

# CSC420 Assignment 2

Alex (Kao-Tsun) Chang

October 14, 2017

1

(a)

```
1 function Harris
2
3     radius = 1;
4     threshold = 0.342;
5
6     img = double(rgb2gray(imread('/h/u9/g6/00/changkao/csc420/assignments/Assignment2/building.jpg')))
7
8     [Ix,Iy] = imgradientxy(img);
9
10    g = fspecial('gaussian', 3, 6);
11
12    Ix2 = conv2(Ix.^2, g, 'same');
13    Iy2 = conv2(Iy.^2, g, 'same');
14    Ixy = conv2(Ix.*Iy, g, 'same');
15    R = (Ix2.*Iy2 - Ixy.^2)./(Ix2 + Iy2);
16
17    figure; imagesc(R); axis image; colormap(gray);
18
19    max_filter = strel('disk', radius).Neighborhood;
20
21    ord_max = nnz(max_filter==1);
22
23    max_pixel = ordfilt2(R, ord_max, strel('disk', radius).Neighborhood);
24
25    n = threshold*max(max_pixel(:));
26
27    [r,c] = find(max_pixel>=n);
28
29    figure, imagesc(img), axis image, colormap(gray),hold on
30    plot(c, r, 'ro');
31    hold off;
32 end
```

Figure 1: Harris Corner w/ maximal suppression

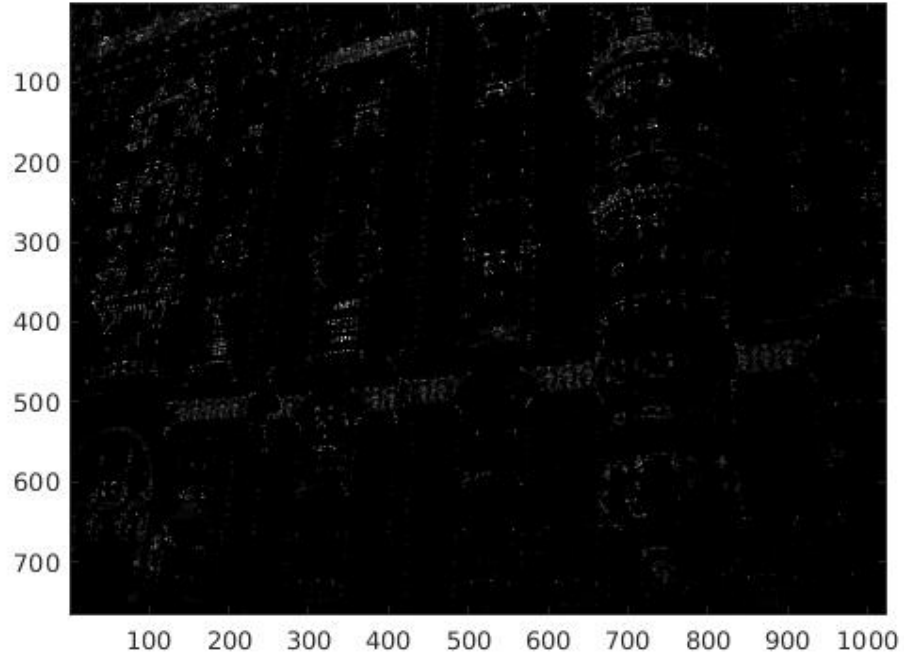


Figure 2: Output

(b) With a bigger  $r$ , the interest point is compared with a bigger area during non-max suppression, thus it is more likely to be pruned during non-max suppression, thus less interest points as  $r$  increases.

