

73.019: Advanced Photogrammetry

Lab #4: Single Photo Resection with Drone Images

전일서

1. 영상 선택하기, GCP 취득하기

서울시립대학교 전농관 주차장 전면 촬영 영상을 선택하였다. 한 영상에 다수의 지상 기준점이 보이며, 총 사용한 지상 기준점은 7개다. 각각의 지상기준점 위치는 다음 영상에 표시하였다. (출처: 박강민)



```
GCP = [205000.384 553767.153 62.837; ...  
       205000.407 553762.250 62.952; ...  
       204998.115 553767.156 62.828;  
       204998.120 553762.259 62.941; ...  
       204990.079 553780.005 69.556; ...  
       204990.075 553780.013 66.118; ...  
       205009.231 553779.912 66.118];
```

(main 함수 설명)

1. 영상(im), GCP, 카메라 파라미터 입력
2. 영상에서 좌표점 대응 함수 만들기 및 좌표점 획득 (M_IP.m)
3. 초기 외부표정요소 계산 (ini.m)
4. SPR
5. 잔차, 공분산, 분산 요소 분석
6. 좌표계 및 영상 바운더리 가시화

2. 영상에다가 좌표점 측정

M_IP.m 라는 함수를 만들어 영상에서 좌표점을 측정할 수 있도록 한다.

처음에 화면이 뜨면, 왼쪽 마우스를 한 번 클릭하면 좌표점을 측정할 수 있게 화면이 전환된다. 해당되는 부근을 클릭하면 화면이 확대되어 픽셀 단위로 측정할 수 있게 된다. GCP 점을 전부 입력한 뒤, 오른쪽 버튼을 마지막으로 클릭하면 영상점과 GCP점이 대응된 출력 값을 얻게 된다.

M_IP.m

입력: 영상

출력: 영상에서의 point 위치 대응점

```
function [IP_ics] = M_IP(img)
im = imread( img, 'jpg');

% Visualize the image point over image 1
figure(1)
imshow(im);
ax = axis;
hold on
axis on

k = 0; %
ws = 50; %
while 1
    figure(1)
    [x1,y1,btn] = ginput(1); %ginput(n) n°³ Á; ÁÂÇ¥ ¼Ä°°
    if btn == 3 % click three times, break
        break
    end
    k = k + 1;
    axis([x1-ws x1+ws y1-ws y1+ws]);
    [x2,y2,btn] = ginput(1);

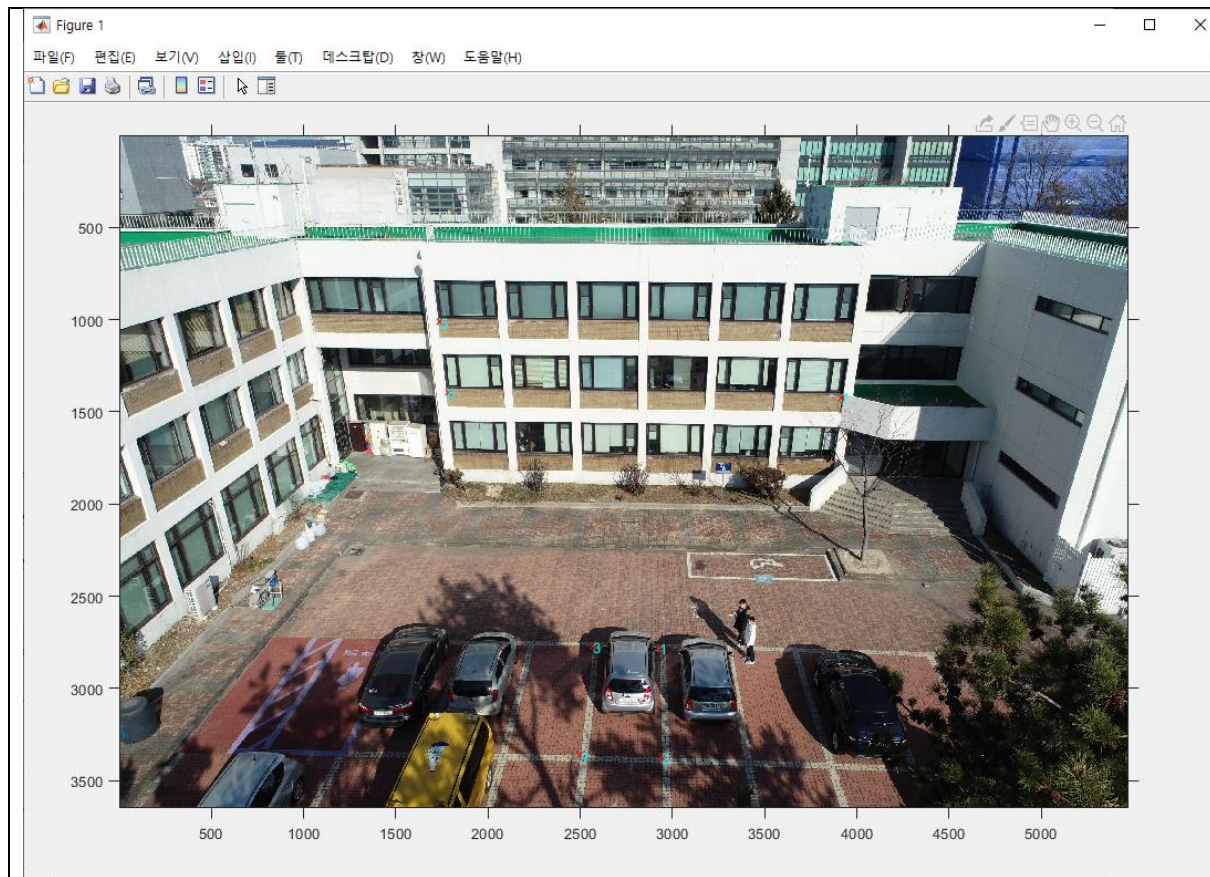
    IP_ics(k,:) = [k, x2, y2];
    plot(IP_ics(k,2), IP_ics(k,3), 'rx');
    h = text(IP_ics(k,2), IP_ics(k,3), sprintf('%d', k) );
    set(h, 'Color', 'c');
    axis ( ax );

end

fid = fopen ('IP_ics.txt', 'w');
fprintf(fid, '%s IP Coordinates in ICS\r\n' );
fprintf(fid, '%d\r\n', size(IP_ics,1) );
fprintf(fid, '%d\t%.3f\t%.3f\r\n', IP_ics' );
fclose(fid);
end
```

