HOMEWORK2

Xinhui Shao

1) Input image

```
image=imread('scene.png');
image_detect=imread('object.png');
```

2) Color space conversion
I convert the color space from PNG to GRAY.

```
imageGRAY=mat2gray(image);
image_detectGRAY=mat2gray(image_detect);
```

Figure 1 imageGRAY

Figure 2 image_detectGRAY

3) Features2D detection and calculate descriptors
Before using “vl_sift”, I convert them to single images.

```
imageGRAY_s=single(imageGRAY);
image_detectGRAY_s=single(image_detectGRAY);
[f_image,d_image] = vl_sift(imageGRAY_s) ;
[f_detect,d_detect] = vl_sift(image_detectGRAY_s) ;
```

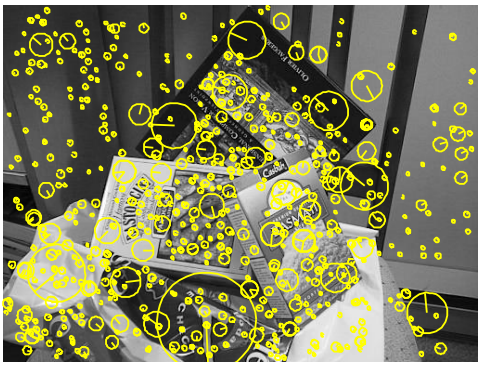

Figure 3 image’s keypoints

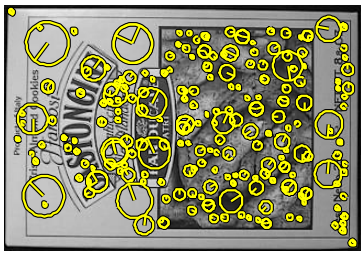

Figure 4 detection’s keypoints

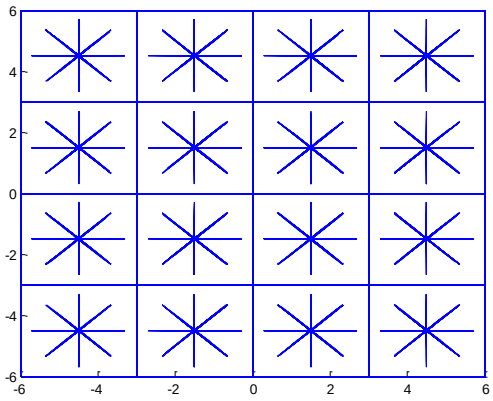

Figure 5 calculate descriptors

Where, f_image and f_detect are the feature2D,. The matrix f_image or f_detect has a column for each feature2D. A feature2D is a disk of center f_image(1:2), scale f_image(3) and orientation f_image(4), the same to f_detect.

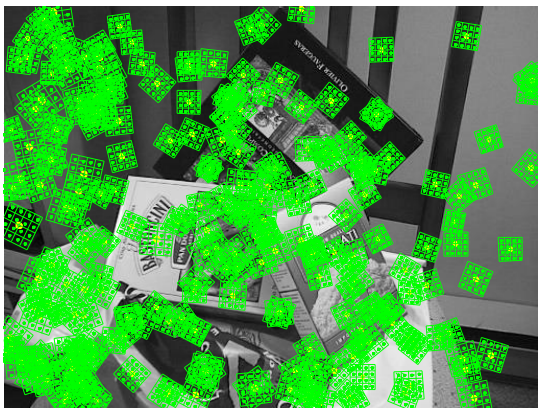

Figure 6 detect's descriptor

4) Match descriptors

```
[matches, scores] =
vl_ubcmatch(descriptor_image,descriptor_detect) ;
num_matches = size(matches,2) ;
X1 = feature_image(1:2,matches(1,:)) ; X1(3,:) = 1 ;
X2 = feature_detect(1:2,matches(2,:)) ; X2(3,:) = 1 ;
```

5) Find homography transformation

```
%%%%%%%%%%%%%求H矩阵
[matchLoc1 matchLoc2] = siftMatch(image, image_detect);
% % use RANSAC to find homography matrix
[H corrPtIdx] = findHomography(matchLoc2',matchLoc1');
```

H <3x3 double>

	1	2	3
1	0.4399	-0.1666	119.0092
2	-9.9219e-04	0.3995	161.2158
3	-2.5139e-04	-3.6470e-04	1
4			

Figure 7 homography transformation matrix

6) Show match results

```
dh1 =
max(size(image_detectGRAY_s,1)-size(imageGRAY_s,1),0) ;
dh2 =
max(size(imageGRAY_s,1)-size(image_detectGRAY_s,1),0) ;
figure;
imshow(image_detectGRAY);clf;
imagesc([padarray(imageGRAY_s,dh1,'post')
padarray(image_detectGRAY_s,dh2,'post')]) ;
o = size(imageGRAY_s,2) ;
line([feature_image(1,matches(1,:));feature_detect(1,matc
hes(2,:))+o], ...

[feature_image(2,matches(1,:));feature_detect(2,matches(2
,:))]) ;
 title(sprintf('%d tentative matches', num_matches)) ;
 axis image off ;
```

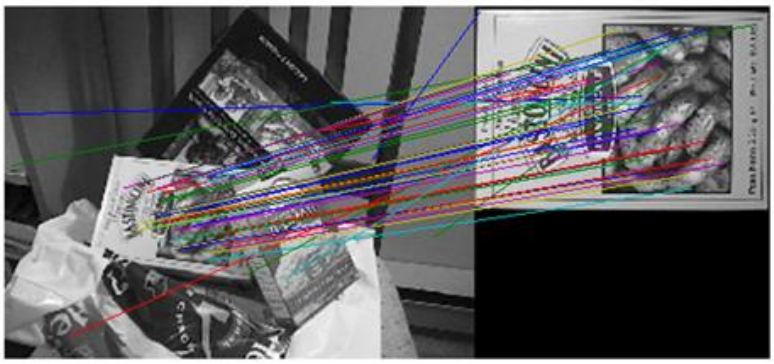

Figure 8 match descriptors

7) Perspective transform

```
%%%%%%%%%%%%%从图中框出目标

[m,n]=size(image_detect);
tform = maketform('projective',H');
```

```
img21 = imtransform(image_detect,tform);
pt = zeros(3,4);
pt(:,1) = H*[1;1;1];
pt(:,2) = H*[n;1;1];
pt(:,3) = H*[n;m;1];
pt(:,4) = H*[1;m;1];
x2 = pt(1,:)./pt(3,:);
y2 = pt(2,:)./pt(3,:);
figure;
imshow(image);
hold on
plot([x2(1),x2(2),x2(3),x2(4),x2(1)],[y2(1),y2(2),y2(3),y
2(4),y2(1)],...
     'Color',[1,0,0]);
```

Figure 9 localize the object