

73.019: Advanced Photogrammetry

Lab #5: Relative Orientation with Drone Images

전일서

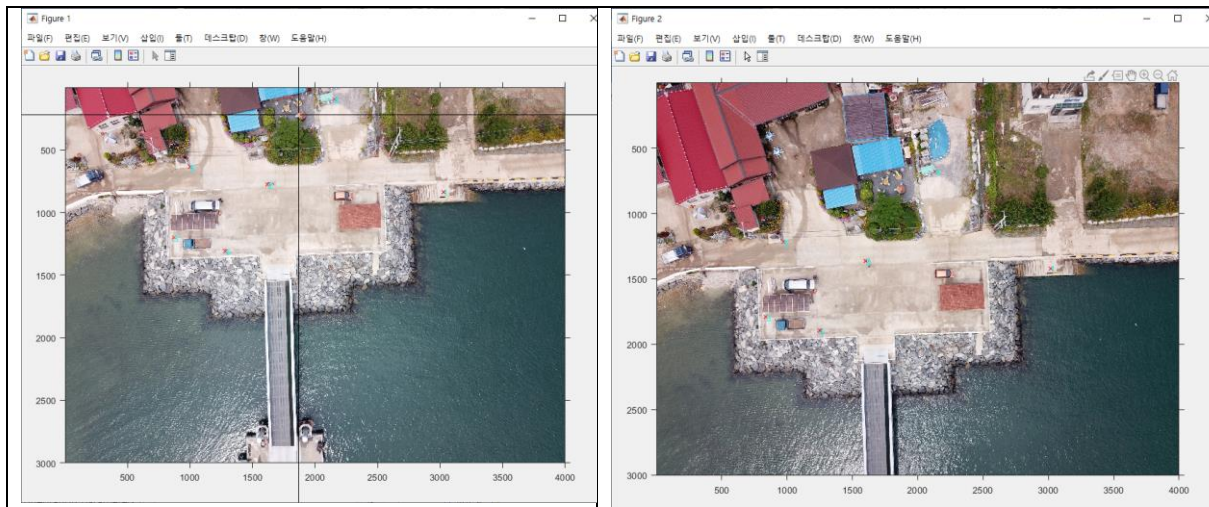
1. 영상 선택, CCS로 좌표 변환하기

연도에서 촬영된 DJI Mavic 영상 2장을 선택하였다.

