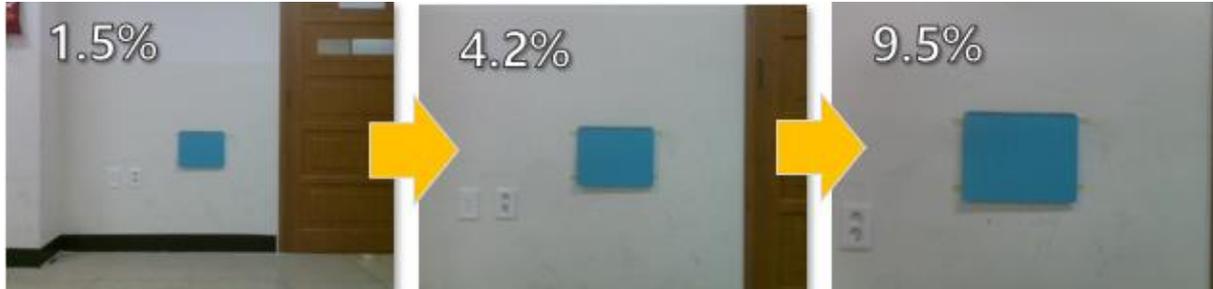


Fig 3 수평상황에에서 타겟과 캔위성 카메라가 가까워질 때 사진 속 피사체의 비율이 증가함을 보이는 사진



Fig 4 수평상황 실험에서 캔위성을 흔들리지 않게 의자에 적재해 이동시키며 피사체를 찍는 모습

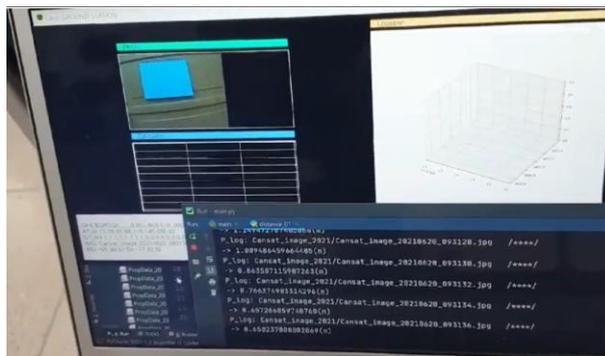


Fig 5 구현한 지상국 프로그램의 모습. 좌상단은 실시간으로 송출받는 이미지, 우상단은 캔위성의 3축 위치를 나타낼 그리드, 좌하단은 실시간으로 촬영하는 이미지의 정보, 우하단은 Python으로 계산한 거리를 실시간으로 송출중이다.

[여기에 입력]

3.4 이미지 왜곡으로 인해 발생하는 오차 보정

타겟의 크기가 일정하더라도 이미지에서 타겟이 중앙에 위치하는지, 말단에 위치하는지, 즉 위치에 따라 같은 피사체라도 이미지 속 픽셀의 차이가 발생할 수 있다. 이를 정 중앙과 타겟의 무게 중심과의 거리에 따라 가중치를 부여하는 방식으로 해결하고자 했으며 각 가중치의 수치는 실험을 통해 결정하였다. 실험은 실험실 환경에서 진행되었으며 이미지의 각 위치에(좌상단, 우하단 등) 타겟을 위치하고 그 픽셀의 수를 비교하는 방식으로 진행되었다.



Fig 6,7 좌측 사진은 실제 캔위성에서 촬영한 사진으로 왜곡이 존재함을 확인할 수 있다. 우측 사진은 인공위성으로 촬영한 사진이며 왜곡이 존재하지 않음을 확인할 수 있다.

