73.019: Advanced Photogrammetry
Lab #6: Registration of Close Range Images

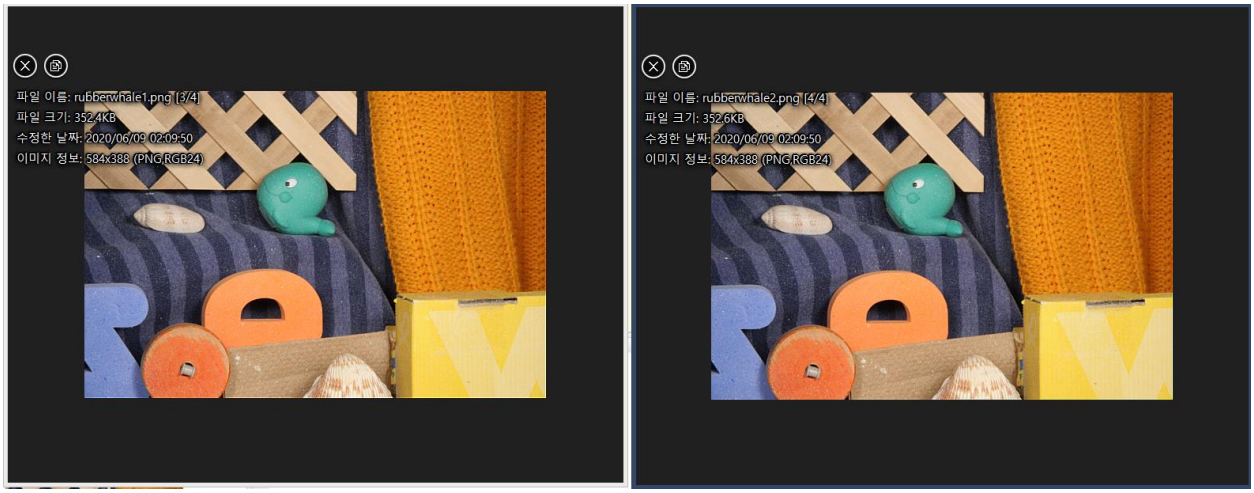
전일서

메인 코드: lab6_ho.m (homography 실행 함수) , lab6_fun.m (fundamental matrix 실행 함수)
특징점 매칭 함수: SURF.m
H 구하는 함수: estimateH.m
F 구하는 함수: estimateF.m

추가로 이용한 코드: findHomography (그 이외 sub 함수들), estimateFundamentalmatrix(및 epipolar geometry 관련 함수)

차례
1. Homography 구하기 1.1 영상 및 타인포인트 구하기 (문제 1 번 해당) 1.2 Homography 식 유도 및 함수 구현 (문제 2 번 해당) 1.3 정확도 비교 (문제 3 번 해당) 가시화는 각 항에 포함 (Optional 문제 5 번 해당) 2. Fundamental matrix 구하기 2.1. 영상 및 타이포인트 구하기 (문제 1 번 해당) 2.2. Fundamental matrix 구하는 식 유도 및 함수 구현 (문제 3 번 해당) 2.3. 정확도 비교(문제 3 번 해당) 가시화는 각 항에 포함 (Optional 문제 5 번 해당)

1. Homography 구하기
1.1 영상 및 타이포인트 구하기
보통 Homography 를 구하는 알고리즘에 이용되는 영상을 이용하였다.
584x388 의 해상도를 가진 사진으로, 같은 시점에서 카메라의 방향을 살짝 회전한 정도로 찍힌 영상이다.



