# Computer Vision HW4


Chien-Cheng Lai
ChBE

## Harris Corner Detector

### I. Implementation

1. I convert the input image to gray scale after loading the image.

```
%convert to gray image
grc = rgb2gray(image);
```

2. Compute Ix and Iy using central difference x-gradient and y-gradient filters respectively. I use meshgrid function to generate dx and dy matrix filters.

```
%Derivates of x and y
[dx,dy] = meshgrid(-1:1,-1:1);
ix = imfilter(double(grc),dx,'replicate');
iy = imfilter(double(grc),dy,'replicate');
```

3. Create a Gaussian smoothing filter (w) using fspecial with a chosen standard deviation (σ) and size 4σ. In this case, I choose 2 as sigma so the size would be 8.

```
%Create gassian filter
s = 2;
g = fspecial('gaussian',4*s,s);
```

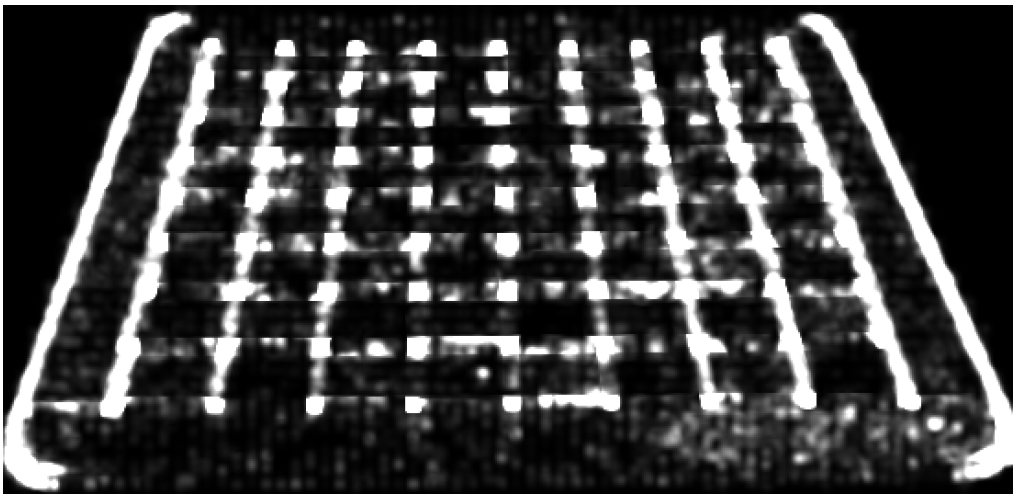
4. Apply the Gaussian filter to Ix2, Iy2 and IxIy using imfilter. I choose replicate in the boundary condition.

```
%Apply to Harris matrix
ix2 = imfilter(ix.^2,g,'replicate');
iy2 = imfilter(iy.^2,g,'replicate');
ixy = imfilter(ix.*iy,g,'replicate');
```

5. Compute the cornerness measure M.

```
%Compute the cornerness measure M
M = (ix2.*iy2-ixy.^2)./(ix2+iy2 + eps);
```

M image shows below,



6. I use ordfilt2 to compute non-maximal suppression on M to find local maximas. Moreover, I add threshold to filter out undesired points. The parameters I use are 3 for radius 2000 for threshold.

```
% Corner extraction
r = 3;
thresh = 2000;

%Perform non-maximal suppression
Ms = ordfilt2(M,r^2,true(r));
corpoints = (M == Ms) & (M > thresh);
```

7. Find the coordinates of the corner points and display the image and superimpose the corners.

```
%Find the coordinates of the corner points
[row,col] = find(corpoints);

%Display the image and superimpose the corners
imshow(image),hold on,
plot(col,row,'ys');
```

8. Finally, put them all in to a function call “harris” with inputs including image, sigma, radius and threshold and with output corner points image. So I could call the function to display the corner image.

```
function harris(image, sigma, radius, threshold)
harris(im, 2, 3, 2000)
```

Image shows below,

We can see the result is pretty good that points out every corners without noises.