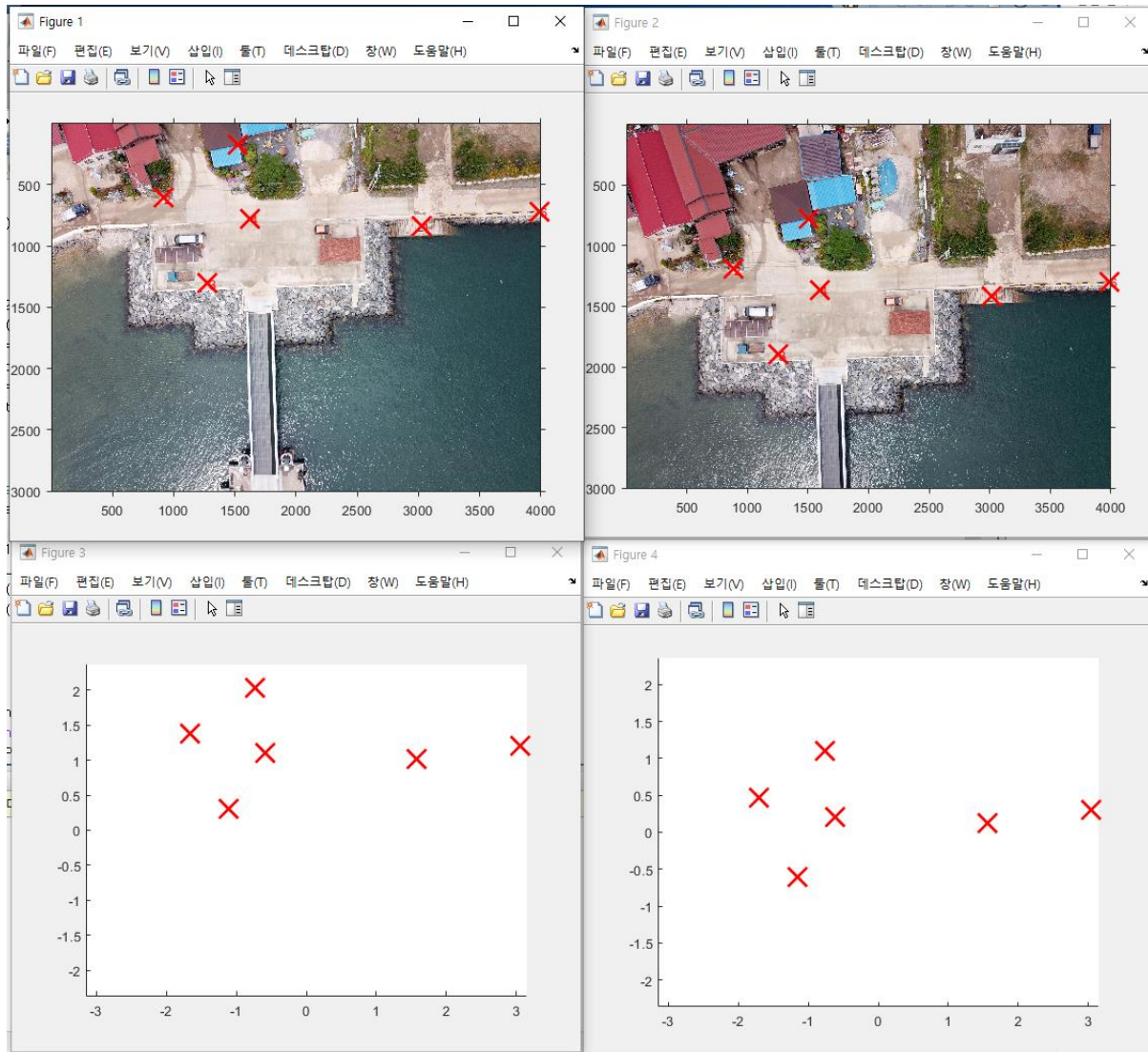
이동 방향은 영상의 세로 방향이고 좌가 1번, 우가 2번 영상이다.

Measure_TP.m을 이용하여 총 6개의 Tie Points를 픽셀 좌표 단위로 측정하였고,

Convert_CCS.m을 이용하여 측정한 Tie Points를 CCS 기준으로 변환하였다.



CCS 변환 결과



+ 사용한 디지털 영상의 바운더리 계산

Lab5_main.m 안에서 실행문

```
[BP(1,2:3), BP(2,2:3), BP(3, 2:3), BP(4, 2:3)]=Cal_IB_CCS(IO, resolution); % Image Boundary in CCS
```

Cal_IB_CCS.m 함수

```
function [ccs_dl, ccs_dr, ccs_ur, ccs_ul]=Cal_IB_CCS(IO, resolution)
```

```
R_IC=[1 0; 0 -1];
ul_ics = [0 0]';
dl_ics = [0 resolution(2)]';
ur_ics = [resolution(1) 0]';
dr_ics = [resolution(1) resolution(2)]';

img_origin =[resolution(1)/2 resolution(2)/2]';
```

```
ccs_ul=(R_IC*(ul_ics-img_origin)*IO(4))';
ccs_dl=(R_IC*(dl_ics-img_origin)*IO(4))';
ccs_dr=(R_IC*(dr_ics-img_origin)*IO(4))';
ccs_ur=(R_IC*(ur_ics-img_origin)*IO(4))';
```

end

2. TP의 좌표점 계산 함수 (&5 가시화 포함)

Input : TP list, IO, 초기 EO

Output : Coordinates of the Object Points(Ground Points or Model Points)

1번에서 결정한 TP는 텍스트 파일로 받고, IO와 EO는 Lab5_main.m에서 다음과 같이 입력한다.

```
% IO
resolution=[4000 3000];
p_size = 0.00153452;
IO = [0 0 4.73 p_size]'; % unit : mm

% Initial EO
EO_i = [0 0 0 0 0 0; 0 5 0 0 0 0]; % initial EO
```