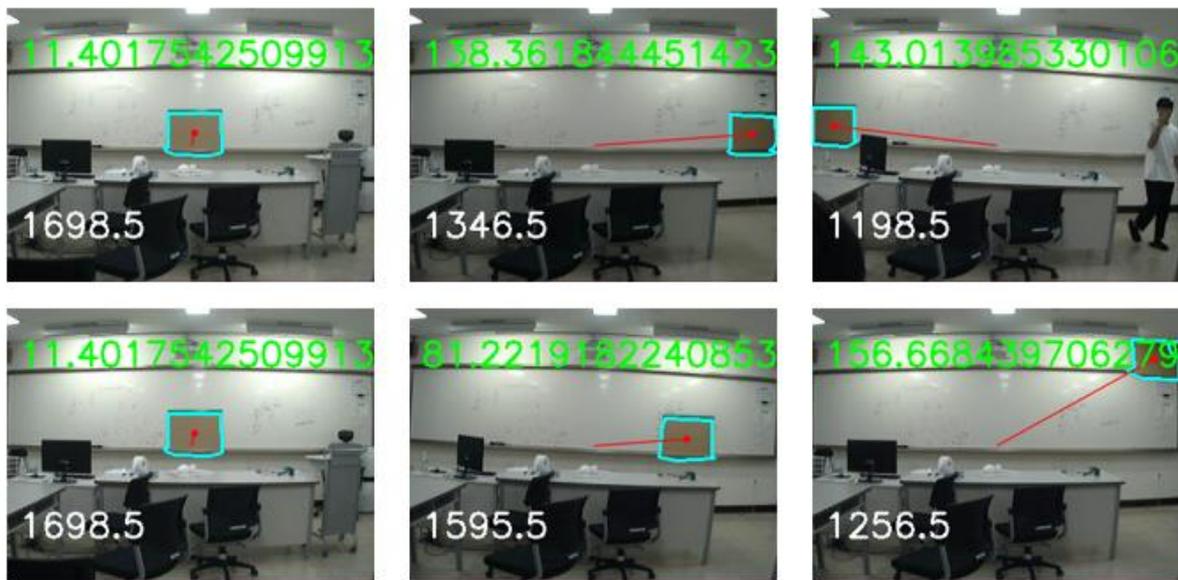


Fig 8 좌하단의 흰색 숫자는 타겟의 현재 픽셀을 나타내고 상단의 녹색 숫자는 카메라의 중심과 피사체의 무게중심 사이의 거리를 나타낸다.

[여기에 입력]

3.5 수직 상황에서의 실험 진행

3.5층고의 교내 외부계단에서 캔위성을 줄에 매달아 일정 거리만큼 늘어뜨리면서 실거리와 계산 거리를 비교하는 실험과 캔위성에 낙하산을 매달아 실제 거리 송출이 가능한지를 관찰하는 두가지 실험을 연이어 진행했다. 실험 1에서는 밧줄에 50cm 간격으로 표시 해 둔 후 이를 단위로 하여 단위거리만큼 늘어뜨린 후 이미지 데이터를 취득하는 방식으로 실험이 진행되었다. 실험 2에서는 반복적으로 낙하산을 매달은 캔위성을 4.5층에서 낙하시키며 낙하산의 성공적인 작동 여부와 낙하시 취득하는 이미지에서 성공적으로 타겟을 추출해낼 수 있는지 등을 확인하였다.

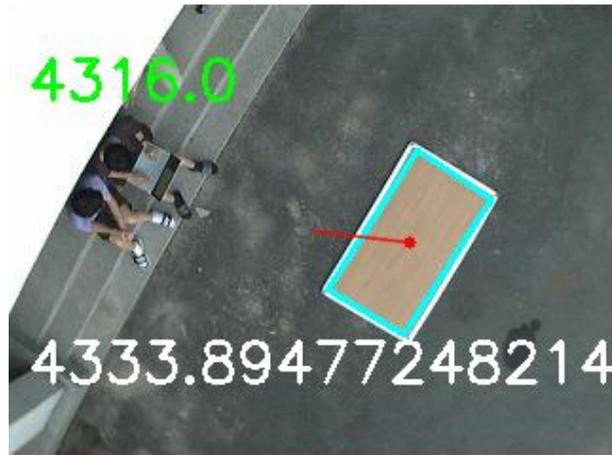


Fig 9 녹색 숫자는 보정 전 타겟의 픽셀 흰 숫자는 보정 후 타겟의 픽셀의 수를 나타낸다.



Fig 10 낙하산이 성공적으로 작동함을 확인 할 수 있었다.

3.6 실제 캔위성 발사

발사는 2021 캔위성 경연대회(CanSat Project)를 통해 2021년 8월 전라남도 고흥 한국항공우주연구원 항공센터에서 이루어졌다.