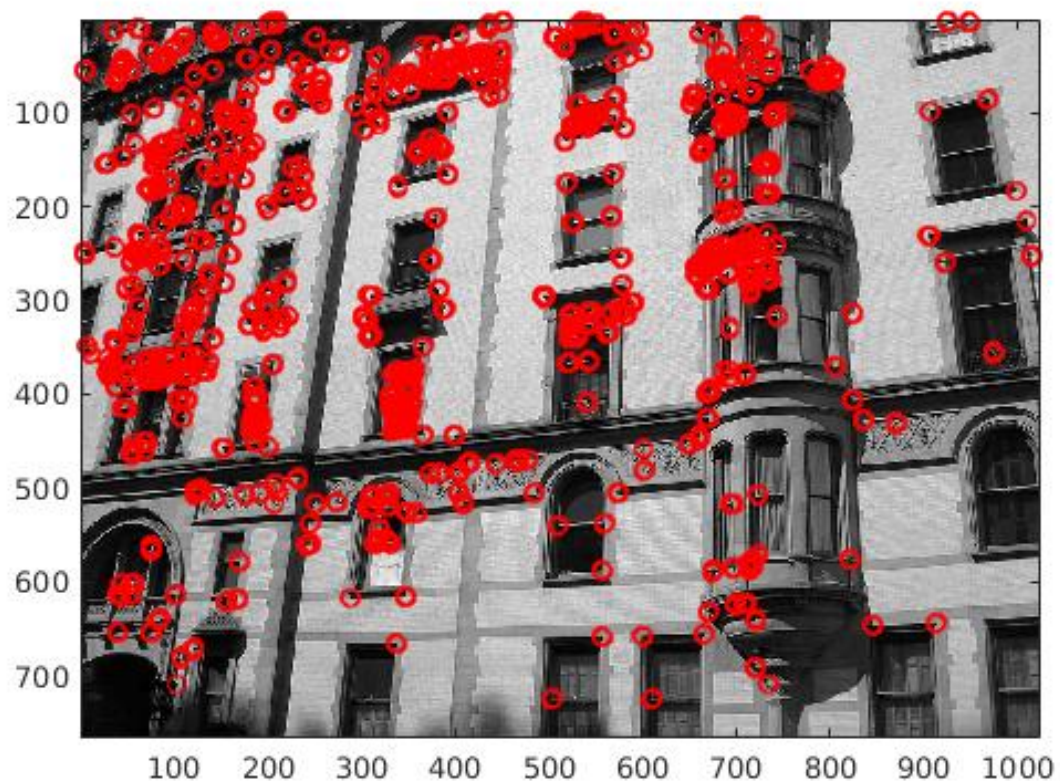


Figure 3: output w/ threshold = 0.342;  $r = 1$

```

1
2 function LoG
3
4     img = double(rgb2gray(imread('synthetic.png')));
5
6     radius = 2;
7
8     threshold = 0.2;
9     k = 1.1;
10    sigma = 5;
11    s = k.^(1:50)*sigma;
12
13    resLoG = zeros(size(img,1),size(img,2),length(s));
14    figure, imagesc(img), axis image, colormap(gray),hold on
15
16    %% Filter over a set of scales - borrowed from tutorial
17    for si = 1:length(s);
18
19        sL = s(si);
20        hs= max(25,min(floor(sL*3),128));
21        HL = fspecial('log',[hs hs],sL);
22        imgFiltL = conv2(img,HL,'same');
23        resLoG(:,:,si) = (sL^2)*imgFiltL;
24    end
25
26    for i = 1:length(s)
27
28        sc = s(i);
29        max_filter = strel('disk', radius).Neighborhood;
30        ord_max = nnz(max_filter==1);
31        n = ordfilt2(resLoG(:,:,i), ord_max, max_filter);
32        t = threshold*min(n(:));
33        [y,x] = find(n<=t);
34        total_pts = size(y, 1);
35
36        for j = 1:length(total_pts)
37
38            if i == 1
39                result = isScaleMax(resLoG(:,:,1:1),resLoG(:,:,2:3), x(j), y(j));
40            elseif i == length(s)
41                result = isScaleMax(resLoG(:,:,i:i), resLoG(:,:,i-2:i-1), x(j), y(j));
42            else
43                result = isScaleMax(resLoG(:,:,i:i),resLoG(:,:,i-1:2:i+1), x(j), y(j));
44            end
45
46            if result == 1
47

```

Figure 4: Laplacian matlab code

```

1  □ function result = isScaleMax(reslog, scalelogs, x, y)
2
3
4  -     Scale3_1 = scalelogs(y-1:y+1, x-1:x+1, 1:1);
5  -     Scale3_2 = scalelogs(y-1:y+1, x-1:x+1, 2:2);
6  -     |
7  -     [min_1, ~] = min(Scale3_1(:));
8  -     [min_2, ~] = min(Scale3_2(:));
9
10 -     % compare with neighborhood in the other scale spaces,
11 -     % return true, if it is the maximum in 3x3x3 neighborhood
12 -     if (reslog(y, x) <= min_1)&&(reslog(y, x) <= min_2)
13 -         result = 1;
14 -     else
15 -         result = 0;
16 -     end
17
18 - end