Lab5_main.m에서 실행은 다음과 같이 한다.

```
GP_i = Comp_GP(TP, IO, EO_i);
```

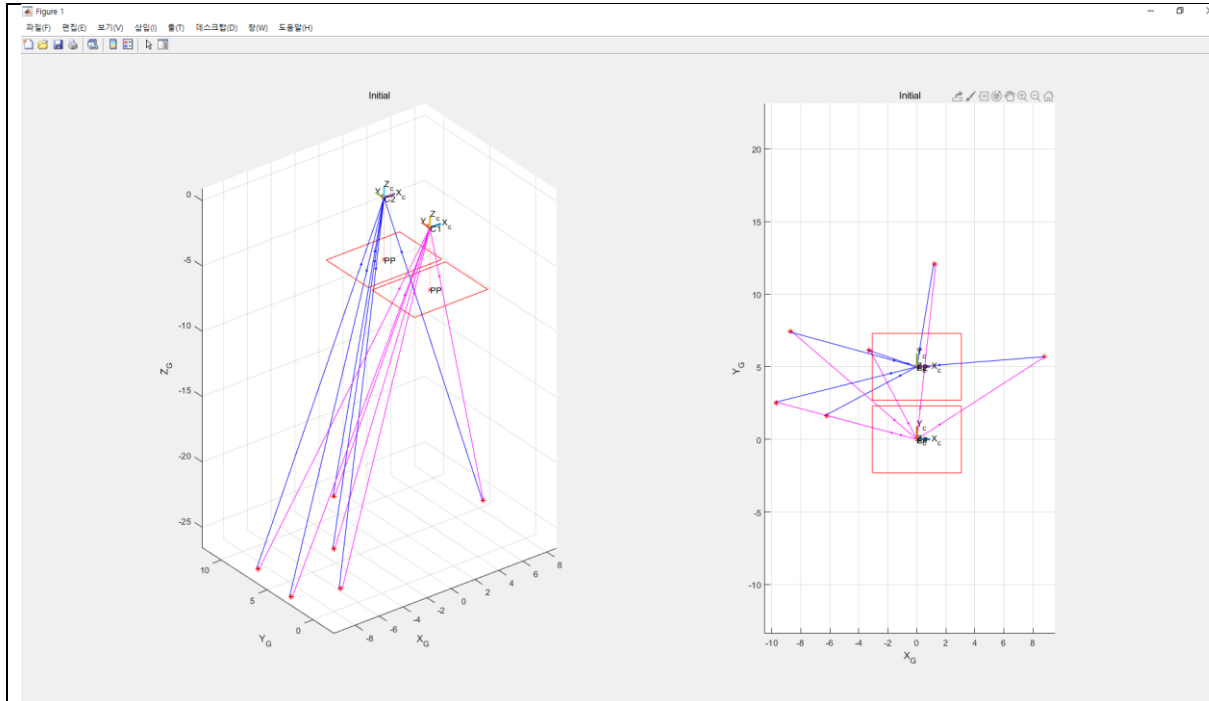
Comp_GP.m을 통해 입력 받은 TP점들을 Model Space를 기반으로 GP초기 값을 계산한다. 이에 대한 계산은 다음의 식에 기반한다.

$$\begin{bmatrix} X_{c2} \\ Y_{c2} \\ Z_{c2} \end{bmatrix} - \begin{bmatrix} X_{c1} \\ Y_{c1} \\ Z_{c1} \end{bmatrix} = \lambda_1 R_1^T \begin{bmatrix} x_1 \\ y_1 \\ -c \end{bmatrix} + \lambda_2 R_2^T \begin{bmatrix} x_2 \\ y_2 \\ -c \end{bmatrix} = \begin{bmatrix} R_1^T & R_2^T \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ -c \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

초기 값들에 대한 값과 가시화 결과는 다음과 같다.

GP_i =

8.79429988781686	5.6920125167016	-26.5508978844906
-3.31305423929082	6.15735852918974	-26.2323527936683
-9.68418381909824	2.5368808473861	-26.0276531825872
-6.2196914181569	1.6224785165432	-26.1853362766471
-8.70071064619978	7.41227696609293	-26.4362127483698
1.23384905437082	12.0706784424133	-26.4911354041237



3. RO 함수 생성 (&5 가시화 포함)

Input : TP list, 초기 GP, IO, 초기 EO

Output : loop에 따른 EO, loop에 따른 GP, statistical results(stat)

Lab5_main.m에서 함수 실행은 다음과 같이 한다.