SURF detector 로 두 영상에서 매칭하기

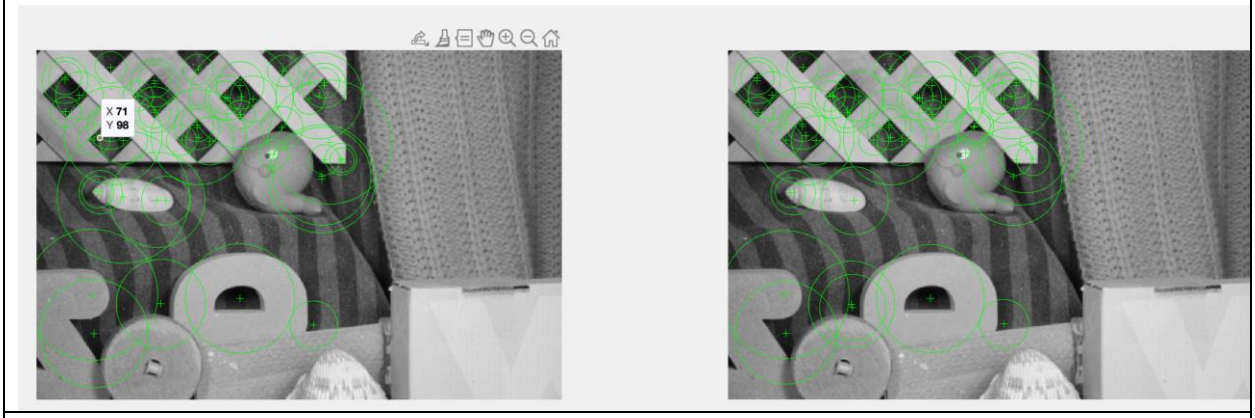
SURF.m

입력: grayscale 로 변환한 한 페어의 두 영상

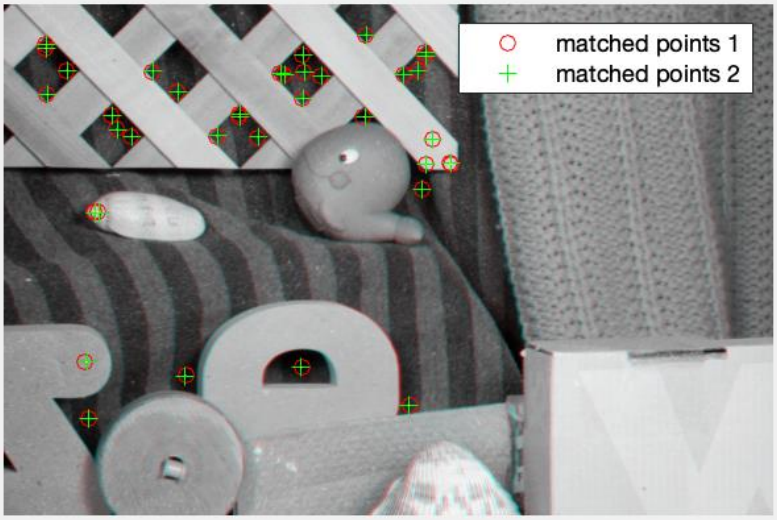
출력: 매칭된 포인트

```
%% Feature matching with SURF
%
% from Lab6, Adv. Photogrammetry, Geoinformatics, Univ. of Seoul
% Ilseo Jeon

%% matching with SURF
function [matchedPoints1, matchedPoints2]=SURF(I1, I2, str, m)

img1pts = detectSURFFeatures(I1, 'MetricThreshold',300);
img2pts = detectSURFFeatures(I2, 'MetricThreshold',300);

img1pts_str = img1pts.selectStrongest(str);
img2pts_str = img2pts.selectStrongest(str);

figure(m);
subplot(1,2,1); imshow(I1); hold on
plot(img1pts_str);
subplot(1,2,2); imshow(I2); hold on
plot(img2pts_str);
hold off

[f1,vpts1] = extractFeatures(I1,img1pts_str);
[f2,vpts2] = extractFeatures(I1,img2pts_str);

indexPairs = matchFeatures(f1,f2, 'Method', 'Approximate','Unique', true,
'MaxRatio', 0.2,'MatchThreshold', 10) ;
matchedPoints1 = vpts1(indexPairs(:,1));
matchedPoints2 = vpts2(indexPairs(:,2));

figure(m+1); showMatchedFeatures(I1,I2,matchedPoints1,matchedPoints2);
legend('matched points 1','matched points 2');
hold off;
end
```

탐지된 features 결과는 blob 으로 나타난다.



각 영상에서 매칭된 포인트 가시화한 모습, 거리가 얼마나 차이 나는지 육안으로 확인할 수 있다.



1.2. 같은 시점에서 촬영된 두 영상에 대한 Homography 식 유도, matlab 구현하기

$$\mathrm{x}' = P\mathrm{x}$$

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \mathrm{x}$$

$$x' = \frac{u'}{w'}, y' = \frac{v'}{w'}, A^T = [p_{11} \quad p_{12} \quad p_{13}], B^T = [p_{21} \quad p_{22} \quad p_{23}], C^T = [p_{31} \quad p_{32} \quad p_{33}]$$

$$x' = \frac{u'}{w'} = \frac{A^T\mathrm{x}}{C^T\mathrm{x}}, y' = \frac{v'}{w'} = \frac{B^T\mathrm{x}}{C^T\mathrm{x}}$$

$$x'C^T\mathrm{x} - A^T\mathrm{x}=0$$
$$y'C^T\mathrm{x} - B^T\mathrm{x}=0$$

unknown 을 벡터화한 식으로 $A^T = [p_1 \quad p_2 \quad p_3], B^T = [p_4 \quad p_5 \quad p_6], C^T = [p_7 \quad p_8 \quad p_9]$에 따라 다음과 같이 $\mathrm{Ap} = 0$ 꼴로 정리한다.

$$\begin{bmatrix} -x & -y & 1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & 1 & xy' & yy' & y \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{31} \\ p_{32} \\ p_{33} \end{bmatrix} = 0$$

A 행렬의 고윳값 분해를 통해 p 벡터를 구할 수 있다.
SVD 함수를 이용하여 다음 과 같이 매틀랩 코드를 구현하였다.

파일명: estimateH.m

입력: 두 영상의 매칭된 포인트 img1pts, img2pts

출력: 디자인 매트릭스 A 와 homography matrix H 출력

```
%% estimate Homography
% Ilseo Jeon
% 2020 Adv. Photogrammetry, Geoinformatics, University of Seoul
% calculate Homography from the same projection

function [A, H] = estimateH(img1pts, img2pts)
format long g
no_TP = size(img1pts,1);
A = zeros(no_TP, 9);
for i = 1:no_TP
    A(2*i-1,:) = [-img1pts(i,1) -img1pts(i,2) 1 0 0 0
img1pts(i,1)*img2pts(i,1) img2pts(i,1)*img1pts(i,2) img2pts(i,1)];
    A(2*i,:) = [0 0 0 -img1pts(i,1) -img1pts(i,2) 1
img1pts(i,1)*img2pts(i,2) img2pts(i,2)*img1pts(i,2) img2pts(i,2)];
end

[U, D, V] = svd(A);

eig_val=diag(D);
[min_eig_val, min_eig_val_idx] = min(eig_val);
H = reshape(V(:,min_eig_val_idx), 3, 3)';

end
```

1.3. Homography 정확도 비교하기

1.1 에서 추출한 특징점을 이용하여 다음의 값을 구하였다.

```
>> myH/myH(end)
```

ans =

0.995740421317452	-0.00730525679554034	-1.89190295141524
-0.00103893440354423	1.00191940671942	0.144070438412745
-4.1171737536904e-06	3.66089847172559e-06	1

새로운 매칭 포인트를 직접 수동으로 취득하여 검사점으로 활용한다. 총 16 개의 점을 직접 취득하여 rubberwhale_test.txt 파일에 작성하였다. 매칭 포인트를 취득하는 코드는 Measure_TP.m 을 이용하였다.

구한 H 를 통해 $x' = Hx$ 를 다시 계산하여 x'이 실제 두 번째 영상의 추출한 특징점과 같은 위치에 있는지 확인한다. 둘의 차 $(x'_{calculated} - x'_{acquired})$ 를 이용한다.