```

Figure 5: Laplacian helper function - isScaleMax matlab code



Figure 6: Laplacian output

(d)

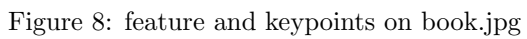
The Harris detector and Laplacian detector seems to be able to detect sharp edge corners well, as well as the corners detected were all similar in size. The difference between the two were that the Laplacian was able to detect rounded edges and circles better then the harris corner detector

## 2

(a)

```
1 function sifttransform
2
3 %% Setup
4
5 %run('/h/u9/g6/00/changkao/Downloads/vlfeat-0.9.20/toolbox/vl_setup')
6 book = imread('book.jpg');
7 findbook = imread('findBook.JPG');
8
9
10
11 %% Part (a)
12
13 imbook = single(rgb2gray(book)) ;
14 imfindbook = single(rgb2gray(findbook)) ;
15
16 [f,d] = vl_sift(imbook) ;|
17 [f2, d2] = vl_sift(imfindbook) ;
18
19 figure, imagesc(imbook), axis image,hold on
20 perm = randperm(size(f,2)) ;
21 sel = perm(1:50) ;
22 h1 = vl_plotframe(f(:,perm)) ;
23 h2 = vl_plotsiftdescriptor(d(:,sel),f(:,sel)) ;
24 set(h1,'color','y','linewidth',3) ;
25 set(h2,'color','g') ;
26 hold off;
27
28 figure, imagesc(imfindbook), axis image,hold on
29 perm2 = randperm(size(f2,2)) ;
30 sel2 = perm2(1:50) ;
31 hb1 = vl_plotframe(f2(:,perm2)) ;
32 hb2 = vl_plotsiftdescriptor(d2(:,sel2),f2(:,sel2));
33 set(hb1,'color','y','linewidth',3) ;
34 %hold off;
35 %figure, image(imfindbook), axis image, hold on
36 set(hb2,'color','g') ;
37 hold off;
38
39
40
```