`[EO, GP, stat] = RO_F(TP, GP, IO, EO);`

RO_F.m 함수에서 iteration 을 통해 pair에 대한 EO와 GP(MP)를 조정한다. 여기서, 언제 수렴했다고 결정하고 iteration을 중단할지 결정하는 delta 값은 $1e-6$ 으로 둔다.

Baseline을 정하는 방법은 Independent RO, Dependent RO 2가지가 있는데, Dependent RO로 설정하여 두 영상간의 원점과 축을 모두 고정하는 방법을 사용한다.

RO_F.m에서 갱신할 때, 다음과 같이 입력한다.

`ci EO = [7 9:12]; % Dependent RO img2, x,z,o,p,k`

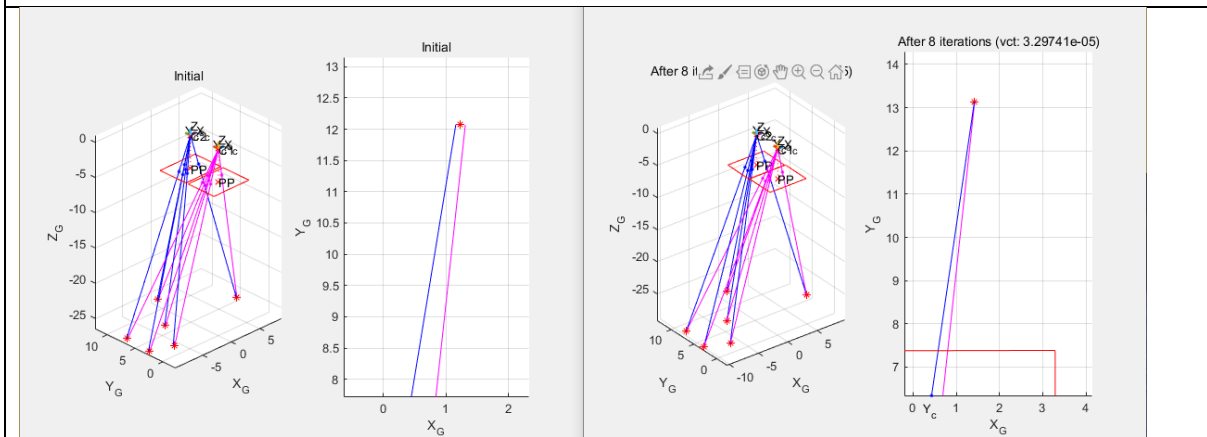
알고리즘 순서

1. 초기 값들을 이용하여 영상 1번(EO1), 영상 2번(EO2) 에 대하여 각각 R행렬의 미분 값을 계산한다.
2. For n= 1:(# of Tie points)
 - For m=1:2 (EO1, EO2)
 - 로 iteration하여 조정된 EO, GP(MP)를 얻는다.
3. `if norm(kt{k}) < delta`
`break`
`end`
 로 iteration을 종료한다.

총 5번 iteration이 수행되었으며, 1번 iteration에서 unknown의 수는 $6*2+3*6(\text{\#oftiepoints})-7(\text{arbitrary parameters})=23$ 개, observations의 수는 $4*6(\text{\#oftiepoints})=24$ 개를 얻을 수 있었다.

초기 값을 이용한 가시화(좌), iteration마지막 가시화(우)의 결과는 다음과 같다. 영상 바운더리

Scale을 위해 초점거리, 픽셀 길이와 실제 단위를 맞춰주지 않았다.