결과 화면

GCP 번호는 하늘색 글씨로 매겨져 있다.



3. SPR의 initial approximations 계산하는 함수 만들기

카메라 중심점의 초깃 값을 계산하기 위해 최소 제곱법을 활용한다. 최소 제곱법의 일반적인 관측방정식인 다음의 식을 세운다.

$$L = A \cdot \xi$$

각 x, y 좌표값을 구하기 위한 식은 다음과 같이 세운다.

$$\begin{aligned} L(x) &= P_{GCS}^{ICS}(x) * pixelsize - c_x \\ L(y) &= P_{GCS}^{ICS}(y) * pixelsize - c_y \end{aligned}$$

디자인 행렬은 다음과 같이 세울 수 있다.

$$A_{\begin{matrix} (for\ x) \\ (for\ y) \end{matrix}} = \begin{bmatrix} P_{GCS}^{GCS}(x) & -P_{GCS}^{GCS}(y) & 1 & 0 \\ P_{GCS}^{GCS}(y) & P_{GCS}^{GCS}(x) & 0 & 1 \end{bmatrix}$$

구하고자 하는 중심점의 초깃값은 다음과 같다.

$$\xi = \begin{bmatrix} C_{GCS}^{ICS} \\ C_{GCS}^{ICS} \end{bmatrix}$$

위에 대한 식을 함수로 구현한 코드 ini.m 은 다음과 같다.

ini.m

입력: GCP, IP(GCP를 입력하는), 내부표정요소(초점 거리, 주점 거리), 영상에서의 point 위치 대응점

출력: image index, 카메라 중심점 (x, y 좌표)

```
function [ini_EO] = ini(GCP, IO, IP_ccs)
```

```

[no_data, no_var] = size(GCP);
Q = eye(no_data*2);

for n = 1:no_data
    %observations; in ccs
    L(2*n-1:2*n, 1) = [IP_ccs(n,1) IP_ccs(n,2)]';
    %design matrix
    A(2*n-1:2*n, :) = [GCP(n,1) -GCP(n,2) 1 0; GCP(n,2) GCP(n,1) 0 1];
end
%cos(a) = -kt(1) sin(a) = -kt(2)
%xm=kt(3), ym=kt(4)
[kt, et, vct, Dkt] = LSE (A, L, Q);
avg_z = mean(GCP(:,3));
XYc = inv([-kt(1) kt(2); -kt(2) -kt(1)])*[kt(3); kt(4)];
K = atan2(kt(2), kt(1));
h = IO(3)*cos(K)/kt(1);
Zc = h + avg_z;
ini_EO = [XYc(1) XYc(2) Zc 0 0 K];
end