Figure 7: setup and 2a matlab code





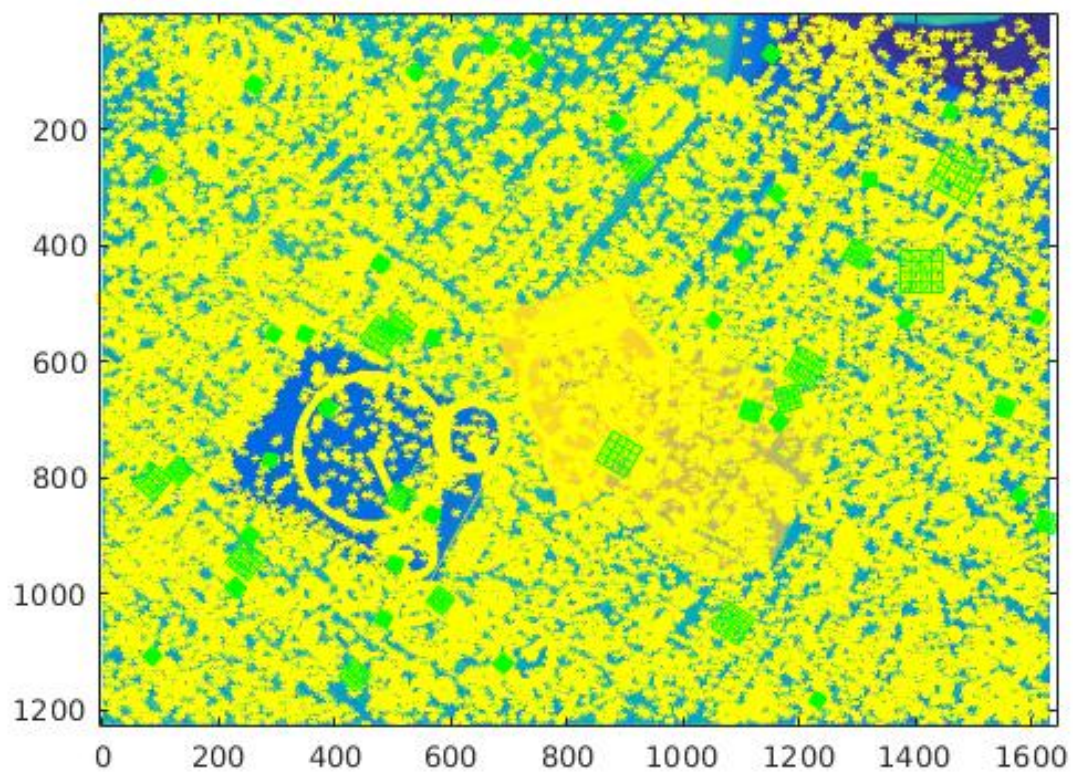


Figure 9: feature and keypoints on findBook.JPG

```

41 %% Part (b)
42
43 % threshold = 0.747
44 % threshold = 0.221
45 % threshold = 0.543
46 threshold = 0.673;
47
48 %offset for montage (displaying 2 images together)
49 offset = size(imfindbook);
50 offset = offset(2);
51
52 e_dist= pdist2(double(d)', double(d2)', 'euclidean');
53 ascending_l = sort(e_dist, 2);
54 r=ascending_l(:,1)./ascending_l(:,2);
55 %ratios=euc(:,1)./euc(:,2);
56
57 t_index = find(threshold>r);
58 num_matches = zeros(size(t_index,1), 3);
59
60 for i = 1:size(e_dist, 1)
61
62     if threshold > r(i)
63
64         num_matches(i,1) = r(i);
65         num_matches(i,2)= i;
66         t_matches = (e_dist(i,:)==ascending_l(i,1));
67         num_matches(i,3)=find(t_matches);
68     end
69 end
70
71 tesmp = any(num_matches, 2);
72 num_matches( ~tesmp, : ) = [];

```

Figure 10: 2b part 1 Matlab code

```

72 num_matches( ~tesmp, : ) = [];
73 mscores = zeros(size(num_matches,1), 1);
74 x_coordinates = zeros(size(num_matches,1),2);
75 y_coordinates = zeros(size(num_matches,1),2);
76
77 for i = 1:size(num_matches,1)
78
79     mscores(i) = num_matches(i,1);
80     x_coordinates(i,1:2) = [f(1,num_matches(i,2)) f2(1,num_matches(i,3))];
81     y_coordinates(i,1:2) = [f(2,num_matches(i,2)) f2(2,num_matches(i,3))];
82
83 end
84
85 figure, imshowpair(imbook, imfindbook, 'montage'), axis image, hold on
86
87 for i = 1:size(mscores,1)
88
89     drawnow;
90     plot([x_coordinates(i,1) offset+x_coordinates(i,2)], [y_coordinates(i,1) y_coordinates(i,2)], 'g'
91
92 end
93
94 hold off;
95
--