메인함수 lab6_ho.m 에서 확인한 결과는 다음과 같다. 차를 구한 뒤 거리 값으로 계산하고, 평균과 분산을 확인하였다. 평균 2.7990 픽셀 정도의 거리 차이를 보였고, 분산은 2.8246 정도를 보였다.

lab6_ho.m

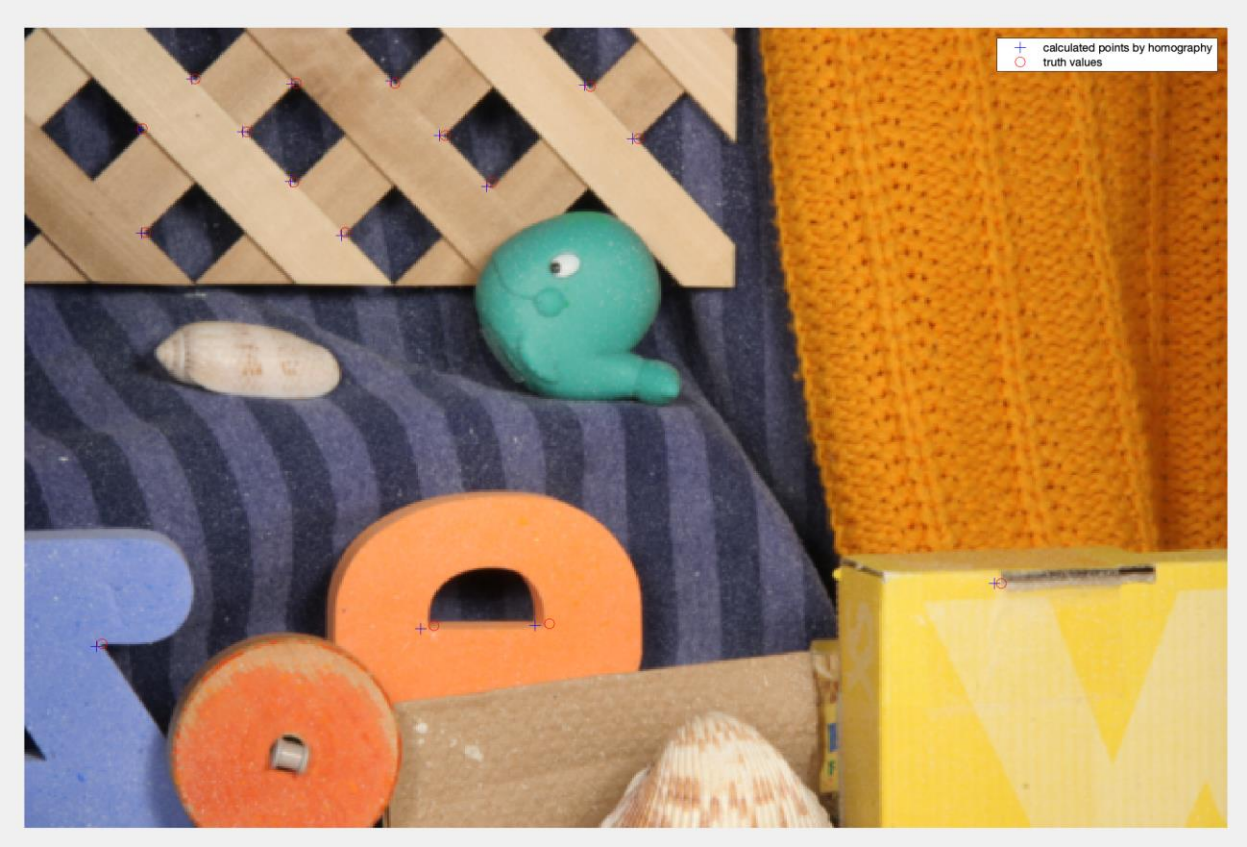
diff_myH 의 결과 값

-1.47320652324527	0.696709215450071	0
-7.15538324680404	0.948096174530804	0
-1.65107534132341	1.29532054659983	0
-1.97838931104531	0.261477177582179	0
-2.08861225876956	0.586928494036194	0
-2.50403984802841	0.0665413822362311	0
-1.50848992700757	0.0496085835406603	0
-2.35442972960090	-0.512707119100050	0
-2.31425938876501	1.83863959347929	0
-6.45809503955277	1.39325401084437	0
-1.48502333471586	-0.468819651224308	0
-1.67089773811014	0.141752099597035	0
-1.70987564841280	-1.00747269385503	0
-2.68023386913256	-0.00630274182524460	0
-2.98082252838225	0.306187183443058	0
-2.67650946656350	0.788860628087775	0

```
>> d = sqrt(power(diff_myH(:,1),2)+power(diff_myH(:,2),2))

d =
```

```
        1.62964449835707
        7.21792184529618
        2.09854833186417
        1.99559379144521
        2.16951290030598
        2.50492381442312
        1.50930542682528
        2.40960742479027
        2.95573883710937
        6.60667452495051
        1.55726881752146
        1.67689979097195
        1.98460977572384
        2.68024127977312
        2.99650688919881
        2.79034120048984

>> mean(d)

ans =

        2.79895869681539

>> var(d)

ans =

         2.8245604446789
```

차이를 가시화한 모습은 다음과 같다.



이상적이라면 diff_myH 은 0 의 값을 가져야 한다. 위의 결과는 0 에 가깝지 않은 결과를 보였기 때문에 올바른 H, F matrix 를 구한 것일까? 라는 의문을 갖게 한다. 통상적으로 Homography, Fundamental matrix 는 RANSAC 을 이용하여 구하게 된다.