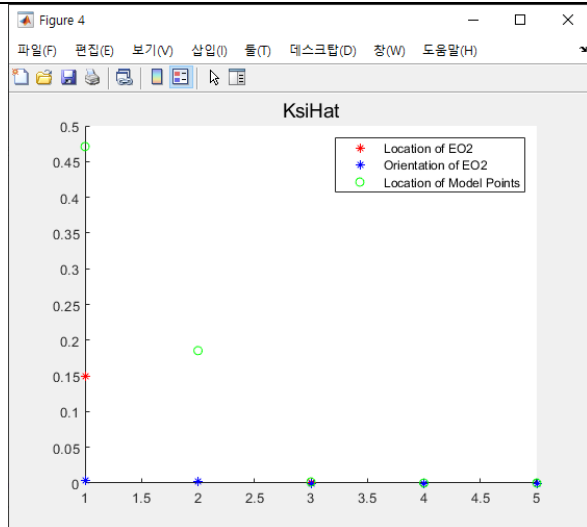
4. 통계 분석

RO_F.m을 수행하는 과정에서 `stat = {kt, et, vct, Dkt};` 을 이용하여 통계 결과에 대한 값을 return하였다. 각각에 대한 결과를 그래프를 그리면서 해석하였다.

kt ; ksi hat ; unknown 벡터를 의미한다. 비선형을 선형화하였기 때문에 0에 가까울수록 수렴하는 것과 관련이 있다.

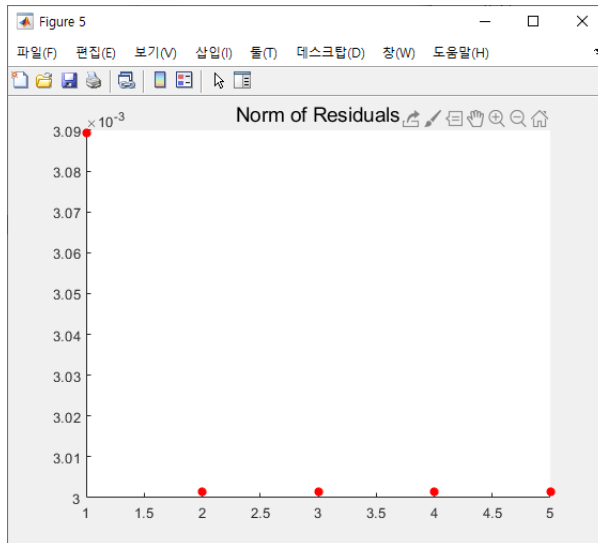
n을 tie point의 개수라고 했을 때, iteration 한 번에 다음과 같은 unknown을 얻을 수 있다.

$$\begin{bmatrix} EO_{1 \times 1} \\ EO_{2 \times 1} \\ MP_{1 \times 1} \\ MP_{2 \times 1} \\ \vdots \\ MP_{6 \times 1} \end{bmatrix}$$



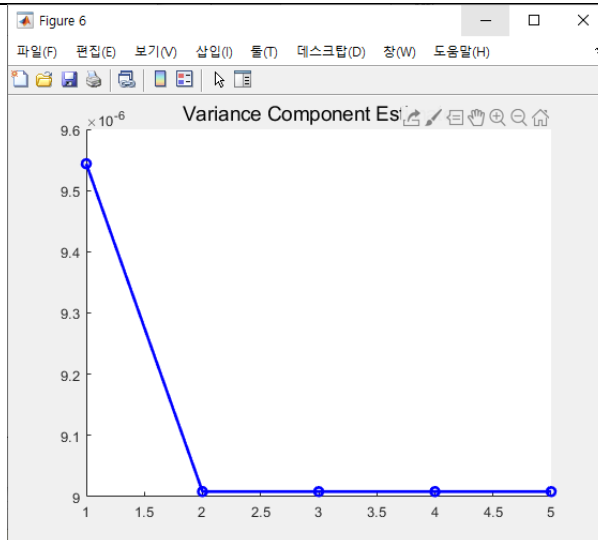
x축은 iteration index를 의미한다. Arbitrary Parameter 7개를 제외하고, unknown이 iteration함에 따라 0에 가까워지는 것을 확인하기 위해 그린 그래프다. Location of EO2는 두 번째 영상의 x, z를 norm한 것에 대한 값을 y축에 표현하였다. 마찬가지로 Orientation of EO2는 opk 회전각(자세)에 대하여 표현하였다. Location of Model Points는 모든 공액점의 x, y, z 값에 대하여 norm한 것에 대한 값을 y축에 표현하였다. 모든 값이 초기 값에 대해서는 격차가 크지만 계산이 iteration 될수록 0에 가까워지는 것을 볼 수 있다. **plot그래프를 그리기 위해 IO단위를 다시 1/1000배 하여 m로 수정해야한다.

et ; residuals ; 관측값과 계산값의 차이를 의미한다.