Fig 11 캔위성이 약 상공 200m에서 촬영한 사진. 이미지 속 붉은 점은 타겟의 무게중심이며 빨간 선으로 무게중심과 카메라의 중앙을 이어주고 있다.

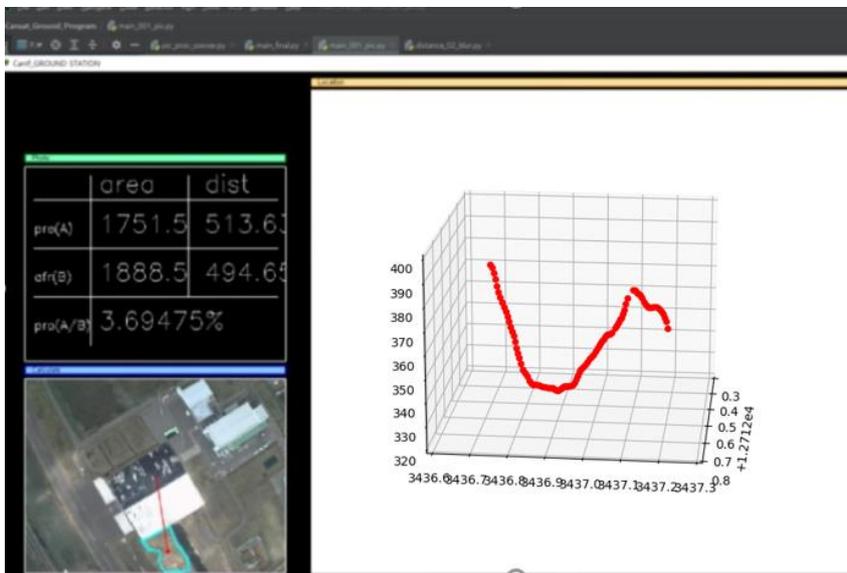
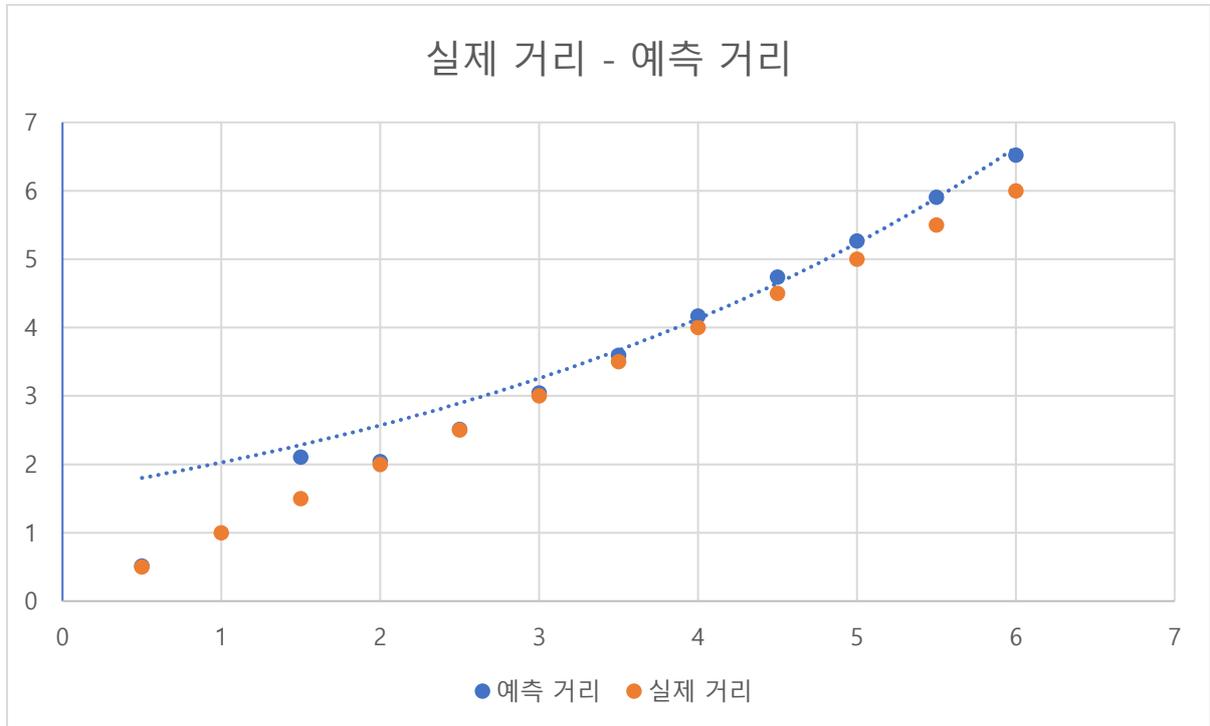