RANSAC 을 사용하는 이유는 inlier 라고 여겨진 포인트들 사이에 이상치를 배제하고 알고리즘을 수행하기위해서다. RANSAC 은 Homography 와 Fundamental matrix 를 구하면서 다음의 문제를 해결하고자 한다. 1. 특징점으로 추출한 포인트들이 매칭된 포인트들이 이상적으로 매칭되었다고 보기 어려운 경우
2. 매칭된 포인트들을 이용하여 Homography, Fundamental matrix 를 유추해줄 수 있는 기하적 제약조건을 만족하지 않는 경우 (ex. Homography 에서는 투영 중심이 같지 않은 경우, Fundamental matrix 에서는 coplanar constraint 를 만들어줄 수 있는 점들이 한 평면에 존재해 버리는 경우나 투영 중심이 같아 baseline 이 생기지 않는 경우 등)

다시 문제에 접근하여, 각각의 Homography 와 Fundamental matrix 를 구하는 문제에서 RANSAC 을 이용한 알고리즘과 비교하고자 한다.

+ RANSAC 의 중요한 요소인 샘플 포인트의 갯수, trial 의 횟수 등의 파라미터는 다음과 같이 설정되었다.

hRANSAC

```
coef.minPtNum = 8;
coef.iterNum = 1000;
coef.thDist = 4;
coef.thInlrRatio = .6;
```

fRANSAC

```
fRANSAC = estimateFundamentalMatrix(img1pts(:,1:2),...
    img2pts(:,1:2),'Method','RANSAC',...
    'NumTrials',1000,'DistanceThreshold',1e-4);
```

마찬가지로 계산한 결과는 다음과 같다. 평균은 myH 와 차이를 보이나 분산은 RANSAC 을 사용했을 때 더 좋은 값을 보인다. RANSAC 은 이상치에 대처할 수 있는 알고리즘이기 때문에 이러한 결과가 나타난 것으로 보인다. 하지만, RANSAC 알고리즘의 파라미터는 상황에 맞게 경우에 맞게 조절했을 때 가장 좋은 성능을 보인다는 것을 알고 있어야 한다.

lab6_ho.m

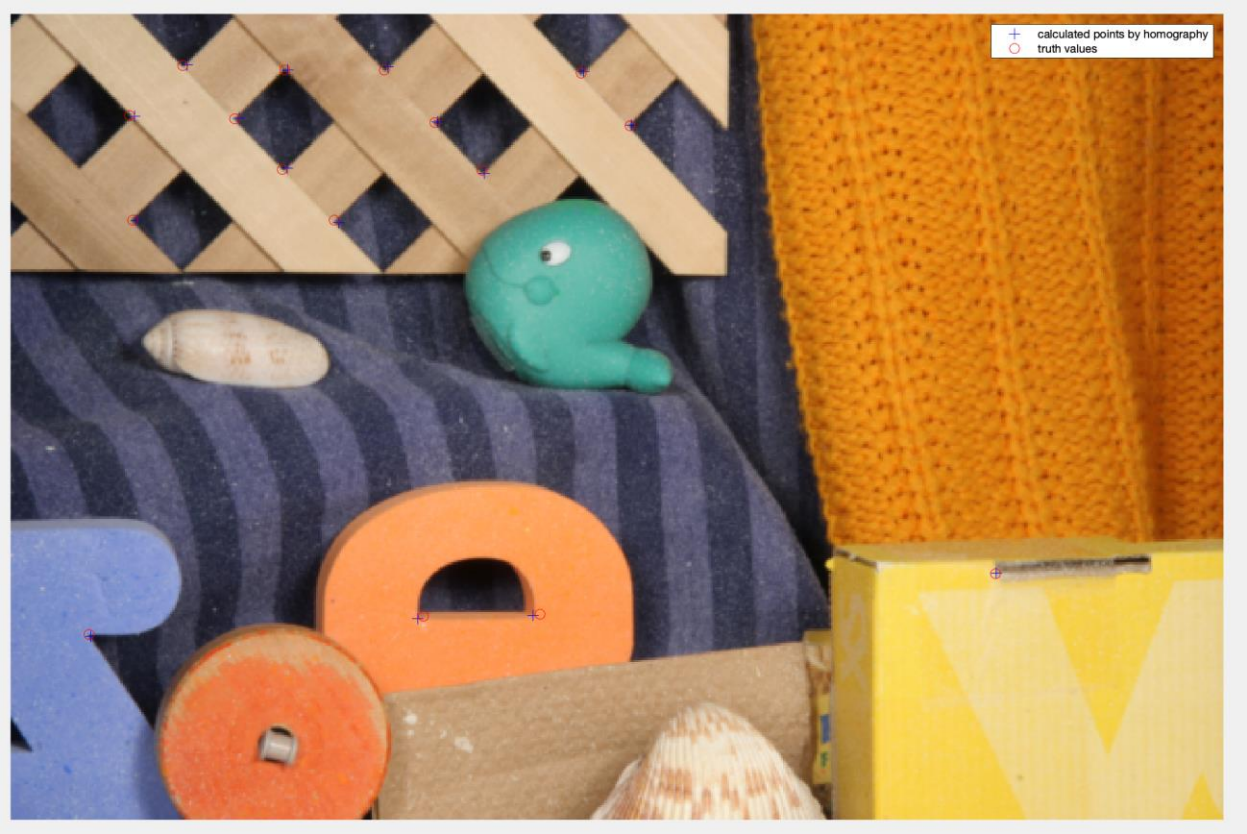
diff_hRANSAC 결과 값

2.31082229261291	0.408546874374913	0
-3.37165008530175	0.659954870549825	0
2.13379604141744	1.00709960744641	0
1.80496752479997	-0.0266269203641656	0
1.69688418345422	0.298655350589808	0
1.28223802835848	-0.221788178047817	0
2.27627997684665	-0.238612021779193	0
1.43327662595823	-0.801143131715719	0
1.47203153808181	1.55030886657836	0
-2.67524419421912	1.10519172506156	0
2.29982053812768	-0.757039885968936	0
2.11390976173492	-0.146467708241005	0
2.07639580390415	-1.29580271284966	0
1.10747132677932	-0.294740388896933	0
0.806597062864171	0.0177177623351668	0
1.10375434596104	0.501028066991466	0

```
>> d = sqrt(power(diff_hRANSAC(:,1),2)+power(diff_hRANSAC(:,2),2))

d =

    2.34665937379039
    3.43563163463107
    2.35952011343142
    1.80516391457136
```

```
          1.72296574269281
           1.3012779708004
          2.28875211194453
          1.64198422769936
          2.13783405130028
          2.89454318465758
          2.42121537590974
          2.11897788339462
          2.44755473995972
          1.14602121991063
         0.806791634143087
          1.21214800257274

>> mean(d)

ans =

          2.00544007383811

>> var(d)

ans =

         0.473952838901446
```