각 iteration에서 observation 24개의 잔차에 대해 norm한 것을 y축에 나타내었다. Iteration 세 번째부터 수렴하는 것을 볼 수 있다. ****plot**그래프를 그리기 위해 IO단위를 다시 1/1000배 하여 m로 수정한다.

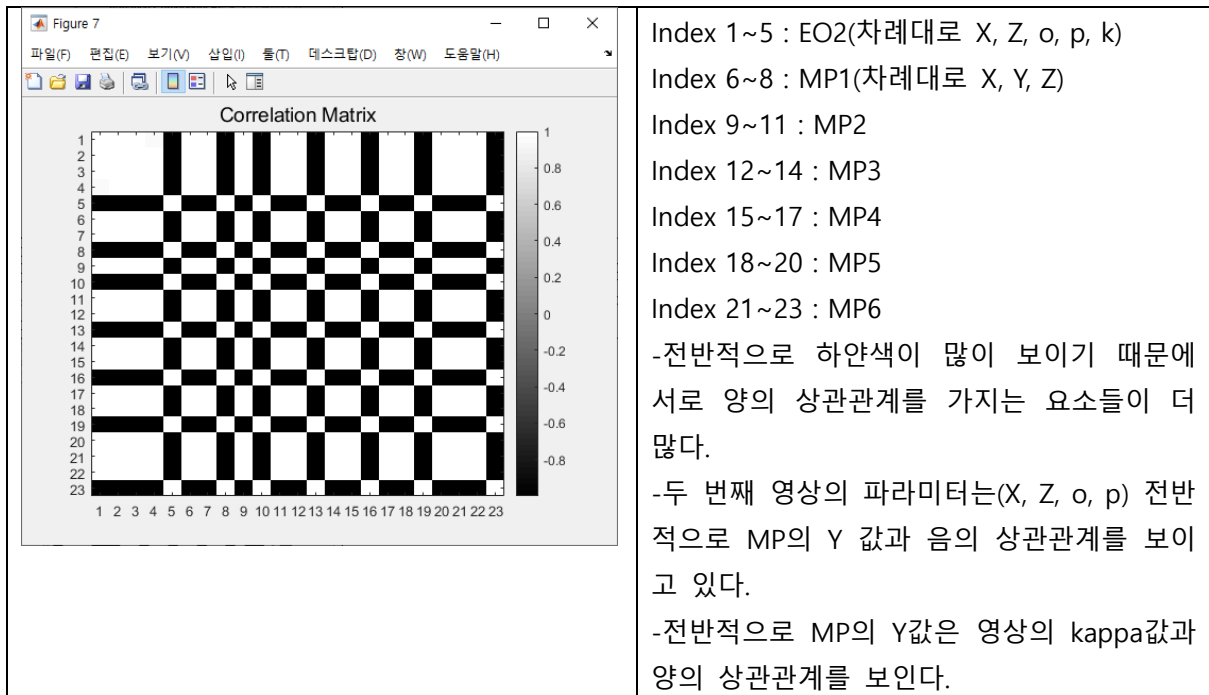
vct ; variance component estimate ; 분산을 실제로 구할 수 없기 때문에 추정으로 구한 분산 값이다.



각 iteration에 대한 vct 값을 y축에 나타내었다. Iteration 세 번째부터 수렴하는 것을 볼 수 있다. ****plot**그래프를 그리기 위해 IO단위를 다시 1/1000배 하여 m로 수정한다.

Dkt ; cofactor matrix

Cofactor matrix 는 구하고자 하는 미지수(unknown)에 대한 각각의 관계를 의미한다. 이를 다시 correlation으로 변환하여 둘의 상관관계를 [-1,1]구간으로 나타내줄 수 있다. 변환한 값은 다음과 같다.



5. 가시화

2,3,4 과정에 포함