## II. Rotation and Scaling

1. Rotate the image by 30 degrees. Moreover, I change the background color to white (black default) since the original image background is white. That is, it will eliminate undesired the points in the boundary.

```
%Change background to white
imr = imrotate(im,30); %Rotate the image
mrot = ~imrotate(true(size(im)),30);
imr(mrot ~= zeros(size(mrot))) = 255;
```

2. Resize the chessboard image by 4 times.

```
ims = imresize(im,4); %Resize the image
```

3. Apply the “harris” function to images.

```
figure(1)
harris(im, 2, 3, 2000) %original image
figure(2)
harris(imr, 2, 3, 5000) %rotated image
figure(3)
harris(ims, 8, 5, 600) %Resized image
```

4. As the results,
   Rotated image (sigma = 2, radius = 3, threshold = 5000):

I think the rotation will increase the cornerness measure M if that point changes its location from on vertical or horizontal line to slash line. So I increase the threshold to filter out the points on the boundary.

Resized image (sigma = 8, radius = 5, threshold = 1000):

It is a four times bigger image. I think resizing will cause difficulty for non-maximal suppression since the gradient of image becomes smoother. The intensity of each pixel becomes as the same as its neighbors. That is, the gradient of each pixel is decreasing comparing to original one. So in order to fit this case, I decrease the threshold and increase the sigma and radius to a better outcome.

# III. SURF

In computer vision, speeded up robust features (SURF) is a robust local feature detector and descriptor based on SIFT. Applications includes object recognition, image registration, classification and 3D reconstruction. It much faster than SIFT and more robust toward several different image transformations than SIFT.

The SURF algorithm is based on the same principles and steps as SIFT; but details in each step are different. The algorithm has three main parts: interest point detection, local neighborhood description and matching.

## Detection

Its detector bases on the Hessian matrix due to its good performance in computation time and accuracy. However, Hessian-Laplace detector uses a different measure for selecting the location and the scale while surf relies on the determinant of the Hessian for both. The determinant of the Hessian matrix is used as a measure of local change around the point and points are chosen where this determinant is maximal.

Furthermore, in other feature detection algorithms, the scale space is usually realized as an image pyramid. Images are repeatedly smoothed with a Gaussian filter, then they are subsampled to get the next higher level of the pyramid. Hence, unlike previous methods, scale spaces in SURF are implemented by applying box filters of different sizes. Owing to the use of box filters and integral images, it does not have to iteratively apply the same filter to the output of a previously filtered layer, but instead can apply such filters of any size at exactly the same speed directly on the original image.

In order to localize interest points in the image and over scales, a non-maximum suppression in a $3 \times 3 \times 3$ neighborhood is applied. The maxima of the determinant of the Hessian matrix are then interpolated in scale and image space with the method proposed by Brown, et al.

## Descriptor

The goal of a descriptor is to provide a unique and robust description of an image feature. It uses relative intensity and orientations of gradients reduces the effect of photometric changes. The first step consists of fixing a reproducible

orientation based on information from a circular region around the interest point. Then construct a square region aligned to the selected orientation and extract the SURF descriptor from it.

In order to achieve rotational invariance, the orientation of the point of interest needs to be found. it calculates the Haar-wavelet responses in x and y directions within a circular neighborhood of radius 6s around the point of interest are computed, where s is the scale at which the point of interest was detected. Again, it uses integral images for fast filtering. The horizontal and vertical responses within the window are summed. The two summed responses then yield a local orientation vector.

For the purpose of describing the region around the point, it constructs a square region centered on the interest point and oriented along the orientation as selected previously. Then the interest region is split into smaller 4x4 square sub-regions, and for each one, the Haar wavelet responses are extracted at 5x5 regularly spaced sample points. The responses are weighted with a Gaussian (to offer more robustness for deformations, noise and translation).

# Matching

Based on previous works, we could match different images by comparing the features we found in each image.

In conclusion, in its experiment, the average recognition rates reflect the results of our performance evaluation. The leader is SURF-128 with 85.7% recognition rate, followed by U-SURF (83.8%) and SURF (82.6%). The other descriptors achieve 78.3% (GLOH), 78.1% (SIFT) and 72.3% (PCA-SIFT).

It is obviously that SURF provides a much faster and more robust way to find the features in images based on the SIFT. It outperforms in both in speed and accuracy.