Fig 12 실제 발사 시 사용했던 지상국 프로그램의 모습이다. 우측 그리드는 캔위성의 궤적을, 좌상단의 표는 보정 전, 후 타겟의 픽셀 수와 그로 인해 계산된 거리 및 보정율을 나타낸다. 좌하단의 이미지는 캔위성에서 송출되어 오던 이미지이다.

[여기에 입력]

IV. 연구 결과 (국문: 바탕, 굵게, 14포인트)

4.1 3.3에 대한 결과



거리가 가까워짐에 따라 오차가 크게 발생하였다. 이는 평면을 구면으로 근사하는 과정에서 발생하는 오류로 판별되어 이후 3.4의 오차 보정 과정에서 일련의 수학적 과정을 Python 코드에 추가하여 보정하였다.

4.2 3.5에 대한 결과

다음은 지상국 프로그램에 추가된 표 및 실제거리가 5m일 때 촬영한 사진과 계산해낸 값들이다.

	area	dist
pre(A)	4250.0	5.019
dfr(B)	4279.9	5.001
pro(A/B)	0.34991%	

Fig 12 좌측 상단부터 보정 전 픽셀(영역), 보정 전 거리,(m) 보정 후 픽셀, 보정 후 거리, 보정율을 나타내는 표이다.

[여기에 입력]