```

Figure 11: 2b part 2

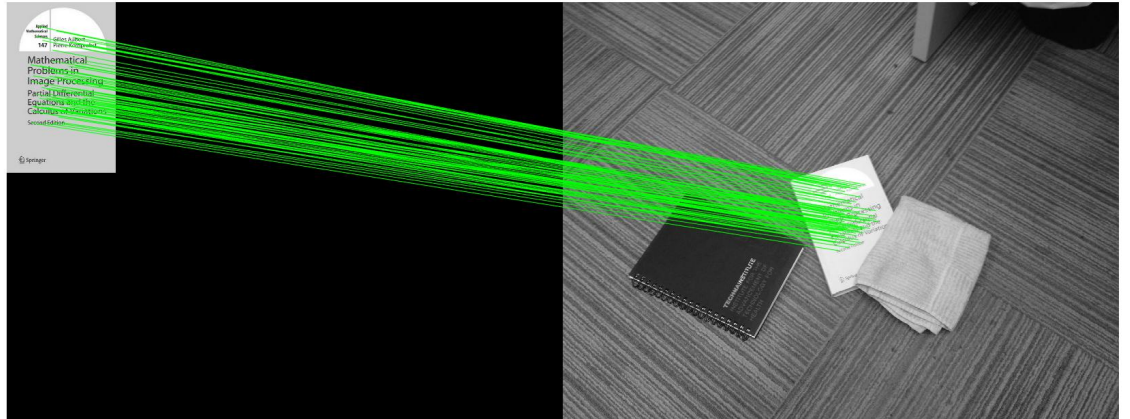


Figure 12: output, threshold = 0.673

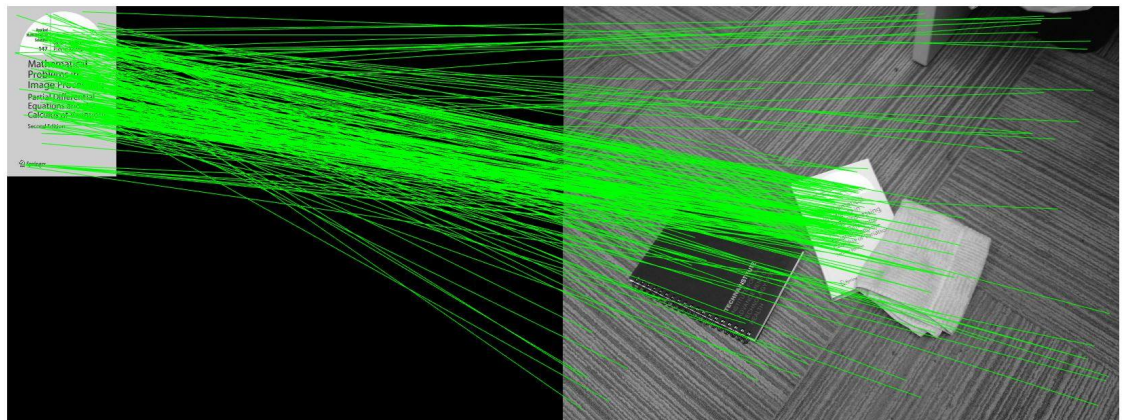


Figure 13: output, threshold = 0.98

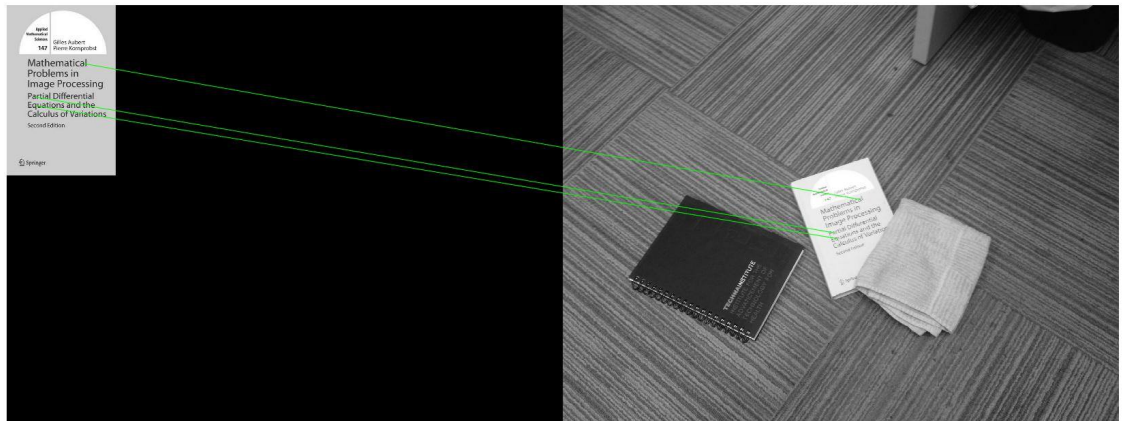


Figure 14: output, threshold = 0.275

```

98 %% Part (c and d)
99
100 k = 1;
101 %k = 4;
102 %k = 12;
103 %k = 50;
104
105 %helper to get k top matches and their coordinates
106 top = topmatches(mscores, x_coordinates, y_coordinates, k);
107
108 image_dim = size(imbook);
109
110 x = [0:image_dim(2);image_dim(2)+1];
111 y = [0:image_dim(1);image_dim(1)+1];
112
113 P = [];
114 PP = [];
115
116 for i = 1:k
117
118     %coordinates from top matches corresponding to image 1
119     x1 = top(i,2);
120     y1 = top(i,4);
121
122     %coordinates from top matches corresponding to image 2
123     x2 = top(i,3);
124     y2 = top(i,5);
125
126     P(size(P,1)+1,:) = [x1 y1 0 0 1 0];
127     P(size(P,1)+1,:) = [0 0 x1 y1 0 1];
128
129     PP(size(PP,1)+1,:) = x2;
130     PP(size(PP,1)+1,:) = y2;
131
132
133 end
134