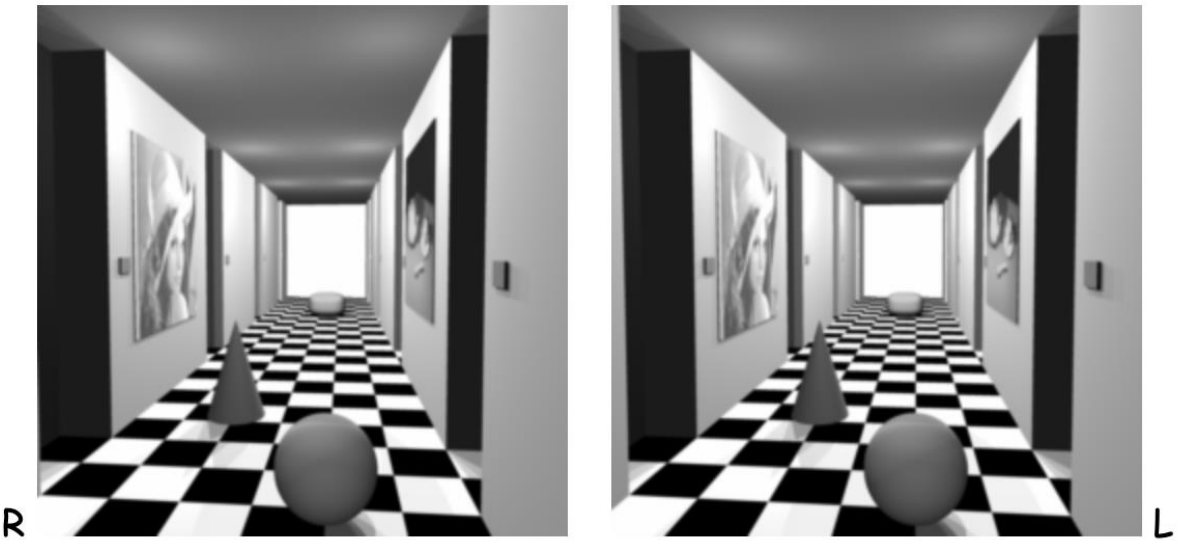
diff_hRANSAC 을 가시화한 모습은 다음과 같다.



2. Fundamental matrix 구하기

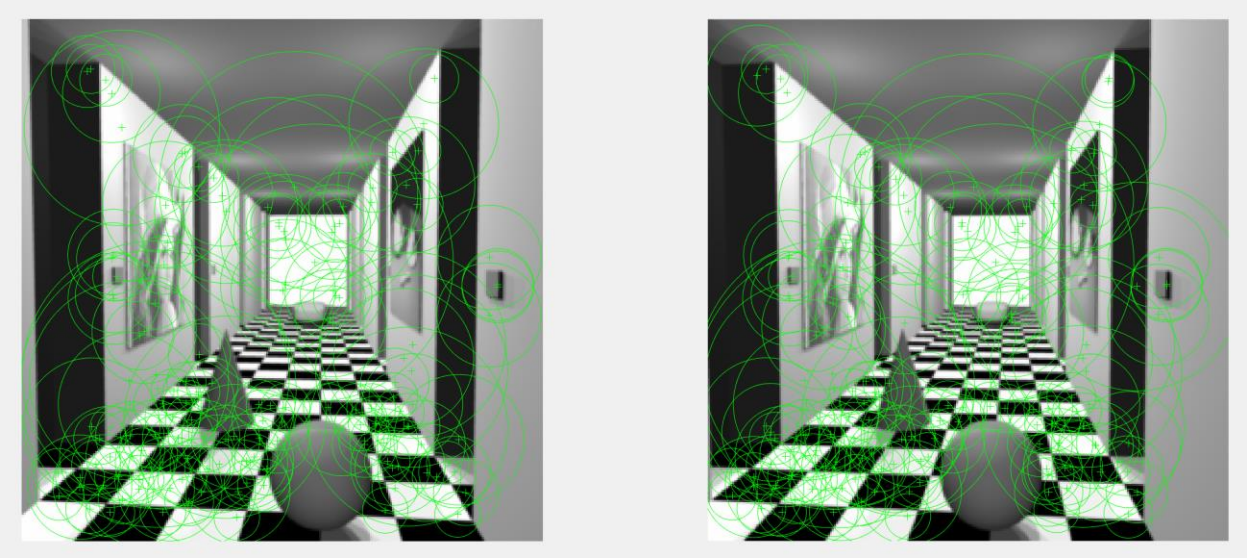
2.1. 영상 및 타이 포인트 구하기

보통 stereo sample 로 많이 사용되는 복도 영상을 이용하였다. 256x256 영상이다.