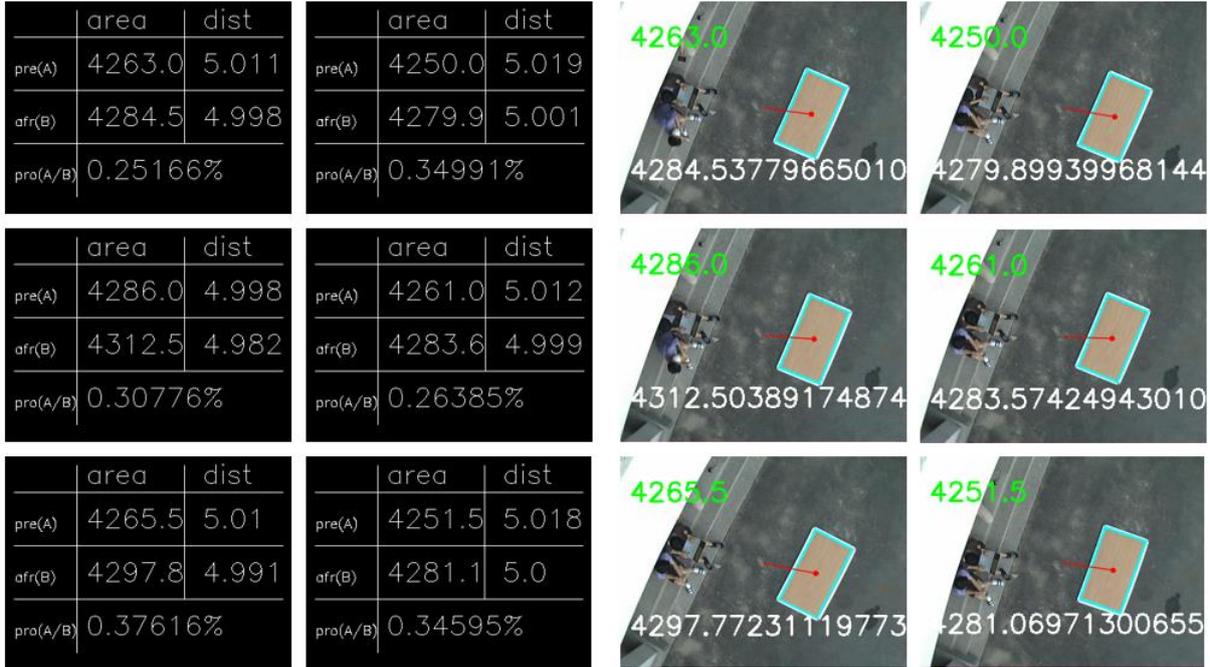
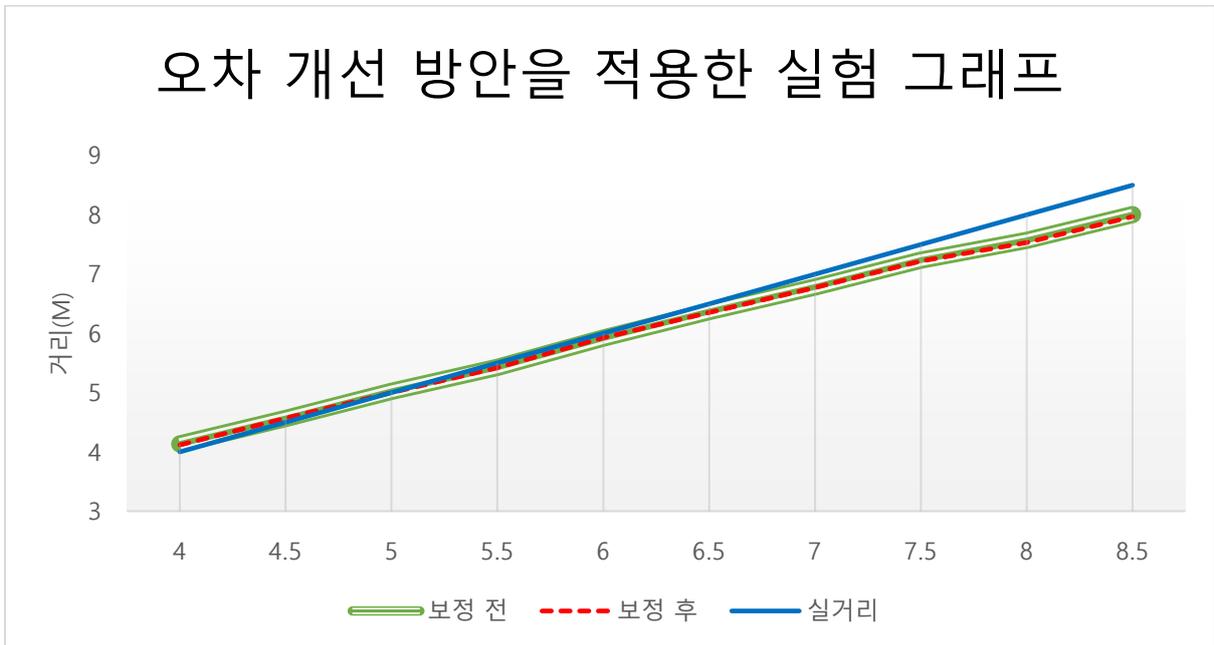