```

Figure 15: 2c/d part 1 Matlab Code

```

1 function top = topmatches(mscores, x, y, k)
2
3     combined = [mscores, x, y];
4     sorted = sortrows(combined, 1);
5
6     top = sorted(1:k, :);
7
8 end

```

Figure 16: 2c/d helper - topmatches

```

136 - affine = (P'*P)'*inv((P'*P)*(P'*P))*P'*PP;
137 - A = [];
138 -
139 - for i = 1:4
140 -     A(size(A,1)+1,:) = [x(i) y(i) 0 0 1 0];
141 -     A(size(A,1)+1,:) = [0 0 x(i) y(i) 0 1];
142 - end
143 -
144 - transform = A*affine;
145 -
146 - new_x = transform(1:2:length(transform));
147 - new_y = transform(2:2:length(transform));
148 - offset_x = new_x + offset;
149 -
150 - figure, imshowpair(imbook, imfindbook, 'montage'), axis image, hold on
151 - for i = 1:4
152 -     drawnow;
153 -     plot([x(i) offset_x(i)], [y(i) new_y(i)], 'g');
154 - end
155 -
156 - connect_x= [offset_x' offset_x(1)];
157 - connect_y= [new_y' new_y(1)];
158 - plot(connect_x,connect_y);
159 - hold off;
160 -
161 - end
162 -
163 -