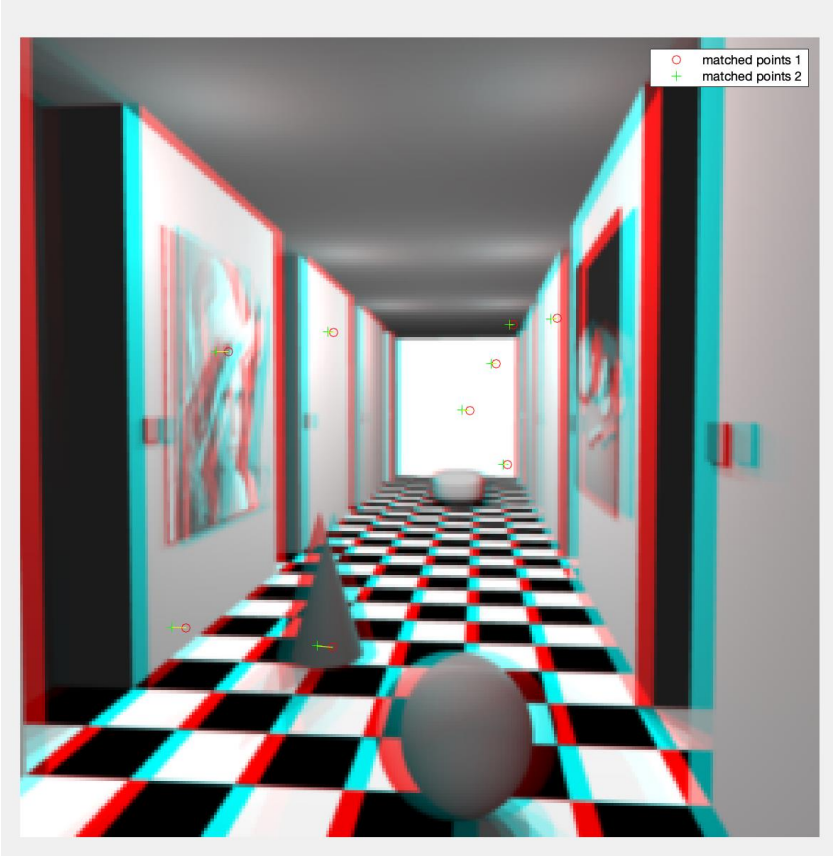


출처: https://people.duke.edu/~ng46/topics/stereo.htm

마찬가지로 SURF.m 영상을 이용하였다. 가장 강한 특징점으로 보이는 점은 150 개를 추출하였다.

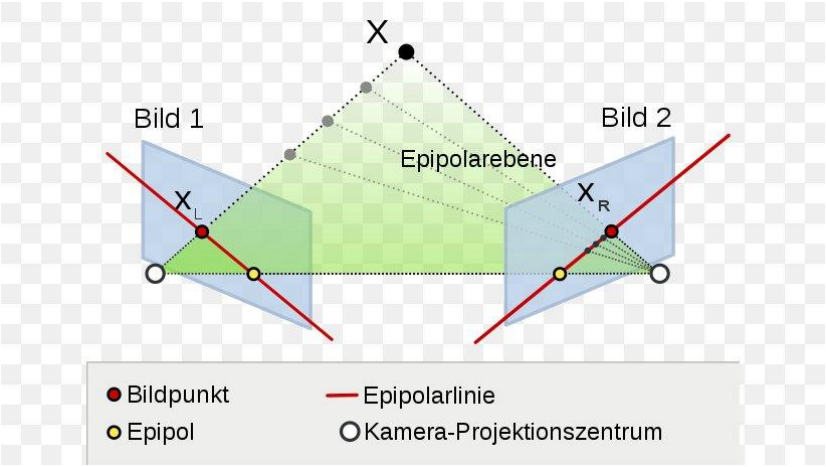


매칭된 포인트간의 차이를 가시화한 영상은 다음과 같다.



2.2. Fundamental matrix 식 유도 및 함수 구현하기

다음의 epipolar geometry 를 나타내는 그림에 따르면,



다음과 같은 수식이 성립한다.

$$x_L F x_R = 0$$

$$[x', y', 1]\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}\begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = 0$$

$$x'' f_{11} x' + x'' f_{21} y' + x'' f_{31} + \cdots = 0$$

af = 0 꼴로 정리하기 위해 다음과 같이 정리한다.
이 때, Kronecker Product 를 이용하여 F 행렬을 벡터화 하여 정리할 수 있다. (Cyrill Stachniss 강의 참고.)

$$(x_R \otimes x_L)^T = a^T = [x''x', x'' y', x'', y'' x', y'' y', y'', x', y', 1]$$

$$vec(F) = f = [f_{11}, f_{21}, f_{31}, f_{12}, f_{22}, f_{32}, f_{13}, f_{23}, f_{33}]^T$$

$$(x_R \otimes x_L)^T vec(F) = a^T f = 0$$

여러 개의 포인트에 대하여 다음과 같이 정리할 수 있다.