```

계산한 초기값은 다음과 같다.

```

ini_EO=[204998.569946372,553775.411712265,114.221961241597,0,0,-
0.0584551365782694]';

```

4. SPR실행 함수 만들기

SPR.m

입력: GCP, (3번에서 계산한) 초기 카메라 EO, IO, delta, 영상에서의 point 위치 대응점

출력: 추정된 EO, 잔차(et), 분산 추정치(vct), cofactor estimate(VEO)

```

function [EstiEO, SavedEO, SaveEt, SaveVct, SaveVEO, F0]= SPR(IP_ics,
GCP, IO, ini_EO, delta)

```

```

EO=ini_EO;

```

```

for k = 1:100

```

```

    % Rotational Matrix
    R = Rot3D(EO(4), EO(5), EO(6));

```

```

    % Derivatives of the Rotational Matrix
    [dR_om, dR_ph, dR_kp] = dRot3D(EO(4), EO(5), EO(6));

```

```

    % Observation Equations
    for n = 1:size(IP_ics,2)
        GC = GCP(:,n) - EO(1:3,1);
        ND = R * GC;
        F0(2*n-1:2*n,1) = IO(1:2,1) - IO(3) / ND(3) * ND(1:2,1);

```

```

        dND(:,1:3) = -R;
        dND(:,4) = dR_om * GC;
        dND(:,5) = dR_ph * GC;
        dND(:,6) = dR_kp * GC;

```

```

        A(2*n-1:2*n,:) = IO(3) / ND(3)^2 * [-ND(3) 0 ND(1); 0 -ND(3)
ND(2)] * dND;

```



```

        Y(2*n-1:2*n,1) = [IP_ics(1,n) IP_ics(2,n)]';
    end
    EstiEO(:,k) = EO(:,1);

    % Estimate based on Least Squares Adjustment
    dEO = inv(A'*A) * A'*(Y-F0);

    % Update the Initial Approximations of EO
    EO = EO + dEO;

    et = (Y-F0) - A * dEO;
    vct = ( et' * et ) / (size(A,1) - rank(A));
    sdt = sqrt(vct);
    VEO = vct * inv(A'*A);

    SavedEO(:,k)=dEO;
    SaveEt=et;
    SaveVct=vct;
    SaveVEO=VEO;
    if norm(dEO) < delta
        break
    end
k
end

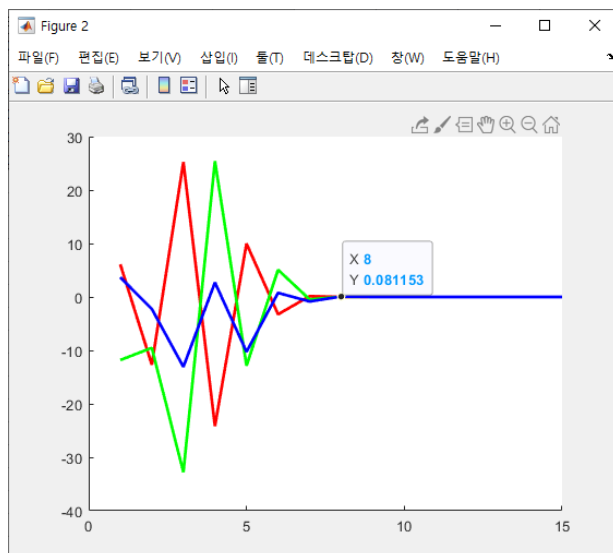
```

5. 통계 분석(잔차, 분산 추정치, 분산공분산추정치, 등)

다음은 외부표정요소 중 x, y, z의 변화량에 대한 그래프다.

R, G, B는 각각 x값, y값, z 값을 의미한다.

8번째 이터레이션부터 거의 0에 수렴하는 것을 확인할 수 있다.



분산공분산추정치 (variance component estimate) 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.

	1	2	3	4	5	6
1	1	-0.2254	-0.2898	-0.2809	-0.9924	-0.9965
2	-0.2254	1	0.9859	0.9913	0.1038	0.3041
3	-0.2898	0.9859	1	0.9993	0.1703	0.3684
4	-0.2809	0.9913	0.9993	1	0.1607	0.3596
5	-0.9924	0.1038	0.1703	0.1607	1	0.9787
6	-0.9965	0.3041	0.3684	0.3596	0.9787	1

Unknown의 순서는 x, y, z, omega, phi, kappa로 되어 있다.

Phi, kappa는 x와 상관도가 높고, z, omega가 y가 상관도가 높고, omega와 z가 서로 상관도가 높은 것을 확인할 수 있다.

6. GCP, 커버리지, 카메라 축, 영상 커버리지, 영상점 등 가시화하기