```

Figure 17: 2c/d part 2

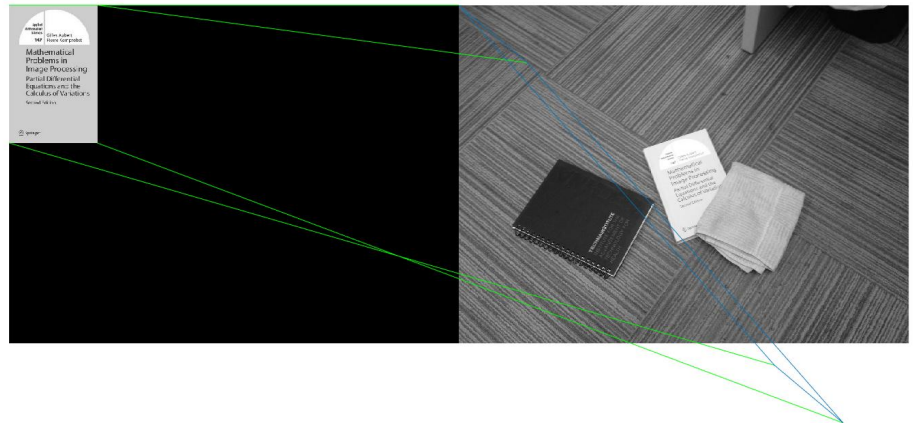


Figure 18: Affine transform w/  $k=1$



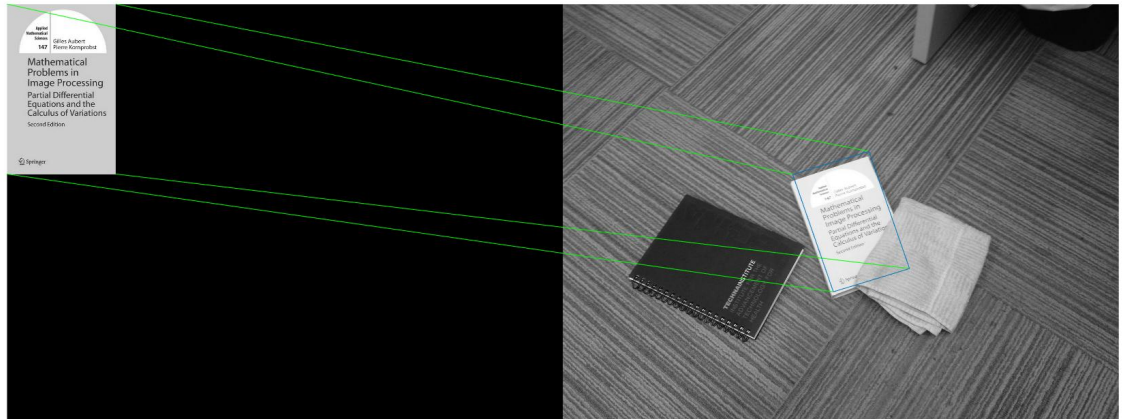


Figure 19: Affine transform w/  $k=4$

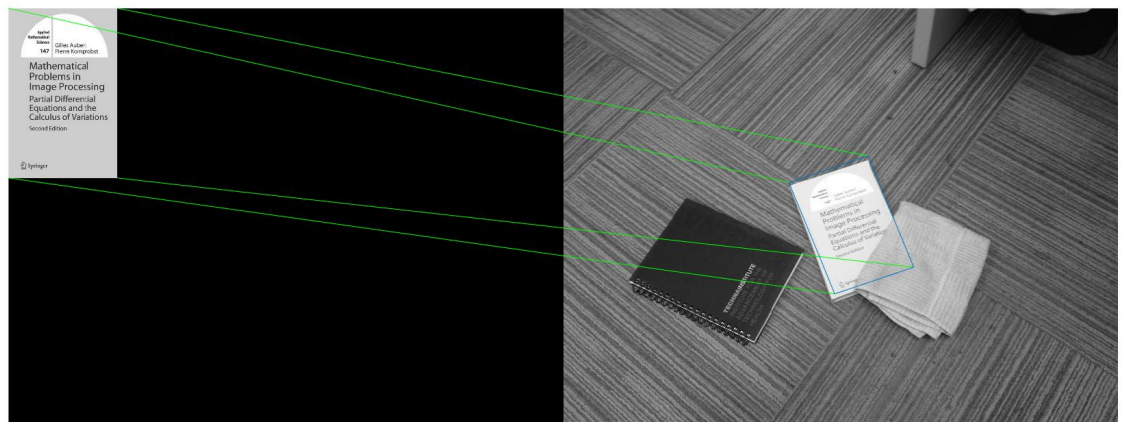


Figure 20: Affine transform w/  $k=12$

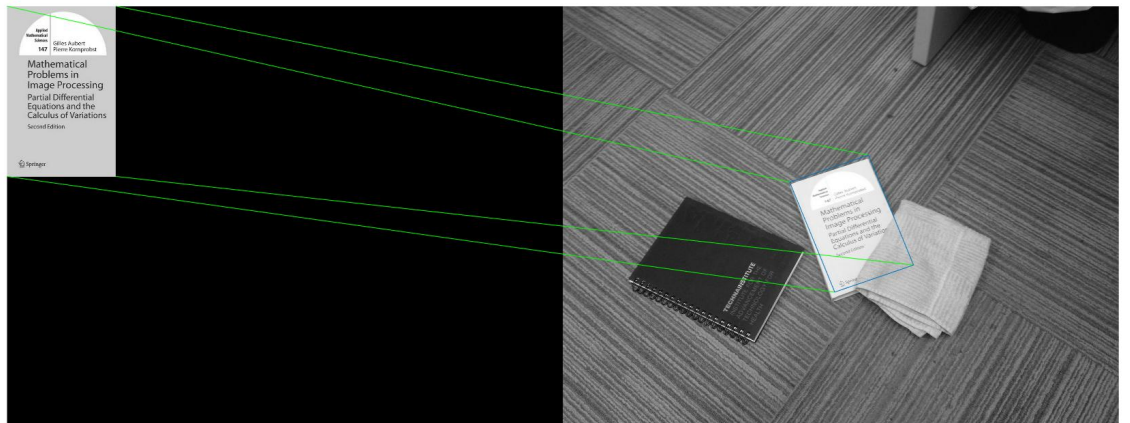


Figure 21: Affine transform w/  $k=50$