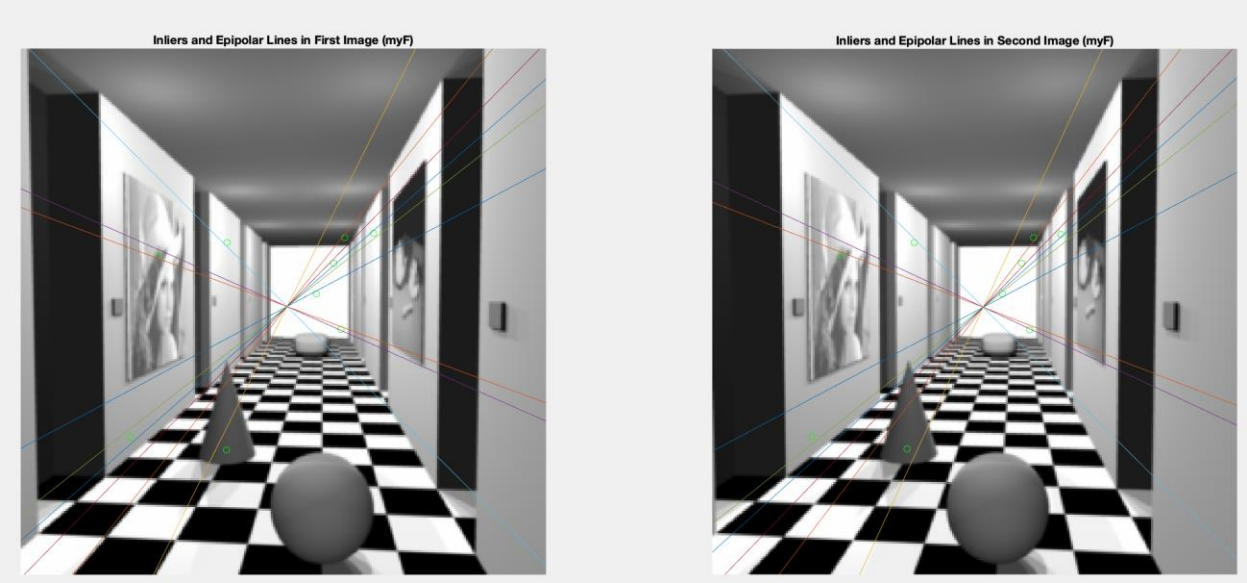
$$a_n^T f = Af = 0$$

f 는 A 행렬의 특잇값 분해(Singular decomposition value) 에서 가장 작은 특이값(the smallest singular value)을 갖는 singular vector 로 선택할 수 있다. 하지만, 실제 특잇값 분해를 구해보면 F 의 rank 가 2 가 아닌 것을 알 수 있다. 가장 작은 특이값(the smallest singular value)을 갖는 singular vector 를 선택한 후, F 행렬의 Rank 가 2 가 되도록 바꿔서 다시 F 행렬을 구해줄 수 있다.

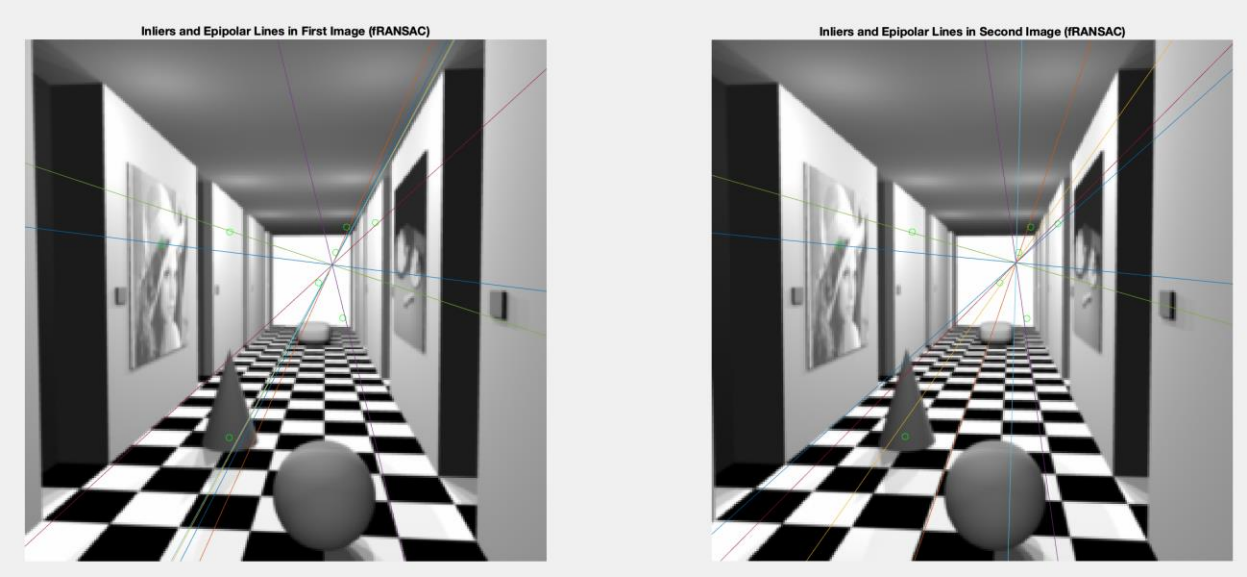
파일명: estimateF.m
입력: 두 영상의 매칭된 포인트 img1pts, img2pts 출력: Fundamental matrix 출력

```
function F = estimateF(img1pts, img2pts)
format long g
for n = 1:size(img1pts, 1)
    A(n,:) = kron(img1pts(n,:), img2pts(n,:));
end

[U, D, V] = svd(A);
eig_val = diag(D);
Fa = reshape(V(:, 9), 3, 3)';

[Ua, Da, Va] = svd(Fa);

F = Ua * diag([Da(1,1), Da(2,2), 0]) * Va';

end
```

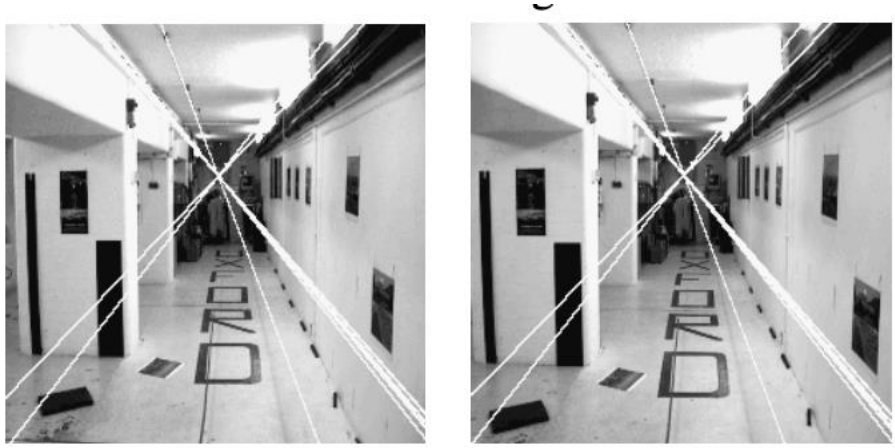
두 영상에 대하여 다음과 같이 myF 로 구한 Epipolarline 을 가시화하였다.