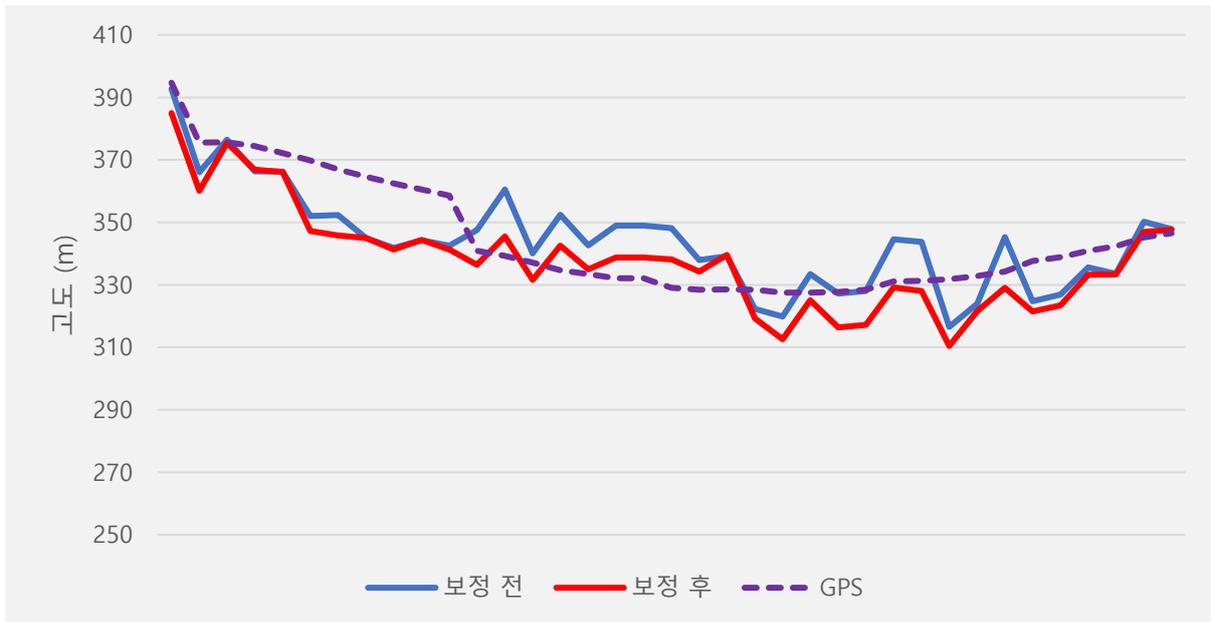
Fig 13 우측 이미지는 캔위성에서 실제 거리 5m 상황에서 촬영한 사진이고 좌측은 각 촬영된 시점의 타겟의 무게중심과 이미지의 중앙 사이의 거리를 기준으로 보정 계산된 값을 보여주는 표이다. 대부분의 경우 보정 이후 실거리와 매우 가까워짐을 알 수 있다.



타겟으로 사용된 물체의 크기와 실제 발사 시 타겟으로 하고자 하는 물체의 크기를 비교했을 때 실제로 사용될 스케일은 실험 환경에서 7m 이하임을 고려했을 때 보정 후 거리가 보정 전보다 실거리에 가까워짐을 알 수 있다. 이를 통해 3.4 실험 및 이전의 실험들이 성공적으로 이루어졌다는 것을 알 수 있다.

[여기에 입력]

4.3 실제 켄위성 발사에서의 알고리즘의 효용성 검증



켄위성 기본 키트에 탑재되어 있는 GPS 센서와 개발한 알고리즘으로 측정한 고도를 비교하였다. 오차가 발생하긴 하지만 충분히 알고리즘을 활용 할 수 있을 정도로 적은 오차라고 판단된다.

V. 결론 및 제언

본 연구는 기존 고도 측정 방식으로 활용되는 GPS와 고도센서를 둘 중 하나를 사용하지 못하는 상황에서 교차검증의 도구로써 사용 할 수 있는 이미지 기반의 거리측정 시스템을 제안하고자 했다. 선행된 실험으로부터 수평상황에서의 실험, 수직 상황에서의 실험, 캔위성 낙하 모사 실험, 실제 캔위성 낙하 시 상황에서 모두 시스템의 효용을 보일 수 있었다.

탐구를 시작 할 때 본 시스템을 주로 사용 할 만한 예시로 외계 행성에서의 위성 착륙을 생각하였다. 외계 행성의 궤도의 GPS 위성이 존재하지 않는다면 당연히 GPS 센서는 사용 불가 할 것이고 대기의 존재가 없을수도, 고도별 대기압을 모를수도 있다. 따라서 기존 고도센서 또한 사용 할 수 없을 것이다. 이러한 상황에서 사용 할 수 있는 시스템을 제안하고자 한다. 물론 외계 행성에는 넓이를 알고 있는 피사체도 존재 하지 않을 것이다. 하지만 이는 위성 착륙에 앞서 위성에서 행성으로 정사형이 항상 같은 구를 낙하하는 방식으로 해결 할 수 있다. 스피어볼과 같은 원리를 이용한다면 매우 큰 피사체를 공간을 효율적으로 활용하면서도 낙하시킬 수 있을것이라 생각한다.

VI. 참고문헌

- (1) 캔위성 체험, 경연대회 http://cansat.kaist.ac.kr/04_01.php