마찬가지로 RANSAC 을 이용하여 구한 fRANSAC 도 같이 확인하였다.



교재에 제공되었던 복도 예제와 비슷한 경향으로 에피폴이 생성된 것을 볼 수 있다.



Corridor scene

2.3. Fundamental matrix 정확도 비교하기

Multiple View Geometry Ch11.5 에 나와 있는 잔차 계산법을 이용한다. 아래의 식은 직선과 점 사이의 거리 구하는 공식에 기반한다.

Symmetric epipolar distance. Equation (11.9) is similar in form to another cost function

$$\sum_i \quad d(\mathbf{x}_i', \mathbf{F}\mathbf{x}_i)^2 + d(\mathbf{x}_i, \mathbf{F}^\mathsf{T}\mathbf{x}_i')^2$$

$$= \sum_i (\mathbf{x}_i'^\mathsf{T}\mathbf{F}\mathbf{x}_i)^2 \left(\frac{1}{(\mathbf{F}\mathbf{x}_i)_1^2 + (\mathbf{F}\mathbf{x}_i)_2^2} + \frac{1}{(\mathbf{F}^\mathsf{T}\mathbf{x}_i')_1^2 + (\mathbf{F}^\mathsf{T}\mathbf{x}_i')_2^2} \right) \quad (11.10)$$

$$\frac{1}{N}\sum_i^N d(\mathbf{x}_i', \mathbf{F}\mathbf{x}_i)^2 + d(\mathbf{x}_i, \mathbf{F}^\mathsf{T}\mathbf{x}_i')^2$$

총 5 개의 점에 대하여 잔차를 계산하였다. 거듭 제곱 형태의 값을 얻기 때문에 제곱근의 값을 구해보면 다음과 같다.

```
>> sqrt(checkmyF/2)

ans =

  single

        190.5186
```

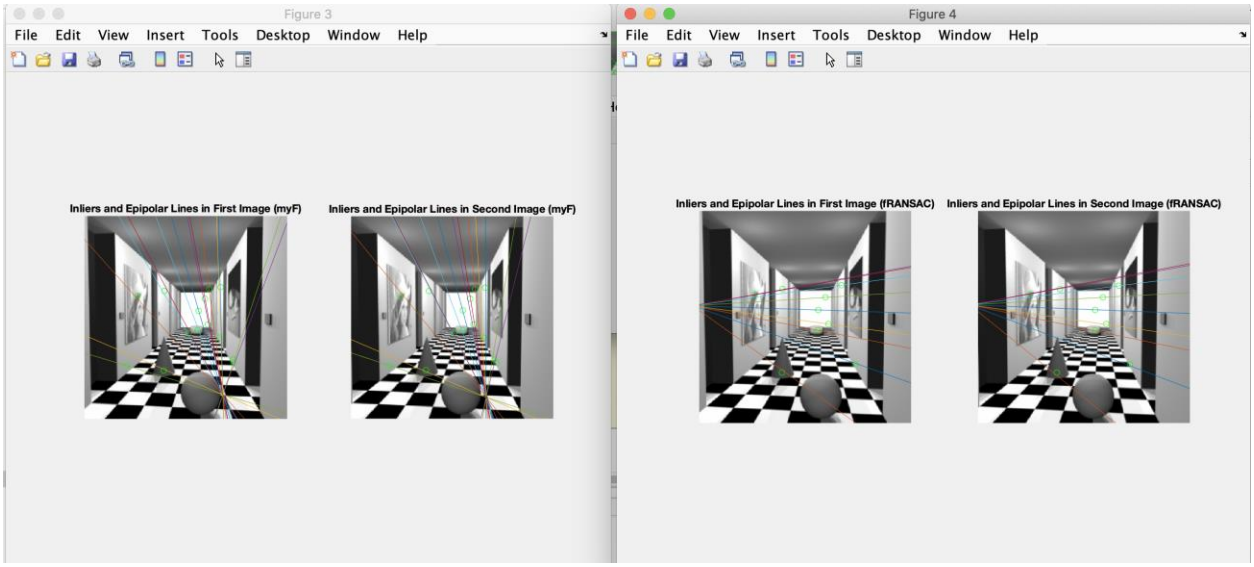
한 영상에서 평균적으로 190 픽셀정도 차이가 나는 것으로 볼 수 있다.

```
>> sqrt(checkfRANSAC/2)

ans =

          12.8080190921287
```

한 영상에서 평균적으로 12.8 픽셀 정도 차이가 난다.

RANSAC 을 사용했을 때 더 좋은 결과를 나타낸 것으로 볼 수 있다. 그렇다면 RANSAC 을 사용하는 것이 가장 올바른 것일까? 앞서 분량의 이유로 특징점 검출 파라미터를 조정한 경우를 보고서에 모두 넣지는 않았지만, epipolarline 을 형성하는 것이 특징점에 따라 영향을 많이 받는 것으로 보인다. 특징점이 많이 있는 것보다 올바른(매칭이 잘 되는) 특징점이 많이 있을수록 epipolarline 결과가 더 좋은 것으로 보인다. 또한 RANSAC 의 경우에도 이상치로 너무 많은 매칭점들을 제거해버리면 오히려 epipolar geometry 에 악영향을 끼칠 수 있다. 특징점을 추출하는 알고리즘의 파라미터와 RANSAC 파라미터를 경우에 맞게 합리적으로 조정했을 때, 이상적인 결과를 얻을수 있다.

특징점을 많이 추출했음에도 불구하고 fundamental matrix 와 epipolar line 이 잘못 결정된 간단한 사례도 다음과 같이 확인할 수 있다.



검사점에 대하여 epipolar line 을 가시화한 모습은 다음과 같다. myF 로 구한 epipolar line(좌), fRANSAC 으로 구한 epipolar line(우) 는 다음과 같다. RANSAC 없이 결정한 myF 는 눈에 띄게 차이를 보이는 것으로 확인된다.